Alexandria University
Faculty of Engineering
Electrical and Electronics Engineering Department
Fall semester, 2021/2022

جامعة الإسكندرية
كلية الهندسة
قسم الهندسة الكهربية
الفصل الدراسى الأول, 2022/2021

DIGITAL COMMUNICATIONS LAB

سيف الدين زكريا محمد	19015813
محمد عيد عبد المجيد	19016463
تهاني ياسر مرسى	18010503

Alexandria University
Faculty of Engineering
Electrical and Electronics Engineering Department
Fall semester, 2021/2022

جامعة الإسكندرية
كلية الهندسة
قسم الهندسة الكهربية
الفصل الدراسى الأول, 2022/2021

Experiment 2

Basics of BER calculations and channel models

Introduction

This is the first experiment in the Digital Communications Lab. In this experiment, we study the most basic method for assessing the performance of digital communication systems, the Bit *Error Rate*. We also study simple models for communication channels and their effect on the communication system from the perspective of bit error rate.

About the lab

The digital communications lab consists of a set of experiments which introduce essential digital communication concepts to students. The lab consists mainly of MATLAB-based experiments in which the students complete the required tasks of the experiment by writing MATLAB codes. Some experiments of the lab may consist of hardware-based experiments which are performed in the digital communication laboratory on campus.

Background

The most basic target of the study of digital communications is to understand digital communication systems and how digital information can be conveyed from a source or transmitter to a destination or receiver over a channel. Depending on the communication systems, channels can be wired circuits, wireless channels, satellite channels and so on. The study of digital communications begins by transforming the digital communication system into an equivalent mathematical model, and then attempts to design transmitters and receivers which achieves the target of information transmission over the channel in an efficient manner.

Figure 1 shows an example of a digital communication system. The goal of the transmitter and receiver is to deliver the digital data from the source to the sink in the best way possible. There are several ways to define what *best* mean: one of the most common and most important methods to assess the performance of a communication system is the *Bit Error Rate (BER)*.

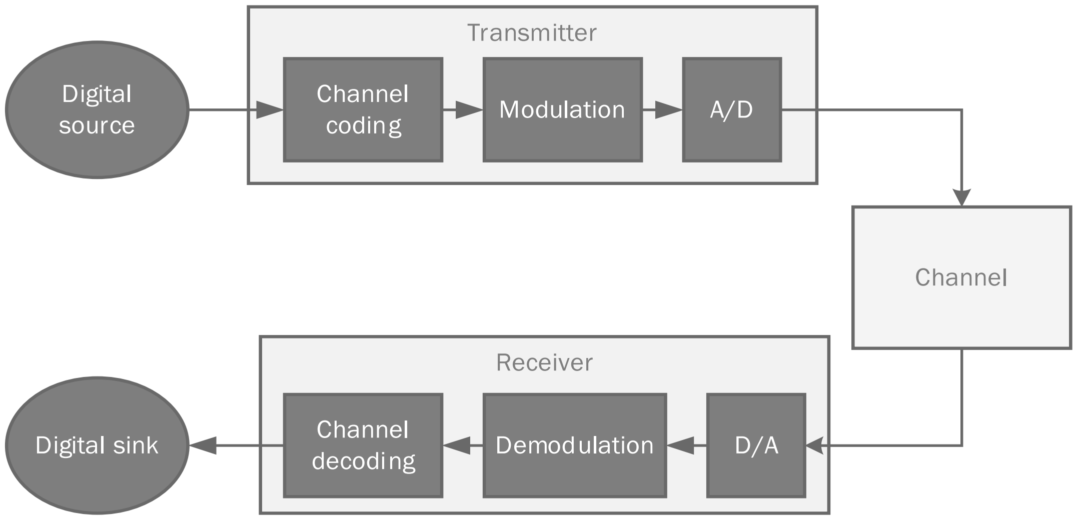


Figure 1 An example of a digital communication system

Bit Error Rate (BER): the rate of error occurrences among an output sequence of bits corresponding to an input sequence of bits.

