GEBZE TECHNICAL UNIVERSITY
ELECTRONICS ENGINEERING DEPARTMENT

DIGITAL COMMUNICATION PROJECT

Project Name:
Digital Communication Error Probability Performance of Signaling

Prepared by: Fouad Aladhami
Student Number: 210102002808

Date of the project: 13.05.2024

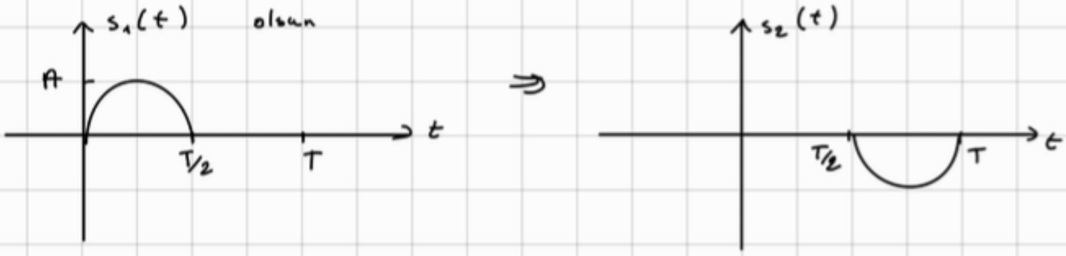
Contents

1 ANALYTIC SOLUTIONS 3
2 MATLAB FOR ANALYZE THEORICAL SOLUTIONS AND MAKE SYSTEM SIMULATIONS. 7
2.1 SOLUTION FOR EQUAL PROBABILITIES P(1) = P(0) = ½ 7
2.1.1 Plotting the theorical SNR vs PB curve in MATLAB 7
2.1.2 Simulation of the system and plotting SNR vs PB simulation 8
2.1.3 Comparison of Theorical and simulated BER Curves 9
2.1.4 MATLAB Code 9
2.2 SOLUTION FOR DIFFERENT PROBABILITIES P(1) =¼ P(0) = ¾ 13
2.2.1 Plotting the theorical SNR vs PB curve in MATLAB 13
2.2.2 Simulation of the system and plotting SNR vs PB simulation 14
2.2.3 Comparison of Theorical and simulated BER Curves 15
2.2.4 MATLAB Code 16
3 REFERENCES 17

1 Analytic Solutions

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

$$s_2(t) = -s_1(t - T/2)$$



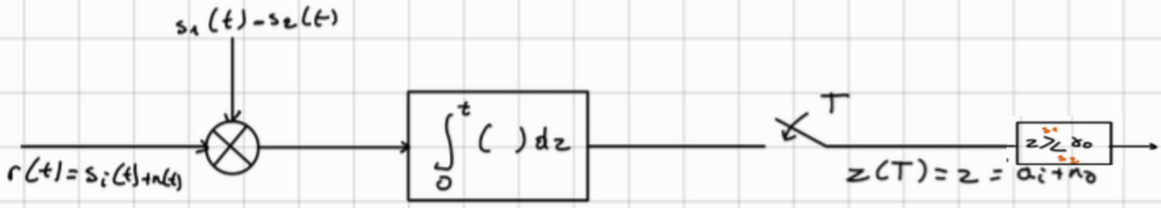
* $s_1(t)$ signal is transmitted for the bit "1"
* $s_2(t)$ signal is transmitted for the bit "0"

Find the bit error (BER) expression of this system over additive white gaussian channel (AWGN) for

a) $P(1) = P(0) = 1/2$

b) $P(1) = 1/4$, $P(0) = 3/4$

and plot it.



$$a_i = a_i(T) = \int_0^T s_i(t)\left[s_1(t) - s_2(t)\right] dt$$

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

$$a_1(T) = \int_0^{T/2} A^2 \sin^2\left(\frac{2\pi t}{T}\right) dt \qquad \boxed{\sin^2 x = \frac{1 - \cos(2x)}{2}}$$

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

$$= \frac{A^2}{2}\left[t - \frac{\sin\left(\frac{4\pi t}{T}\right)}{\frac{4\pi}{T}}\right]_0^{T/2}$$

$$= \frac{A^2}{2}\left[t - T\,\frac{\sin\left(\frac{4\pi t}{T}\right)}{4\pi}\right]_0^{T/2}$$

$$= \frac{A^2}{2}\left[\left(\frac{T}{2}-0\right) - \left[\frac{T\sin\left(\frac{\cancel{4}^{2}\pi}{\cancel{T}}\cdot\frac{\cancel{T}}{\cancel{2}}\right)}{4\pi} - \frac{T\sin(0)}{4\pi}\right]\right]$$

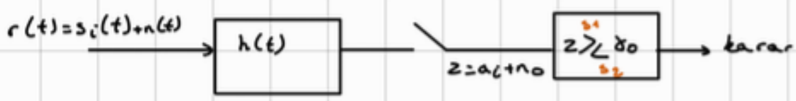
$$= \frac{A^2}{2}\left[\frac{T}{2} - \left[\cancelto{0}{\frac{T\sin(2\pi)}{4\pi}} - \cancelto{0}{\frac{T\sin(0)}{4\pi}}\right]\right]$$

