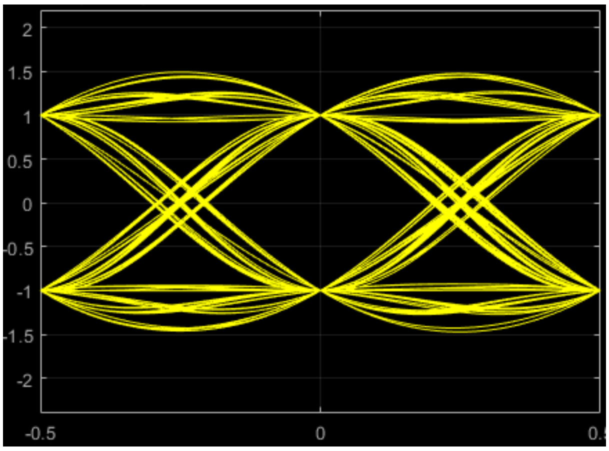
EN2074 Communication Systems Engineering

Lab Assignment – Eye Diagrams and Equalization

Name	**Index Number**
Dilshan N.L.	210129P
Gunawardana W.N.M	210199D

Content

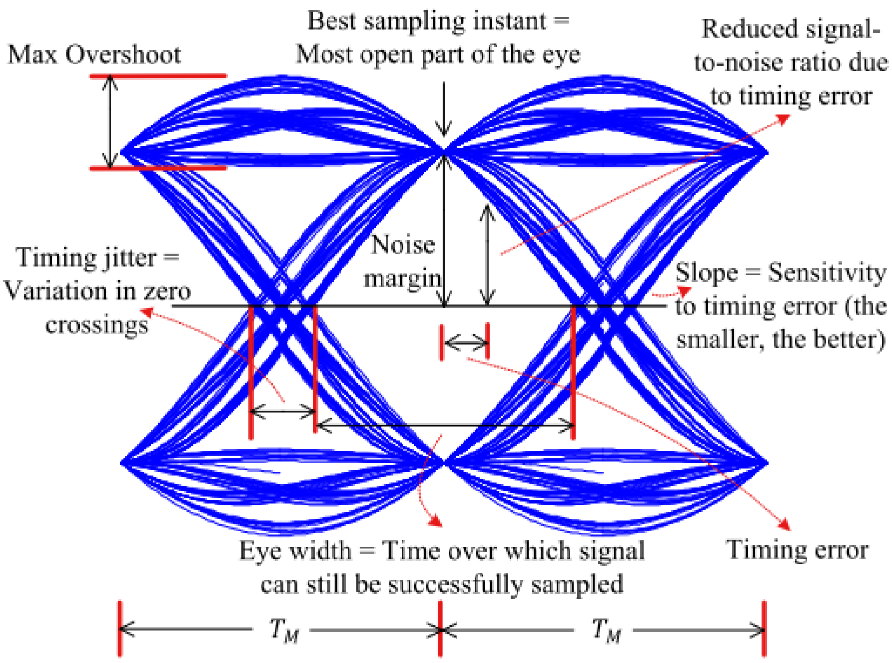
Eye Diagram and Analysis



An eye diagram is a powerful tool used in the analysis of digital communication systems to evaluate the performance of signal transmission and reception. An eye diagram is generated by overlaying many periods of a signal waveform on a single graph. This overlay is usually done by:

1. Sampling the Signal: Capturing multiple instances of the signal at a consistent phase of its clock cycle.
2. Superimposing: These samples are then overlaid on top of each other on a single time axis.

The following features can be extracted from an eye diagram.



An "eye diagram" is named for its resemblance to a human eye when used in analyzing digital signals, particularly in 2-PAM (Pulse Amplitude Modulation) systems. Here's how to analyze an eye diagram: An eye diagram allows for the analysis of various key aspects of a digital communication system.

These include inter-symbol interference (ISI), noise immunity, sensitivity to synchronization errors, and variations in delay, commonly referred to as jitter. This makes the eye diagram an essential tool for assessing and optimizing the performance of communication links.

- Vertical Opening:
    - This analysis provides insights into the noise immunity of the system. A large vertical opening in the eye diagram indicates high noise immunity.
    - The point with the largest vertical opening should be used as the optimal sampling instant.
- Horizontal opening:
    - The horizontal opening of the eye diagram indicates the ISI-free sampling region.
    - Higher sampling rates result in a narrower horizontal opening, while lower sampling rates produce a wider horizontal opening.
    - A wider horizontal opening suggests that errors due to sampling are reduced.
- Slope of the diagram:
    - The slope of the eye diagram is proportional to the synchronization error.

$$slope \propto syncronization\ error$$

    - Having a large slope can lead to significant errors if proper synchronization does not occur.
- Level crossing ambiguity:
    - This provides insight into the variation of delays, known as jitter.
    - A larger ambiguity indicates that the system experiences significant jitter.
- Max overshoot:
    - This provides a measure of the change in the peak value of the signal at the receiver. The higher the maximum overshot, the greater the deviation of the peak value.

Task 01

In this task,

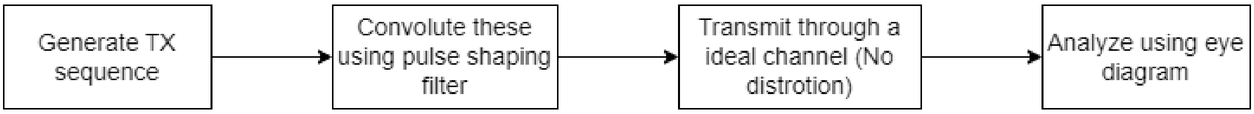


Figure 1.1 – Task block diagram

we just compare 3 pulse shapes assuming ideal channel conditions:

- Sinc pulse
- Raised cosine pulse with 0.5 roll-off and 1.0 roll-off.

First, let's look at the generated impulse train. We generated an impulse train from t=0 to t=10 seconds with 1000 samples. Then, we up-sampled it by a factor of 100 before sending it to the pulse-shaping filter.

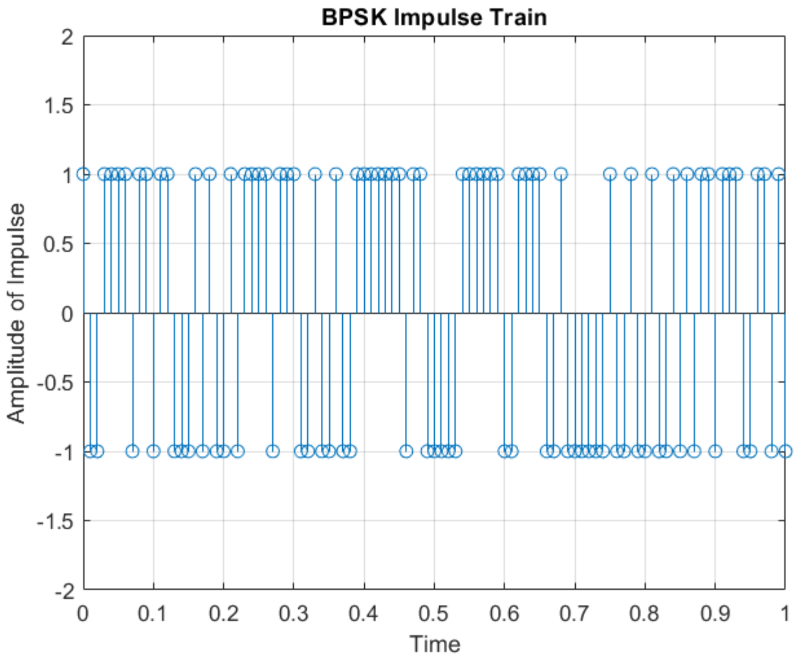


Figure 1. 2 – BPSK impulse train

1. Shaping with Sinc pulse:

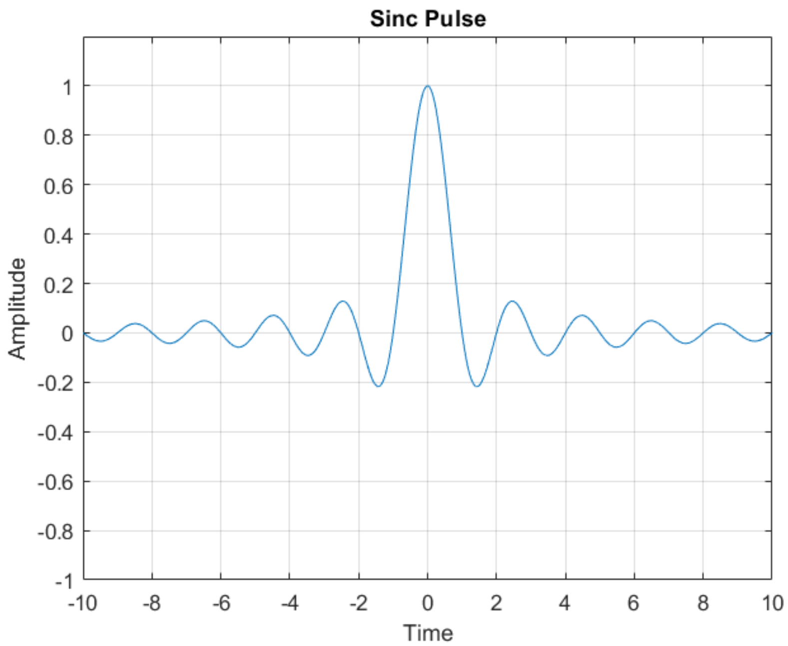


Figure 1.3 – Sinc pulse

The convoluted Sinc pulse with the up-sampled impulse train is given below:

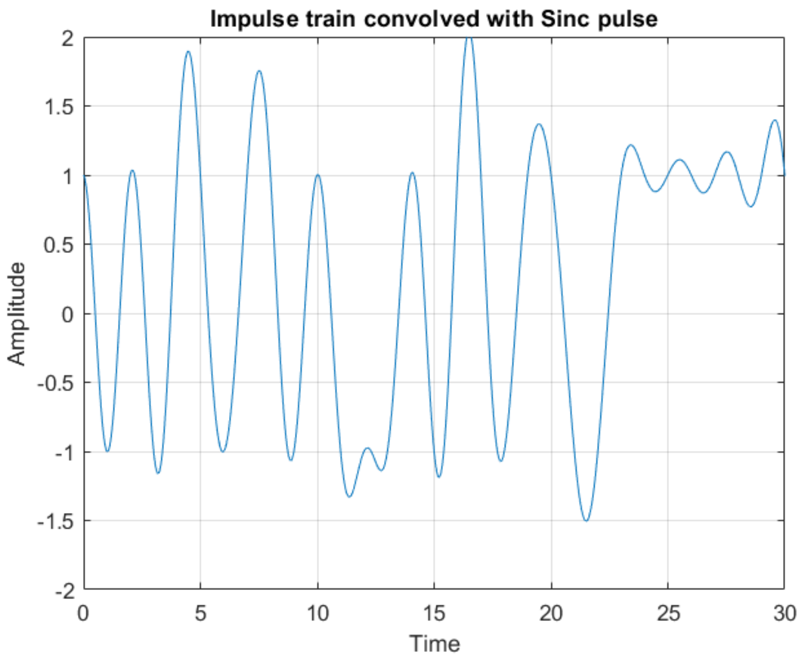


Figure 1.4 – Impulse train convoluted with Sinc pulse.

Eye diagram:

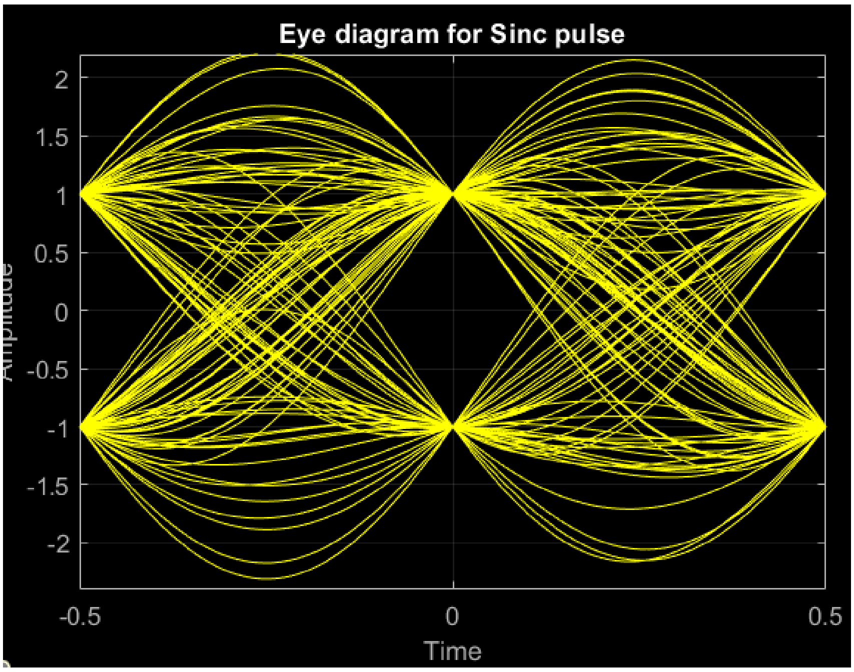


Figure 1.5 – Eye diagram for Sinc pulse.

2. Shaping with raised cosine pulse (roll-off = 0.5):

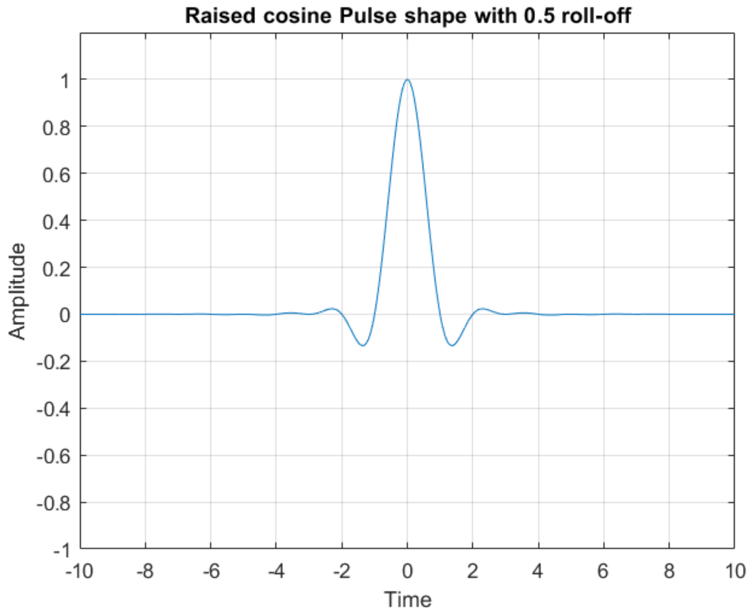


Figure 1.6 – raised cosine pulse shape with 0.5 roll-off.

The convoluted raised cosine pulse with the up-sampled impulse train is given below:

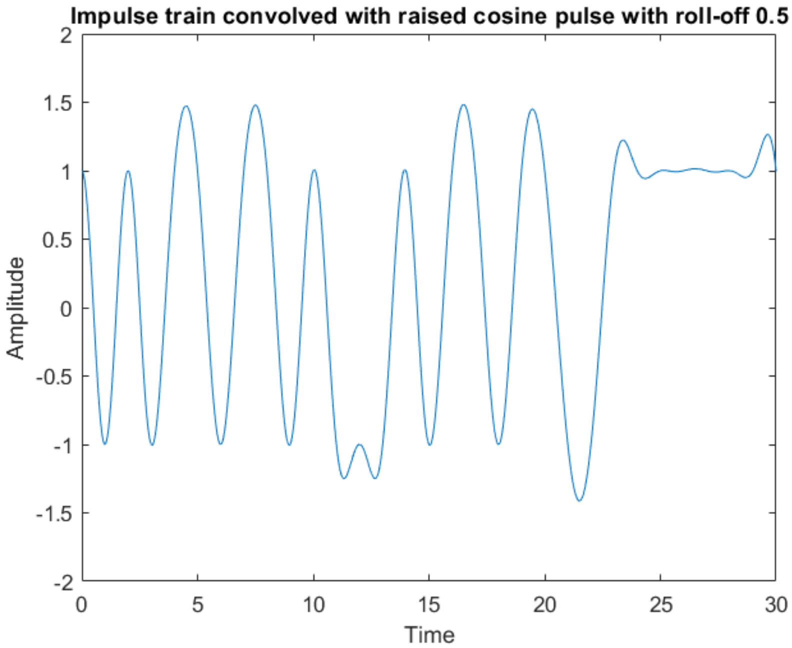


Figure 1.7 – Impulse train convoluted with pulse in Figure 1.6

Eye diagram:

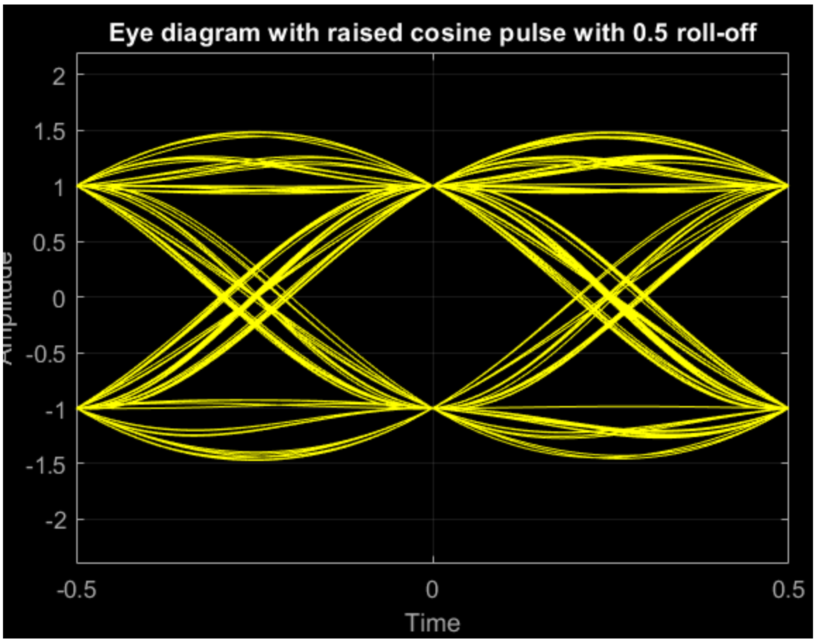


Figure 1.8 – Eye diagram for signal in Figure 1.7

3. Shaping with raised cosine pulse (roll-off = 1.0):

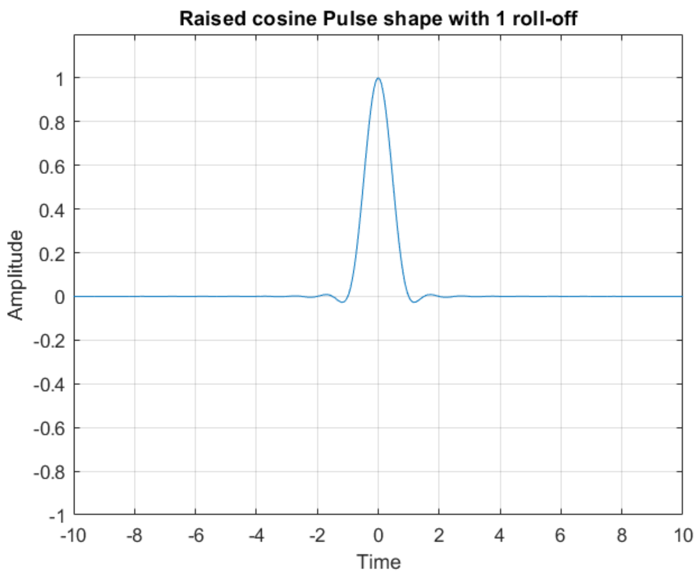


Figure 1.9 – Raised cosine pulse shape with 1 roll-off.