An empirical method of computing the BER in a communication system is as follows:

1. Generate a sequence of N bits at the input side.
2. Pass the sequence of input bits through the system to receive a corresponding output sequence.
3. Count the number of errors in the output sequence by comparing it to the input sequence; call that number of errors E
4. The BER is given by $BER = \frac{E}{N}$

In this experiment, we will compute the BER of different digital communication systems. These systems differ in their respective channel models and therefore their corresponding transmitter and receiver designs.

Experiment

Part 1 (3 Marks)

In this part, we consider a very simple digital communication system, in which the channel takes as input binary digits $b = \{0,1\}$, and produces the corresponding output according to the following equation.

$$y = \{b \ \ with\, probability\ 1 - p\ \underline{b} \ \ with\, probability\ p$$

The channel described above simply flips the input bit with probability p or passes the input bit unchanged with probability $1-p$; this channel is referred to as the *Binary Symmetric Channel (BSC)*. The system is shown in Figure 2. In this system, we assume that the transmitter takes the input bits coming from the source and passes them unchanged to the channel (i.e., the transmitter does nothing). However, we would like to investigate how the receiver can be designed to produce a good BER.

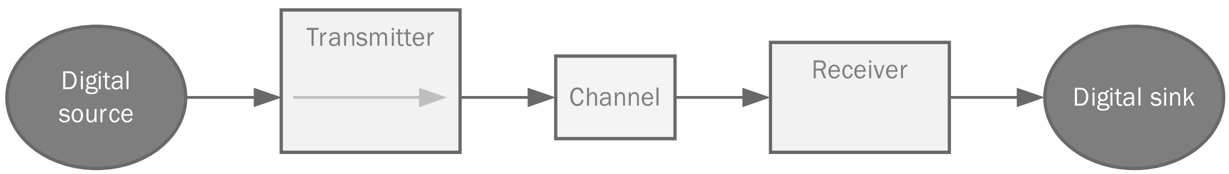


Figure 2 A digital communication system with a Binary Symmetric Channel

Your goal in this task is to design the receiver. You know that the channel takes the data, flips it randomly (with probability p) and gives you the output. What would the receiver do with that output?

Think about the following two receivers and say what is the expected performance of these receivers. As a hint to start, these two receivers are not very good.

Example 1: the receiver gives a 0 bit as output. This output does not depend at all on what the channel is giving out.

Questions	
What is the corresponding BER for that receiver? You do not need to implement it in the m-file to answer.	**It has a BER of no of ones over the total input number of samples. This is because the receiver ignores the channel output and always outputs 0, so it will make an error if the channel flips the transmitted bit from 0 to 1 with a probability of p.**
What is the reason behind the performance of this receiver?	**It does not use any information about the channel output to make its decision. It ignores the noisy channel completely and makes a fixed decision, which is not related to the received signal. So, it has no ability to correct the errors introduced by the channel and will make errors with probability p**

Example 2: the receiver gives random output, i.e., 0s and 1s with a probability of 0.5. Again, this output is not based on what the channel is giving out.

Questions	
What is the corresponding BER for that receiver? You do not need to implement it in the m-file to answer.	**It has a BER of 0.5. This is because the receiver's decision is independent of the channel output, and it makes an error with probability 0.5 for each bit transmitted.**
What is the reason behind the performance of this receiver?	**It uses a random process to decide the transmitted bit, which has no relation to the channel output. So, it has no ability to correct the errors introduced by the channel**

The above two receivers are examples of receivers which clearly would not be considered as good receivers from a BER perspective (why?). In the following part of the experiment, you would design the best receiver and assess its performance by computing the corresponding BER.

EXP. ***Complete PART 1 in the experiment M-file*** **Lab1_script.m** ***and the missing implementation of all included functions. Then answer the following questions:***

Questions	
What is the corresponding BER for receivers 1 and 2 above? You do not need to implement the two receivers to answer.	**They have a high BER compared to my designed receiver.**
What is the reason behind the performance of these two receivers?	**This receivers' poor performance is due to its complete ignorance of the channel's behavior. The receivers are not even attempting to make use of the information it receives from the channel.**
What is the BER of the best receiver?	`Bit error rate (BER) for simple BSC channel with p = 0.2 is 0.1999` **It is 1.9 and less than the error of the channel**

Part 1-a (2 Marks) In this part, we study the impact of the BSC channel parameter p on the BER of the digital communication system. Namely, we vary the value of p from 0 to 1, and for each value of p we compute the corresponding BER, we save these values in an array, then, later in Part 3-a, plot the values of BER versus their corresponding parameter value p.

