DEPARTMENT OF ELECTRONICS AND TELECOMMUNICATION ENGINEERING

UNIVERSITY OF MORATUWA

EN2074 – Communication Systems Engineering

Eye diagrams and Equalization

Simulation Assignment

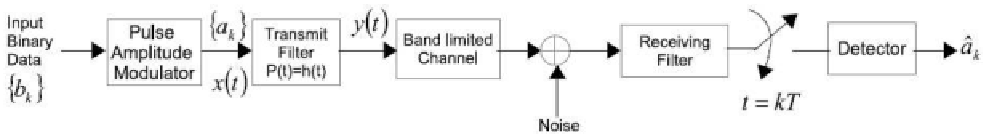
NAME		INDEX NO
Samarakoon S.M.R.K	-	200556L
De Silva A.L.U.P	-	200105F

ABSTRACT

In digital communication systems, modulated baseband signals need to be passed through a pulse-shaping filter with the necessary characteristics before transmission. The selected filter should be robust against noise, and synchronization errors. Here we are analyzing the performance of sinc and raised cosine pulse shaping filters. For analyzing the signal through the channel, we are using an ‘eye diagram’. We investigate the performance of 2-PAM (2-Level Pulse Amplitude Modulation) in a multipath fading channel with intersymbol interference (ISI) and additive white Gaussian noise (AWGN). The objective is to assess the Bit Error Rate (BER) performance of 2-PAM and evaluate the effectiveness of a Zero Forcing (ZF) equalizer in mitigating ISI.

INTRODUCTION

In the communication system messages are generated, transmitted, and received. The process in abstract view is as follows.



The source generates the impulses which are converted to BPSK symbols. Impulses cannot be transmitted as they are. Impulses are converted to suitable signals by transmitting through a pulse-shaping filter.

$$y(t) = \sum_{k=-M}^{M} a_k P(t - kT)$$

Here $a_k = The\ BPSK\ symbols$
$T = pulse\ duration$

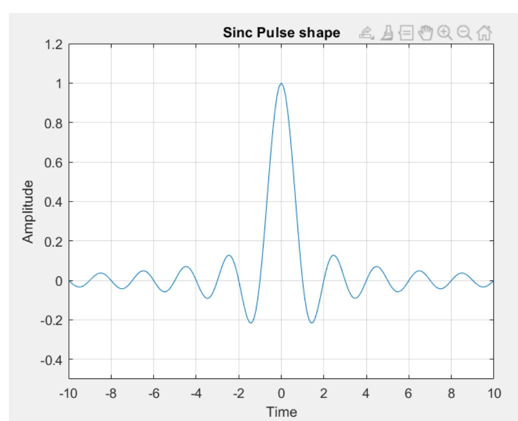
The channel is bandlimited. The bandwidth of the transmitting signal must therefore be smaller or equal to that of the channel. Bandwidth of the filter should be considered when choosing the filter.

Three pulse shaping filters are considered.

1. Sinc pulse

$$p(t) = \begin{cases} 1 & t = 0 \\ \frac{\sin(\pi R_b t)}{\pi R_b t} & t \neq 0 \end{cases}$$

Sinc pulse is not time limited. The decay is $\frac{1}{t}$ the side lobes of the pulse are large, resulting with time jitters.

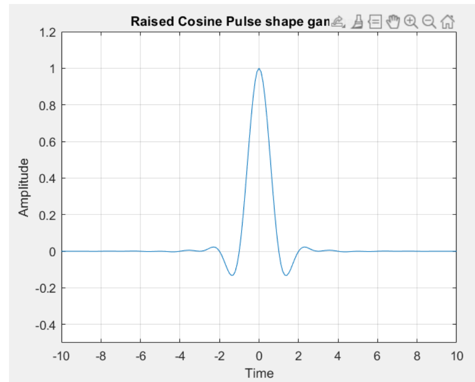


2. Raised cosine pulse with $\gamma = 0.5$

$$p(t) = \begin{cases} R_b & t = 0 \\ \dfrac{R_b \operatorname{sinc}(\pi R_b t) \cos(\pi \gamma R_b t)}{1-(2\gamma R_b t)^2} & t \neq 0 \end{cases}$$

