GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF ELECTRONICS ENGINEERING

ELEC 365

Fundamentals of Digital Communications

MATLAB Project

Due: 13.5.2024

Name	Arda
Last Name	DERİCİ
No	200102002051

Contents

Introduction

In a baseband communication systems, the evaluation of Bit Error Rate (BER) is important to see the performance of the system under various conditions. In this project, it is observed the BER performance of a baseband communication system operating over an Additive White Gaussian Noise (AWGN) channel. The system uses two different waveforms, $s_1(t)$ and $s_2(t)$ to represent bits "1" and "0" respectively.

Analytical derivations for the BER expression are important for simulation. In this project, analytical expressions for BER are derived under two different probability. Firstly, when the probabilities of transmitting "1" and "0" are equal $(P(1) = P(0) = 1/2)$ and secondly, when the probabilities are unequal $(P(1) = 1/4,\ P(0) = 3/4)$. These expressions are found to compare the results obtained with simulation.

The objective of this project is to analyze and compare the theoretical BER expressions with the simulated BER curves across a range of SNR values. This comparison process gives a general idea about the accuracy of theoretical expressions and the practical performance of the communication system used in this project.

In this project; analytical derivations of BER expressions, MATLAB code with explanations written for the simulation, simulation results of the system, BER curve vs SNR and comments on the observed differences and similarities between theoretical and simulated BER curves are presented.

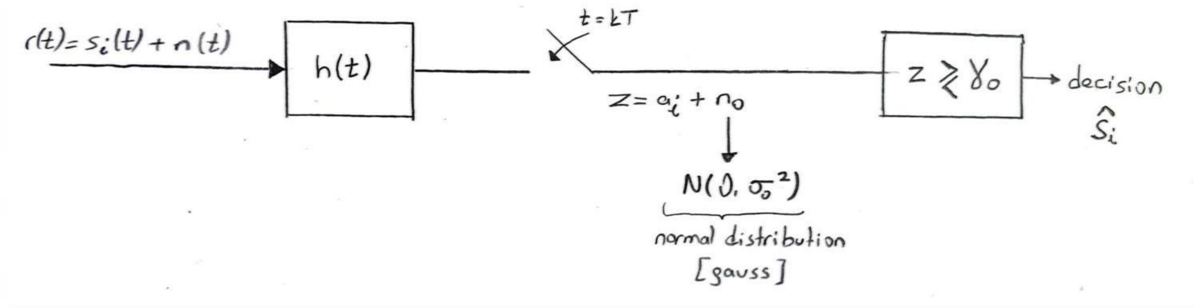
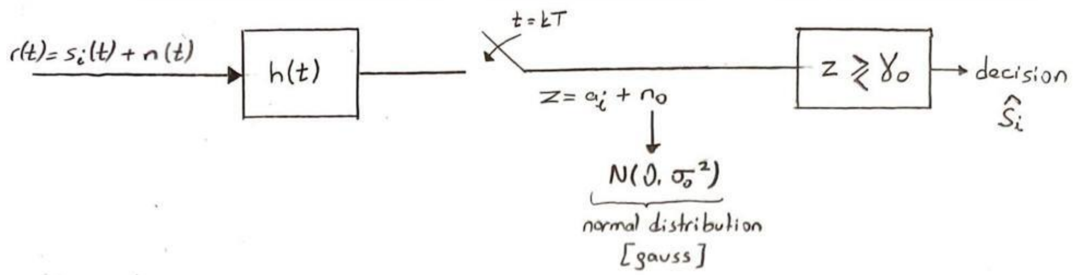
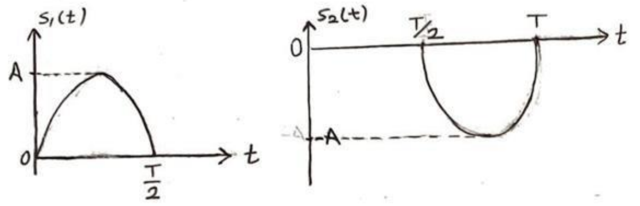


Figure 1 - Baseband communication system block diagram.

