GEBZE TEKNİK ÜNİVERSİTESİ

ELEKTRONİK MÜHENDİSLİĞİ

SAYISAL HABERLEŞME SİSTEMLERİ - PROJE ÖDEVİ

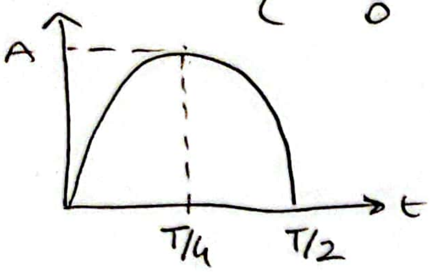
ADI SOYADI: AHMET ALİ TİLKİCİOĞLU

NUMARASI: 210102002163

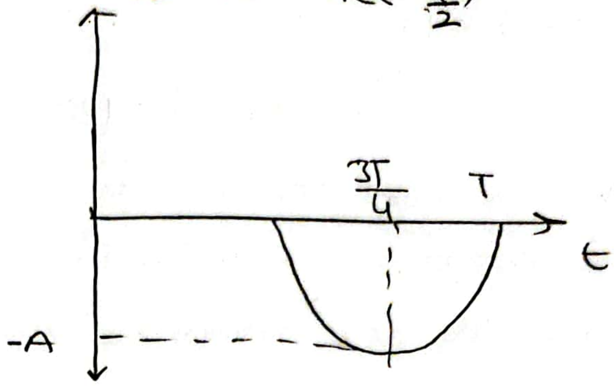
BÖLÜM : ELEKTRONİK MÜHENDİSLİĞİ

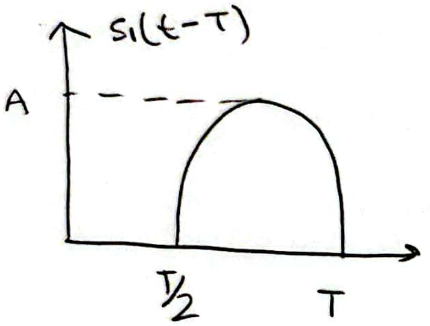
Analytic Solutions;

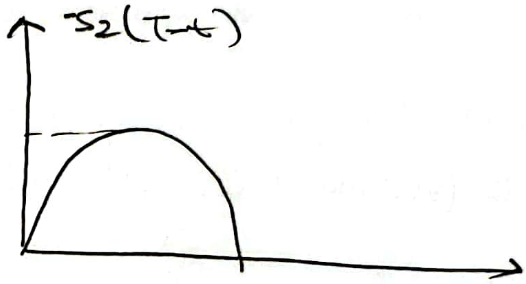
$$s_1(t) = \begin{cases} A\sin\left(\frac{2\pi t}{T}\right), & 0 \le t \le T/2 \\ 0, & \text{Else} \end{cases}$$



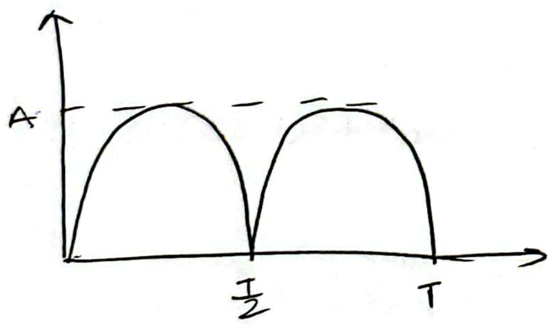
$$s_2(t) = -s_1\left(t - \frac{T}{2}\right)$$



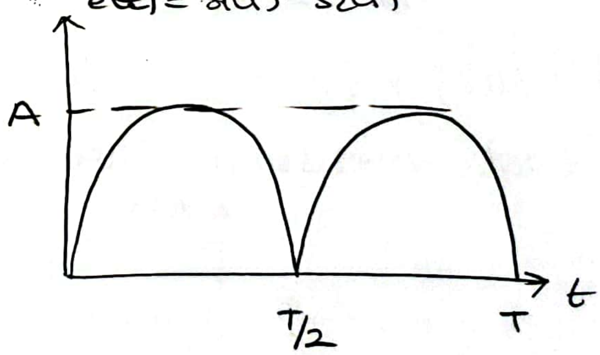




$$h(t) = s_1(T-t) - s_2(T-t)$$



$$e(t) = s_1(t) - s_2(t)$$



$$a_1(T) = \int_0^T [s_1(t) - s_2(t)]\, s_1(t)\, dt$$

$$a_1(T) = \int_0^T [s_1(t) - s_2(t)] s_1(t)\, dt = \int_0^{T/2} s_1(t)^2 dt + \int_{T/2}^{T} s_1(t)[\cancelto{0}{s_1(t)} - s_2(t)]\, dt$$