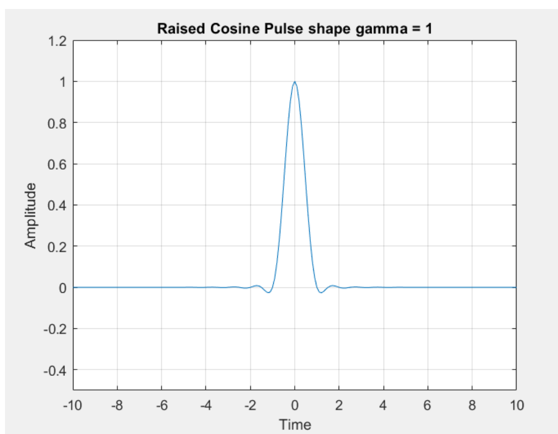
The decay is $\frac{1}{t^3}$ which makes the side lobes small.

Hence robust to synchronization errors.



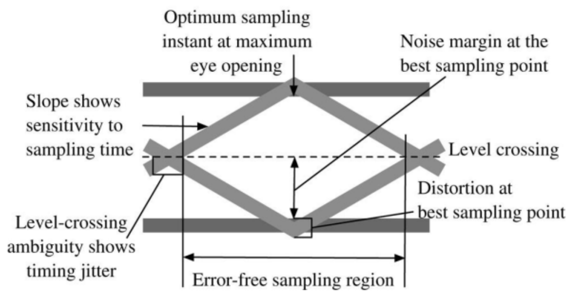
3. Raised cosine pulse with $\gamma = 1$

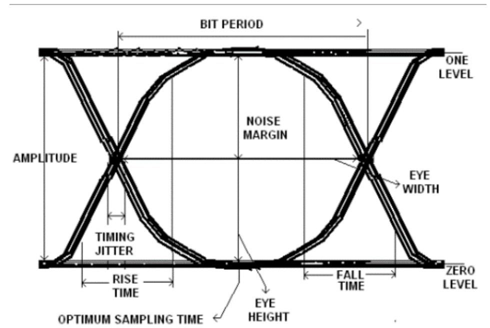
The decay is higher. The side lobes are even smaller.



Eye diagram

An eye diagram is a graphical representation of a digital signal's quality and integrity. It provides insights into various signal characteristics such as noise, timing, and distortion. The eye diagram is created by superimposing multiple consecutive signal waveform segments, aligned with the timing reference.





Some of the characteristics of the eye diagram are as follows.

a) Eye Height
   - Eye height is an indication of the noise margin.
   - Noise margin is defined as half the eye height.
   - Higher the height, higher the noise margin, better the channel.
   - This implies the high robustness to the AWGN introduced by the channel.

b) Eye Width
   - The eye width refers to the horizontal distance between the points where the eye-opening starts to close.
   - It represents the time duration during which the received signal transitions from one symbol to another.
   - Higher the data transmission rates are narrower the eye diagram is.
   - Wider the eye, lower the probability of happening errors due to sampling.

c) The slope of the eye
   - Slope is directly proportional to the immunity to synchronization errors.
   - Higher the slope, higher the synchronization errors that will occur.

d) Optimum sampling instant
   - This is the position with the maximum eye opening.
   - It is the best sampling point at which we should sample the pulses to extract the information.

e) Time variation of level-crossing
   - This shows the time jitter of the pulses transmitted through the channel.
   - This occurs when there is a mismatch between the rise time and the fall time of the pulses.
   - Lower the width of the level-crossing ambiguity, better robustness will be three for the ISIs.

f) Peak distortion
    - This shows the variation of peak amplitude of the pulse during transmission due to the AWGN.
    - Lower the value better robustness can be obtained for the AWGN.

Using the eye diagram and its' characteristics we can decide which pulse shaping filter is to be used and how the channel is reacting to it.

Additive White Gaussian Noise

The term "white" refers to the fact that AWGN has a flat power spectral density across all frequencies. This means that it has equal power at all frequencies. The term "Gaussian" refers to the probability distribution of the noise samples, which follows a Gaussian or normal distribution. The Gaussian distribution is characterized by a mean of zero and a standard deviation that determines the noise power. This noise affects the output signal to distort.

Zero Force Equalization

Zero Forcing Equalizer refers to a form of linear equalization algorithm used in communication systems which applies the inverse of the frequency response of the channel. The Zero-Forcing Equalizer applies the inverse of the channel frequency response to the received signal, to restore the signal after the channel. The name Zero Forcing corresponds to bringing down the intersymbol interference (ISI) to zero in a noise free case. This will be useful when ISI is significant compared to noise.