Analytical Derivations



The communication system is as above.



$$s_1(t) = \begin{cases} A\sin\left(\frac{2\pi t}{T}\right), & 0 \le t \le \frac{T}{2} \\ 0, & \text{else.} \end{cases}$$

$$s_2(t) = \begin{cases} -A\sin\left(\frac{2\pi\left(t-\frac{T}{2}\right)}{T}\right), & \frac{T}{2} \le t \le \frac{T}{4} \\ 0, & \text{else.} \end{cases}$$

=> $s_1(t)$ is transmitted for the bit "1".
=> $s_2(t)$ is transmitted for the bit "0".

$a_i = \int_T s_i(t)\,[s_1(t) - s_2(t)]\,dt, \quad i = 1, 2.$ so, a_1 and a_2 are calculated as follows.

$$a_1 = \int_0^{T/2} s_1(t)\,[s_1(t) - \underbrace{s_2(t)}_{0}]\,dt = \int_0^{T/2} s_1^2(t)\,dt.$$

$$a_1 = \int_0^{T/2} A^2 \sin^2\left(\frac{2\pi t}{T}\right) dt \underset{\sin^2(x) = \frac{1}{2}(1-\cos(2x))}{=} \frac{A^2}{2}\int_0^{T/2}\left[1 - \cos\left(\frac{4\pi t}{T}\right)\right] dt = \frac{A^2}{2}\left[\int_0^{T/2} 1\,dt - \underbrace{\cancel{\int_0^{T/2}\cos\left(\frac{4\pi t}{T}\right) dt}}_{\text{period } \frac{T}{2}}\right]$$

integral over a period "0"

$$a_1 = \frac{A^2}{2}\left[t\Big|_0^{T/2}\right] = \frac{A^2 T}{4}$$

$$a_2 = \int_{T/2}^{T} s_2(t)\,(\underbrace{s_1(t)}_{0} - s_2(t))\,dt = -\int_{T/2}^{T} s_2^2(t)\,dt$$

$$a_2 = -\int_{T/2}^{T} A^2 \sin^2\left(\frac{2\pi\left(t-\frac{T}{2}\right)}{T}\right) dt = -A^2\int_{T/2}^{T}\sin^2\left(\frac{2\pi t}{T} - \pi\right) dt = -A^2\int_{T/2}^{T}\sin^2\left(\frac{2\pi t}{T}\right) dt$$

$$a_2 = -\frac{A^2}{2}\left[\int_{T/2}^{T} 1\,dt - \cancel{\int_{T/2}^{T}\cos\left(\frac{4\pi t}{T}\right) dt}^{\,0}\right] = -\frac{A^2 T}{4}$$

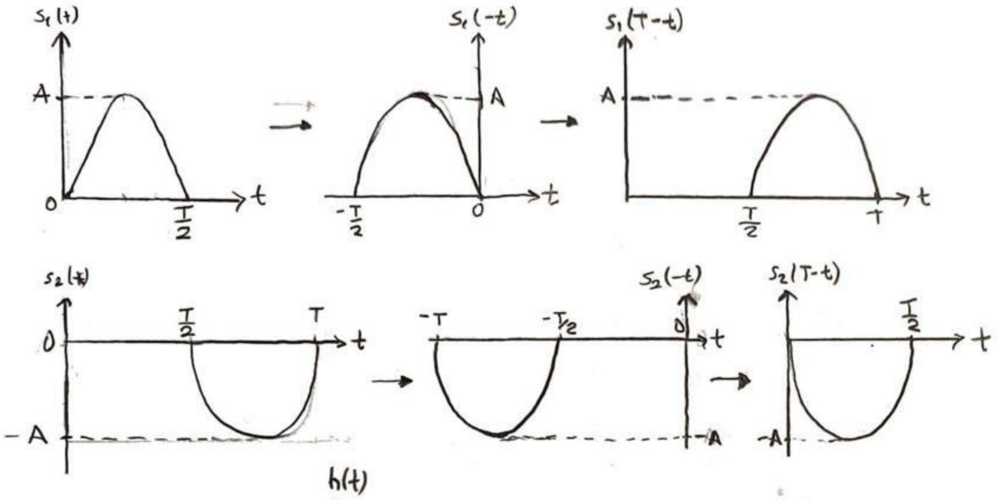
$$E_1 = \int_T |s_1(t)|^2\,dt = \int_0^{T/2} A^2 \sin^2\left(\frac{2\pi t}{T}\right) dt = \frac{A^2}{2}\left[\int_0^{T/2} 1\,dt - \cancel{\int_0^{T/2}\cos\left(\frac{4\pi t}{T}\right) dt}^{\,0 \text{ (one period)}}\right] = \frac{A^2 T}{4}$$