The convoluted raised cosine pulse with the up-sampled impulse train is given below:

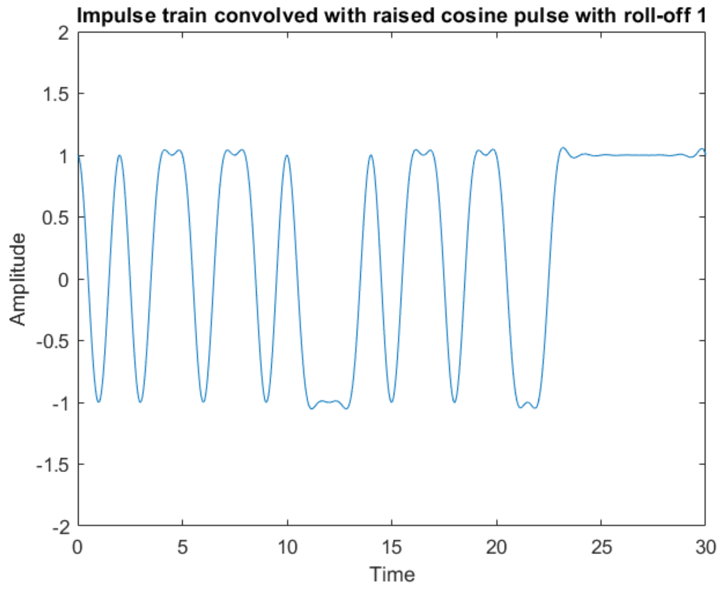


Figure 1.10 – Impulse train convoluted with the signal in Figure 1.9

Eye Diagram:

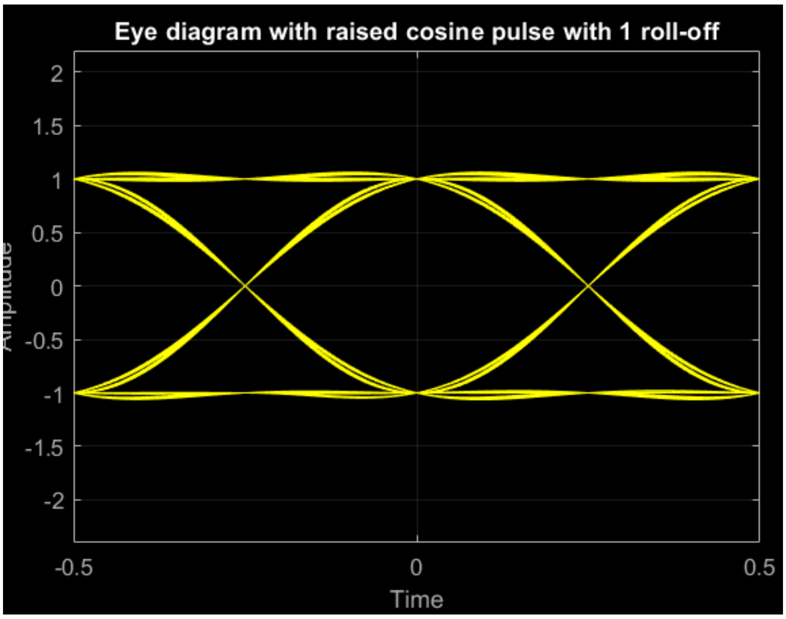


Figure 1.11 – Eye diagram for the signal in Figure 1.10

Property of the eye diagram	Information	Sinc Pulse	Raised cosine pulse (roll-off = 0.5)	Raised cosine pulse (roll-off = 1.0)
Vertical opening	Noise immunity	Since the pulse has a less vertical opening. So, the noise immunity is less.	The vertical opening has been increased significantly w.r.t. Sinc pulse. So, having a higher noise immunity.	This pulse has a higher vertical opening. i.e. This pulse has the highest noise immunity.
Horizontal opening	ISI-free sampling region	It is the lowest of the three. Has less range to the error-free sampling. Less robust to the sampling errors due to ISI.	It is wider than the sinc pulse but narrower than the 1.0 roll-off raised cosine pulse. Better than sinc pulse for error-free sampling	Widest of all three and has the largest region to sample with ISI-free communication.
Slope	Synchronization error	The highest slope of the three has the highest probability of Synchronization errors.	Slope Varies between a range. Compared to the other two consist of the lowest slope as well. More robust to the timing errors.	Average Slope making this to be robust to the timing errors than sinc pulse but more errors than raised cosine with roll-off =0.5
Level crossing ambiguity	Jitter	Verry high jitter as the time variation at zero crossing is high, with very high variations between rise and fall times.	Still, there is some jitter much less than sinc pulse.	There is no visible jitter in the diagram. High robustness to timing offset. Also, highly robust to ISI.
Max overshoot	Variation of the peak value of the received signal	Max overshoot is significantly higher than others.	It's much less overshoot than the Sinc pulse.	Almost negligible overshoot is in the graph.

Table 1.1 – Comparison between three pulses.

Task 02

In this task, we will be simulating the eye diagram with the AWGN.

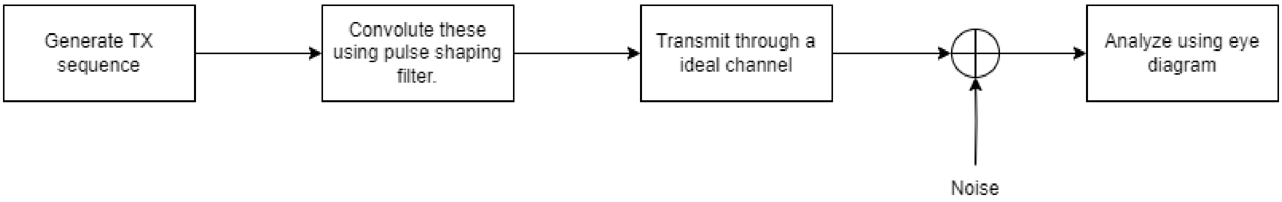


Figure 2.1 – Task 2 block diagram.

The variance of the noise = $\frac{E_b}{N_0}$ = 10dB

$E_b = \frac{d^2}{12}(M^2-1) = \frac{2^2}{12}(2^2-1) = 1$ and $N_0 = \frac{E_b}{10^{0.1Variance}} = \frac{1}{10^{0.1\times 10}} = 0.1$

SNR = $\frac{P}{N_0 w} = \frac{E_b}{N_0}\frac{R_s}{w} = 10 \times \log_2(M)\,/N = 10 = 10\text{dB}$

1. Shaping with Sinc pulse:

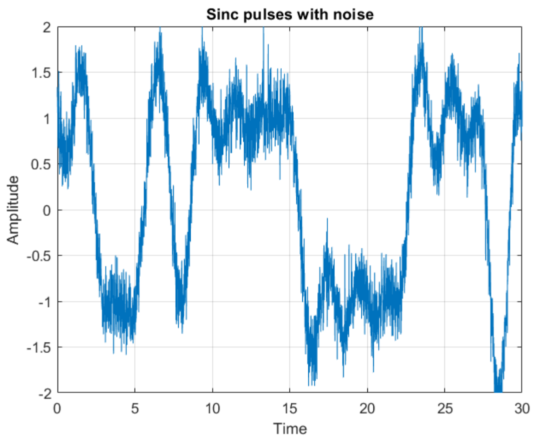


Figure 2.2 – impulse train convoluted with Sinc pulse with noise.

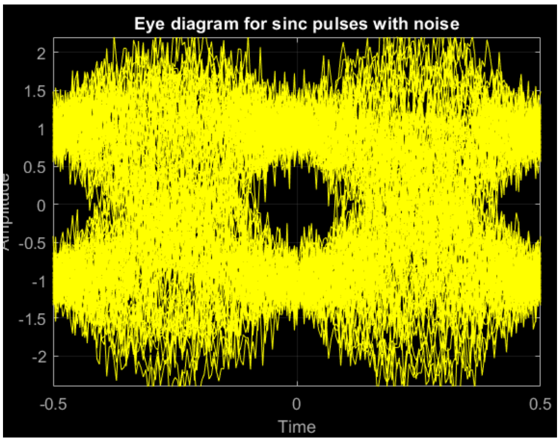


Figure 2.3 – Eye diagram for Figure 2.1

2. Shaping with raised cosine pulse (roll-off = 0.5):

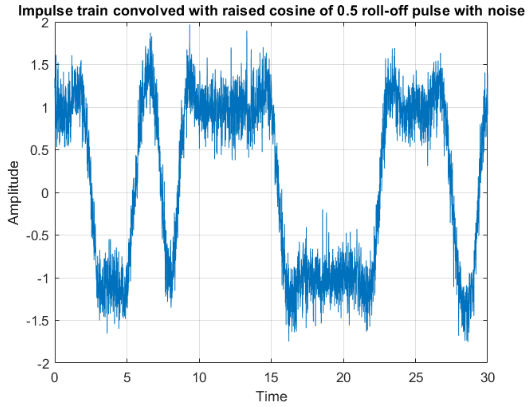


Figure 2.4 – impulse train convoluted with raised cosine of 0.5 roll-off.