$$= \int_0^{T/2} \left(A\sin\left(\frac{2\pi t}{T}\right)\right)^2 dt = A^2 \int_0^{T/2} \sin^2\left(\frac{2\pi t}{T}\right) dt$$

$$= \frac{A^2}{2} \int_0^{T/2} 1 - \cos\left(\frac{4\pi t}{T}\right) dt = \frac{A^2}{2}\left(t - \frac{T}{4\pi}\sin\left(\frac{4\pi t}{T}\right)\Big|_0^{T/2}\right)$$

$$= \frac{A^2}{2}\left[T/2 - \underbrace{\frac{T}{4\pi}\sin(2\pi)}_{0} - \Big(0 - \underbrace{\frac{T}{4\pi}\sin(0)}_{0}\Big)\right] = \boxed{\frac{A^2 T}{4} = a_1}$$

$$a_2(T) = \int_0^T [s_1(t) - s_2(t)]\, s_2(t)\, dt = \int_{T/2}^{T} s_2(t)\, A \sin\left(\frac{2\pi t}{T}\right) dt$$

$$= -A^2 \int_{T/2}^{T} \sin^2\left(\frac{2\pi t}{T}\right) dt = \frac{-A^2}{2} \int_{T/2}^{T} 1 - \cos\left(\frac{4\pi t}{T}\right) dt$$

$$= \frac{-A^2}{2} \left[t - \frac{T}{4\pi} \sin\left(\frac{4\pi t}{T}\right) \right] \Bigg|_{T/2}^{T}$$

$$= \frac{-A^2}{2} \left[T - \frac{T}{4\pi} \sin(4\pi) - \left(\frac{T}{2} - \frac{T}{4\pi} \sin(2\pi) \right) \right] = \frac{-A^2}{2} \left[T - T/2 \right]$$

$$= \boxed{\frac{-A^2 T}{4} = a_2}$$

$$E_{s_1} = \int_0^{T/2} |s_1(t)|^2 dt = A^2 \int_0^{T/2} \sin\left(\frac{2\pi t}{T}\right) dt = \frac{A^2 T}{4} \text{ joule}$$

$$E_{s_2} = \int_{T/2}^{T} |s_2(t)|^2 dt = A^2 \int_{T/2}^{T} \sin\left(\frac{2\pi t}{T}\right) dt = \frac{A^2 T}{4} \text{ joule}$$

i) $\longrightarrow$ $P(s_1) = \frac{1}{2}$, $P(s_2) = \frac{1}{2}$

$$E_b = E_{s_1} P(s_1) + E_{s_2} P(s_2) = \frac{A^2 T}{4} \cdot \frac{1}{2} + \frac{A^2 T}{4} \cdot \frac{1}{2} = \frac{A^2 T}{4} \text{ joule}$$

$$\boxed{E_b = 1 = \frac{A^2 T}{4} \longrightarrow \begin{array}{l} a_1 = 1 \\ a_2 = -1 \end{array}}$$

$$E_h = \int_0^T h(t)^2 dt = A^2 \int_0^T \sin^2\left(\frac{2\pi t}{T}\right) dt = \frac{A^2 T}{2} = \underline{\underline{2}}$$

$$\gamma_0 = \frac{a_1 + a_2}{2} = \frac{1-1}{2} = \underline{\underline{0}}$$

$$P_b = Q\left(\frac{a_1 - a_2}{2\sigma_0}\right) = Q\left(\frac{1+1}{2\sigma_0}\right) = \boxed{Q\left(\frac{1}{\sigma_0}\right)}$$ for i)