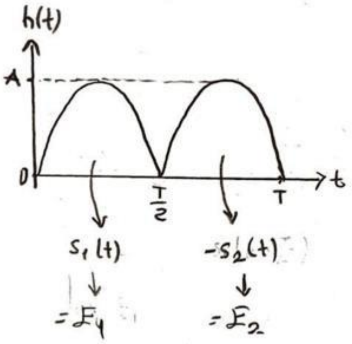
$$E_2 = \int_T |s_2(t)|^2\,dt = \int_{T/2}^{T}\left|-A\sin\left(\frac{2\pi\left(t-\frac{T}{2}\right)}{T}\right)\right|^2 dt = \frac{A^2}{2}\left[\int_{T/2}^{T}\sin^2\left(\frac{2\pi t}{T} - \pi\right) dt\right] = \frac{A^2}{2}\left[\int_{T/2}^{T}\sin^2\left(\frac{2\pi t}{T}\right) dt\right]$$

$$E_2 = \frac{A^2}{2}\left[\int_{T/2}^{T} 1\,dt - \cancel{\int_{T/2}^{T}\cos\left(\frac{4\pi t}{T}\right) dt}^{\,0}\right] = \frac{A^2 T}{4}$$

$\sigma_0^2 = \frac{N_0}{2} E_h$. so, E_h must be calculated.

$h(t) = s_1(T-t) - s_2(T-t)$





$$E_h = \int_0^T |h(t)|^2 \, dt$$

$$E_h = \underbrace{E_1}_{\frac{A^2T}{4}} + \underbrace{E_2}_{\frac{A^2T}{4}}$$

$$E_h = \frac{A^2T}{2}$$

$$\sigma_0^2 = \frac{N_0}{2} E_h = \frac{N_0}{2} \cdot \frac{A^2T}{2} = \frac{N_0 A^2 T}{4}$$

a) for $P(1) = P(0) = \frac{1}{2}$,

$$\gamma_0 = \frac{a_1 + a_2}{2} = \frac{\frac{A^2T}{4} - \frac{A^2T}{4}}{2} = 0$$

$$E_b = \sum_{i=1}^{2} E_i \cdot P(s_i) = \underbrace{E_1}_{\frac{A^2T}{4}} \underbrace{P(s_1)}_{\frac{1}{2}} + \underbrace{E_2}_{\frac{A^2T}{4}} \underbrace{P(s_2)}_{\frac{1}{2}} = \frac{A^2T}{8} + \frac{A^2T}{8} = \frac{A^2T}{4}$$

