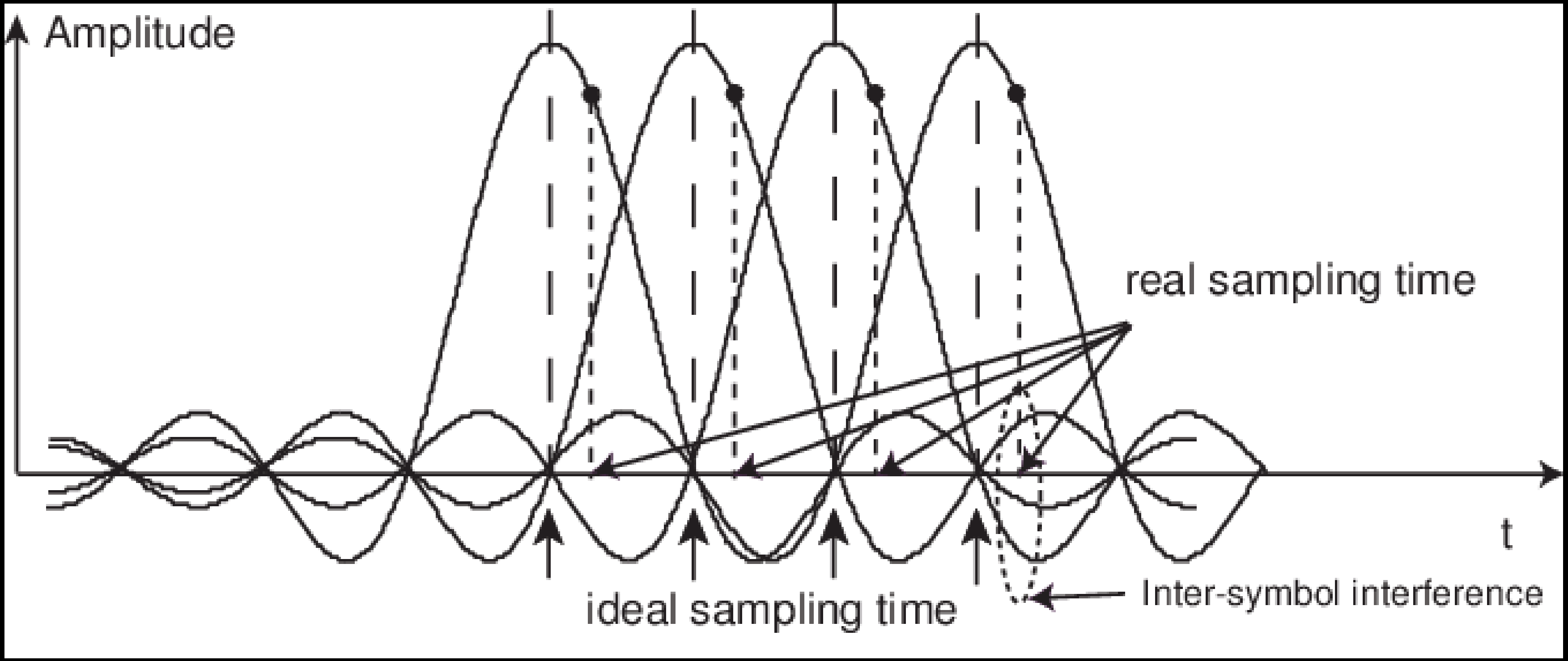
DIGITAL COMMUNICATIONS

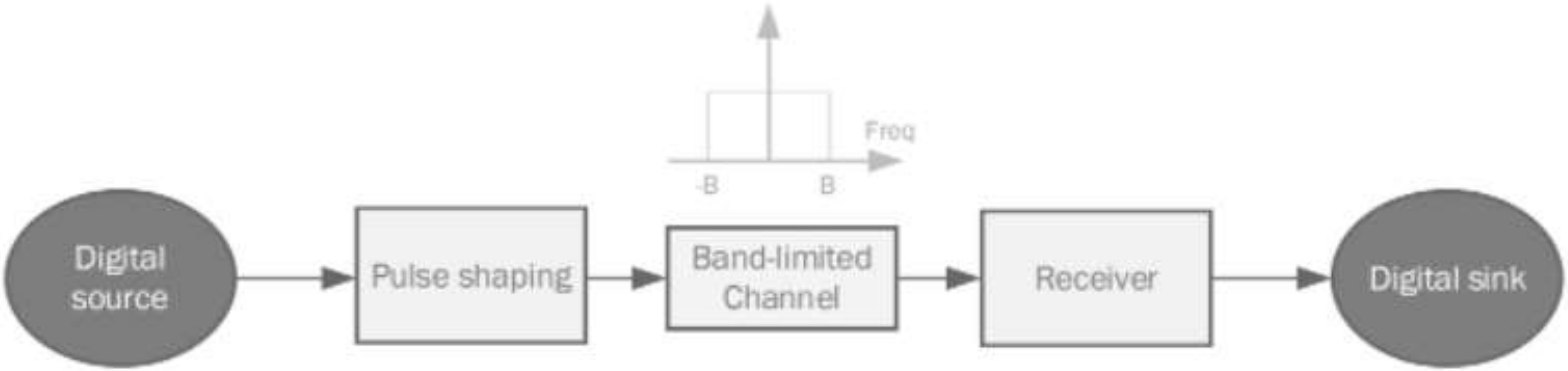
LAB 3

19.5.2022

WHAT IS INTER-SYMBOL INTERFERENCE (ISI)?

- Inter-symbol interference (ISI) is **a form of distortion of a signal in which one symbol interferes with subsequent symbols**.
- It's an unwanted phenomenon as the previous symbols have similar effect as noise.
- It makes the communication less reliable.



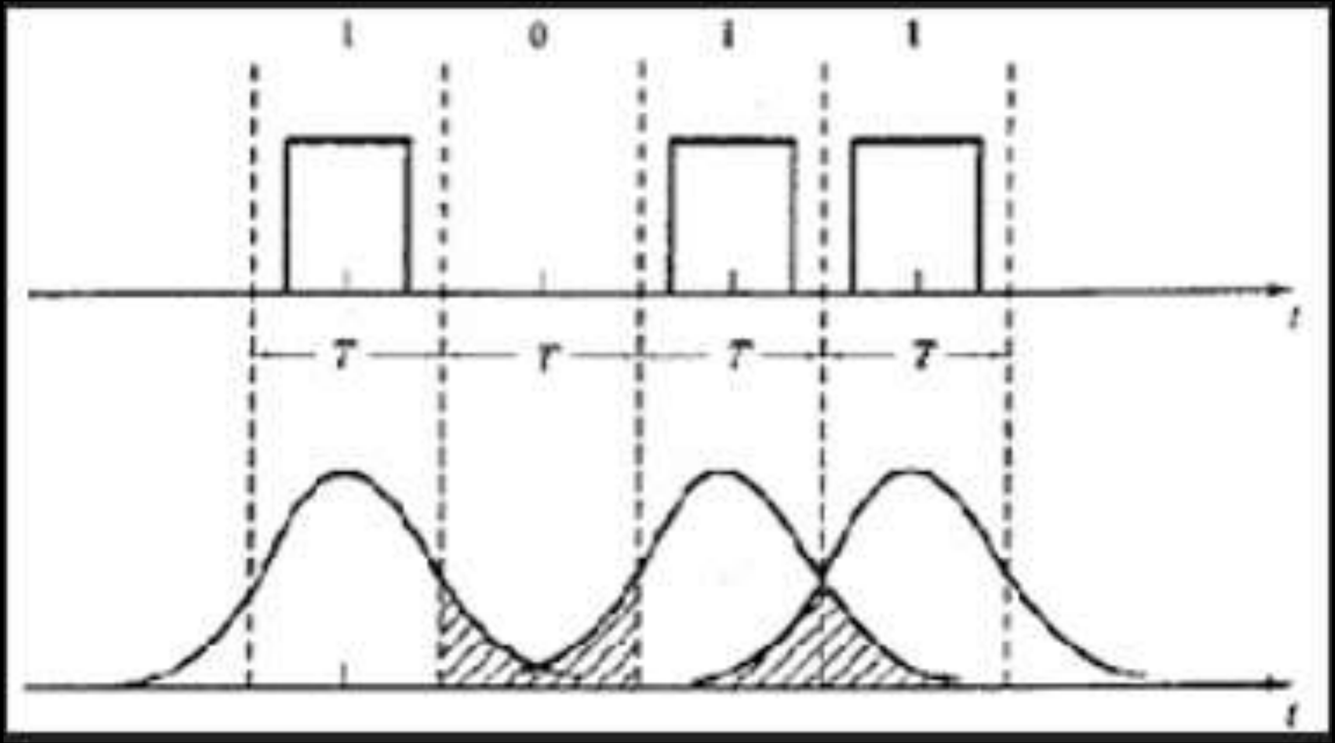


INTER-SYMBOL INTERFERENCE DUE TO BAND-LIMITED CHANNELS

PART (1)

WHAT'S A BAND-LIMITED CHANNEL?