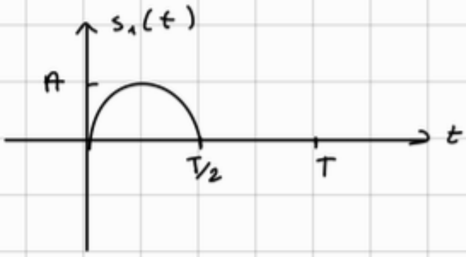
$$a_1(T) = \frac{A^2T}{4}$$

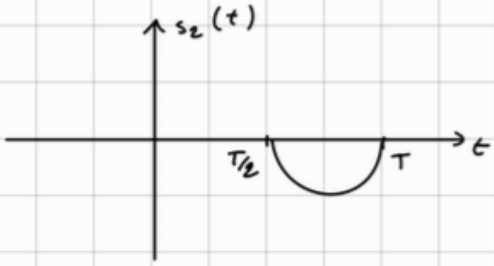
$$a_2(T) = \int_0^T s_2(t)\left[s_1(t) - s_2(t)\right]dt$$

$$a_2(T) = \int_0^T \cancelto{0}{s_1(t)s_2(t)} - \int_0^T s_2^2(t)\,dt = -\int_{T/2}^{T} s_2^2(t)\,dt$$

$$= -\int_0^{T/2} s_1^2(t)\,dt = -a_1(T) = -\frac{A^2T}{4}$$







$$E_{s_1} = \frac{A^2}{2}\cdot\frac{T}{2} = \frac{A^2T}{4} \qquad E_{s_2} = \frac{A^2}{2}\cdot\frac{T}{2} = \frac{A^2T}{4}$$

$E_b = E_{s_1}P(s_1) + E_{s_2}P(s_2)$ and $E_{s_1} = E_{s_2}$

$$E_b = E_{s_1}\big(\underbrace{P(s_1)+P(s_2)}_{1}\big) = E_{s_1} = \frac{A^2T}{4}$$

$$a_1(T) = \frac{A^2T}{4} \quad , \quad a_2(T) = -a_1(T)$$

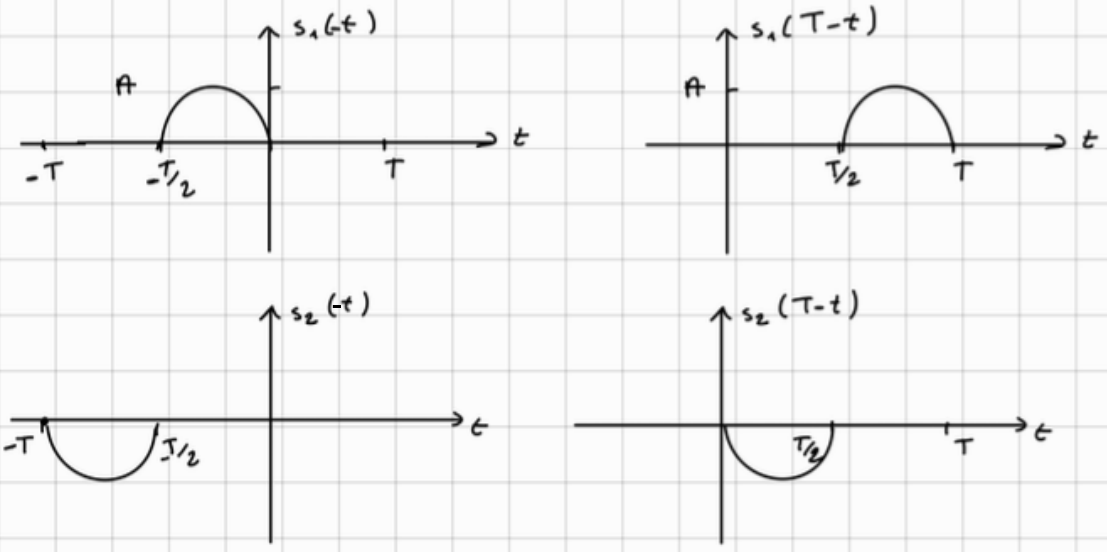
if $E_b = 1$ $\quad a_1(T) = 1$, $\quad a_2(T) = -1$

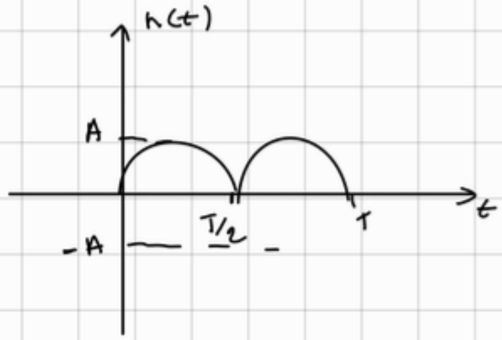
$$\gamma_0 = \frac{a_1+a_2}{2} = 0 \quad \text{when } P(s_1) = P(s_2) = \frac{1}{2}$$