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

$$\Rightarrow \sigma_0 = \sqrt{N_0}$$

ii) $P(s_1) = \frac{1}{4}$, $P(s_2) = \frac{3}{4}$

$$E_b = E_{s1} P(s_1) + E_{s2} P(s_2) = \frac{A^2T}{4} \cdot \frac{1}{4} + \frac{A^2T}{4} \cdot \frac{3}{4}$$

$$= \frac{A^2T}{16} + \frac{3A^2T}{16} = \boxed{\frac{A^2T}{4} = 1} \longrightarrow \begin{array}{l} a_1 = 1 \\ a_2 = -1 \end{array} \quad E_h = 2$$

$$\sigma_0^2 = \frac{N_0}{2} \cdot E_h = N_0 \longrightarrow \sigma_0 = \sqrt{N_0}$$

$$\gamma_0 = \frac{\sigma_0^2}{a_1 - a_2} \ln \frac{P(s_1)}{P(s_2)} + \cancel{\frac{a_1 + a_2}{2}}^{\,0}$$

$$= \frac{N_0}{2} \cdot \ln \frac{1}{3} = \boxed{\gamma_0 \cong -0{,}55 N_0}$$

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

$$\boxed{P_b = \left[1 - Q\left(\frac{-0{,}55 N_0 - 1}{\sqrt{N_0}}\right)\right] \cdot \frac{1}{4} + Q\left(\frac{-0{,}55 N_0 + 1}{\sqrt{N_0}}\right) \cdot \frac{3}{4}}$$

For ii) situation

Gebze Technical University

Digital Communication Project Solutions.

by Ahmet Ali Tilkıcıoğlu

Lecturer : Prof. Dr. Oğuz Kucur.

Simulation and Analytical Results:

a) side code:

```
% Ahmet Ali Tilkicioğlu - 210102002163 - Digital Communication
clc;
clear all;
close all;

% a side code
N = 10^7;                              % 10 million bit size
Eb = 1;                                % average bit energy
Ps1 = 0.5;                             % Probability bit 1
Ps2 = 0.5;                             % Probability bit 0
si = randsrc(1,N,[1,0;Ps1,Ps2]);       % si values
a1 = 1;                                % bit 1 signal value
a2 = -1;                               % bit 0 signal value
ai = 2*si - 1;                         % all ai signal values
max_dB = 17;                           % db value maximum limit

siAllValue = zeros(1,N);               % si vector
simPbValue = zeros(0,max_dB);          % simulation vector
AnltyPbValue = zeros(0,max_dB);        % analytic summaries vector

for db_SNR = 0:max_dB                  % loop for all process (0:18 db)

  SNR = 10^(db_SNR/10);               % SNR values
  N0 = Eb/SNR;                        % No values
  sigma0 = sqrt(N0);                  % sigma value

  z = ai + sigma0*randn(1,N);         % z = (ai + no) equations
  gamma0 = 0;                         % gamma formula

  siAllValue = z > gamma0;            % decision circuit simulation

  errorVal = si ~= siAllValue;        % Total error for simulation
  Pb = sum(errorVal)/N;               % Pb for simulation

  simPbValue = [simPbValue Pb];       % Pb simulation vector for graph
  analytical_sol = qfunc(1/sigma0);       % analytic solution
  AnltyPbValue = [AnltyPbValue analytical_sol]; % analytic solution

end

% Solution graphs code for a)
dB_SNR = 0:max_dB;
figure()
semilogy(dB_SNR,simPbValue,'go-');
hold on;
semilogy(dB_SNR,AnltyPbValue,'b*');
hold on;
legend('simulation', 'analytical')
grid on;
title('BER curve vs. SNR (Tilkicioğlu)');
xlabel('SNR(dB)');
ylabel('Pb');
ylim([10^-7, 10^0]);
```

The code structure written for a part of the project is given in the figure. The desired probability value of 10 million bits of the code was obtained. Then a1 and a2 are written onto the ai labels according to the turn-on output bits. The average bit energy is given as 1 joule. Other values can be derived from the results obtained in the analytical solution. Our simulation and analytical result is divided from 0 to 18 dB and its cycle is in the table in the code. The cyclic power of each dB costs, the mixture itself simulates sigma and the comparator circuit. The comparator circuit simulation is carried out on the gamma calculated by the code on the 'siAllValues' variable. To obtain the mixture (z = ai + no) noise, the problems posed by Gaussian arise. The strength of his body and his equations are obtained through the z variable. The 'randn()' function was used to obtain the Gaussian state. In the last part of the code, each analytical result was obtained by writing the Q() function we obtained analytically within the loop. The simulation shape obtained for a) is as shown in 1.