- The channel only allows a limited range of frequency
- components to pass.
- It blocks frequency components outside this range.
- The channel obviously limits the kind of signals that can pass unchanged through the channel.
- Most signals will have an output signal different from their input signal after passing through it.

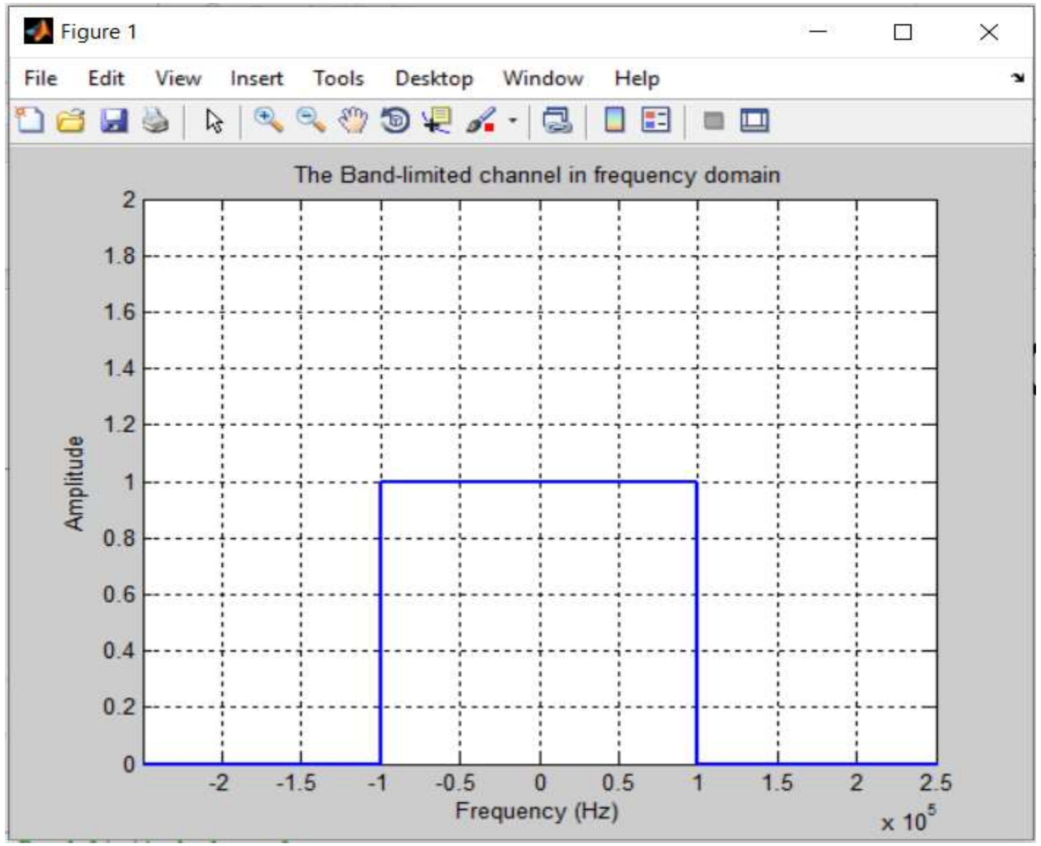


STEPS TAKEN TO ACHIEVE THE LAB GOAL

(1)	(2)	(3)	(4)
Created Band-limited channel	Generated first square pulse and passed it through the Band-limited channel	Generated second square pulse and passed the 2 square pulses through the Band-limited channel	Created Sinc function and passing it through the band-limited channel

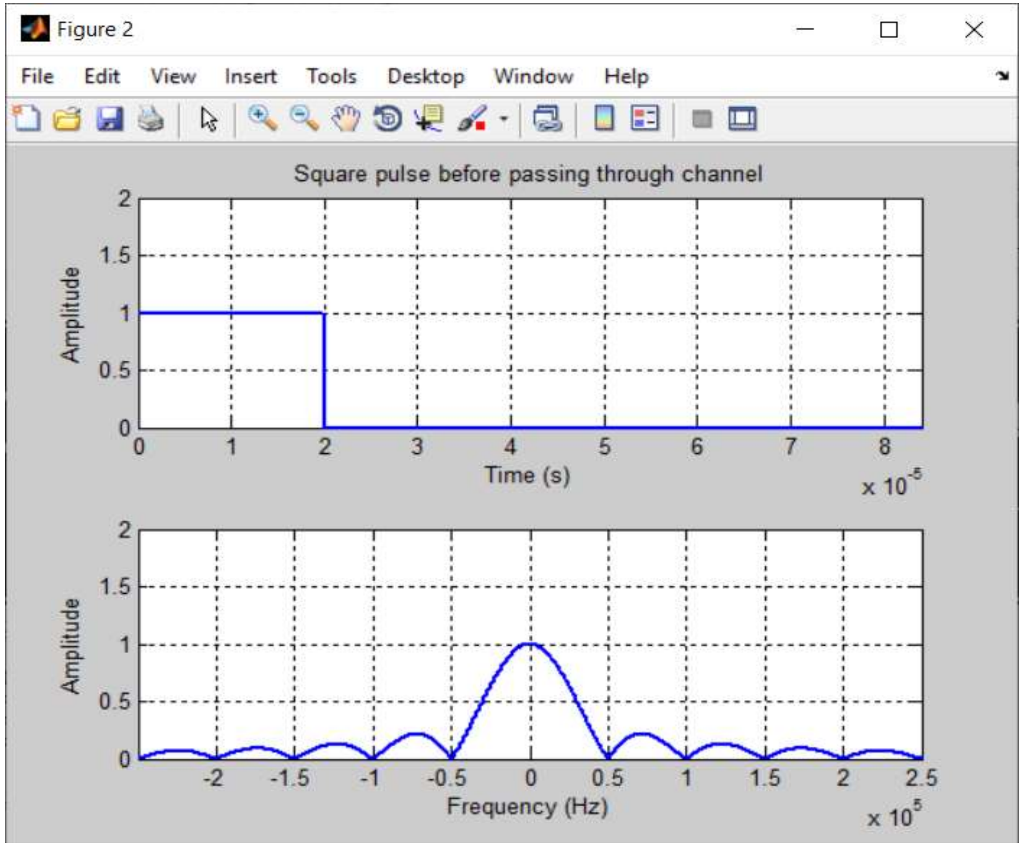
CREATED BAND-LIMITED CHANNEL

```
clc
close all
clear all

fs = 1e7;
Ts = 1/fs;
N = 1e7;
time_axis = (0:N-1)*Ts;

freq_axis = -fs/2:fs/N:fs/2-1/N;
B = 100e3;
T = 2/B;

%% Creating Band-limited channel

one_square = ones(1,200e3);
zero_me = zeros(1,9800e3/2);
Band_limited_channel= [zero_me one_square zero_me];

figure
plot(freq_axis,Band_limited_channel,'linewidth',2)
grid on
ylim([0 2])
xlim([-1/T 1/T]*5)
xlabel('Frequency (Hz)','linewidth',2)
ylabel('Amplitude','linewidth',2)
title('The Band-limited channel in frequency domain','linewidth',10)
```