EXP. ***Complete PART 1-a in the experiment M-file*** **Lab1_script.m*****. The final figure containing the required plot will be generated at the end of Part 3-a of the experiment.***

Part 2 (3 Marks)

In this part, we again consider the system proposed in Figure 2 but we try to improve the transmitter a bit. Namely, the transmitter works as follows: for each input bit b, the transmitter generates a set of 5 copies of the bit b which are then passed sequentially through the channel. Note that this behavior leads to the increase in the number of bits being passed through the channel (is that good or bad?). The system is shown in Figure 3. For this transmitter, the receiver expects to receive a sequence of 5 channel outputs, all corresponding to the same input bit. Therefore, we expect that the receiver can use these outputs for a better decoding performance. In this part, we investigate how to design the best receiver and the corresponding BER performance.

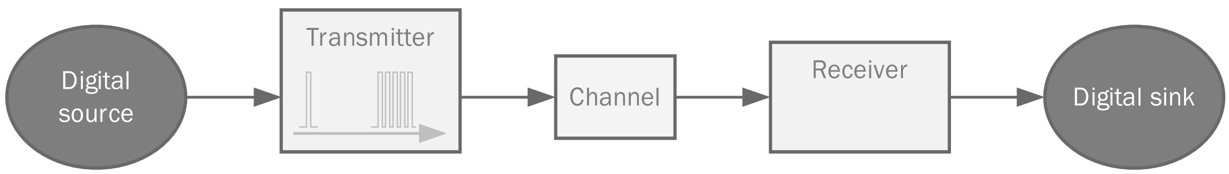


Figure 3 A digital communication system with a Binary Symmetric Channel and a modified transmitter

EXP. ***Complete PART 2 in the experiment M-file*** **Lab1_script.m** ***and the missing implementation of all included functions. Then answer the following questions:***

Questions	
What is the BER of the best receiver?	`Bit error rate (BER) for part 2 with p = 0.2 is 0.0602`
What is the expected (theoretical) BER if the number of r epetitions is increase to 10?	**The theoretical BER for 10 repetitions is expected to be lower than that of 5 repetitions, as more repetitions increase the probability of correct reception of the bit**
What is the cost/downside of using the transmitter in Part 2?	**The cost of increased bandwidth and transmission time must be considered when deciding on the appropriate number of repetitions to use in a communication system.**

Part 2-a (2 Marks)

Similar to Part 1-a, in this part, we study the impact of the BSC channel parameter p on the BER of the digital communication system in Part 2.

EXP. ***Complete PART 2-a in the experiment M-file*** **Lab1_script.m.** ***The final figure containing the required plot will be generated at the end of Part 3-a of the experiment.***

Part 3 (3 Marks)

In part 3, we consider the same system in Part 2. However, the channel in Part 3 generates correlated outputs among the 5 transmitter outputs that correspond to the same input bit. For example, for a 0 input bit to the transmitter and a corresponding five copies of the bit 0, the channel output either generates a set of five 0's with probability $1-p$ or a set of five 1's with probability p. In this case, we investigate the design of the best receiver and the corresponding BER.

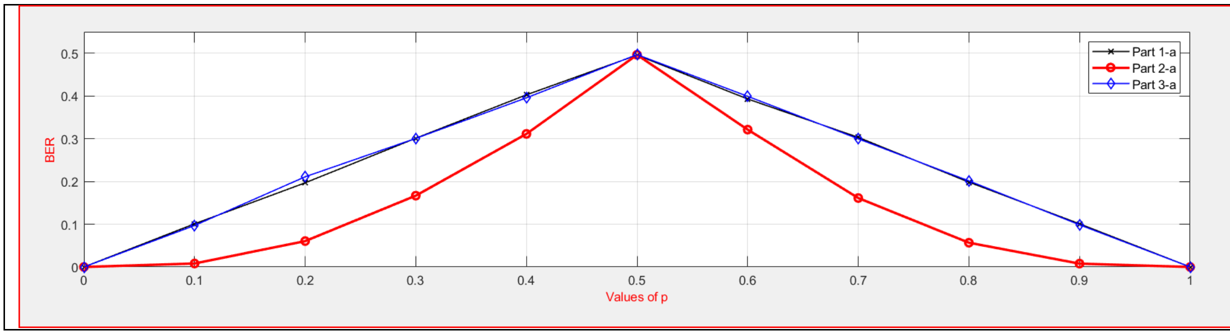
EXP. ***Complete PART 3 in the experiment M-file*** **Lab1_script.m** ***and the missing implementation of all included functions. Then answer the following questions:***