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

$\sigma_0^2 = \frac{N_0}{2} E_h$ so we have to find E_h

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





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

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

GEBZE TEKNİK ÜNİVERSİTESİ

$$\text{if } P(s_1) = P(s_2) \Rightarrow P_b = Q\left(\frac{a_1 - a_2}{2\sigma_0}\right) = Q\left(\frac{\frac{A^2T}{4} - \left(-\frac{A^2T}{4}\right)}{2\frac{A}{2}\sqrt{N_0T}}\right) = Q\left(\frac{A^{\cancel{2}}T}{2\cancel{A}\sqrt{N_0T}}\right) =$$

$$= Q\left(\sqrt{\frac{A^2T}{4N_0}}\right) = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

$$\boxed{P(1) = P(0) = \frac{1}{2} \Rightarrow P_b = Q\left(\sqrt{\frac{E_b}{N_0}}\right)}$$

special case

$$\text{if } E_b = 1 \Rightarrow P_b = Q\left(\sqrt{\frac{1}{N_0}}\right)$$

Then if $P(1) = \frac{1}{4} = P(s_1)$, $P(0) = \frac{3}{4} = P(s_2)$

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

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

$$\Rightarrow \gamma_0 = \frac{\frac{N_0 A^2 T}{4}}{\frac{A^2T}{4} - \left(\frac{A^2T}{4}\right)} \ln \frac{\left(\frac{3}{\cancel{4}}\right)}{\frac{1}{\cancel{4}}} + \frac{\cancel{\frac{A^2T}{4} - \frac{A^2T}{4}}^{\to 0}}{2}$$

$$\Rightarrow \gamma_0 = \frac{\frac{N_0 A^2 T}{4}}{\frac{A^2T}{2}} \ln(3) = \frac{N_0}{2} \ln(3) \approx (0.549) N_0$$

$$P_b = \left[1 - Q\left(\frac{\frac{N_0\ln(3)}{2} - \frac{A^2T}{4}}{\frac{A}{2}\sqrt{N_0T}}\right)\right]\frac{1}{4} + Q\left(\frac{\frac{N_0\ln(3)}{2} - \left(-\frac{A^2T}{4}\right)}{\frac{A}{2}\sqrt{N_0T}}\right)\cdot\frac{3}{4}$$

$$\boxed{P_b = \left[1 - Q\left(\frac{\frac{N_0\ln(3)}{2} - \frac{A^2T}{4}}{\frac{A}{2}\sqrt{N_0T}}\right)\right]\frac{1}{4} + Q\left(\frac{\frac{N_0\ln(3)}{2} + \frac{A^2T}{4}}{\frac{A}{2}\sqrt{N_0T}}\right)\cdot\frac{3}{4}}$$

special case

$$E_b = \frac{A^2T}{4} \quad \text{if } E_b = 1 \quad \frac{A^2T}{4} = 1 \Rightarrow \frac{A}{2}\sqrt{T} = 1$$

$$\Rightarrow P_b = \left[1 - Q\left(\frac{\frac{N_0\ln(3)}{2} - 1}{\sqrt{N_0}}\right)\right]\frac{1}{4} + Q\left(\frac{\frac{N_0\ln 3}{2} + 1}{\sqrt{N_0}}\right)\frac{3}{4}$$

2 MATLAB for analyze theorical solutions and make system simulations.

2.1 Solution for equal probabilities P(1) = P(0) = ½

2.1.1 Plotting the theorical SNR vs PB curve in MATLAB

From theorical solution the Bit Error Rate Pb formula is funded as following:

$$P_B = Q\left(\sqrt{\frac{E_b}{N_0}}\right) \quad and \quad SNR = \frac{E_b}{N_0}$$

So P_B can be rewritten as following:

$$P_B = Q\left(\sqrt{SNR}\right) \quad and \quad SNR_{dB} = 10\log_{10} SNR$$

$$SNR = 10^{\frac{SNR_{dB}}{10}}$$

$$\Longrightarrow P_B = Q\left(\sqrt{10^{\frac{SNR_{dB}}{10}}}\right)$$

The SNR_{dB} vector is chosen as a vector from 0 to 14 with 1 dB steps. Then the Pb value is calculated and drawn using "semiology" function as shown in Figure 1. The y axis is limited to 10^{-6} value using "ylim" function.