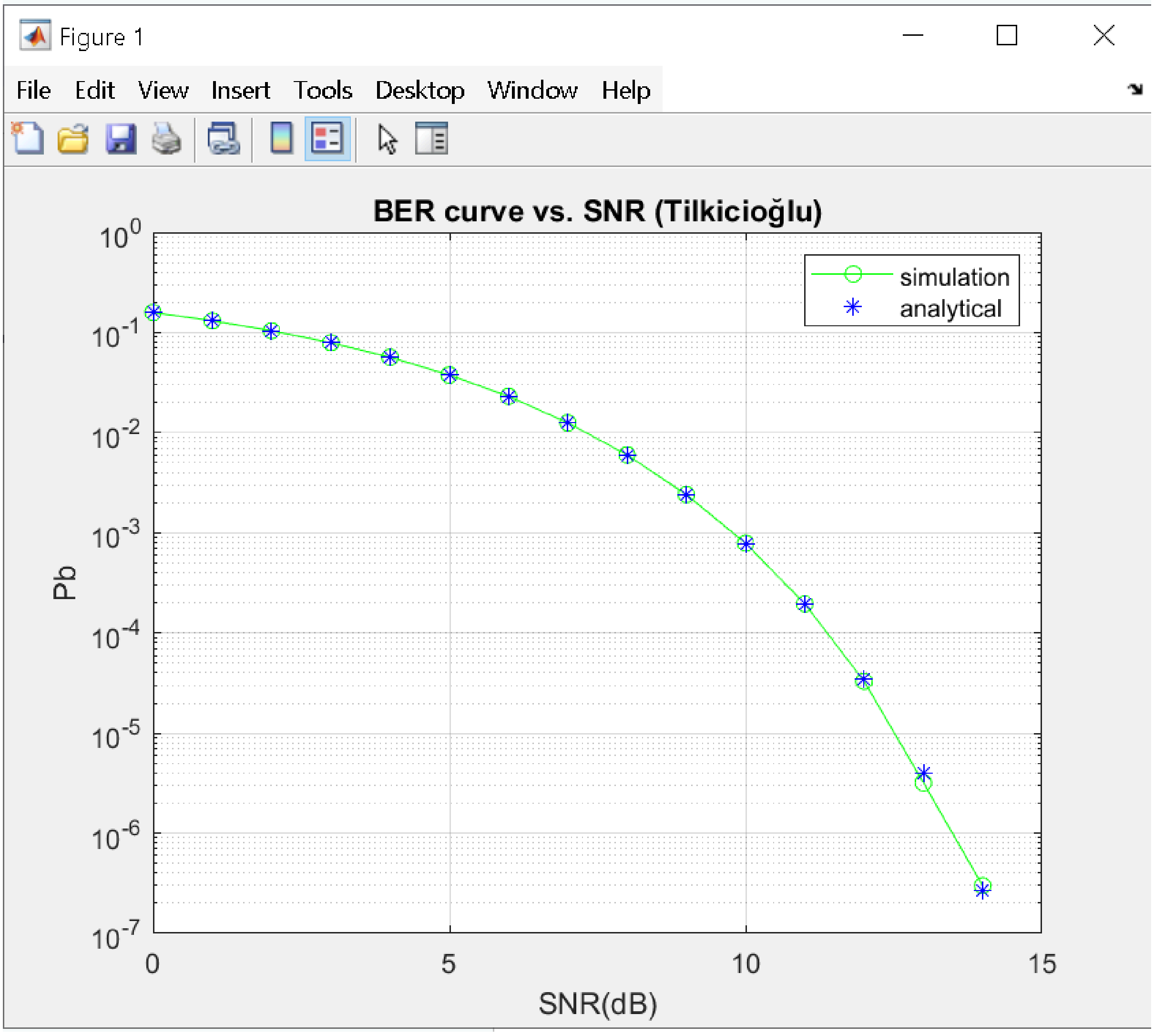


Figure 1: a) Simulation and Analytical Results

b) side code:

```
% b) side code
N = 10^7;                              % 10 million bit size
Eb = 1;                                % average bit energy
```

```matlab
Ps1 = 0.25;                          % Probability bit 1
Ps2 = 0.75;                          % Probability bit 0
si = randsrc(1,N,[1,0;Ps1,Ps2]);     % si values
a1 = 1;                              % bit 1 signal value
a2 = -1;                             % bit 0 signal value
ai = 2*si - 1;                       % all ai signal values
max_dB = 17;                         % db value maximum limit

siAllValue = zeros(1,N);             % si vector
simPbValue = zeros(0,max_dB);        % simulation vector
AnltyPbValue = zeros(0,max_dB);      % analytic summaries vector

for db_SNR = 0:max_dB                % loop for all process (0:18 db)

  SNR = 10^(db_SNR/10);              % SNR values
  N0 = Eb/SNR;                       % No values
  sigma0 = sqrt(N0);                 % sigma value

  z = ai + sigma0*randn(1,N);        % z = (ai + no) equations
  gamma0 = (N0/2)*log(Ps2/Ps1);      % gamma formula

  siAllValue = z > gamma0;           % decision circuit simulation

   errorVal = si ~= siAllValue;      % Total error for simulation
  Pb = sum(errorVal)/N;              % Pb for simulation

  simPbValue = [simPbValue Pb];      % Pb simulation vector for graph
  % analytic solution

analytical_sol=(1-qfunc((gamma0-a1)/sigma0))*Ps1+(qfunc((gamma0-a2)/sigma0))*Ps2;
  AnltyPbValue = [AnltyPbValue analytical_sol];    % analytic solution

end

% Solution graphs code for b)
dB_SNR = 0:max_dB;
figure()
semilogy(dB_SNR,simPbValue,'go-');
hold on;
semilogy(dB_SNR,AnltyPbValue,'b*');
hold on;
legend('simulation', 'analytical')
grid on;
title('BER curve vs. SNR (Tilkicioğlu)');
xlabel('SNR(dB)');
ylabel('Pb');
ylim([10^-7, 10^0]);
```

For option b, the exact code in option a was used. Only the desired bit probabilities have been changed. Figure 2 shows the output of the code for option b.