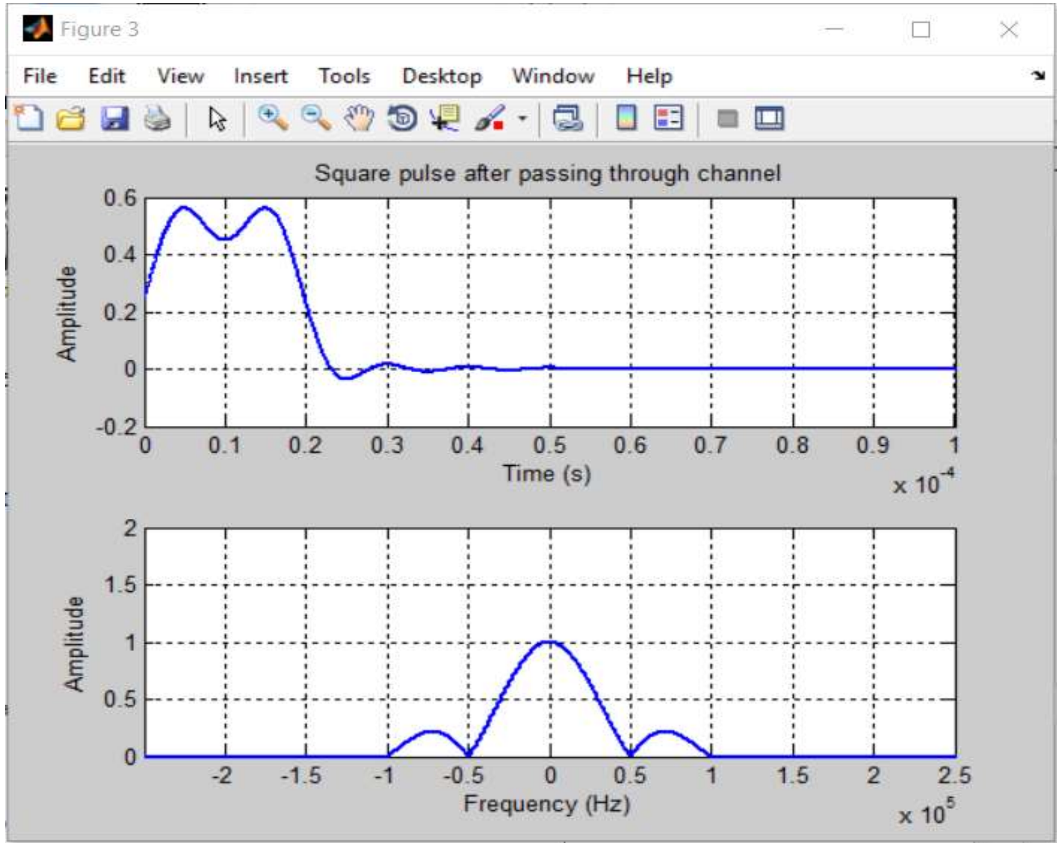
GENERATED FIRST SQUARE PULSE

```
%% Generating first square pulse

x_bits = [1];
pulse1 = rectpuls(time_axis-1/B,T);
pulse1_length = length(pulse1);
pulse1_fft = (1/200)*fftshift(fft(pulse1));
freq_axis = -fs/2:fs/pulse1_length:fs/2-1/pulse1_length;

%% Plotting first square pulse

figure
subplot(2,1,1)
plot(time_axis,pulse1,'b','linewidth',2); hold on;
grid on
xlim([0 T*4.2])
ylim([0 2])
xlabel('Time (s)','linewidth',2)
ylabel('Amplitude','linewidth',2)

subplot(2,1,2)
plot(freq_axis,abs(pulse1_fft),'b','linewidth',2); hold on;
grid on
ylim([0 2])
xlim([-1/T 1/T]*5)
xlabel('Frequency (Hz)','linewidth',2)
ylabel('Amplitude','linewidth',2)
subplot(2,1,1)
title('Square pulse before passing through channel','linewidth',10)
```



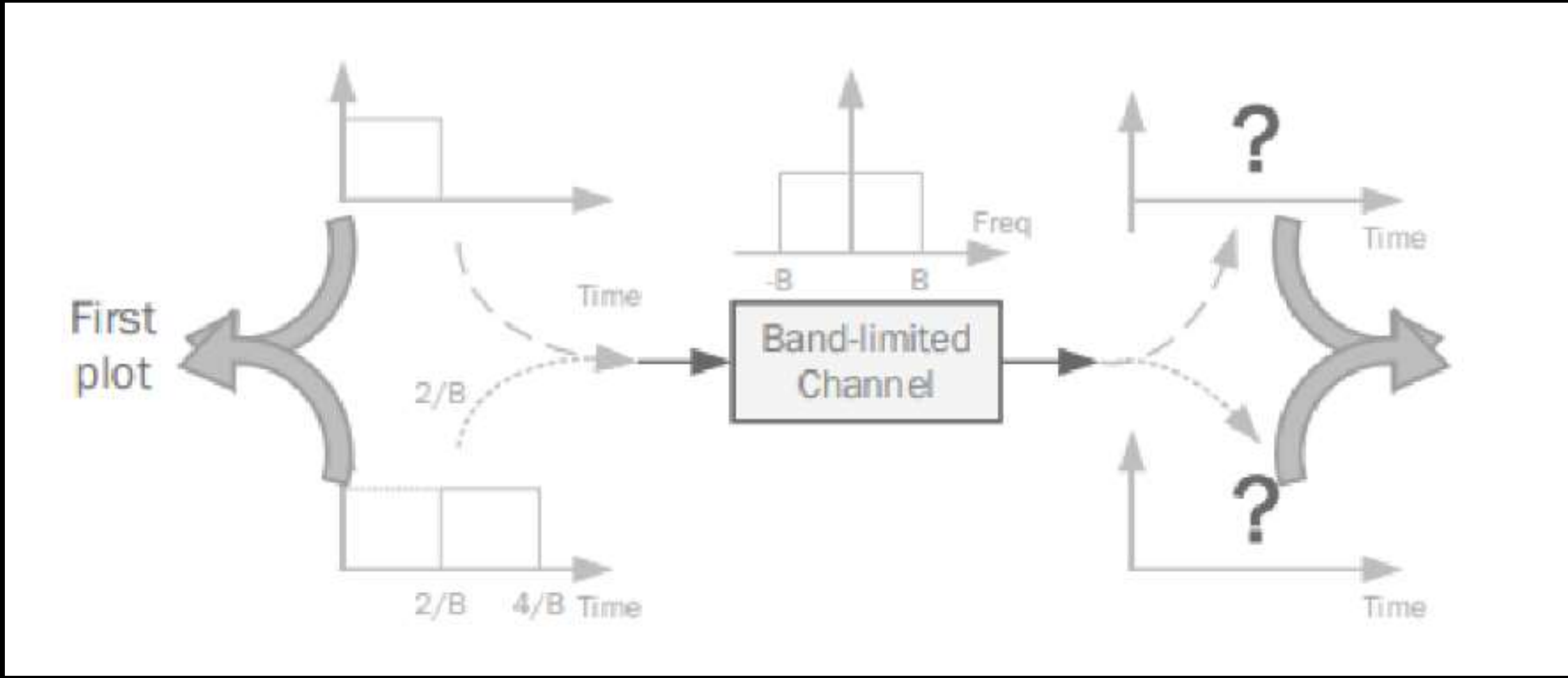
PASSED FIRST SQUARE PULSE THROUGH THE BAND-LIMITED CHANNEL

```
%% Passing first square pulse through the Band-limited channel

pulse1_after_chann = pulse1_fft .* Band_limited_channel;
pulse1_after_chann_T =100* ifft(ifftshift(pulse1_after_chann));

figure
subplot(2,1,1)
plot(time_axis,pulse1_after_chann_T,'b','linewidth',2); hold on;
grid on
xlim([0 T*5])
xlabel('Time (s)','linewidth',2)
ylabel('Amplitude','linewidth',2)

subplot(2,1,2)
plot(freq_axis,abs(pulse1_after_chann),'b','linewidth',2); hold on;
grid on
xlim([-1/T 1/T]*5)
ylim([0 2])
xlabel('Frequency (Hz)','linewidth',2)
ylabel('Amplitude','linewidth',2)
subplot(2,1,1)
title('Square pulse after passing through channel','linewidth',10)
```