If $E_b = 1$ is determined,

$$E_b = \frac{A^2T}{4} = 1 \Rightarrow A^2T = 4$$

$$a_1 = \frac{A^2T}{4} = \frac{4}{4} = 1, \quad a_2 = \frac{-A^2T}{4} = \frac{-4}{4} = -1, \quad \sigma_0^2 = \frac{N_0 \overbrace{A^2T}^{4}}{4} = N_0$$

$$E_1 = \frac{A^2T}{4} = \frac{4}{4} = 1, \quad E_2 = \frac{A^2T}{4} = \frac{4}{4} = 1, \quad E_h = \frac{A^2T}{2} = \frac{4}{2} = 2.$$

$$P_b = Q\left(\frac{a_1 - a_2}{2\sigma_0}\right) = Q\left(\frac{1-(-1)}{2\sqrt{N_0}}\right) = Q\left(\sqrt{\frac{1}{N_0}}\right)$$

b) for $P(1) = \frac{1}{4}$, $P(0) = \frac{3}{4}$

$$\gamma_0 = \frac{\sigma_0^2}{a_1 - a_2} \ln \frac{P(s_2)}{P(s_1)} + \frac{a_1 + a_2}{2}$$

so, σ_0^2 must be found for this case.

$$E_b = \sum_{i=1}^{2} E_i P(s_i) = \underbrace{E_1}_{\frac{A^2T}{4}} \underbrace{P(s_1)}_{\frac{1}{4}} + \underbrace{E_2}_{\frac{A^2T}{4}} \underbrace{P(s_2)}_{\frac{3}{4}} = \frac{A^2T}{16} + \frac{3A^2T}{16} = \frac{A^2T}{4}$$

If $E_b = 1$ is determined,

$$E_b = \frac{A^2T}{4} = 1 \Rightarrow A^2T = 4. \quad \text{Then,}$$

$$a_1 = \frac{A^2T}{4} = \frac{4}{4} = 1, \quad a_2 = \frac{-A^2T}{4} = \frac{-4}{4} = -1, \quad \sigma_0^2 = \frac{N_0 \overbrace{A^2T}^{4}}{4} = N_0$$

$$\gamma_0 = \frac{N_0}{1-(-1)} \ln \frac{\frac{3}{4}}{\frac{1}{4}} + \underbrace{\frac{1-1}{2}}_{0} = \frac{N_0}{2} \cdot 1.099 = 0.549 N_0$$

$$P_b = \left[1 - Q\left(\frac{\gamma_0 - a_1}{\sigma_0}\right)\right] P(s_1) + Q\left(\frac{\gamma_0 - a_2}{\sigma_0}\right) P(s_2)$$

$$P_b = \left[1 - Q\left(\frac{0.549N_0 - 1}{\sqrt{N_0}}\right)\right] \cdot \frac{1}{4} + Q\left(\frac{0.549N_0 + 1}{\sqrt{N_0}}\right) \cdot \frac{3}{4}$$

Simulation is done with MATLAB with the values found in the analytical derivations. Thus, BER curve vs SNR is obtained for both equal probability and different probability situations.