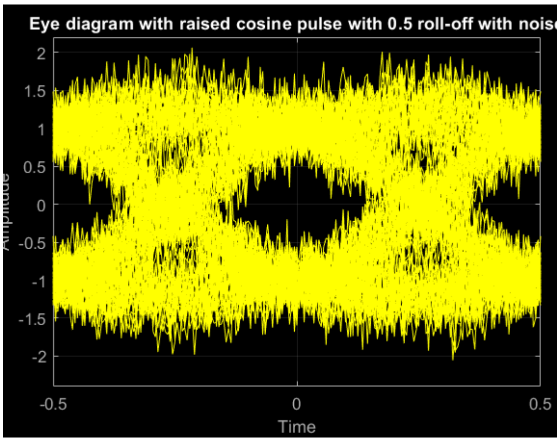


Figure 2.5 – Eye diagram for Figure 2.4

3. Shaping with raised cosine pulse (roll-off = 1.0):

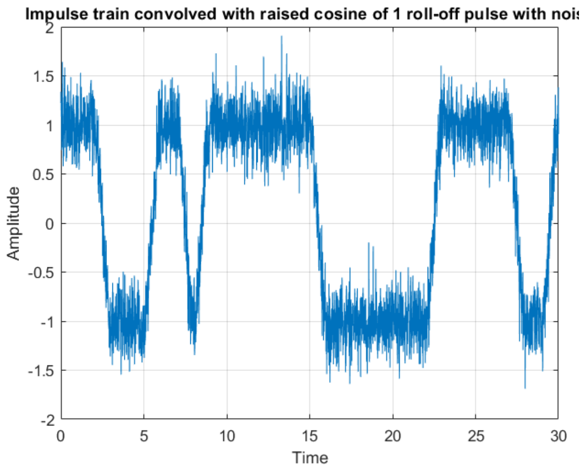


Figure 2.6 – impulse train convoluted with raised cosine of 1 roll-off.

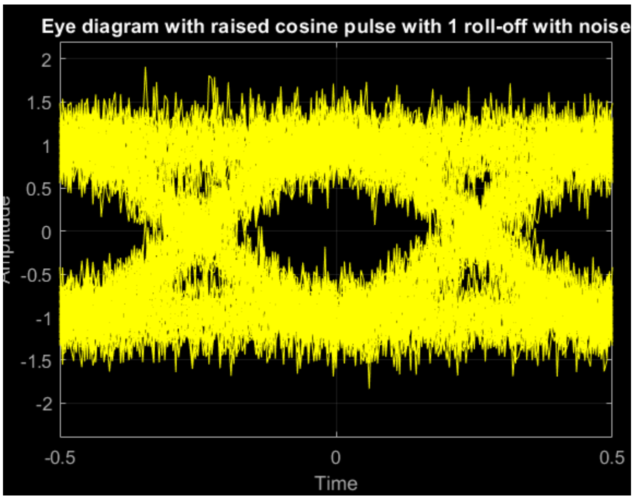


Figure 2.7 – Eye diagram for Figure 2.6

Property of the eye diagram	**Information**	**Sinc Pulse**	**Raised cosine pulse (roll-off = 0.5)**	**Raised cosine pulse (roll-off = 1.0)**
Vertical opening	Noise immunity	It is lower than when there is no noise. Which means low SNR at the sampling. High bit error occurrences at sampling.	Height has reduced but it has an opening greater than the sinc pulse. But there can be sampling errors more than before.	Height has reduced than the without noise channel. But among the channels with noise, this is the more robust one.

Horizontal opening	ISI-free sampling region	Width has further reduced. Almost all the regions are not good for sampling without errors.	Horizontal opening has significantly reduced but it has some regions for error-free sampling. Better than the sinc pulse.	Reduced but it maintains an adequate width for error-free sampling. High robustness to sampling errors than the other two.
Slope	Synchronization error	Very high slope. High possibility of synchronization errors.	Low slope. Low possibility of synchronization error occurrence.	Average slope, average chance of synchronization errors.
Level crossing ambiguity	Jitter	Jitter has further increased. Lowes robustness to jitter or timing offsets.	Jitter has increased. Average robustness to jitter.	Earlier there was no jitter present but now some jitter is visible. Still, it is the lowest of the three hence better robustness for jitter is shown.
Max overshoot	Variation of the peak value of the received signal	The peak deviation is preset due to the presence of the SNR value at the sampling region. This causes some distortions in the pulse amplitudes which in turn can cause sampling errors.	The peak deviation is preset due to the presence of the SNR value at the sampling region. This causes some distortions in the pulse amplitudes. Which in turn can cause sampling errors.	The peak deviation is preset due to the presence of the SNR value at the sampling region. This causes some distortions in the pulse amplitudes which in turn can cause sampling errors.

Table 2.1 – Comparison between three pulses.

Multipath Channel and Zero Force Equalizer

Up to now we have not had simulated channel effects other than the noise. Let us discuss the effect of the multipath channel for the eye diagram and how to reduce that effect using a zero forcing equalizer and simulation.

Effect of the n-tap multipath channel

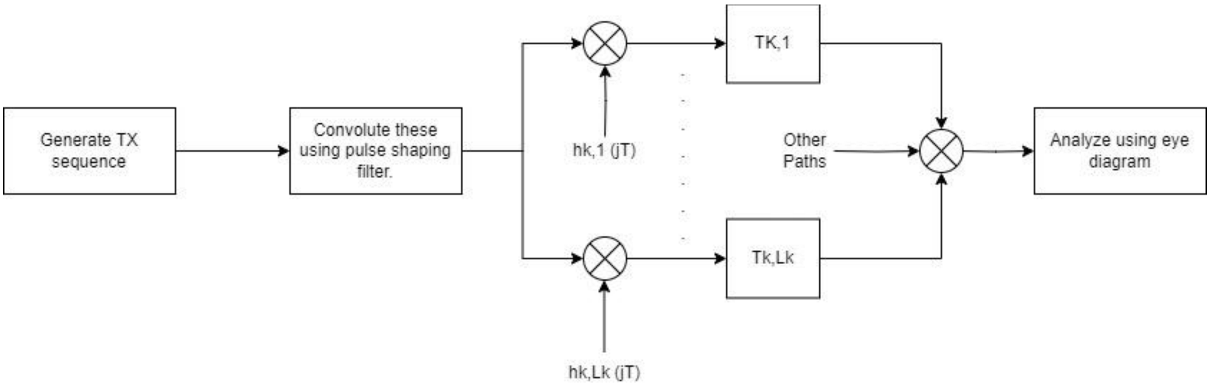


Figure 2.8 – N-tap multipath channel block diagram.

For the simulation, let us analyze the effect on the eye diagram from the 3-tap multipath channel.

Effect on the received pulse trains by 3-tap multipath channel:

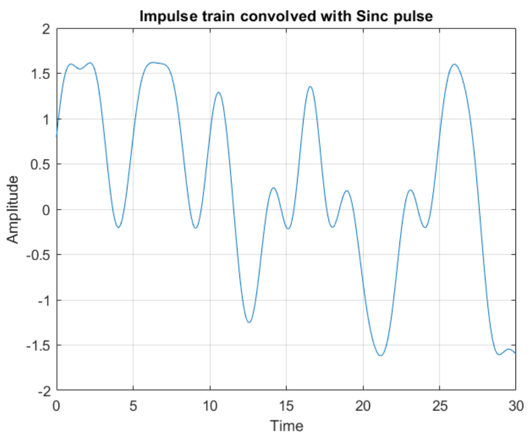


Figure 2.9 – impulse train convoluted with sinc pulse.

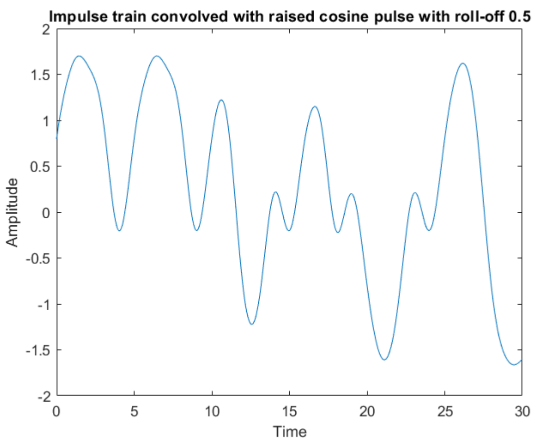


Figure 2.10 – impulse train convoluted with raised cosine of 0.5 roll-off.

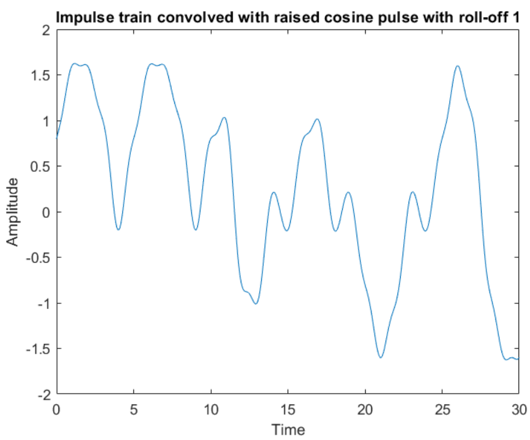


Figure 2.11 – impulse train convoluted with raised cosine of 1 roll-off.