PASSING 2 SQUARE PULSES THROUGH THE CHANNEL

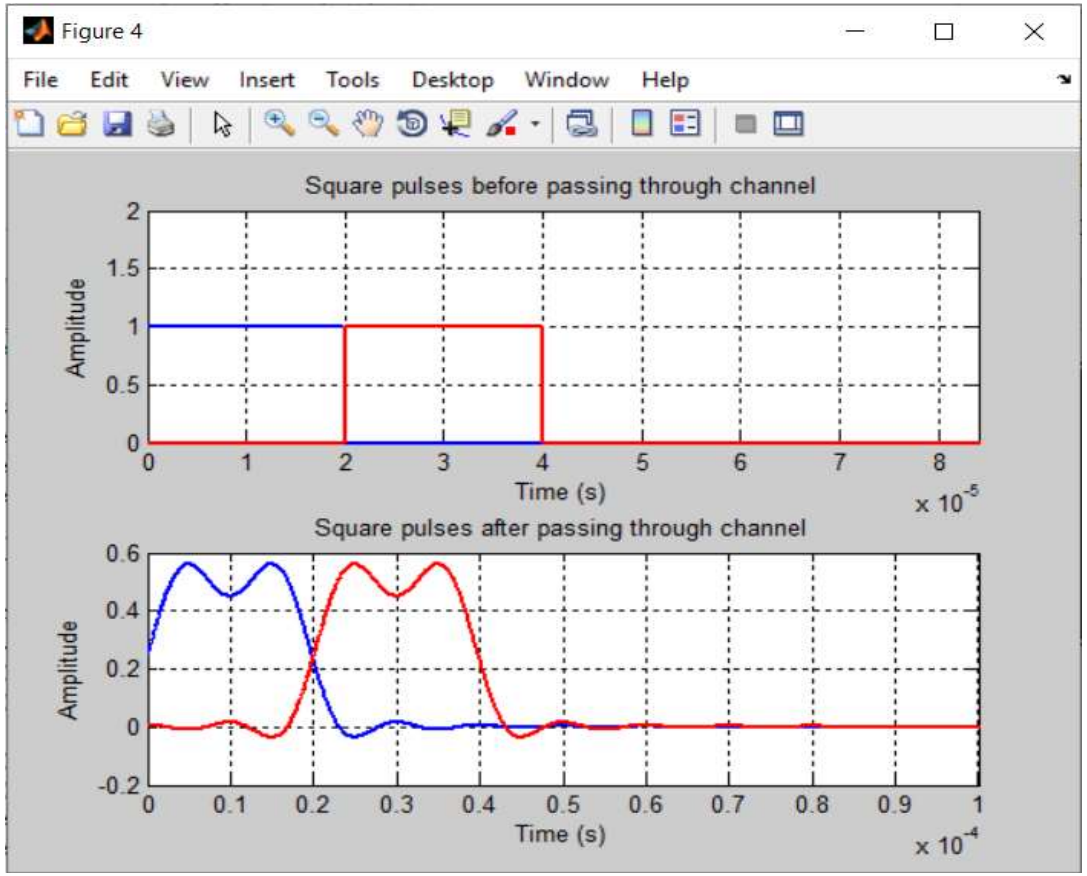
- Our expectation is that inter-symbol interference is going to be clear.
- Each one of the square pulses change shape so they will interfere with one another.
- The output signal will differ from the original signal.



GENERATED SECOND SQUARE PULSE AND

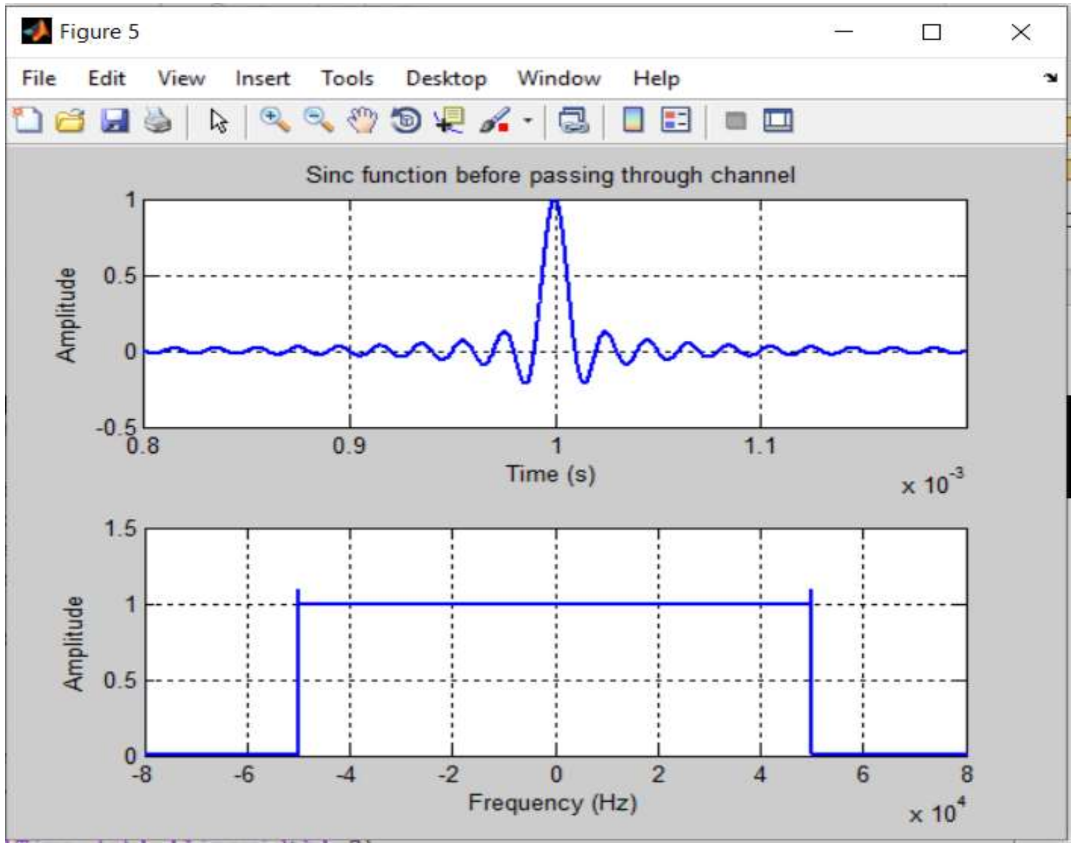
PASSED THE 2 SQUARE PULSES THROUGH THE BAND-LIMITED CHANNEL

```
%% Generating second square pulse

x_bits = [0 1];
pulse2 = rectpuls(time_axis-3/B,T);
pulse2_length = length(pulse2);
pulse2_fft =(1/200)* fftshift(fft(pulse2));

%% Plotting the 2 square pulses

figure
subplot(2,1,1)
plot(time_axis,pulse1,'b','linewidth',2); hold on;
plot(time_axis,pulse2,'r','linewidth',2); hold on;
grid on
xlim([0 T*4.2])
ylim([0 2])
xlabel('Time (s)','linewidth',2)
ylabel('Amplitude','linewidth',2)
title('Square pulses before passing through channel','linewidth',10)

%% Passing the 2 square pulses through the Band-limited channel

pulse2_after_chann = pulse2_fft .* Band_limited_channel;
pulse2_after_chann_T =100* ifft(ifftshift(pulse2_after_chann));

subplot(2,1,2)
plot(time_axis,pulse1_after_chann_T,'b','linewidth',2); hold on;
plot(time_axis,pulse2_after_chann_T,'r','linewidth',2); hold on;
grid on
xlim([0 T*5])
xlabel('Time (s)','linewidth',2)
ylabel('Amplitude','linewidth',2)
title('Square pulses after passing through channel','linewidth',10)
```