MATLAB Code with Explanations

For a) $P(1) = P(0) = 1/2$, MATLAB code with explanations are as follows.

```
% ELEC 365 - Fundamentals of Digital Communications %
% Spring 2024 %
% MATLAB Project %
% Arda DERİCİ - 2001020020251 %

% BER expression of this system over AWGN for a) P(1) = P(0) = 1/2

% SNR values in dB are determined
SNRdB= 0:1:15;
% Let Eb = 1 and using the equation SNR_dB = 10*log(Eb/N0), 16 numerical
% values of N0 are found
N0 = 1./(10.^(SNRdB/10));

% for each SNR value, number of bits is determined as 100 million
num_bits = 10^8;
```

```
% for each SNR value, 'num_bits' of bits are generated with randi()
% command. randi() is uniformly distributed for equal probability.
% 'bits' consist of bits 0 and 1.
bits = randi([0, 1], length(SNRdB), num_bits);

% in the analytical derivations, it is calculated as a1 = 1 and a2 = -1
a1 = 1;
a2 = -1;
% a matrix of size 16 x 100 million is generated
ai = zeros(length(SNRdB), num_bits);
% if the bit is 1, then ai equals a1; if the bit is 0, then ai equals a2
ai(bits == 0) = a2;
ai(bits == 1) = a1;

% a normal distribution of the form N(0, σ=1) is generated as noise
% with randn() command
n0 = randn(length(SNRdB), num_bits);
% for each SNR value, a normal distribution of the form N(0, σ^2=N0)
% is generated. also, σ^2=N0 is found in the analytical derivations
% it is known that if Y = α.X, then μY=α.μX, σY^2=α^2.σX^2
% thus, the α value is found as the square root N0
% so that the variance of the n0 Gaussian noise can be
% at the N0 value found in the analytical derivations
n0 = sqrt(N0') .* n0;

% it is calculated z = ai + n0 for each SNR value
z = ai + n0;
% in the analytical derivations, it is calculated as γ = 0
gama = 0;

% a zeros matrix is genrated to hold
% the number of error bits for each SNR value
num_bit_err = zeros(length(SNRdB), 1);
% the decision part for error detection is made with a 'for' loop
for i = 1:length(SNRdB)
    % for each SNR value, the number of bits received correctly
    % at the receiver when bit 1 is transmitted is found
    ctrl_for_bit1 = sum(ai(i,z(i,:) > gama) == 1);
    % for each SNR value, the number of bits received correctly
    % at the receiver when bit 0 is transmitted is found
    ctrl_for_bit0 = sum(ai(i,z(i,:) < gama) == -1);
    % the number of bit errors is calculated for each SNR value
    num_bit_err(i) = num_bits - (ctrl_for_bit1 + ctrl_for_bit0);
end

% it is the bit error rate (BER) expression in the analytical derivations
% it is used qfunc() command
theo_ber = qfunc(sqrt(1./N0));
% plotting process
figure;
% theoretical BER curve plotted
semilogy(SNRdB, theo_ber, 'LineWidth', 2);
hold on
% simulated BER curve plotted
semilogy(SNRdB, num_bit_err./num_bits, 'rh', 'LineWidth', 4);
legend('theoretical BER curve', 'simulated BER curve', ...
    'FontWeight', 'bold', 'FontSize', 14);
xlabel('SNR (dB)', 'FontWeight', 'bold', 'FontSize', 14);
```

```
xticks(SNRdB);
ylabel('Bit Error Probability (P_b)', 'FontWeight', 'bold', ...
    'FontSize', 14);
set(gca, 'FontSize', 14);
title('Theoretical and Simulated BER Curves for Equal Probability', ...
    'FontSize', 14, 'FontWeight','bold');
grid on;
```