Effect on the eye-diagrams by 3-tap multipath channel:

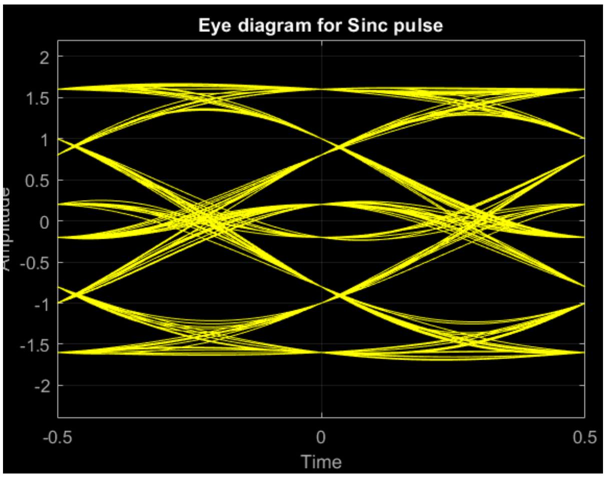


Figure 2.12 – Eye diagram for Figure 2.9.

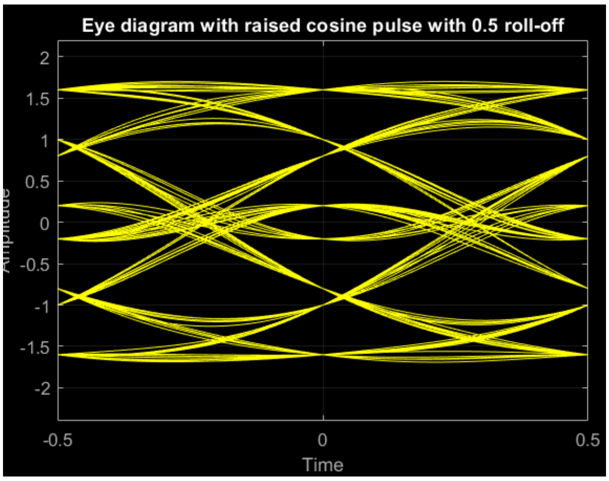


Figure 2.13 – Eye diagram for Figure 2.10.

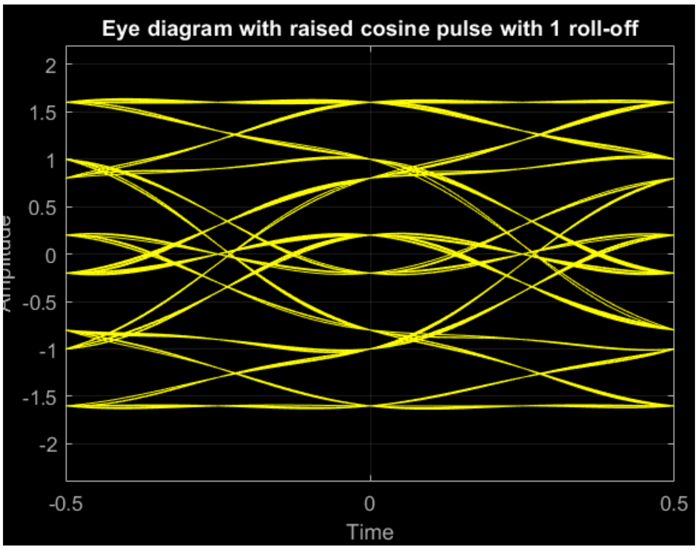


Figure 2.14 – Eye diagram for Figure 2.11.

Compensate the effect of the n-tap multipath channel using Zero-Forcing Equalizer

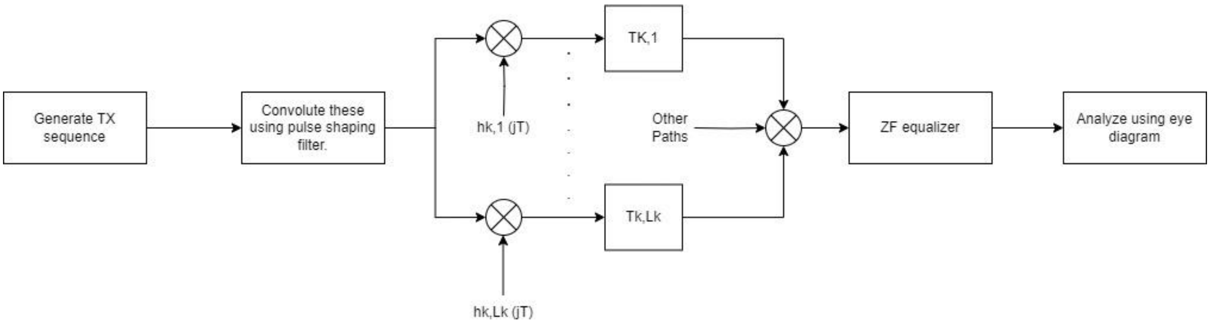


Figure 2.15 – Block diagram for multipath channel using ZF equalizer.

Using the ZF equation, we can compensate for the effect of the multipath channel. By increasing the number of taps of the equalizer (M-tap equalizer), we can reduce the effect of ISI due to the multipath channel. For now, let us discuss 9-tap equalizer.

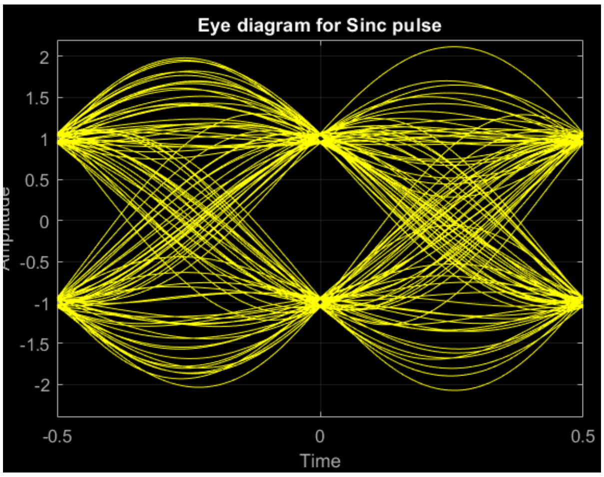


Figure 2.16 – Eye diagram for sinc pulse with ZF equalizer.

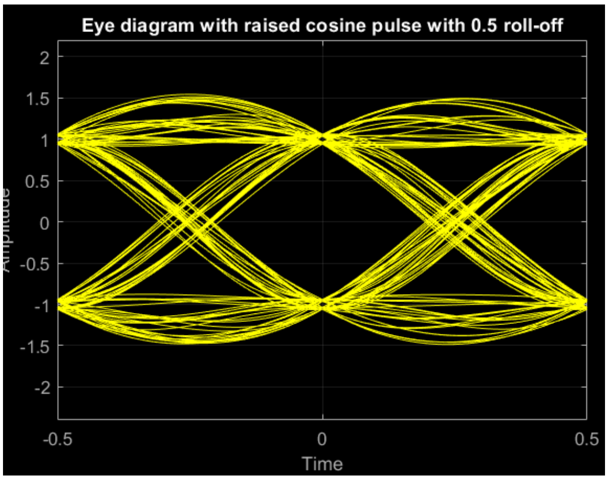


Figure 2.17 – Eye diagram for raised cosine with 0.5 roll-off with ZF equalizer.

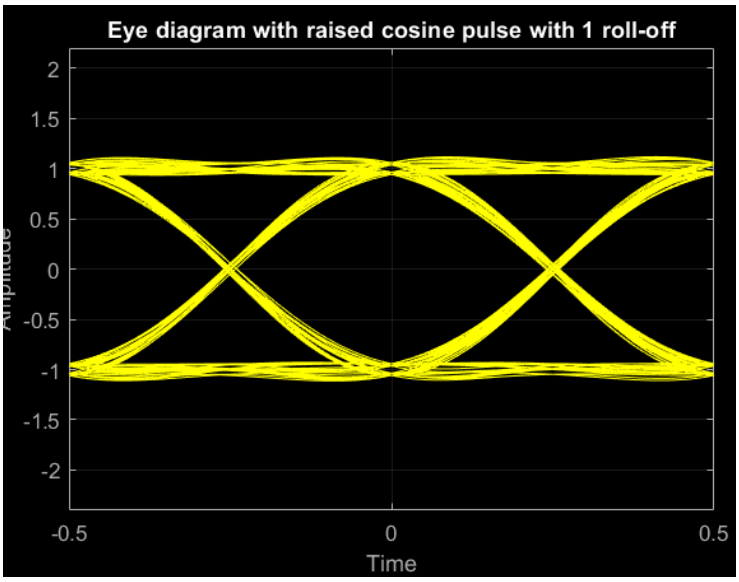
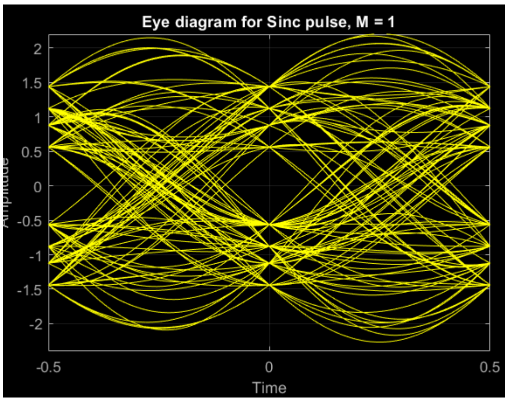
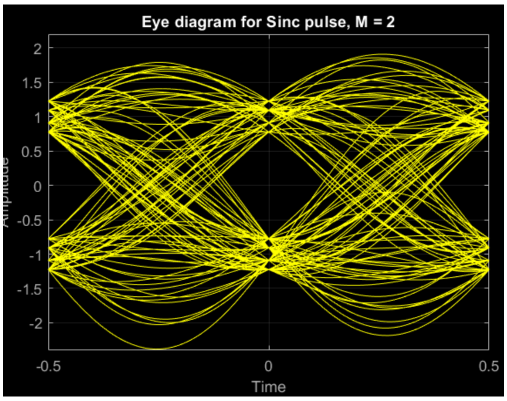


Figure 2.18 – Eye diagram for raised cosine with 1 roll-off with ZF equalizer.