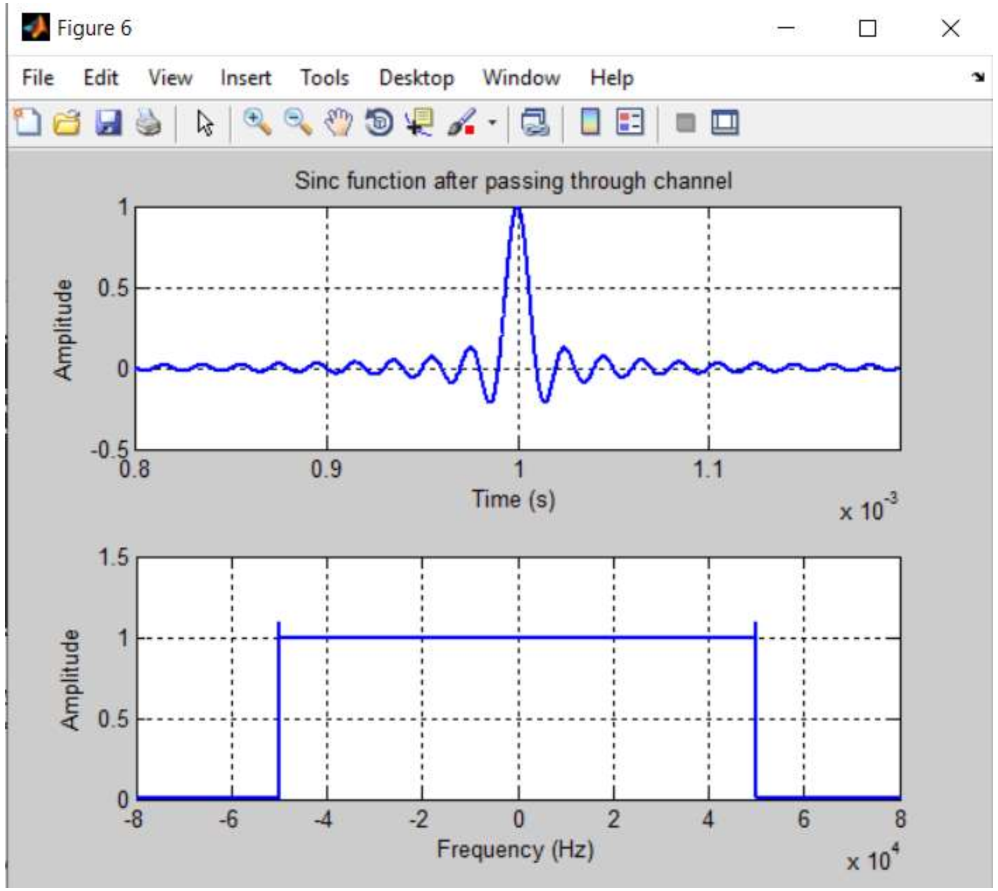
CREATED SINC FUNCTION

```
%% creating Sinc function and passing it through the band-limited channel

time_axis = (-N/2:N/2-1)*Ts;
y = sinc(time_axis*B);
y1 = [zeros(1,10000) y(1:9990000)];
y1_length = length(y1);
y1_f = (1/100)*fftshift(fft(y1));
freq_axis = -fs/2:fs/y1_length:fs/2-1/y1_length;

figure
subplot(2,1,1)
plot(time_axis,y1,'b','linewidth',2); hold on;
xlim([0.0008 0.0012])
grid on
xlabel('Time (s)','linewidth',2)
ylabel('Amplitude','linewidth',2)
subplot(2,1,2)
plot(freq_axis,abs(y1_f),'b','linewidth',2); hold on;
xlim([-80000 80000])
grid on
xlabel('Frequency (Hz)','linewidth',2)
ylabel('Amplitude','linewidth',2)
subplot(2,1,1)
title('Sinc function before passing through channel','linewidth',10)
```



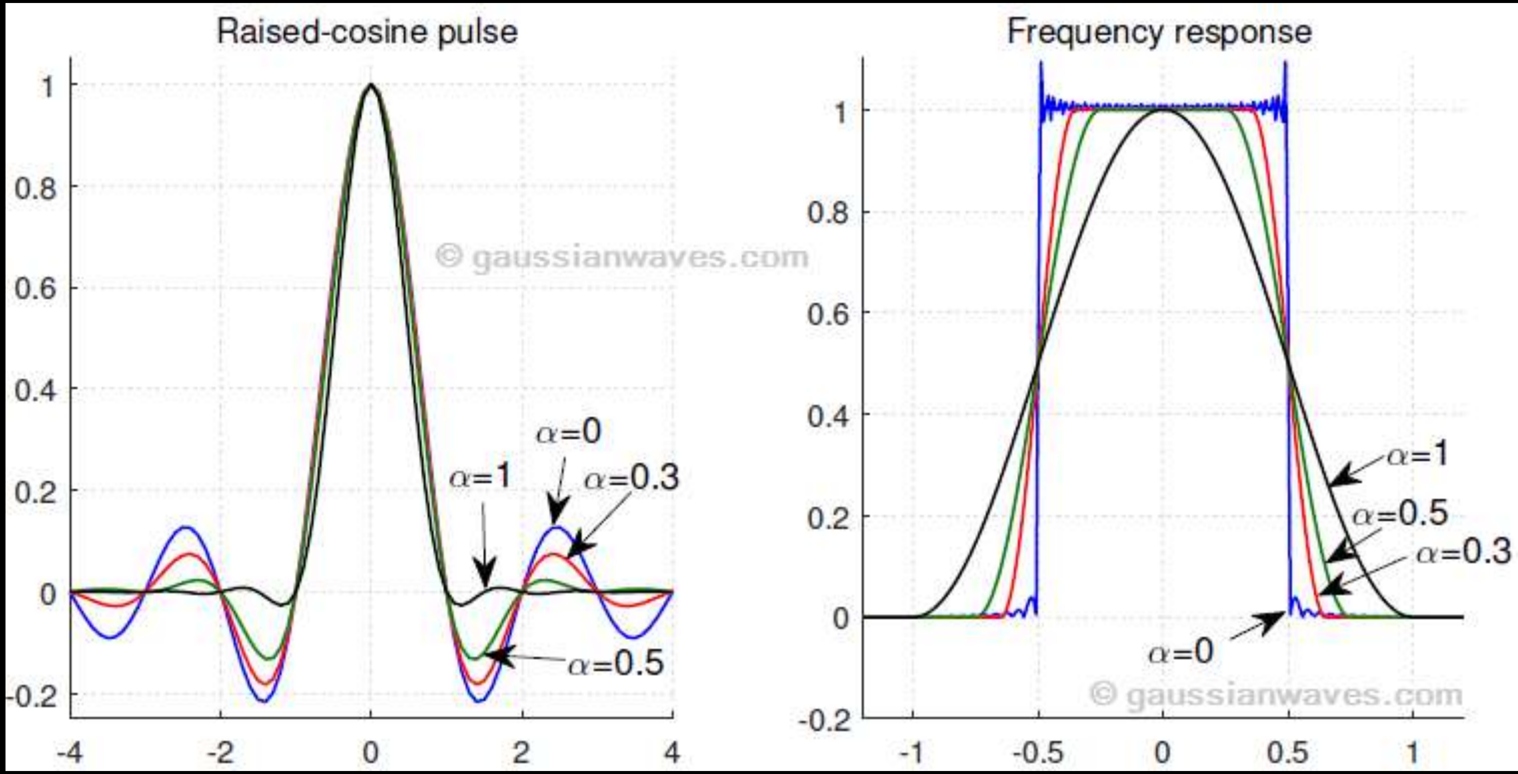
PASSED SINC FUNCTION THROUGH THE BAND-LIMITED CHANNEL

```
yl_after_ch = yl_f .* Band_limited_channel;
yl_after_ch_T = ifft(ifftshift(yl_after_ch));

figure
subplot(2,1,1)
plot(time_axis,yl_after_ch_T,'b','linewidth',2); hold on;
xlim([0.0008 0.0012])
grid on
xlabel('Time (s)','linewidth',2)
ylabel('Amplitude','linewidth',2)
subplot(2,1,2)
plot(freq_axis,abs(yl_after_ch),'b','linewidth',2); hold on;
xlim([-80000 80000])
grid on
xlabel('Frequency (Hz)','linewidth',2)
ylabel('Amplitude','linewidth',2)
subplot(2,1,1)
title('Sinc function after passing through channel','linewidth',10)
```



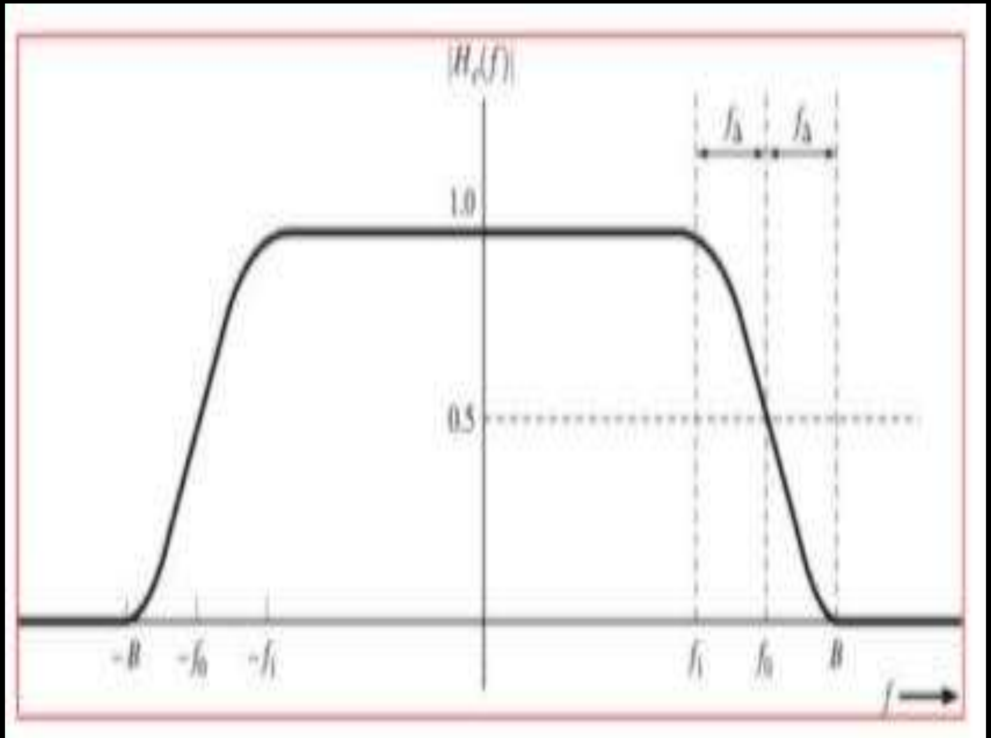
OVERCOME SQUARE PULSE PROBLEM WITH BAND-LIMITED CHANNELS

- According to previous outputs, we concluded that we can't use square pulse signals with band-limited channels.
- Other pulse shapes are better suited for band-limited channels.
- We used raised cosine to solve the problem of ISI.
- The raised-cosine filter is an implementation of a low-pass Nyquist filter.