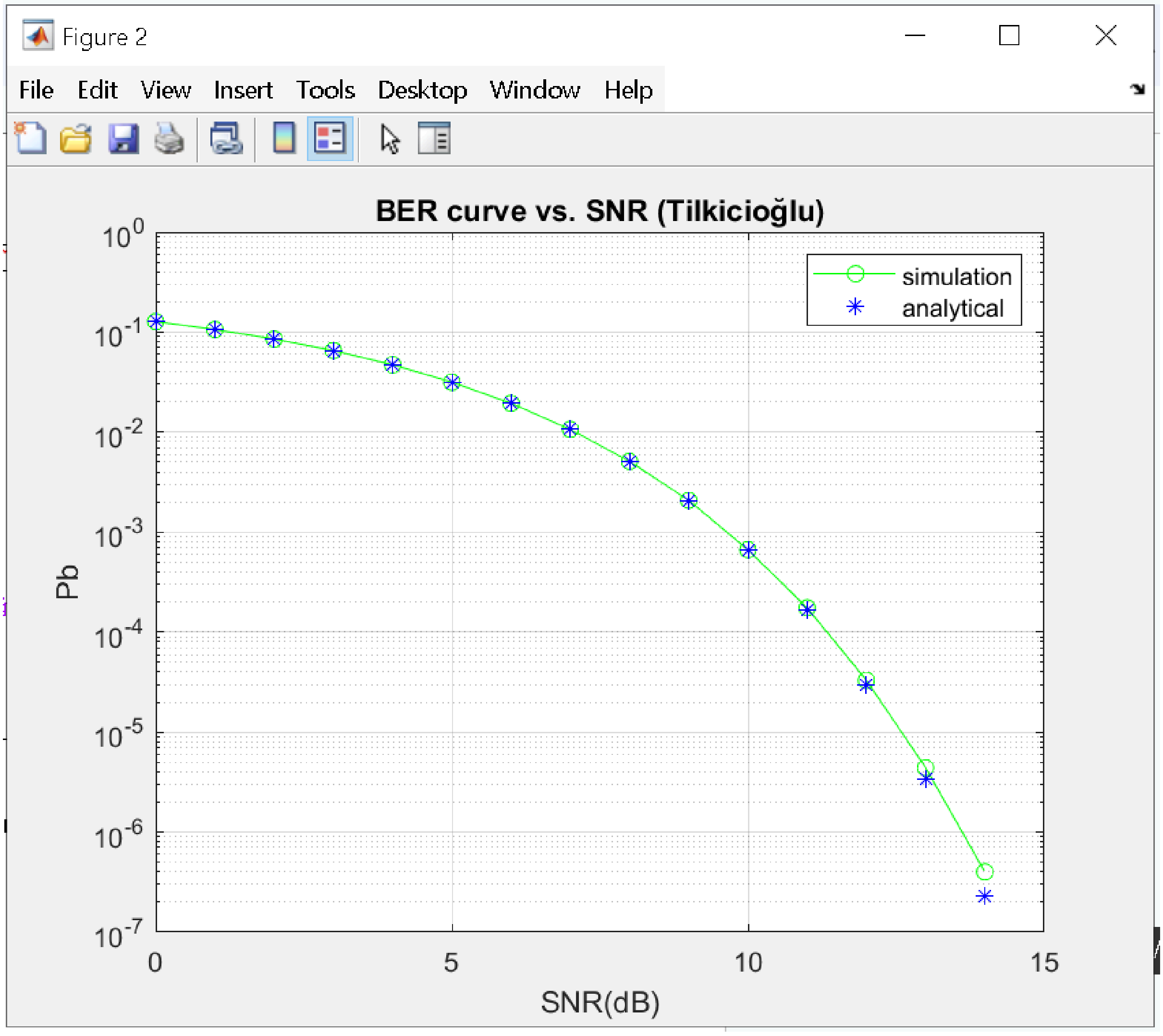


Figure 2: b) Simulation and Analytical Results

Interpretation of Simulation and Analytical Results:

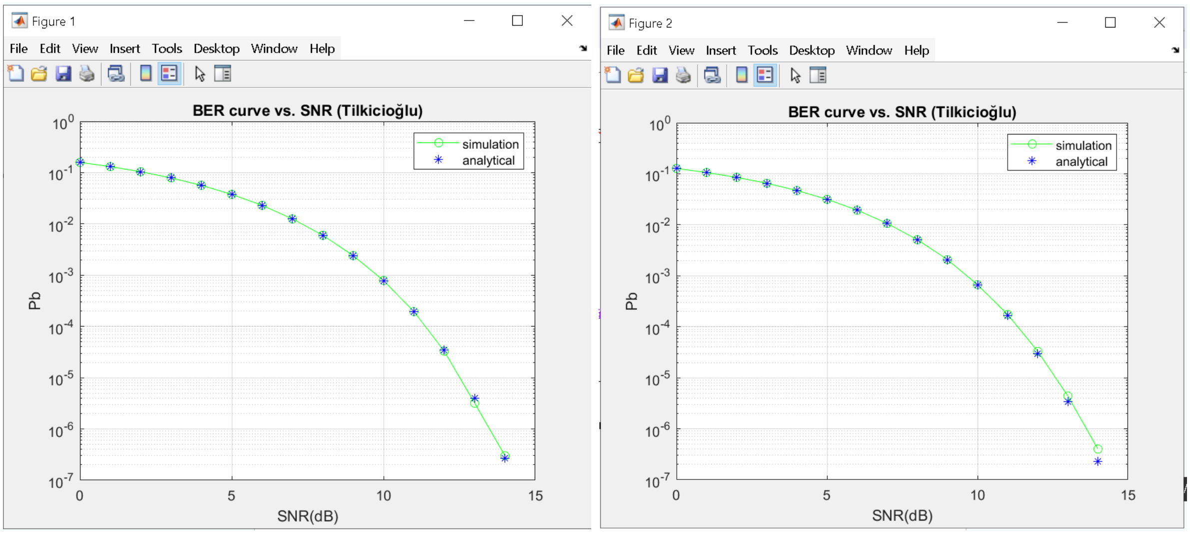


Figure 3: Analytical and Simulation Results in Both Cases

Similar results were obtained from simulation and analytical results for both probability cases. This is because their Eh's are the same. We can see that the bit error appearance of the SNR values of the table decreases, and this is valid for both cases. The analytical function calculated with the Q function was predicted very well in the simulation. Due to its characteristics, very small deviations occur. Tables for the values in the table When the SNR is 10 dB, our bit error probability is approximately 1 bit in 1000 bits.