Questions	
What is the BER of the best receiver?	**The BER of receiver f part 2 is the best since it is 0.06**
What is the reason behind such a performance?	**The repetition of the bits in transmitter**

Part 3-a (2 Marks)

Finally, we study the impact of the BSC channel parameter p on the BER of the digital communication system in Part 3.

EXP. ***Complete PART 3-a in the experiment M-file*** **Lab1_script.m*****. The final figure containing the plots from all three parts can now be generated. Please add the generated plot in the box below.***

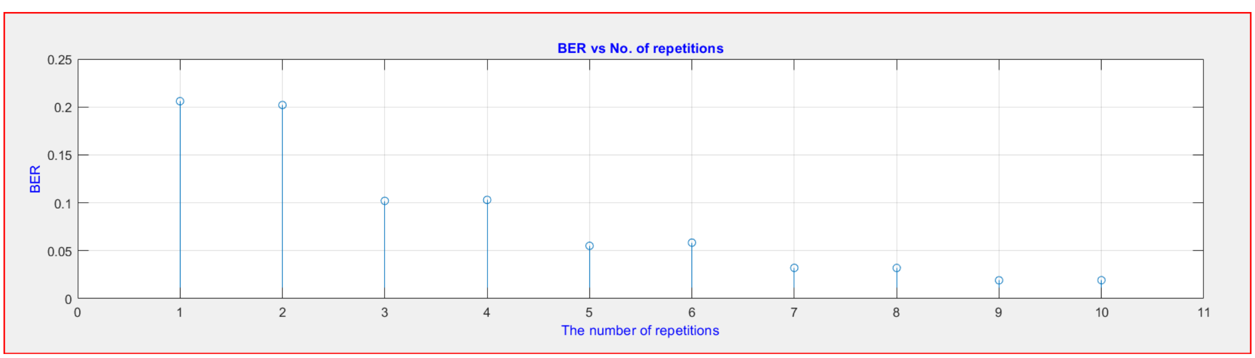


Questions	
Which of the three systems have the best performance in terms of BER?	**The system in part 2 has the best performance in terms of BER**
If the receiver you designed in any of the previous parts attain a BER more than 0.5, how can it be changed to attain a maximum of 0.5 BER?	**By flipping all the received bits so the BER will be 1-$(BER)_{old}$**

Part 4 (8 Marks)

In this part, we go back to the system considered in Part 2, namely the system with a transmitter which generated a set of 5 repetitions to the input bit. Now, we would like to investigate the effect of changing the number of repetitions on the decoding performance. You need to generate a figure where the x-axis shows the number of repetitions, and the y-axis shows the corresponding BER. In this part, you can consider p = 2.

EXP. ***Write your own code in PART 4 in the experiment M-file* Lab1_script.m*. Your code should generate a figure as described in the discussion above.***