RAISED-COSINE FILTER

NYQUIST'S FIRST METHOD (ZERO ISI)



$$h_e(t) = \frac{\sin \pi f_s t}{\pi f_s t} \leftrightarrow H_e(f) = \frac{1}{f_s}\Pi\left(\frac{f}{f_s}\right)$$

RAISED-COSINE FILTER

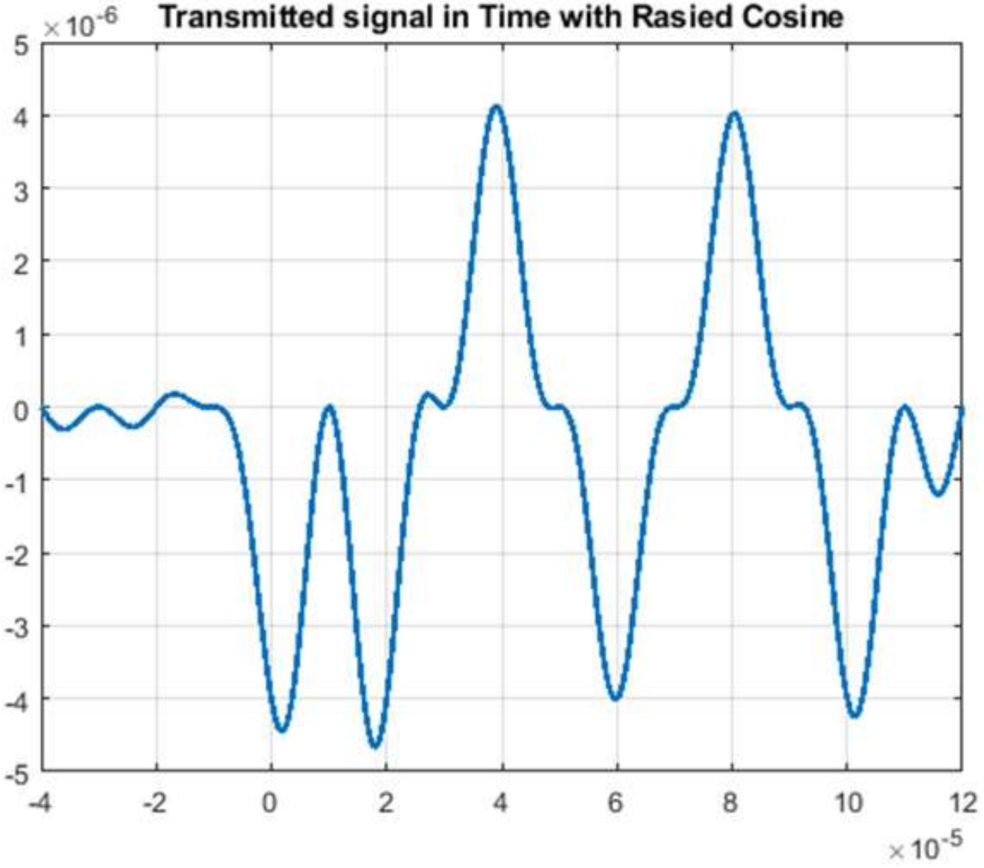
TIME DOMAIN

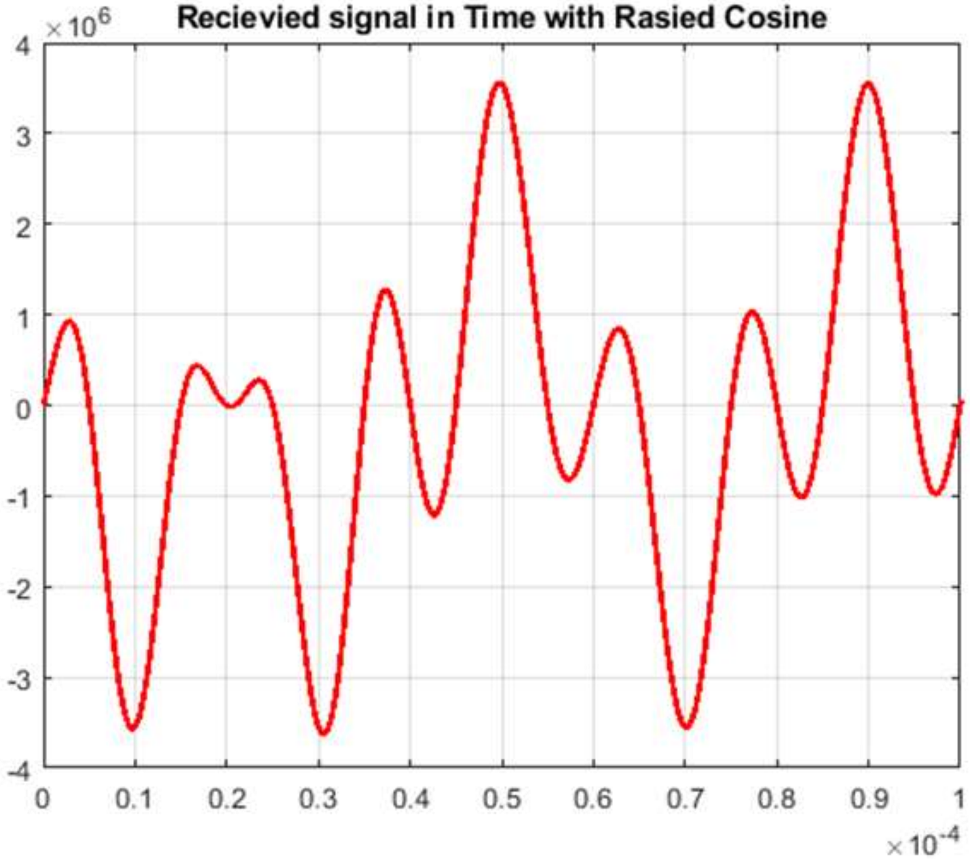
$$p(t) = \operatorname{sinc}(2Wt)\frac{\cos(2\pi\alpha Wt)}{1-16\alpha^2W^2t^2}$$

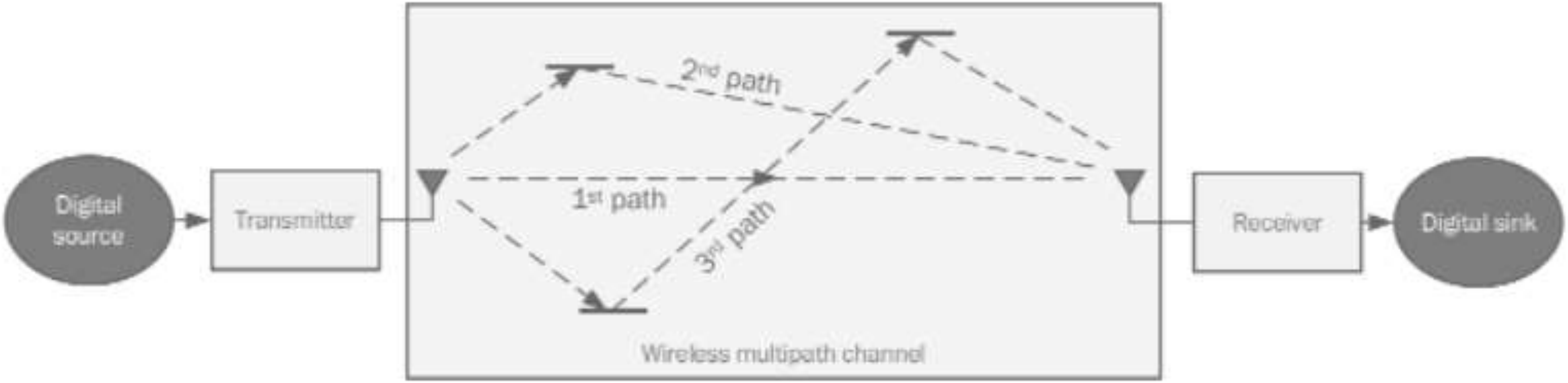
FREQUENCY DOMAIN

$$P(f) = \begin{cases} \dfrac{1}{2W}, & 0 \le |f| < f_1 \\ \dfrac{1}{4W}\left\{1+\cos\left[\dfrac{\pi}{2W\alpha}(|f|-f_1)\right]\right\}, & f_1 \le |f| < 2W-f_1 \\ 0, & |f| \ge 2W-f_1 \end{cases}$$