For b) $P(1) = 1/4$, $P(0) = 3/4$, MATLAB code with explanations are as follows.

```
% BER expression of this system over AWGN for b) P(1) = 1/4, P(0) = 3/4

% SNR values in dB are determined
SNRdB= 0:1:15;
% Let Eb = 1 and using the equation SNR_dB = 10*log(Eb/N0), 16 numerical
% values of N0 are found
N0 = 1./(10.^(SNRdB/10));

% for each SNR value, number of bits is determined as 100 million
num_bits = 10^8;

% probabilities are determined for bits 1 and 0.
P_1 = 1/4;
P_0 = 3/4;
% random bits are generated for each SNR value with rand() command
bits = rand(length(SNRdB), num_bits);
% 0 and 1 bits are determined according to probabilities
% 'bits' consist of bits 0 and 1.
bits = bits < P_1;

% in the analytical derivations, it is calculated as a1 = 1 and a2 = -1
a1 = 1;
a2 = -1;
% a matrix of size 16 x 100 million is generated
ai = zeros(length(SNRdB), num_bits);
% if the bit is 1, then ai equals a1; if the bit is 0, then ai equals a2
ai(bits == 0) = a2;
ai(bits == 1) = a1;

% a normal distribution of the form N(0, σ=1) is generated as noise
% with randn() command
n0 = randn(length(SNRdB), num_bits);
% for each SNR value, a normal distribution of the form N(0, σ^2=N0)
% is generated. also, σ^2=N0 is found in the analytical derivations
% it is known that if Y = α.X, then μY=α.μX, σY^2=α^2.σX^2
% thus, the α value is found as the square root N0
% so that the variance of the n0 Gaussian noise can be
% at the N0 value found in the analytical derivations
n0 = sqrt(N0') .* n0;
% it is calculated z = ai + n0 for each SNR value
z = ai + n0;
% in the analytical derivations, it is calculated as γ = 0.549*N0
gama = 0.549*N0;
```

```
% a zeros matrix is genrated to hold
% the number of error bits for each SNR value
num_bit_err = zeros(length(SNRdB), 1);
% the decision part for error detection is made with a 'for' loop
for i = 1:length(SNRdB)
    % for each SNR value, the number of bits received correctly
    % at the receiver when bit 1 is transmitted is found
    ctrl_for_bit1 = sum(ai(i,z(i,:)>gama(i)) == 1);
    % for each SNR value, the number of bits received correctly
    % at the receiver when bit 0 is transmitted is found
    ctrl_for_bit0 = sum(ai(i,z(i,:)<gama(i)) == -1);
    % the number of bit errors is calculated for each SNR value
    num_bit_err(i) = num_bits - (ctrl_for_bit1 + ctrl_for_bit0);
end

% it is the bit error rate (BER) expression in the analytical derivations
% it is used qfunc() command
theo_ber = (1 - qfunc((0.549*N0 - 1) ./ sqrt(N0)))*1/4 + qfunc( ...
    (0.549*N0 + 1) ./ sqrt(N0))*3/4;

% plotting process
figure;
% theoretical BER curve plotted
semilogy(SNRdB, theo_ber, 'LineWidth', 2);
hold on
% simulated BER curve plotted
semilogy(SNRdB, num_bit_err./num_bits, 'rh', 'LineWidth', 4);
legend('theoretical BER curve', 'simulated BER curve', ...
    'FontWeight', 'bold', 'FontSize', 14);
xlabel('SNR (dB)', 'FontWeight', 'bold', 'FontSize', 14);
xticks(SNRdB);
ylabel('Bit Error Probability (P_b)', 'FontWeight', 'bold', ...
    'FontSize', 14);
set(gca, 'FontSize', 14);
title('Theoretical and Simulated BER Curves for P(1) = 1/4, P(0) = 3/4', ...
    'FontSize', 14, 'FontWeight','bold');
grid on;
```

Simulation Results of the System and BER Curve vs SNR

In order to obtain better simulation results for each SNR value, simulations is done with 100 million bits in MATLAB. Simulations for equal probability and unequal probability situations take a total of 2 minutes. In this case, it can be concluded that the code works effectively. The simulation result for the case with equal probability is in Figure 2, and the simulation result for the case with unequal probability is in Figure 3.