Let us see how the effect of the multipath channel reduces with increasing the number of taps of the ZF equalizer:





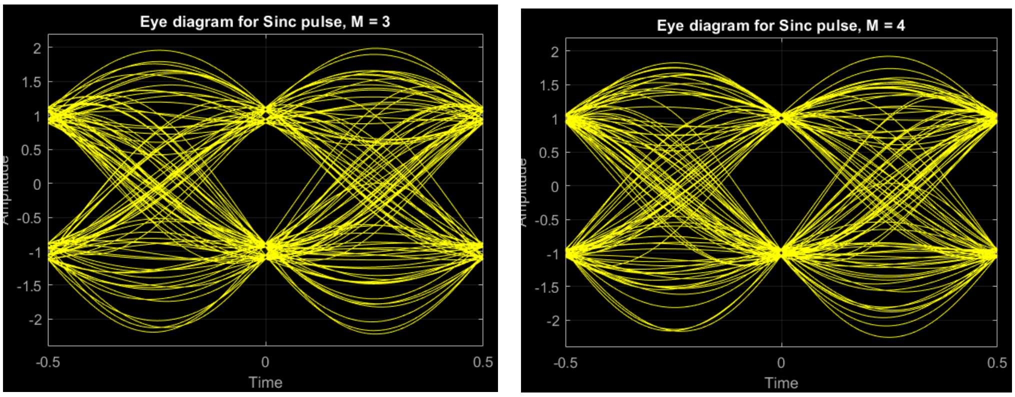


Figure 2.19 – Eye diagrams for sinc pulses for different no. of taps.

The effect of ISI has significantly reduced with the increment of the number of taps in the equalizer.

Task 03

In this task we will be calculating the BER values for 3,4,5, and 9-tap equalizer output samples and, we will be looking at the BER values for the AWGN channel (i.e. without multipath). We will be comparing each BER values w.r.t. SNR (Here E_b/N_0).

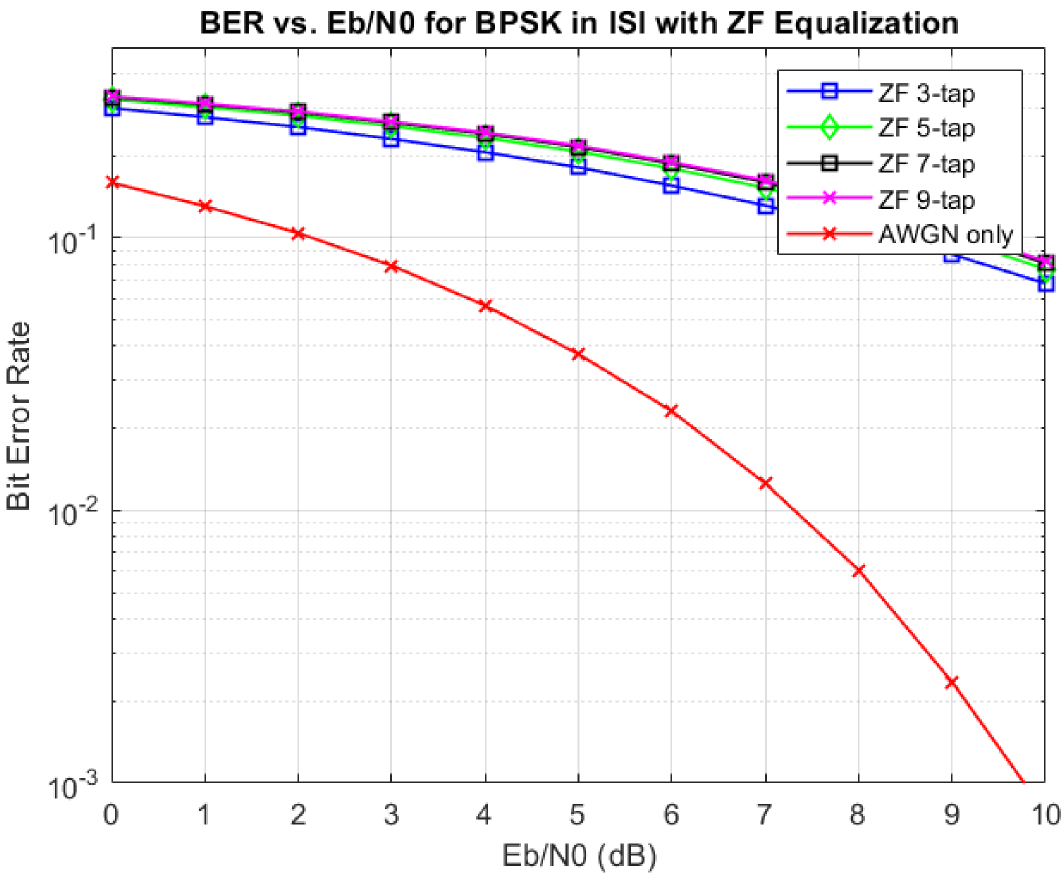


Figure 3.1 – BER vs SNR for BPSK with ZF equalizer.

Q11

In the AWGN channel, no ISI happens due to pulse shapes (Note: ISI can happen due to noise. But ISI does not happen due to transmitted pulses). Therefore, the amount of ISI that can happen in a multipath channel is very high w.r.t AWGN channel even though we use an equalizer. Also, there can be many corrupted samples due to this multipath effect before inserting it into the equalizer. That means the sample can go to the opposite side of the zero-crossing due to higher ISI and the effect of the noise. Also, when we are doing the equalizing, this might cause errors. The reason is the noise. The equalizer does not manage the noises. So, this addition of noises exactly can cause errors. These reasons will make a significant difference between BER values and with AWGN channel case. Alternative equalization techniques like Minimum Mean Square Error (MMSE) equalizers, which consider both ISI and noise, might provide better performance in such scenarios by balancing the trade-offs between noise enhancement and ISI elimination.

Q12

In BPSK the 2 symbols are represented by using 2 antipodal points on the constellation. This setup will maximize the distance between 2 signals within a given energy. Having a larger distance will be more robust to errors due to multipath and noises.
On the other hand, Binary Orthogonal Signaling use 2 orthogonal signal schemes. Let us see Sin and Cos on the constellation as an example. They are orthogonal to each other. But the distance between them is not at the maximum. Therefore, the probability to predict One pulse as the other is high. Error probability due to noise and multipath channel is high. Therefore, BER is high for a given energy w.r.t. BPSK.
Moreover, when considering multipath effects compounded using a Zero Forcing (ZF) equalizer, both BPSK and BOS face challenges, but differently. Even though the ZF equalizer can compensate the errors of pulses, it cannot compensate the noises. Sometimes it might add more noises for a particular sample in both the BPSK and BOS modulation. As we illustrated previously, since the distance between 2 signals is low in BOS, BOS has a higher error probability due to this. This could lead to scenarios where signals become less orthogonal due to channel and equalizer distortions, exacerbating error rates more significantly than in BPSK.
Therefore, using BPSK is more robust and BER value will be less compared to Binary orthogonal signaling including both AWGN and multipath scenarios with ZF equalization.

Simulations in MATLAB

MATLAB R2022a is the simulation tool for all the simulations in this assignment.