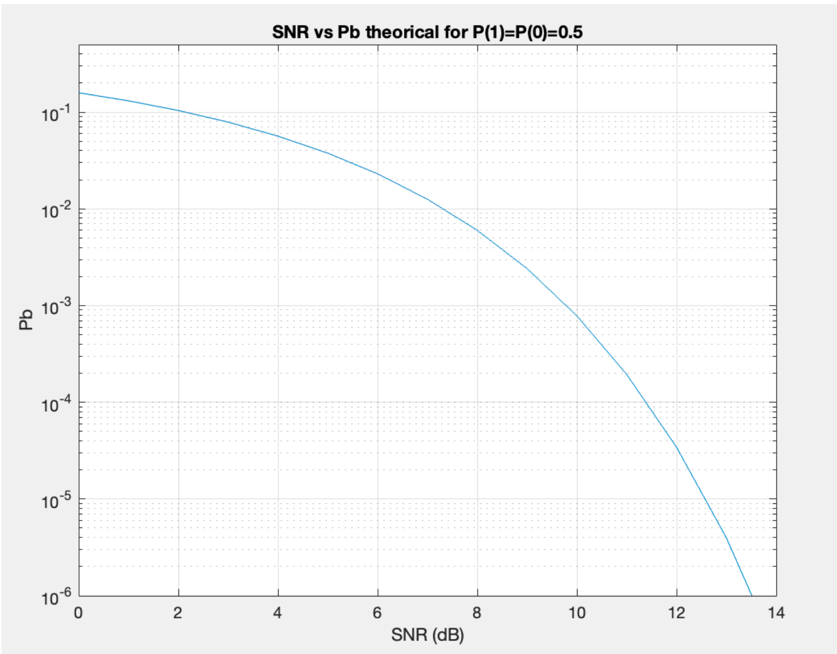


Figure 1: SNR vs Pb from theorical solution for P(1) = P(0) = ½

2.1.2 Simulation of the system and plotting SNR vs PB simulation

To simulate the Bit Error Rate, vector simulation is used. To make the simulation 10^{-8} bit generated with "***randi***" [1], it is a MATLAB function that generate a Uniformly distributed pseudorandom integers. Then if the bit is 1 a_i will be equal to a_1, and if the bit is 0 a_i will be qual to a_2. Then to find z the input of the comparator a gaussian noise added. z is formalized as following:

$$z = a_i + n$$

For each SNR value, a gaussian noise with different variance must be generated. The variance of the noise is found as following from the theorical solution.

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

To generate a gaussian noise with MATLAB ***"randn"*** [2] function is used, this function returns a random scalar vector drawn from the standard normal distribution. And as is known that in the standard normal distribution:

$$Standard\ Normal\ Distrubution = N(0,1)$$
$$so\ Mean = \mu = 0\ and\ Variance = \sigma^2 = 1$$

To change the variance the Variance properties can be used. If all values are scaled by a constant, the variance is scaled by the square of that constant:

$$Var(aX) = a^2 Var(X)$$
$$\text{So } N(0, \sigma_0^2) = \sqrt{\frac{N_0 A^2 T}{4}}\, N(0,1)$$

With this formula the gaussian noise for every SNR is obtained and added to a_i signal to obtain z signal. Then the signal is passed by a comparator with $gama = \gamma = 0$. İf $z > \gamma$ the output signal of the comparator $\hat{s}$ signal will be qual to "1" else it will be equal to "0". Than to find the bit Error the following formula can be used:

$$PEB = \frac{\sum |s - \hat{s}|}{Bit\ number}$$

because the signal is binary.

The SNR vs Pb (BER) curve obtained and drawn as shown at figure 2.

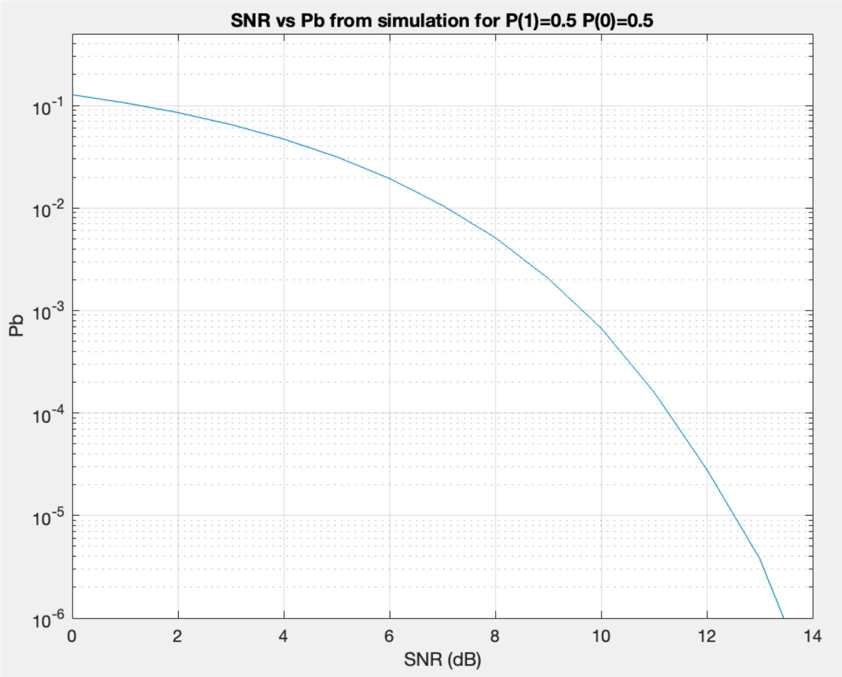


Figure 2: SNR vs Pb from simulation for P(1) = P(0) = ½

2.1.3 Comparison of Theorical and simulated BER Curves

The SNR vs BER curves obtained from Theorical solution and simulation plotted over for comparison as shown at Figure 3. It is clearly seen that the simulation matches the theoretical solution.