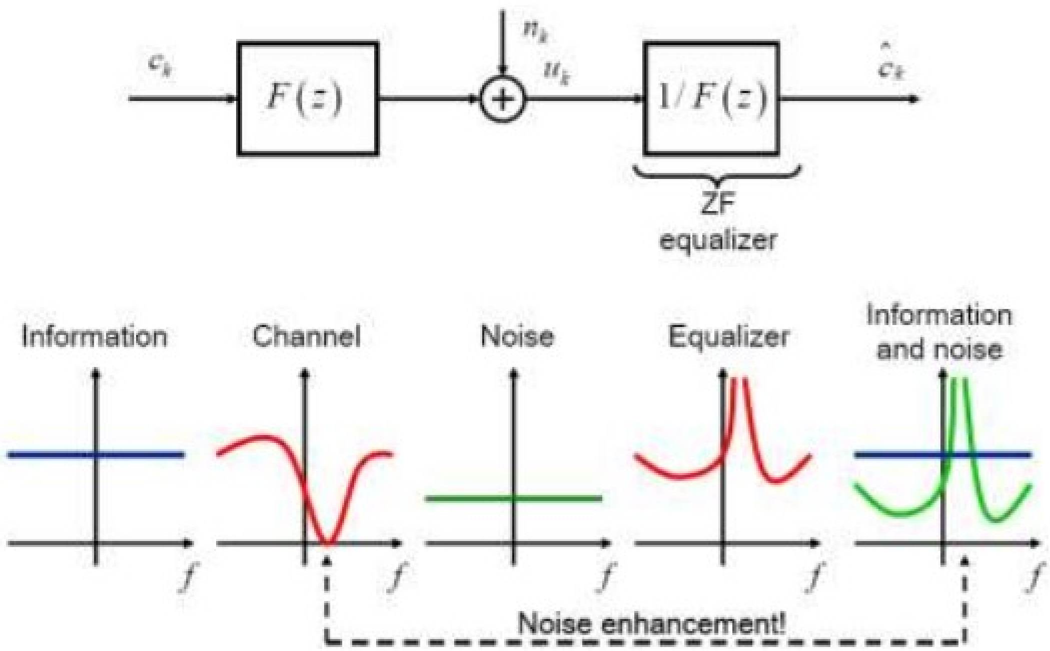
For a channel with frequency response $F(f)$ the zero forcing equalizer is constructed by:

$$C(f) = \frac{1}{F(f)}$$

Thus the combination of channel and equalizer gives a flat frequency response and linear phase $C(f)F(f) = 1$.

Zero-forcing equalization does not work in some applications, for the following reasons:

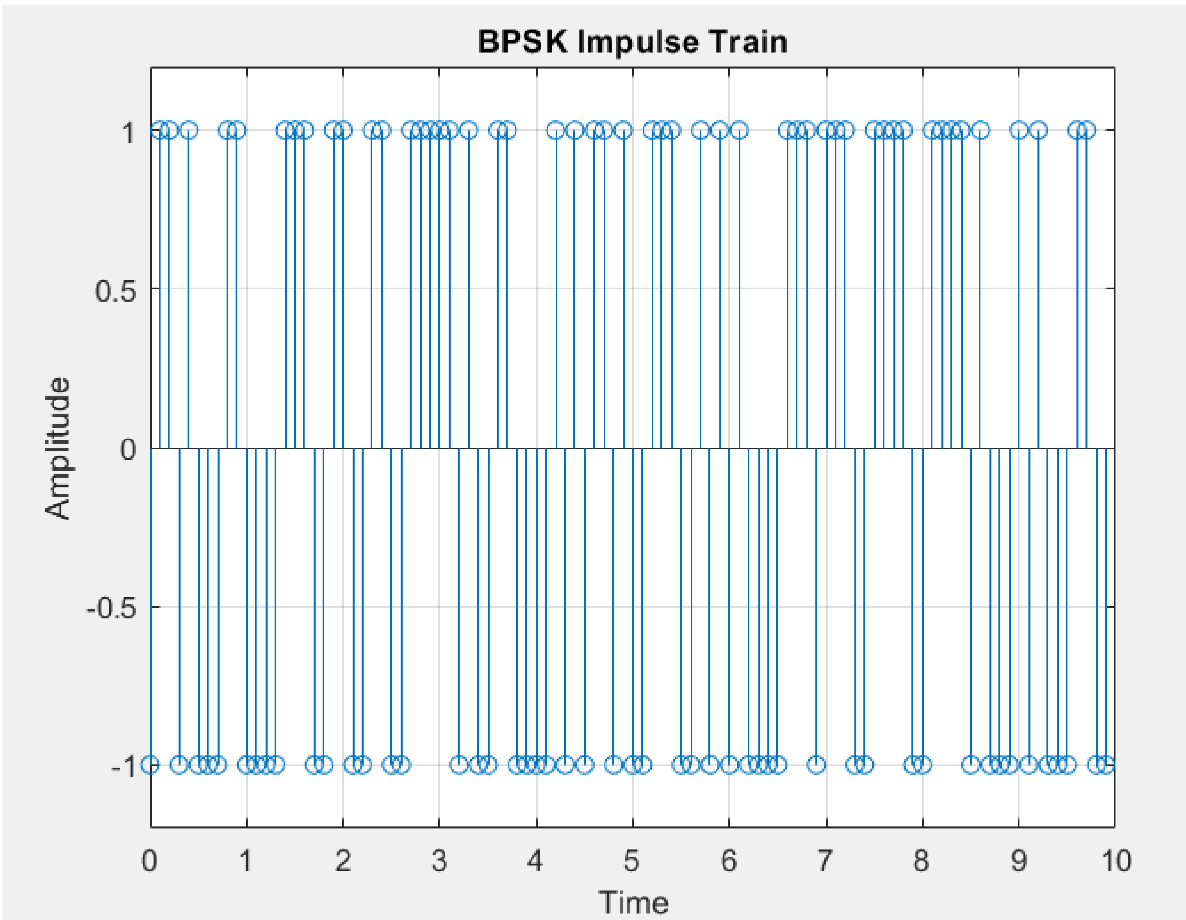
- Even though the channel impulse response has finite length, the impulse response of the equalizer needs to be infinitely long.
- At some frequencies the received signal may be weak. To compensate, the magnitude of the zero-forcing filter ("gain") grows very large. Consequently, any noise added after the channel gets boosted by a large factor and destroys the overall signal-to-noise ratio. Furthermore, the channel may have zeroes in its frequency response that cannot be inverted at all.



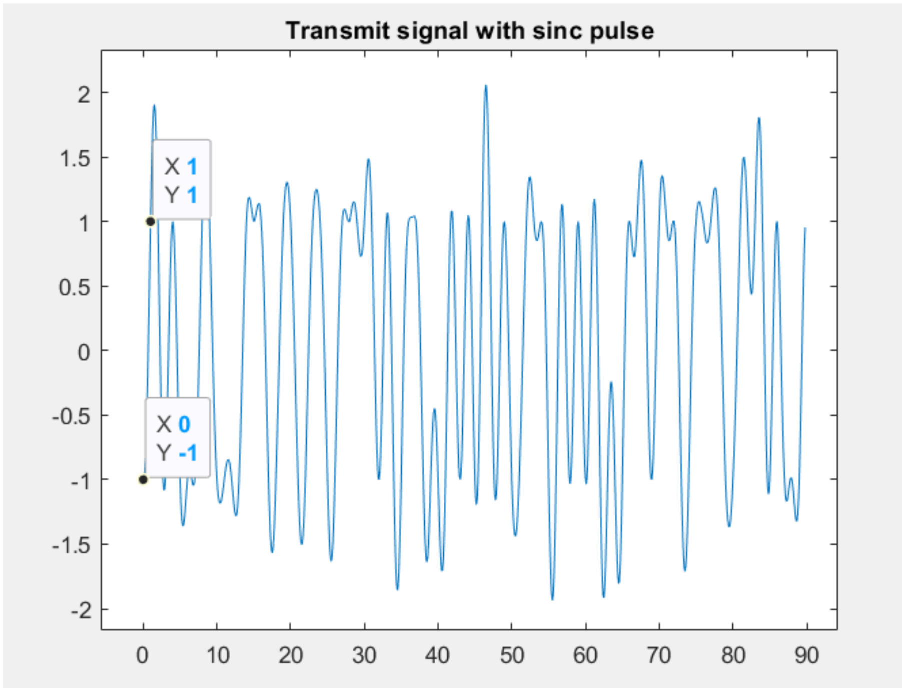
c_k
$F(z)$
n_k
u_k
$1/F(z)$
$\hat{c}_k$
ZF
equalizer
Information
Channel
Noise
Equalizer
Information
and noise
f
Noise enhancement!