Tasks 01 & 02

```
% Task 1 & 2
clear;
close all;
clc;

% Initializing the needed parameters
samp_freq = 100;
no_trans_bits = 10^3;
time = 0:1/samp_freq:999/samp_freq;
t = -samp_freq:1/(samp_freq):samp_freq;

% Creating 0,1ly and converting them to 1,-1
```

```
BPSK = 2*(rand(1,no_trans_bits)>0.5)-1;


% Plotting the BPSK Impulse train
figure;
stem(time, BPSK(1:1000)); xlabel('Time'); ylabel('Amplitude of Impulse');
title('BPSK Impulse Train');
axis([0 10 -2 2]);
grid on;

% Sinc function
Sinc = sinc(t);

% Plotting the Sinc function
figure;
plot(t, Sinc);
title('Sinc Pulse');
xlabel('Time');
ylabel('Amplitude');
axis([-10 10 -1 1.2]);
grid on;

% Upsampling the BPSK impulse array to adjust to the sampling frequency
N = length(BPSK);
upsampleFactor = 100; % Since you are appending 99 zeros after each BPSK element

% Pre-allocate BPSK_U with the correct size
BPSK_U = zeros(1, N * upsampleFactor);

% Index to keep track of the insertion point in BPSK_U
index = 1;

% Loop through each element in BPSK
for i = 1:N
    BPSK_U(index) = BPSK(i);  % Insert the BPSK value
    index = index + upsampleFactor; % Increment index by 100 to skip 99 zeros
end

time_u = 0:1/samp_freq:99999/samp_freq;
% Plotting the upsampled BPSK Impulse train
figure;
stem(time_u, BPSK_U);
xlabel('Time');
ylabel('Amplitude');
title('BPSK Upsampled impulse train');
axis([0 5 -1.2 1.2]);
grid on;

% Plotting the diagram for impulse train convolved with sinc pulse
figure;
sinc_draw = conv(Sinc,BPSK_U,'same');
plot(t,sinc_draw);
title('Impulse train convolved with Sinc pulse');
xlabel('Time');
ylabel('Amplitude');
axis([0 30 -2 2]);
grid on;

% Plotting the eye diagram for Sinc pulse transmission
eyediagram(sinc_draw, 2*samp_freq);
```

```
title('Eye diagram for Sinc pulse');
xlabel('Time');
ylabel('Amplitude');
axis([-0.5 0.5 -2.4 2.2]);
grid on;

% Developing the raised cosine with 0.5 roll-off
roll_off = 0.5;
cos_num = cos(roll_off*pi*t);
cos_den = (1 - (2 * roll_off * t).^2);
cos_denzero = abs(cos_den)<10^-10;
Raised_cosine = cos_num./cos_den;
Raised_cosine(cos_denzero) = pi/4;
rc_roll5 = Sinc.*Raised_cosine;

% Plotting the raised cosine with roll off 0.5
figure;
plot(t,rc_roll5);
title('Raised cosine Pulse shape with 0.5 roll-off');
xlabel('Time');
ylabel('Amplitude');
axis([-10 10 -1 1.2]);
grid on;

% Plotting the diagram for impulse train convolved with raised cosine pulse of 0.5
roll-off factor
figure;
rc_roll5_draw = conv(rc_roll5,BPSK_U,'same');
plot(t,rc_roll5_draw);
title('Impulse train convolved with raised cosine pulse with roll-off 0.5');
xlabel('Time');
ylabel('Amplitude');
axis([0 30 -2 2]);

% Plotting the eye diagram for raised cosine pulse with 0.5 roll-off transmission
eyediagram(rc_roll5_draw,2*samp_freq);
title('Eye diagram with raised cosine pulse with 0.5 roll-off');
xlabel('Time');
ylabel('Amplitude');
axis([-0.5 0.5 -2.4 2.2]);
grid on;

% Developing the raised cosine with 1 roll-off
roll_off = 1;
cos_num = cos(roll_off*pi*t);
cos_den = (1 - (2 * roll_off * t).^2);
cos_denzero = abs(cos_den)<10^-10;
Raised_cosine = cos_num./cos_den;
Raised_cosine(cos_denzero) = pi/4;
rc_roll_1 = Sinc.*Raised_cosine;

% Plotting the raised cosine with roll off 1
figure;
plot(t,rc_roll_1);
title('Raised cosine Pulse shape with 1 roll-off');
xlabel('Time');
ylabel('Amplitude');
axis([-10 10 -1 1.2]);
grid on;
```

```
% Plotting the diagram for impulse train convolved with raised cosine pulse of 1
roll-off factor
figure;
rc_roll1_draw = conv(rc_roll_1,BPSK_U,'same');
plot(t,rc_roll1_draw);
title('Impulse train convolved with raised cosine pulse with roll-off 1');
xlabel('Time');
ylabel('Amplitude');
axis([0 30 -2 2]);

% Plotting the eye diagram for raised cosine pulse with 1 roll-off transmission
eyediagram(rc_roll1_draw, 2*samp_freq);
title('Eye diagram with raised cosine pulse with 1 roll-off');
xlabel('Time');
ylabel('Amplitude');
axis([-0.5 0.5 -2.4 2.2]);
grid on;

% Task 2

SNR=10; %given SNR
power_noise= 1./(10.^(0.1*SNR));
%generating AWGN noise
sinc_conv=conv(Sinc,BPSK_U,'same');
noise =((power_noise/2)^0.5)*randn(1,length(sinc_conv));
sinc_noise=sinc_conv+noise;

% Plotting the convolved sinc pulse with noise
figure;
plot(t,sinc_noise(1:20001));
title('Sinc pulses with noise');
xlabel('Time');
ylabel('Amplitude');
axis([0 30 -2 2]);
grid on;

% Drawing the eye diagram for sinc pulse with noise
eyediagram(sinc_noise, samp_freq*2);
title('Eye diagram for sinc pulses with noise');
xlabel('Time');
ylabel('Amplitude');
axis([-0.5 0.5 -2.4 2.2]);
grid on;

% Adding the noise to the convoluted raised cosine pulse with 0.5 roll-off
rc_roll5_conv = conv(rc_roll5,BPSK_U,'same');
rc_5_noise=rc_roll5_conv+noise;

% Plotting the convolved raised cosine pulse of 0.5 roll-off with noise
figure;
plot(t,rc_5_noise(1:20001));
title('Impulse train convolved with raised cosine of 0.5 roll-off pulse with
noise');
xlabel('Time');
ylabel('Amplitude');
axis([0 30 -2 2]);
grid on;

% Drawing the eye diagram for raised cosine pulse of 0.5 roll-off with noise
eyediagram(rc_5_noise, samp_freq*2);
```

```
title('Eye diagram with raised cosine pulse with 0.5 roll-off with noise');
xlabel('Time');
ylabel('Amplitude');
axis([-0.5 0.5 -2.4 2.2]);
grid on;

% Adding the noise to the convoluted raised cosine pulse with 1 roll-off
rc_roll1_conv = conv(rc_roll_1,BPSK_U,'same');
rc_1_noise=rc_roll1_conv+noise;

% Plotting the convolved raised cosine pulse of 1 roll-off with noise
figure;
plot(t,rc_1_noise(1:20001));
title('Impulse train convolved with raised cosine of 1 roll-off pulse with noise');
xlabel('Time');
ylabel('Amplitude');
axis([0 30 -2 2]);
grid on;

% Drawing the eye diagram for raised cosine pulse of 1 roll-off with noise
eyediagram(rc_1_noise, 2*samp_freq);
title('Eye diagram with raised cosine pulse with 1 roll-off with noise');
xlabel('Time');
ylabel('Amplitude');
axis([-0.5 0.5 -2.4 2.2]);
grid on;
```

Task 03

```
% Task 3

clear;
close all;
clc;

% Simulation parameters
numTransmitBits = 10^6; % Number of bits to transmit
SNR_dB = 0:10; % SNR range from 0 to 10 dB
M = 4; % Maximum number of taps in equalizers used

% Pre-allocate the error matrix
errors = zeros(M+1, length(SNR_dB)); % Rows for different equalizers, columns for SNR values

% Transmitter
bits = rand(1, numTransmitBits) > 0.5; % Random bit generation
BPSK = 2*bits - 1; % BPSK modulation (0 mapped to -1, 1 mapped to 1)

% Define the multipath channel
numTaps = 3;
channelTaps = [0.3, 0.9, 0.4];
channelOutput = conv(BPSK, channelTaps);

for i = 1:length(SNR_dB)
    % Apply AWGN to the channel output
    noisyOutput = awgn(channelOutput, SNR_dB(i), 'measured');
```

```
    for k = 1:M
        % Construct the diagonal matrix for equalizer
        equalizerMatrix = toeplitz([channelTaps(2:end), zeros(1, 2*k+1-
numTaps+1)], ...
            [channelTaps(2:-1:1), zeros(1, 2*k+1-numTaps+1)]);
        targetImpulse = zeros(1, 2*k+1);
        targetImpulse(k+1) = 1; % Target impulse response
        equalizerCoeffs = equalizerMatrix \ targetImpulse'; % Least squares
solution for equalizer coefficients

        % Filter the noisy output with the equalizer
        equalizedOutput = conv(noisyOutput, equalizerCoeffs);
        equalizedOutput = equalizedOutput(k+2:end); % Compensate for filter delay

        % Sample and decode
        decodedBits = real(equalizedOutput(1:numTransmitBits)) > 0;

        % Count the errors
        errors(k+1, i) = sum(bits ~= decodedBits);
    end
end

% Additional AWGN channel processing without multipath
for i = 1:length(SNR_dB)
    awgnOutput = awgn(BPSK, SNR_dB(i));
    decodedBitsAWGN = real(awgnOutput(1:numTransmitBits)) > 0;
    errors(1, i) = sum(bits ~= decodedBitsAWGN);
end

% Calculate Bit Error Rate (BER)
simulatedBER = errors / numTransmitBits;

% Plot BER vs SNR for different equalizers
figure;
semilogy(SNR_dB, simulatedBER(2,:), 'bs-', 'Linewidth', 1); hold on;
semilogy(SNR_dB, simulatedBER(3,:), 'gd-', 'Linewidth', 1);
semilogy(SNR_dB, simulatedBER(4,:), 'ks-', 'Linewidth', 1);
semilogy(SNR_dB, simulatedBER(5,:), 'mx-', 'Linewidth', 1);
semilogy(SNR_dB, simulatedBER(1,:), 'rx-', 'Linewidth', 1);
axis([0 10 10^-3 0.5]);
grid on;
legend('ZF 3-tap', 'ZF 5-tap', 'ZF 7-tap', 'ZF 9-tap', 'AWGN only');
xlabel('Eb/N0 (dB)');
ylabel('Bit Error Rate');
title('BER vs. Eb/N0 for BPSK in ISI with ZF Equalization');
```