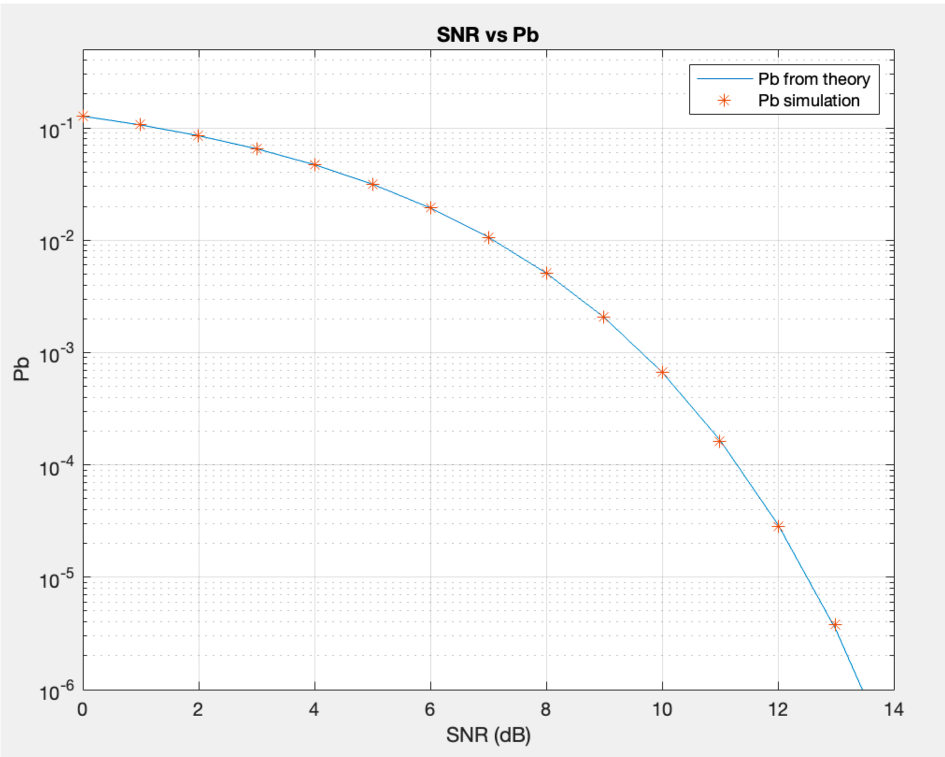


Figure 3: Comparison of SNR vs BER curves from theorical solution and simulation for P(0)=P(1)=1/2

2.1.4 MATLAB Code

The MATLAB code is used for this section is as follow.

```
A = 2;
T = 1;

s1 = @(t) A.*sin((2*pi*t)/T) .* (0<=t & t<=T/2);
s2 = @(t) -s1(t- T/2) .* (T/2<=t & t<=T);
Ts = 0.00001;
t = 0:Ts:T-Ts;
fs = 1/Ts;
figure;
% make subplot for the 2 signals
subplot(2,1,1);
plot(t,s1(t));
xlabel('time');
ylabel('s1(t)');
title('s1(t)');
grid on;

subplot(2,1,2);
plot(t,s2(t));
xlabel('time');
ylabel('s2(t)');
title('s2(t)');
grid on;

Eb = A^2*T/4;
% make a vector of SNR values from 0 to 11 dB
SNRdb = 0:1:14;
Pb =qfunc(sqrt(10.^(SNRdb/10)));
%%
%plot SNRdb vs Pb
figure;
semilogy(SNRdb,Pb);

ylim([10^-6,5*10^-1])
xlabel('SNR (dB)');
ylabel('Pb');
title('SNR vs Pb theorical for P(1)=P(0)=0.5');
grid on;
%%
% make random bits 100 thousand
bit_num = 10^8;
bits = randi([0 1],1,bit_num);
%calculate a1 and a2
a1 = A^2*T/4;
a2 = -a1;
%if the bit is 1 then z = a1 + n if 0 then z = a2 + n
% if bit in bits is 1 then ai(T) = a1 else ai(T) = a2
ai = zeros(1,bit_num);
ai(bits == 1) = a1;
ai(bits == 0) = a2;

%let's calculate z = ai + n by adding gaussian noise for different SNR
%values
z = zeros(1,bit_num);
shat = zeros(1,bit_num);
Pb_sim = zeros(1,length(SNRdb));
comparator_gama = 0;
comparator_gamas = comparator_gama * ones(1,bit_num);
%find N0's value for given SNR
N0 = Eb.*(10.^(-SNRdb/10));
for i = 1:length(N0)

    z = ai + (sqrt(N0(i)) ).*randn(1,bit_num);
    %find shat by using comparator
```

```
    shat = double(z>comparator_gamas);
    %calculate Pb_sim
    Pb_sim(i) = sum(abs(bits-shat))/bit_num;


end
%%
%plot SNRdb vs Pb
figure;
semilogy(SNRdb,Pb_sim);

ylim([10^-6,5*10^-1])
xlabel('SNR (dB)');
ylabel('Pb');
title('SNR vs Pb from simulation for P(1)=0.5 P(0)=0.5');
grid on;

%plot the two graphs
figure;
semilogy(SNRdb,Pb);
hold on;
semilogy(SNRdb,Pb_sim,"*");

ylim([10^-6,5*10^-1])
legend('Pb from theory','Pb simulation')
xlabel('SNR (dB)');
ylabel('Pb');
title('SNR vs Pb');
grid on;
%%
%% For different Probobilities


A = 2;
T = 1;

s1 = @(t) A.*sin((2*pi*t)/T) .* (0<=t & t<=T/2);
s2 = @(t) -s1(t- T/2) .* (T/2<=t & t<=T);
Ts = 0.00001;
t = 0:Ts:T-Ts;
fs = 1/Ts;
figure;
% make subplot for the 2 signals
subplot(2,1,1);
plot(t,s1(t));
xlabel('time');
ylabel('s1(t)');
title('s1(t)');
grid on;

subplot(2,1,2);
plot(t,s2(t));
xlabel('time');
ylabel('s2(t)');
title('s2(t)');
grid on;

Eb = A^2*T/4;
% make a vector of SNR values from 0 to 11 dB
SNRdb = 0:1:14;
%calculate a1 and a2
a1 = A^2*T/4;
a2 = -a1;
N0 = Eb.*(10.^(-SNRdb/10));
```

```
gama0 = 0.549.*N0;
sigma0 = A.*sqrt(N0.*T)/2;


Pb =0.25*(1-qfunc((gama0 - a1) ./sigma0))+...
    0.75*qfunc( (gama0 - a2) ./sigma0 ) ;
%%
%plot SNRdb vs Pb
figure;
semilogy(SNRdb,Pb);
xlabel('SNR (dB)');
ylabel('Pb');
title('SNR vs Pb from theoric for P(1)=0.25 P(0)=0.75');
grid on;
%%

% make random bits 100 thousand
bit_num = 10^8;
random_numbers = rand(1,bit_num);
P1 = 1/4;
bits = random_numbers< P1;
%calculate a1 and a2
a1 = A^2*T/4;
a2 = -a1;
%if the bit is 1 then z = a1 + n if 0 then z = a2 + n
% if bit in bits is 1 then ai(T) = a1 else ai(T) = a2
ai = zeros(1,bit_num);
ai(bits == 1) = a1;
ai(bits == 0) = a2;

%let's calculate z = ai + n by adding gaussian noise for different SNR
%values
z = zeros(1,bit_num);
shat = zeros(1,bit_num);
Pb_sim = zeros(1,length(SNRdb));
%find N0's value for given SNR
N0 = Eb.*(10.^(-SNRdb/10));

for i = 1:length(N0)
    comparator_gamas = 0.549*N0(i) * ones(1,bit_num);


     z = ai + (sqrt(N0(i)*(A^2)*T/4) ).*randn(1,bit_num);
    %find shat by using comparator

    shat = double(z>comparator_gamas);
    %calculate Pb_sim
    Pb_sim(i) = sum(abs(bits-shat))/bit_num;


end
```