RAISED-COSINE FILTER BEFORE AND AFTER PASSING THROUGH BAND-LIMITED CHANNEL





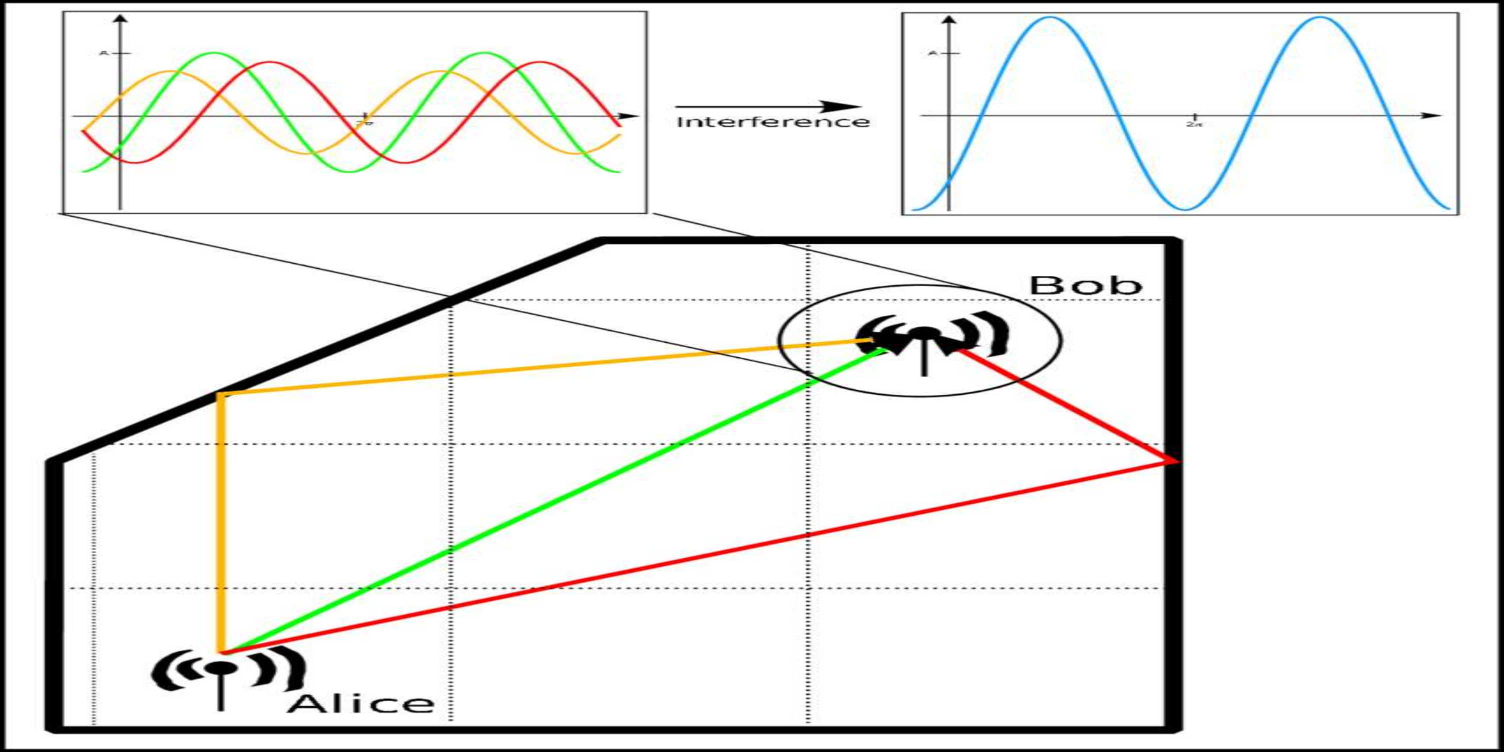


INTER-SYMBOL INTERFERENCE DUE TO MULTI-PATH CHANNELS

PART (2)

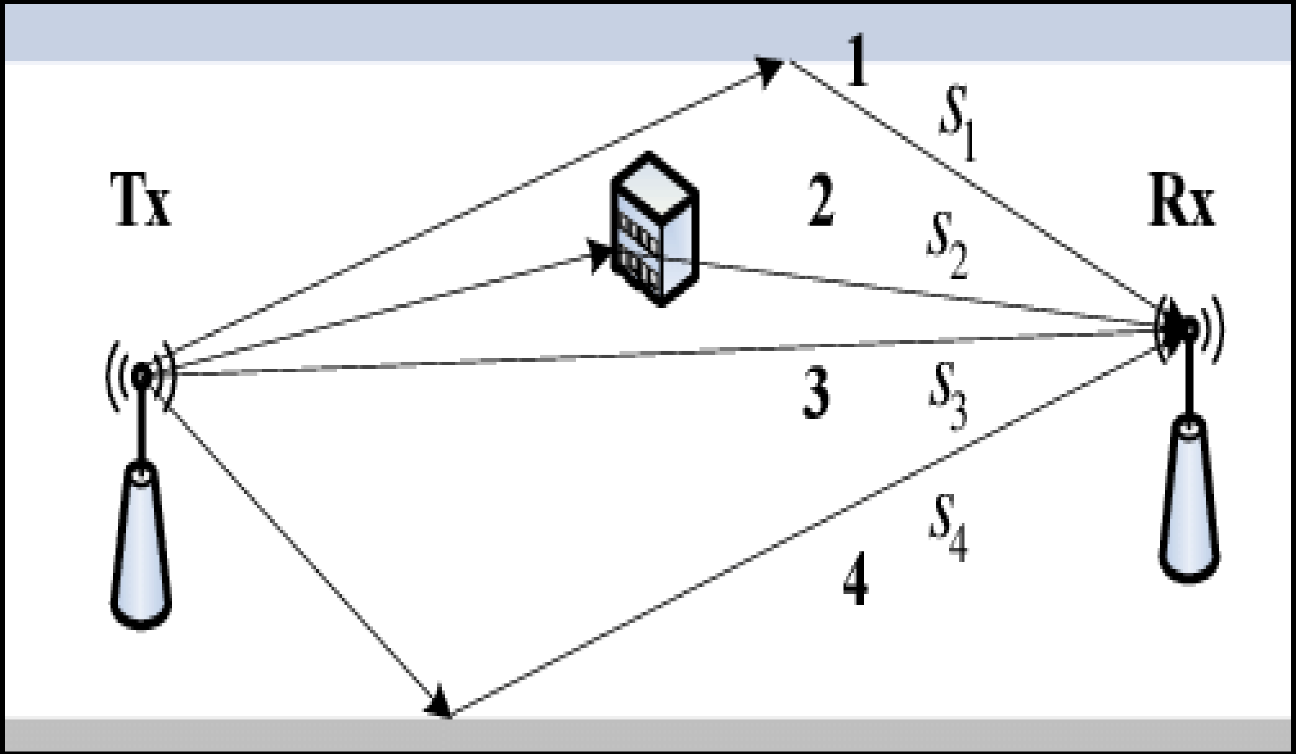
WHAT ARE MULTI-PATH CHANNELS?

- In wireless channels, signals are transmitted via electromagnetic waves which propagate through the air until it reaches the receiver.
- The nature of electromagnetic waves allow that multiple copies of the signal would travel around and reach the receiver at different times.
- A symbol transmitted by the transmitter would traverse multiple paths until it reaches the receiver.



WHAT ARE MULTI-PATH CHANNELS?

- Therefore, the receiver is expected to receiver multiple copies of the same transmitted signal.
- Each of these copies would arrive at a different time and with a different magnitude.
- The time is determined by how long the path is.
- The magnitude is determined by how much attenuation that the signal suffered from during the transmission across the path.