Task 03 – Multipath Effect Analysis on Eye-Diagram

```
%Task 3 - Multipath Effect on Eye-Diagrams

clear;
close all;
clc;

% Initializing the needed parameters
samp_freq = 100;
no_trans_bits = 10^3;
```

```
time = 0:1/samp_freq:999/samp_freq;
t = -samp_freq:1/(samp_freq):samp_freq;

% Creating 0,1ly and converting them to 1,-1
BPSK = 2*(rand(1,no_trans_bits)>0.5)-1;

% Define the multipath channel
numTaps = 3;
channelTaps = [0.3, 0.9, 0.4];
channelOutput = conv(BPSK, channelTaps);

% Sinc function
Sinc = sinc(t);

% Upsampling the BPSK impulse array to adjust to the sampling frequency
N = length(BPSK);
upsampleFactor = 100; % Since you are appending 99 zeros after each BPSK element

% Pre-allocate BPSK_U with the correct size
BPSK_U = zeros(1, N * upsampleFactor);

% Index to keep track of the insertion point in BPSK_U
index = 1;

% Loop through each element in BPSK
for i = 1:N
    BPSK_U(index) = channelOutput(i);  % Insert the BPSK value
    index = index + upsampleFactor; % Increment index by 100 to skip 99 zeros
end

time_u = 0:1/samp_freq:99999/samp_freq;
% Plotting the upsampled BPSK Impulse train
figure;
stem(time_u, BPSK_U);
xlabel('Time');
ylabel('Amplitude');
title('BPSK Upsampled impulse train');
axis([0 5 -1.2 1.2]);
grid on;

% Plotting the diagram for impulse train convolved with sinc pulse
figure;
sinc_draw = conv(Sinc,BPSK_U,'same');
plot(t,sinc_draw);
title('Impulse train convolved with Sinc pulse');
xlabel('Time');
ylabel('Amplitude');
axis([0 30 -2 2]);
grid on;

% Plotting the eye diagram for Sinc pulse transmission
eyediagram(sinc_draw, 2*samp_freq);
title('Eye diagram for Sinc pulse');
xlabel('Time');
ylabel('Amplitude');
axis([-0.5 0.5 -2.4 2.2]);
grid on;

% Developing the raised cosine with 0.5 roll-off
roll_off = 0.5;
```

```
cos_num = cos(roll_off*pi*t);
cos_den = (1 - (2 * roll_off * t).^2);
cos_denzero = abs(cos_den)<10^-10;
Raised_cosine = cos_num./cos_den;
Raised_cosine(cos_denzero) = pi/4;
rc_roll5 = Sinc.*Raised_cosine;

% Plotting the raised cosine with roll off 0.5
figure;
plot(t,rc_roll5);
title('Raised cosine Pulse shape with 0.5 roll-off');
xlabel('Time');
ylabel('Amplitude');
axis([-10 10 -1 1.2]);
grid on;

% Plotting the diagram for impulse train convolved with raised cosine pulse of 0.5
roll-off factor
figure;
rc_roll5_draw = conv(rc_roll5,BPSK_U,'same');
plot(t,rc_roll5_draw);
title('Impulse train convolved with raised cosine pulse with roll-off 0.5');
xlabel('Time');
ylabel('Amplitude');
axis([0 30 -2 2]);

% Plotting the eye diagram for raised cosine pulse with 0.5 roll-off transmission
eyediagram(rc_roll5_draw,2*samp_freq);
title('Eye diagram with raised cosine pulse with 0.5 roll-off');
xlabel('Time');
ylabel('Amplitude');
axis([-0.5 0.5 -2.4 2.2]);
grid on;

% Developing the raised cosine with 1 roll-off
roll_off = 1;
cos_num = cos(roll_off*pi*t);
cos_den = (1 - (2 * roll_off * t).^2);
cos_denzero = abs(cos_den)<10^-10;
Raised_cosine = cos_num./cos_den;
Raised_cosine(cos_denzero) = pi/4;
rc_roll_1 = Sinc.*Raised_cosine;

% Plotting the raised cosine with roll off 1
figure;
plot(t,rc_roll_1);
title('Raised cosine Pulse shape with 1 roll-off');
xlabel('Time');
ylabel('Amplitude');
axis([-10 10 -1 1.2]);
grid on;

% Plotting the diagram for impulse train convolved with raised cosine pulse of 1
roll-off factor
figure;
rc_roll1_draw = conv(rc_roll_1,BPSK_U,'same');
plot(t,rc_roll1_draw);
title('Impulse train convolved with raised cosine pulse with roll-off 1');
xlabel('Time');
ylabel('Amplitude');
```

```
axis([0 30 -2 2]);

% Plotting the eye diagram for raised cosine pulse with 1 roll-off transmission
eyediagram(rc_roll1_draw, 2*samp_freq);
title('Eye diagram with raised cosine pulse with 1 roll-off');
xlabel('Time');
ylabel('Amplitude');
axis([-0.5 0.5 -2.4 2.2]);
grid on;

% With noise addition

SNR=10; %given SNR
power_noise= 1./(10.^(0.1*SNR));
%generating AWGN noise
sinc_conv=conv(Sinc,BPSK_U,'same');
noise =((power_noise/2)^0.5)*randn(1,length(sinc_conv));
sinc_noise=sinc_conv+noise;

% Plotting the convolved sinc pulse with noise
figure;
plot(t,sinc_noise(1:20001));
title('Sinc pulses with noise');
xlabel('Time');
ylabel('Amplitude');
axis([0 30 -2 2]);
grid on;

% Drawing the eye diagram for sinc pulse with noise
eyediagram(sinc_noise, samp_freq*2);
title('Eye diagram for sinc pulses with noise');
xlabel('Time');
ylabel('Amplitude');
axis([-0.5 0.5 -2.4 2.2]);
grid on;

% Adding the noise to the convoluted raised cosine pulse with 0.5 roll-off
rc_roll5_conv = conv(rc_roll5,BPSK_U,'same');
rc_5_noise=rc_roll5_conv+noise;

% Plotting the convolved raised cosine pulse of 0.5 roll-off with noise
figure;
plot(t,rc_5_noise(1:20001));
title('Impulse train convolved with raised cosine of 0.5 roll-off pulse with
noise');
xlabel('Time');
ylabel('Amplitude');
axis([0 30 -2 2]);
grid on;

% Drawing the eye diagram for raised cosine pulse of 0.5 roll-off with noise
eyediagram(rc_5_noise, samp_freq*2);
title('Eye diagram with raised cosine pulse with 0.5 roll-off with noise');
xlabel('Time');
ylabel('Amplitude');
axis([-0.5 0.5 -2.4 2.2]);
grid on;

% Adding the noise to the convoluted raised cosine pulse with 1 roll-off
rc_roll1_conv = conv(rc_roll_1,BPSK_U,'same');
```

```
rc_1_noise=rc_roll1_conv+noise;

% Plotting the convolved raised cosine pulse of 1 roll-off with noise
figure;
plot(t,rc_1_noise(1:20001));
title('Impulse train convolved with raised cosine of 1 roll-off pulse with
noise');
xlabel('Time');
ylabel('Amplitude');
axis([0 30 -2 2]);
grid on;

% Drawing the eye diagram for raised cosine pulse of 1 roll-off with noise
eyediagram(rc_1_noise, 2*samp_freq);
title('Eye diagram with raised cosine pulse with 1 roll-off with noise');
xlabel('Time');
ylabel('Amplitude');
axis([-0.5 0.5 -2.4 2.2]);
grid on;
```

Task 03 – Effect of the Zero-Forcing Equalizer on Eye-Diagrams

```
%Task 3 - ZF Equalizer Effect on Eye-Diagram

clear;
close all;
clc;

% Initializing the needed parameters
samp_freq = 100;
no_trans_bits = 10^3;
time = 0:1/samp_freq:999/samp_freq;
t = -samp_freq:1/(samp_freq):samp_freq;

% Creating 0,1ly and converting them to 1,-1
BPSK = 2*(rand(1,no_trans_bits)>0.5)-1;

% Sinc function
Sinc = sinc(t);

% Upsampling the BPSK impulse array to adjust to the sampling frequency
N = length(BPSK);
upsampleFactor = 100; % Since you are appending 99 zeros after each BPSK element

% Pre-allocate BPSK_U with the correct size
BPSK_U = zeros(1, N * upsampleFactor);

% Index to keep track of the insertion point in BPSK_U
index = 1;

% Loop through each element in BPSK
for i = 1:N
    BPSK_U(index) = BPSK(i);  % Insert the BPSK value
    index = index + upsampleFactor; % Increment index by 100 to skip 99 zeros
end

time_u = 0:1/samp_freq:99999/samp_freq;
```

```
% Transmitter
bits = rand(1, no_trans_bits) > 0.5; % Random bit generation
BPSK = 2*bits - 1; % BPSK modulation (0 mapped to -1, 1 mapped to 1)

% Define the multipath channel
numTaps = 3;
channelTaps = [0.3, 0.9, 0.4];
channelOutput = conv(BPSK, channelTaps);

for k = 1:4
    % Construct the diagonal matrix for equalizer
    equalizerMatrix = toeplitz([channelTaps(2:end), zeros(1, 2*k+1-numTaps+1)],
...
        [channelTaps(2:-1:1), zeros(1, 2*k+1-numTaps+1)]);
    targetImpulse = zeros(1, 2*k+1);
    targetImpulse(k+1) = 1; % Target impulse response
    equalizerCoeffs = equalizerMatrix \ targetImpulse'; % Least squares solution
for equalizer coefficients

    % Filter the output with the equalizer
    equalizedOutput = conv(channelOutput, equalizerCoeffs);
    equalizedOutput = equalizedOutput(k+2:end); % Compensate for filter delay

    % Upsampling the BPSK impulse array to adjust to the sampling frequency
    N = length(BPSK);
    upsampleFactor = 100; % Since you are appending 99 zeros after each BPSK
element

    % Pre-allocate BPSK_U with the correct size
    BPSK_U = zeros(1, N * upsampleFactor);

    % Index to keep track of the insertion point in BPSK_U
    index = 1;

    % Loop through each element in BPSK
    for i = 1:N
        BPSK_U(index) = equalizedOutput(i);  % Insert the BPSK value
        index = index + upsampleFactor; % Increment index by 100 to skip 99 zeros
    end

    sinc_draw = conv(Sinc,BPSK_U,'same');

    % Plotting the eye diagram for Sinc pulse transmission
    eyediagram(sinc_draw, 2*samp_freq);
    title(sprintf('Eye Diagram for Sinc Pulse at M = %d ', k));
    xlabel('Time');
    ylabel('Amplitude');
    axis([-0.5 0.5 -2.4 2.2]);
    grid on;

end
```