Lab1_scribt

```
clear,clc,close all;
%% Simulation parameters
N_bits = 10000; % Total number of bits
p      = 0.2;   % Channel parameter (probability of bit flipping)
%% Part 1: BER for simple BSC channel
% Generate a bit sequence
bit_seq = GenerateBits(N_bits); % IMPLEMENT THIS: Generate a sequence of bits equal to the
total number of bits

% Pass the bit sequence through the channel
rec_sample_seq = BSC(bit_seq,1,p);  % Generate the received samples after passing through
the bit flipping channel

% Decode bits from received bit sequence
rec_bit_seq = DecodeBitsFromSamples(rec_sample_seq,'part_1');  % IMPLEMENT THIS: Decode the
received bits

% Compute the BER
BER_case_1 = ComputeBER(bit_seq,rec_bit_seq); % IMPLEMENT THIS: Calculate the bit error rate

% Display the BER
disp(['Bit error rate (BER) for part 1 with p = ',num2str(p),' is ',num2str(BER_case_1)]);
%% Part 1-a: Effect of bit flipping probability on BER
% GOAL: Make a plot for the BER versus different values of the channel
% parameter p

p_vect          = 0:0.1:1;              % Use this vector to extract different values of p
in your code
BER_case_1_vec  = zeros(size(p_vect));  % Use this vector to store the resultant BER

%%% WRITE YOUR CODE HERE
for p_index = 1:length(p_vect)
    rec_sample_seq = BSC(bit_seq,1,p_vect(p_index));
    rec_bit_seq = DecodeBitsFromSamples(rec_sample_seq,'part_1');
    if ComputeBER(bit_seq,rec_bit_seq)>0.5
        flippedArray = bitxor(rec_bit_seq, ones(size(rec_bit_seq)));
        rec_bit_seq = flippedArray;
    end
    BER_case_1_vec(p_index) = ComputeBER(bit_seq,rec_bit_seq);
end
%%%

%% Part 2: BER for simple bit-flipping channel with multiple samples

% System parameters
fs  = 5;    % Number of samples per symbol (bit)

% Generate a bit sequence
bit_seq = GenerateBits(N_bits); % Generate a sequence of bits equal to the total number of
bits

% Generate samples from bits
sample_seq = GenerateSamples(bit_seq,fs); % IMPLEMENT THIS: Generate a sequence of samples
for each bit

% Pass the sample sequence through the channel
rec_sample_seq = BSC(sample_seq,fs,p);   % Generate the received samples after passing
through the bit flipping channel

% Decode bits from received bit sequence
rec_bit_seq = DecodeBitsFromSamples(rec_sample_seq,'part_2',fs);    % IMPLEMENT THIS: Decode
the received bits

% Compute the BER
BER_case_2 = ComputeBER(bit_seq,rec_bit_seq);   % Calculate the bit error rate
disp(['Bit error rate (BER) for part 2 with p = ',num2str(p),' is ',num2str(BER_case_2)]);
```

```
%% Part 2-a: Effect of bit flipping probability on BER
% GOAL: Make a plot for the BER versus different values of the channel
% parameter p

p_vect          = 0:0.1:1;              % Use this vector to extract different values of p in
your code
BER_case_2_vec  = zeros(size(p_vect));  % Use this vector to store the resultant BER

%%% WRITE YOUR CODE HERE
for p_index = 1:length(p_vect)
    rec_sample_seq = BSC(sample_seq,fs,p_vect(p_index));
    rec_bit_seq = DecodeBitsFromSamples(rec_sample_seq,'part_2',fs);
    if ComputeBER(bit_seq,rec_bit_seq)>0.5
        flippedArray = bitxor(rec_bit_seq, ones(size(rec_bit_seq)));
        rec_bit_seq = flippedArray;
    end
    BER_case_2_vec(p_index) = ComputeBER(bit_seq,rec_bit_seq);
end
%%%

%% Part 3: BER for simple bit-flipping channel with multiple samples and correlated channel

% Generate a bit sequence
bit_seq = GenerateBits(N_bits); % Generate a sequence of bits equal to the total number of
bits

% Generate samples from bits
sample_seq = GenerateSamples(bit_seq,fs); % Generate a sequence of samples for each bit

% Pass the sample sequence through the channel
rec_sample_seq  = BSC(sample_seq,fs,p,'correlated'); % Generate the received samples after
passing through the bit flipping channel

% Decode bits from received bit sequence
rec_bit_seq = DecodeBitsFromSamples(rec_sample_seq,'part_3',fs);    % IMPLEMENT THIS: Decode
the received bits

% Compute the BER
BER_case_3 = ComputeBER(bit_seq,rec_bit_seq);  % Calculate the bit error rate
disp(['Bit error rate (BER) for part 3 with p = ',num2str(p),' is ',num2str(BER_case_3)]);

%% Part 3-a: Effect of bit flipping probability on BER
% GOAL: Make a plot for the BER versus different values of the channel
% parameter p

p_vect          = 0:0.1:1;              % Use this vector to extract different values of p in
your code
BER_case_3_vec  = zeros(size(p_vect));  % Use this vector to store the resultant BER

for p_index = 1:length(p_vect)
    rec_sample_seq = BSC(sample_seq,fs,p_vect(p_index),'correlated');
    rec_bit_seq = DecodeBitsFromSamples(rec_sample_seq,'part_3',fs);
    if ComputeBER(bit_seq,rec_bit_seq)>0.5
        flippedArray = bitxor(rec_bit_seq, ones(size(rec_bit_seq)));
        rec_bit_seq = flippedArray;
    end
    BER_case_3_vec(p_index) = ComputeBER(bit_seq,rec_bit_seq);
end
% Plotting results
figure
subplot(211)
plot(p_vect,BER_case_1_vec,'x-k','linewidth',1); hold on;
plot(p_vect,BER_case_2_vec,'o-r','linewidth',2); hold on;
plot(p_vect,BER_case_3_vec,'d-b','linewidth',1); hold on;
grid on
xlabel('\color{red}Values of p','fontsize',10)
ylabel('\color{red}BER','fontsize',10),ylim([0 0.55])
legend('Part 1-a','Part 2-a','Part 3-a','fontsize',10)
```

```
%% Part 4: Effect of number of repetitions on BER
% GOAL: Make a plot for the BER versus the number of repetitions used in
% the transmitter of part 2
% There is no template code for this part. Please write your own complete
% code here. You can re-use any of the codes in the previous parts
f_vec = 1:10;
% Generate a bit sequence
bit_seq = GenerateBits(N_bits); % Generate a sequence of bits equal to the total number of
bits
% Generate samples from bits
BER_case_4 = zeros(1,10);
for i =1:length(f_vec)
    sample_seq = GenerateSamples(bit_seq,f_vec(i)); % IMPLEMENT THIS: Generate a sequence of
samples for each bit

    % Pass the sample sequence through the channel
    rec_sample_seq = BSC(sample_seq,f_vec(i),p);   % Generate the received samples after
passing through the bit flipping channel

    % Decode bits from received bit sequence
    rec_bit_seq = DecodeBitsFromSamples(rec_sample_seq,'part_2',f_vec(i));    % IMPLEMENT
THIS: Decode the received bits

    % Compute the BER
    BER_case_4(i) = ComputeBER(bit_seq,rec_bit_seq);   % Calculate the bit error rate
    disp(['Bit error rate (BER) for part 4 with p = ',num2str(p),' is
',num2str(BER_case_4(i))]);
end
subplot(212)
stem(f_vec,BER_case_4),title('\color{blue}BER vs No. of repetitions'),xlim([0
11]),xlabel('\color{blue}The number of repetitions'),ylim([0
0.25]),ylabel("\color{blue}BER"),grid on
```