MATHEMATICAL POINT OF VIEW

- Let the first symbol transmitted by the transmitter be labelled as x[0].
- Let y[0] be the received signal.
- Let h0 be the channel effect of the first path on the transmitted signal x[0].
- Let n[0] be the noise component.

$$y[0] = h_0 x[0] + n[0]$$

$$y[1] = h_0 x[1] + h_1 x[0] + n[1]$$

$$y[L-1] = h_0 x[L-1] + h_1 x[L-2] + h_2 x[L-3] + \cdots + h_{L-2} x[1] + h_{L-1} x[0] + n[L-1]$$

$$\begin{aligned}
y[0] &= h_0 x[0] + n[0] \\
y[1] &= h_0 x[1] + h_1 x[0] + n[1] \\
y[2] &= h_0 x[2] + h_1 x[1] + h_2 x[0] + n[2] \\
&\vdots \\
y[L-1] &= h_0 x[L-1] + h_1 x[L-2] + h_2 x[L-3] + \cdots + h_{L-2} x[1] + h_{L-1} x[0] + n[L-1]
\end{aligned}$$

$$\underbrace{\begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ \vdots \\ y[L-1] \end{bmatrix}}_{Y} = \underbrace{\begin{bmatrix} h_0 & & & & & & \\ h_1 & h_0 & & & & & \\ h_2 & h_1 & h_0 & & & & \\ \vdots & & & \ddots & h_1 & h_0 & \\ h_{L-1} & h_{L-2} & h_{L-3} & \cdots & h_2 & h_1 & h_0 \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ \vdots \\ x[L-3] \\ x[L-2] \\ x[L-1] \end{bmatrix}}_{X} + \underbrace{\begin{bmatrix} n[0] \\ n[1] \\ n[2] \\ \vdots \\ n[L-1] \end{bmatrix}}_{N}$$

KNOWING Y, H, AND THE STATISTICS OF THE AWGN NOISE (I.E., MEAN AND VARIANCE), WHAT IS THE BEST WAY OF ESTIMATING THE TRANSMITTED SYMBOLS X?

- THE GOAL HERE IS TO GIVE AN ANSWER TO THE PREVIOUS QUESTION.
- THERE ARE SEVERAL TECHNIQUES TO SOLVE THIS EQUATION.
- WE INVESTIGATED ONE OF THESE TECHNIQUES IN OUR FOLLOWING TRIAL.

STEPS TAKEN TO ACHIEVE THE LAB GOAL

(1)	(2)	(3)	(4)
Created the channel function, path and AWGN noise.	Generating the mathematical matrix to be computed.	Calculated the bit error rate (BER).	Plotted the BER vs Eb/No performance.

CREATED THE CHANNEL FUNCTION, PATH AND AWGN NOISE

```
%Transmitted signal is x
% received signal is y
% N is the awgn noise that corrupts y
% our function is Y = HX+N where H(i) is the channel effect of the i+1
% channel and is represented by a 2D matrix of dimension L*L

%initializing variables
Eb = 1;
%dimension of matrix, could be changed, I used 5 for debugging
L = 50;
H = zeros(L,L);
%channel function that we're going to use  % e^-0.5*x^2
channel_function = exp(-0.5*[0:L-1].^2)';
%channel_function = repmat(channel_function,1,1);
%creating an array of normally distributed numbers, mean =0, variance =1
h = randn(L,1);
%i added the other randn as the graph looked weird
% awgn noise 11 values
No = [0 0.001 0.005 0.01 0.05 0.1 0.25 0.5 0.7 0.9 1];
% initializing empty array to store the BER values inside the loop for plotting
temporary_BER = [];
% initializing empty array for BER/noise
BER = [];
h = abs(h).*channel_function;
```

GENERATING THE MATHEMATICAL MATRIX TO BE COMPUTED

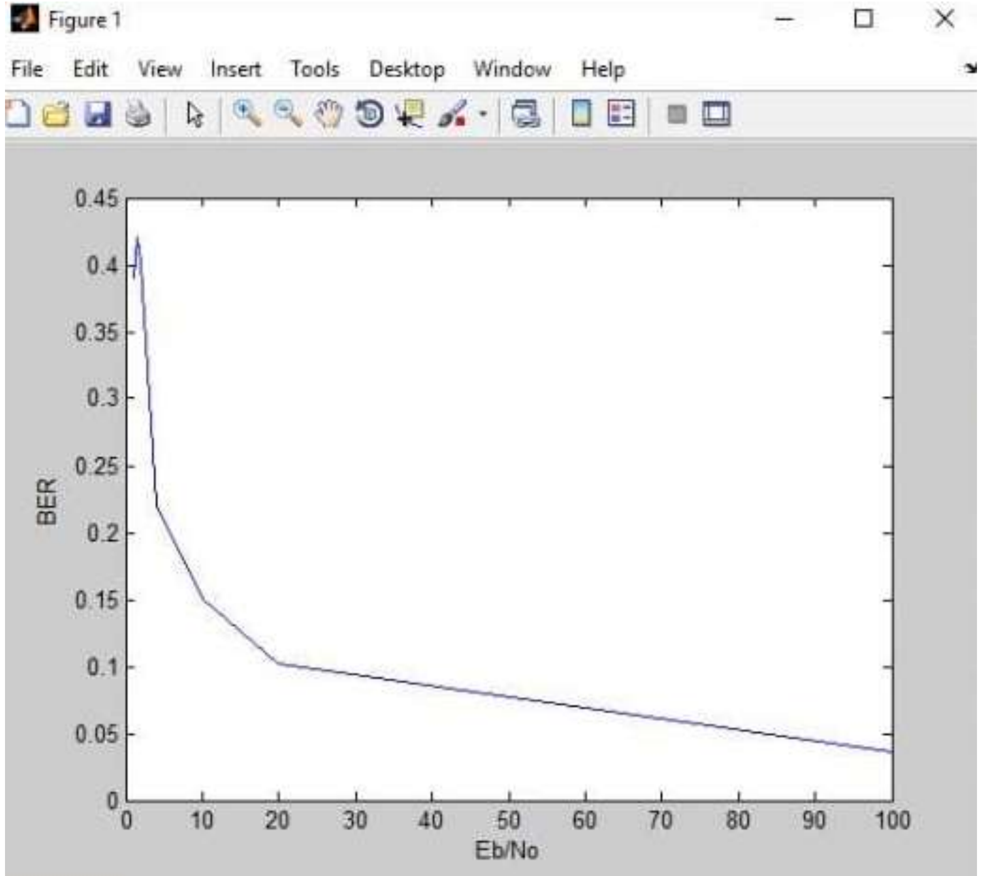
```
% initializing empty array for BER/noise
BER = [];
h = abs(h).*channel_function;
%filling matrix
i=1;
j = 1;
for k = 1:L
    for m = i:-1:1
        H(k,j) = h(m);
        j = j +1;
    end
    j = 1;
    i = i + 1;
end
%disp(h)
%disp(H)
%inverse_H = inv(H);   %inversing the matrix
```

CALCULATED THE BIT ERROR RATE (BER).

```
for a = No
    %calculating noise power
    N = sqrt(a/2)*randn(L,1);
    %calculating the BER 11 times for each noise
    for i = 1:11
        %returning the non zero values with no repitition
        non_zero_values = setdiff(-1:1, 0);
        x = non_zero_values( randi(length(non_zero_values), L, 1) );
        %recieved bits/ signal
        y = H*x' + N;
        %recieved bits plus noise
        x_received =inv(H)*y;
        D = zeros(size(x_received));
        for k = 1:L
            if x_received(k) <= 0
        %polar type, if it's less than zero it's -1, if it's more it's -1
                D(k)= -1 ;
            else
                D(k) = 1;
            end
        end
        n = 0;
        for k = 1:L
            %checking with initial x to calculate BER
            if D(k) ~= x(k)
                n = n + 1;
            end
        end
        temporary_BER = [temporary_BER n/L];
    end
    %calculating the mean BER for each noise
    BER = [BER mean(temporary_BER)];
    %reseting the array for the new noise
    temporary_BER = [];
end
```

PLOTTED THE BER VS EB/NO PERFORMANCE.

```
plot(Eb./No, BER)
xlabel('Eb/No')
ylabel('BER')
xlim([0,100])
```



THANK YOU

Team Members

Mennatallah Moustafa 6234

Farah Ahmed 6274

Alaa Mohamed Abdel Hamid 6473

Rawan Hindawy 6491

Nouran Hisham 6532

Kareem Sabra 6594

Esraa Mahmoud 6597

Nourhan Waleed 6609

Seif Mohamed 6624

Fatema Moharam 6655