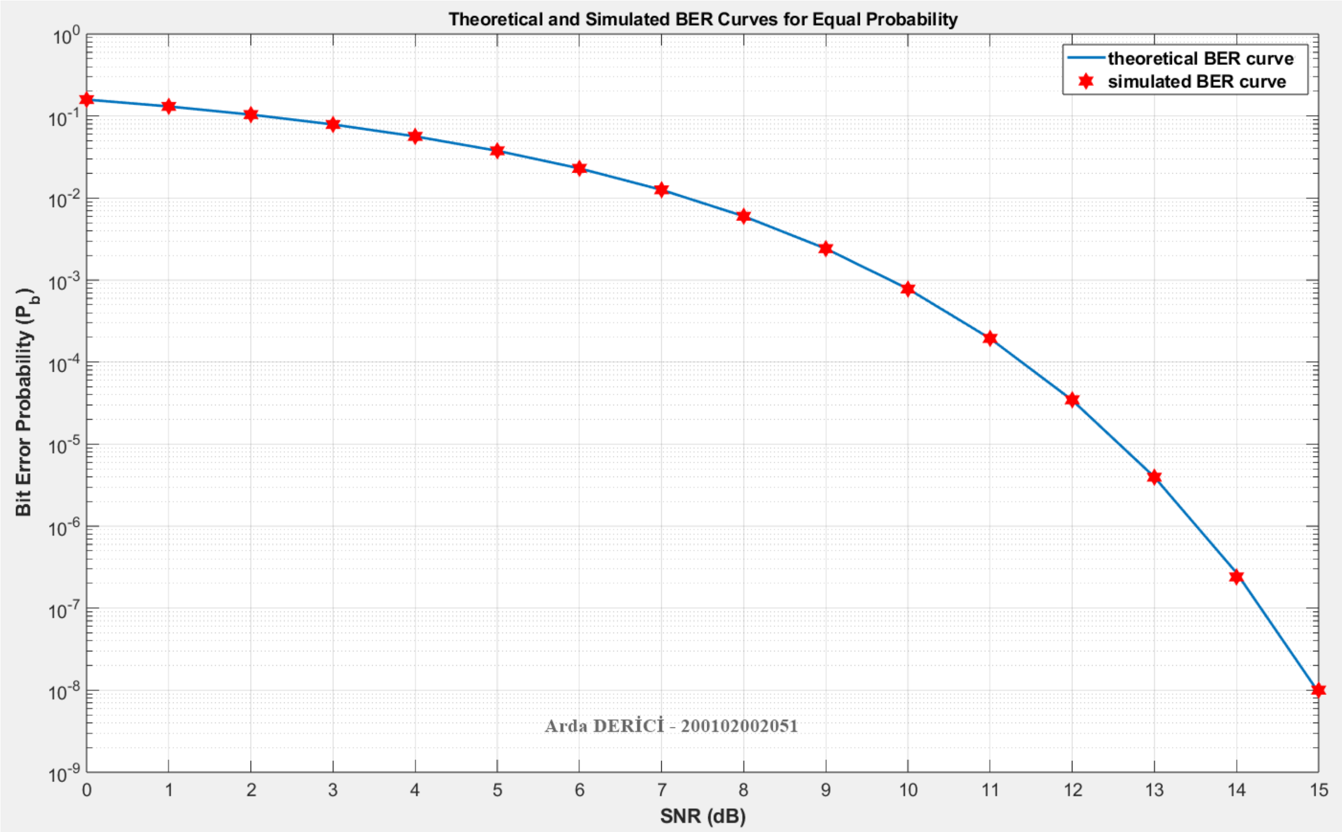


Figure 2 – Theoretical and simulated BER curves for a) $P(1) = P(0) = 1/2$.