2.2 Solution for different probabilities P(1) =¼ P(0) = ¾

2.2.1 Plotting the theorical SNR vs PB curve in MATLAB

To plot the graph from theorical solution the following formulas funded in the theorical solution is used (The bit energy is taken as $E_b = 1$):

$$a_1 = \frac{A^2\,T}{4};$$
$$a_2 = -a_1;$$
$$N_0 = E_b\left(10^{-\frac{SNRdb}{10}}\right)$$

$$gama = \gamma = 0.549\,N_0$$

$$sigma = \sigma_0 = A\frac{\sqrt{(N_0 T)}}{2}$$

$$P_b = 0.25\left(1 - Q\left(\frac{(\gamma - a_1)}{\sigma_0}\right)\right) + 0.75\,Q\left(\frac{(\gamma - a_2)}{\sigma_0}\right)$$

And the SNR vs Pb graph is plotted as shown at Figure 4.

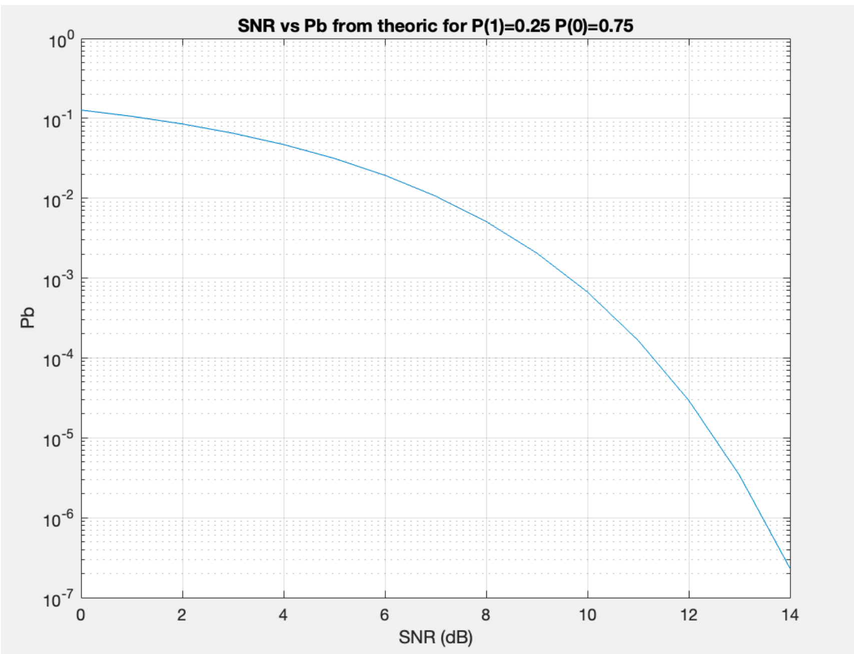


Figure 4:SNR vs Pb from theorical solution for P(1) =0.25 P(0) =0.75

2.2.2 Simulation of the system and plotting SNR vs PB simulation

To simulate the Bit Error Rate, vector simulation is used. To make the simulation 10^{-8} bit has to be generated with different probabilities. To do this with ***"rand"*** [3], it is a MATLAB function that generate a Uniformly distributed random numbers, a numbers section between 0 and 1 its generated and to set the probability of bit 1 to 0.25 if the number generated is less then 0.25 this number is set to 1 else to zero.