GenerateBits

```
trans_samples = 1000;
bits = randi([0, 1], [1, N_bits+trans_samples]);
bit_seq = bits(1001:end);
```

GenerateSamples

```
sample_seq = zeros(size(bit_seq*fs));
% Define the pulse shape
pulse_shape = ones(1,fs);
% Map each bit to a specific amplitude level and generate the waveform
for i = 1:length(bit_seq)
    if bit_seq(i) == 0
        sample_seq((i-1)*fs+1:i*fs) = 0;
    else
        sample_seq((i-1)*fs+1:i*fs) = pulse_shape;
    end
end
```

DecodeBitsFromSamples

```
case 'part_1'
        threshold = 0.5;
        rec_bit_seq = (rec_sample_seq > threshold);
    case 'part_2'
        % Reshape the array into blocks of fs samples
        block_samples = reshape(rec_sample_seq, fs, []);
        % Count the number of ones in each block
        block_counts = sum(block_samples, 1);
        rec_bit_seq = zeros(1,length(block_counts));
        for i = 1:length(block_counts)
            if (block_counts(i)>fs/2)
                rec_bit_seq(i)=1;
            end
        end
    case 'part_3'
        % Reshape the array into blocks of fs samples
        block_samples = reshape(rec_sample_seq, fs, []);
        rec_bit_seq = block_samples(1, :);
```

ComputeBER

```
% Compute number of bit errors
no_errors = sum(bit_seq ~= rec_bit_seq);
% Compute BER
BER = no_errors/length(bit_seq);
```