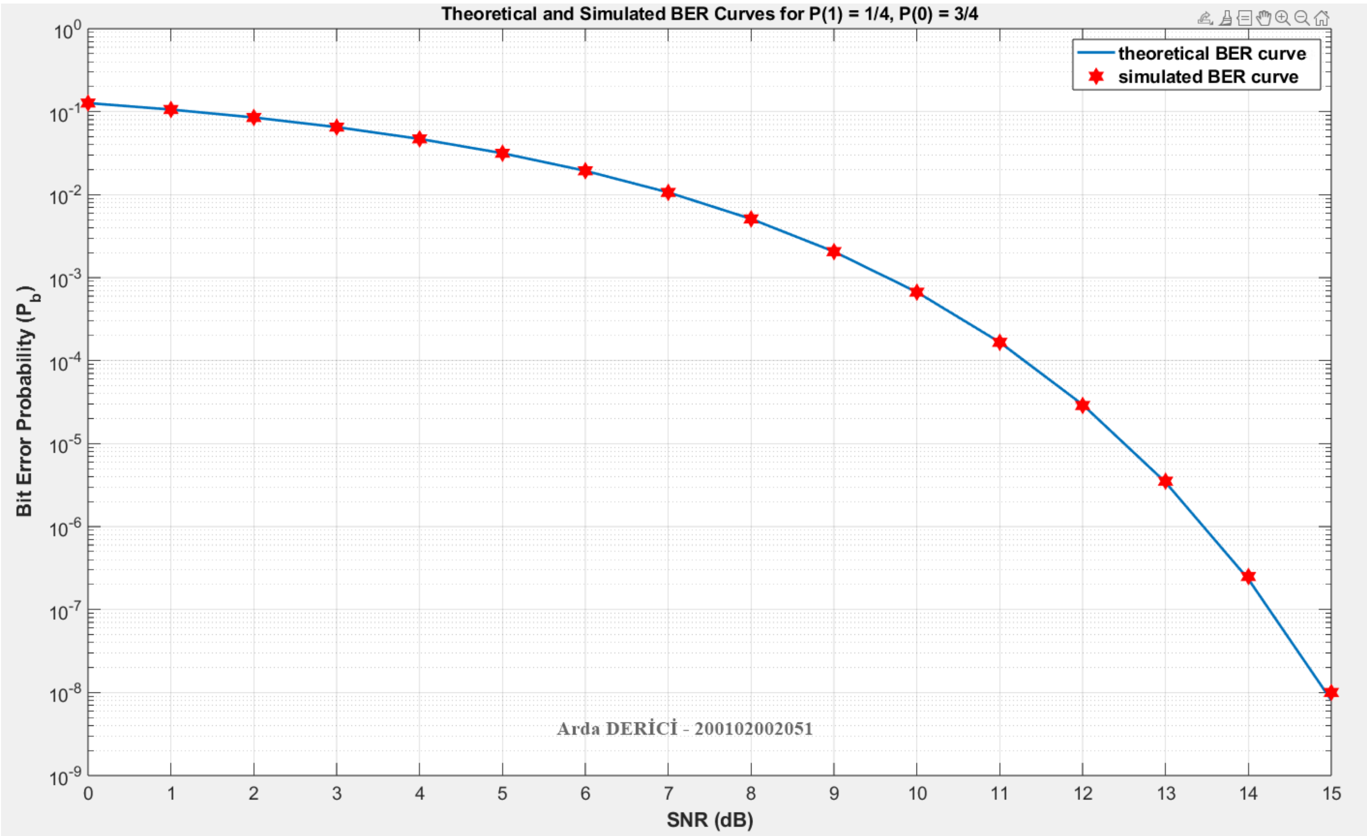


Figure 3 – Theoretical and simulated BER curves for b) $P(1) = 1/4, P(0) = 3/4$.

Comparison and Comments

Looking the results in Figures 2 and 3, it is seen that the agreement between the theoretical and simulation BER curves obatains different probability situation. In Figure 2, where equal probabilities are assumed, the largely overlap between the theoretical and simulation BER curves is apparent. Similarly, in Figure 3, where the probabilities are unequal with $P(1) = 1/4, P(0) = 3/4$, it is observed a similar level of overlap between the theoretical and simulation curves. This consistency for different probabilities shows the reliability of both the MATLAB vector simulation and the analytical derivations for bipolar signaling system. In coclusion, the close similarity between theoretical and simulation results shows the accuracy of analytical and computational approaches in modeling bipolar signaling systems.

Conclusions

In this project, bipolar signaling in the baseband communication system, which is learned theoretically, is implemented practically by performing vector simulation in MATLAB. Since the objective of this project is to analyze and compare theoretical BER expressions with simulated BER curves for each SNR value, it can be said that this project has been successfully completed. This comparison process gives a general idea about the accuracy of theoretical statements and the practical performance of the communication system used in this project. The ability to make analytical derivations about what should be used in baseband communication system design and use them in MATLAB for simulation is obtained. So, the project is done.

References

Digital Communications: Fundamentals and Applications, Bernard Sklar, Prentice Hall, 2001, 2nd Ed., ISBN-10: 0130847887

gtu.sharepoint.com/:v:/s/ELM365Spring2024/Ef7X_ucZJX5NnBx4Fvd_U6kBafDf7mTfh0zFgcBSouTkUA?e=aRz6Po