Then if the bit is 1 a_i will be equal to a_1, and if the bit is 0 a_i will be qual to a_2. Then to find z the input of the comparator a gaussian noise added. z is formalized as following:

$$z = a_i + n$$

For each SNR value, a gaussian noise with different variance must be generated. The variance of the noise is found as following from the theorical solution.

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

To generate a gaussian noise with MATLAB ***"randn"*** function is used, this function returns a random scalar vector drawn from the standard normal distribution. And as is known that in the standard normal distribution:

$$Standard\ Normal\ Distrubution = N(0,1)$$
$$so\ Mean = \mu = 0\ and\ Variance = \sigma^2 = 1$$

To change the variance the Variance properties can be used. If all values are scaled by a constant, the variance is scaled by the square of that constant:

$$Var(aX) = a^2 Var(X)$$
$$\text{So } N(0, \sigma_0^2) = \sqrt{\frac{N_0 A^2 T}{4}}\, N(0,1)$$

With this formula the gaussian noise for every SNR is obtained and added to a_i signal to obtain z signal. Then the signal is passed by a comparator with $gama = \gamma = 0.549\ N_0$. If $z > \gamma$ the output signal of the comparator $\hat{s}$ signal will be qual to "1" else it will be equal to "0". Than to find the bit Error the following formula can be used:

$$PEB = \frac{\sum |\, s - \hat{s}|}{Bit\ number}$$

because the signal is binary.

The SNR vs Pb (BER) curve obtained and drawn as shown at Figure 5.

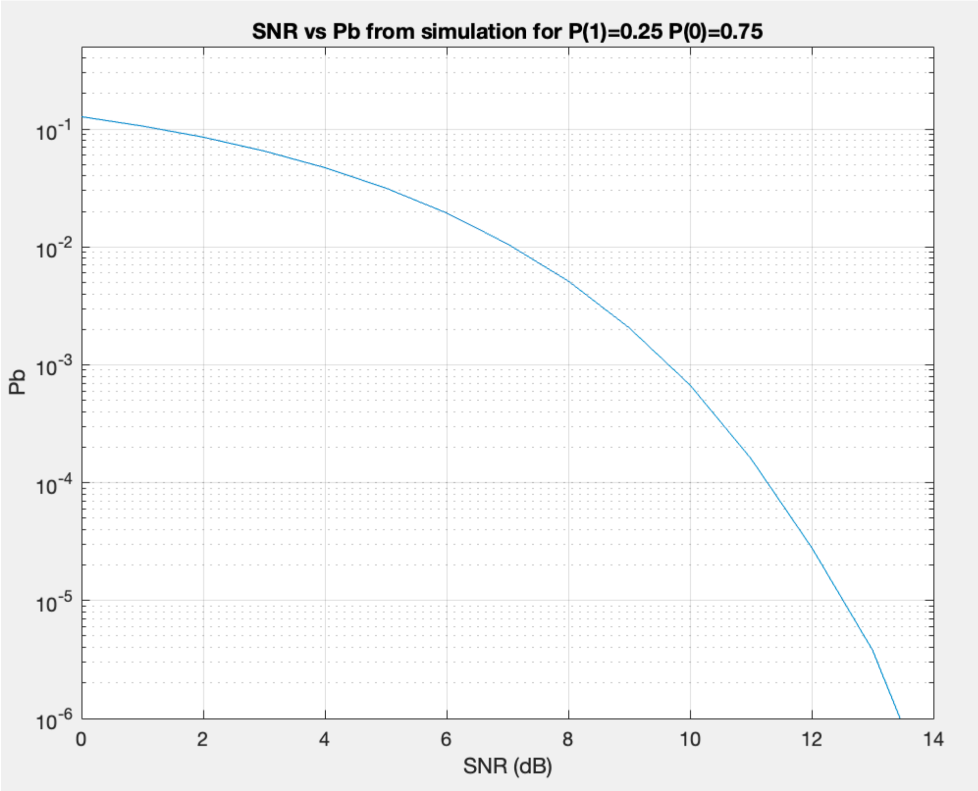


Figure 5: SNR vs Pb from simulation for P(1) =0.25 P(0) =0.75

2.2.3 Comparison of Theorical and simulated BER Curves

The SNR vs BER curves obtained from Theorical solution and simulation for P(1)= 0.25 and P(0)= 0.75 plotted over for comparison as shown at Figure 6. It is clearly seen that the simulation matches the theoretical solution.

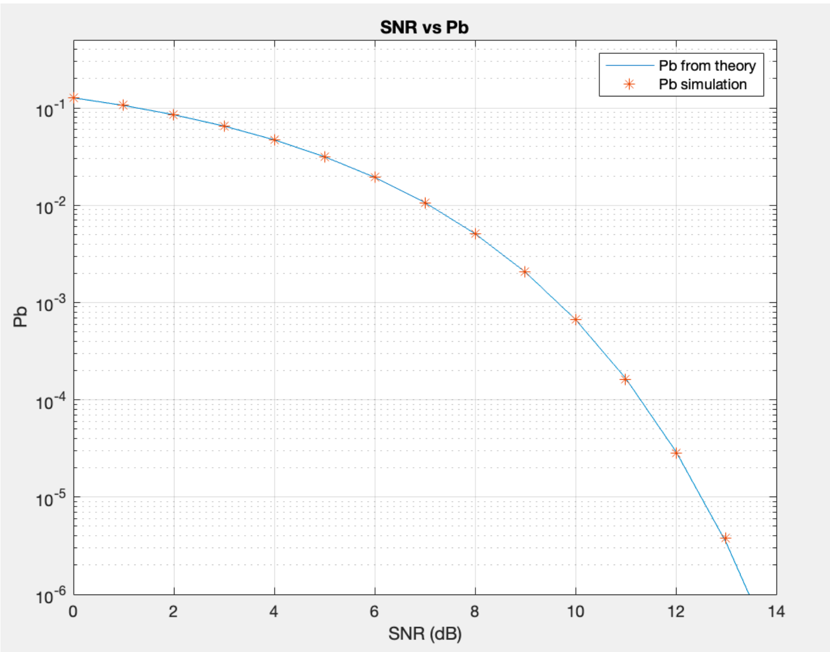


Figure 6:Comparison of SNR vs BER curves from theorical solution and simulation for P(1)=0.25 P(0)=0.75

2.2.4 MATLAB Code

The MATLAB code is used for this section is as follow.

```
%% For different Probobilities


A = 2;
T = 1;

s1 = @(t) A.*sin((2*pi*t)/T) .* (0<=t & t<=T/2);
s2 = @(t) -s1(t- T/2) .* (T/2<=t & t<=T);
Ts = 0.00001;
t = 0:Ts:T-Ts;
fs = 1/Ts;
figure;
% make subplot for the 2 signals
subplot(2,1,1);
plot(t,s1(t));
xlabel('time');
ylabel('s1(t)');
title('s1(t)');
grid on;

subplot(2,1,2);
plot(t,s2(t));
xlabel('time');
ylabel('s2(t)');
title('s2(t)');
grid on;

Eb = A^2*T/4;
% make a vector of SNR values from 0 to 11 dB
SNRdb = 0:1:14;
%calculate a1 and a2
a1 = A^2*T/4;
a2 = -a1;
N0 = Eb.*(10.^(-SNRdb/10));

gama0 = 0.549.*N0;
sigma0 = A.*sqrt(N0.*T)/2;


Pb =0.25*(1-qfunc((gama0 - a1) ./sigma0))+...
    0.75*qfunc( (gama0 - a2) ./sigma0 ) ;
%%
%plot SNRdb vs Pb
figure;
semilogy(SNRdb,Pb);
xlabel('SNR (dB)');
ylabel('Pb');
title('SNR vs Pb from theoric for P(1)=0.25 P(0)=0.75');
grid on;
%%

% make random bits 100 thousand
bit_num = 10^8;
random_numbers = rand(1,bit_num);
P1 = 1/4;
bits = random_numbers< P1;
%calculate a1 and a2
a1 = A^2*T/4;
a2 = -a1;
%if the bit is 1 then z = a1 + n if 0 then z = a2 + n
```

```
% if bit in bits is 1 then ai(T) = a1 else ai(T) = a2
ai = zeros(1,bit_num);
ai(bits == 1) = a1;
ai(bits == 0) = a2;

%let's calculate z = ai + n by adding gaussian noise for different SNR
%values
z = zeros(1,bit_num);
shat = zeros(1,bit_num);
Pb_sim = zeros(1,length(SNRdb));
%find N0's value for given SNR
N0 = Eb.*(10.^(-SNRdb/10));

for i = 1:length(N0)
    comparator_gamas = 0.549*N0(i) * ones(1,bit_num);


     z = ai + (sqrt(N0(i)*(A^2)*T/4) ).*randn(1,bit_num);
    %find shat by using comparator

    shat = double(z>comparator_gamas);
    %calculate Pb_sim
    Pb_sim(i) = sum(abs(bits-shat))/bit_num;


end
%%
%plot SNRdb vs Pb
figure;
semilogy(SNRdb,Pb_sim);
ylim([10^-6,5*10^-1])
xlabel('SNR (dB)');
ylabel('Pb');
title('SNR vs Pb from simulation for P(1)=0.25 P(0)=0.75');
grid on;

%plot the two graphs
figure;
semilogy(SNRdb,Pb);
hold on;
semilogy(SNRdb,Pb_sim,'*');
ylim([10^-6,5*10^-1])
legend('Pb from theory','Pb simulation')
xlabel('SNR (dB)');
ylabel('Pb');
title('SNR vs Pb');
grid on;
```

3 References

1- MathWorks documentation randi function :
https://www.mathworks.com/help/matlab/ref/randi.html
2- MathWorks documentation randn function:
https://www.mathworks.com/help/matlab/ref/randn.html
3- MathWorks documentation rand function:
https://www.mathworks.com/help/matlab/ref/rand.html