Task 1

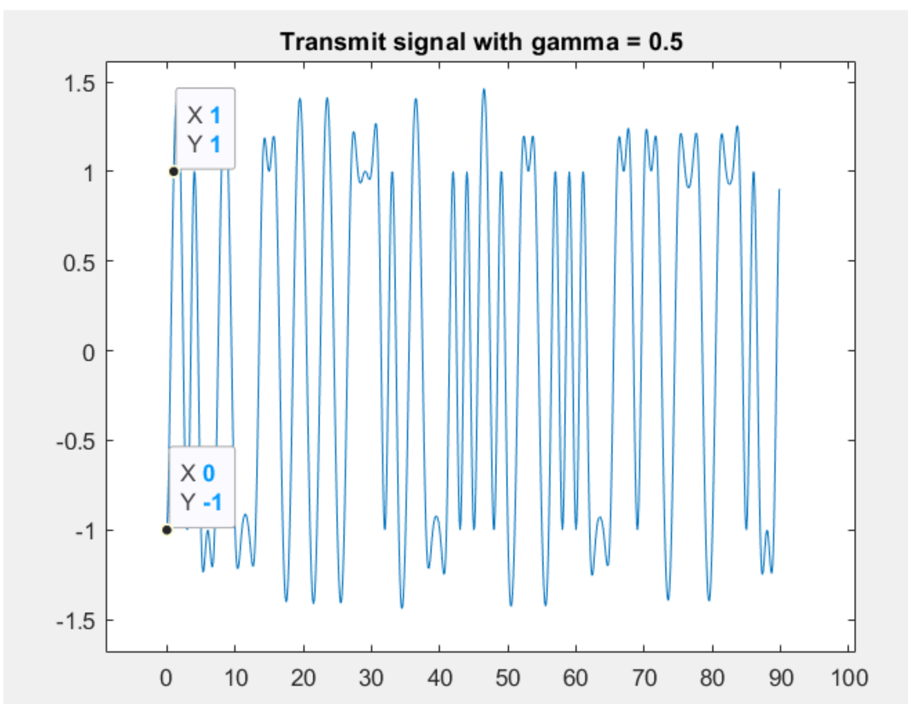
1. Impulse train representing BPSK symbols.



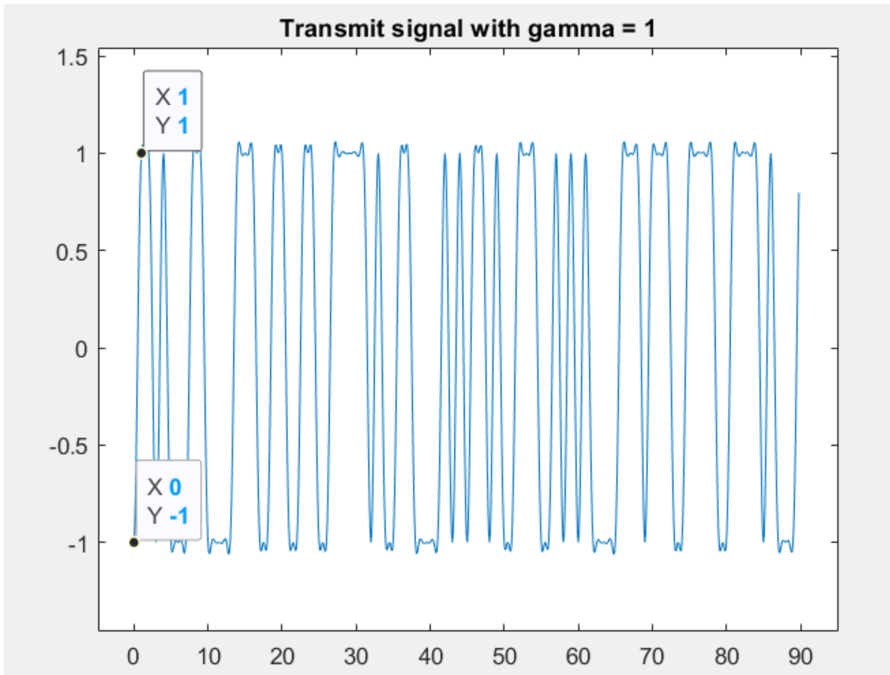
2. Transmit signal when,
   - Sinc pulse shaping filter



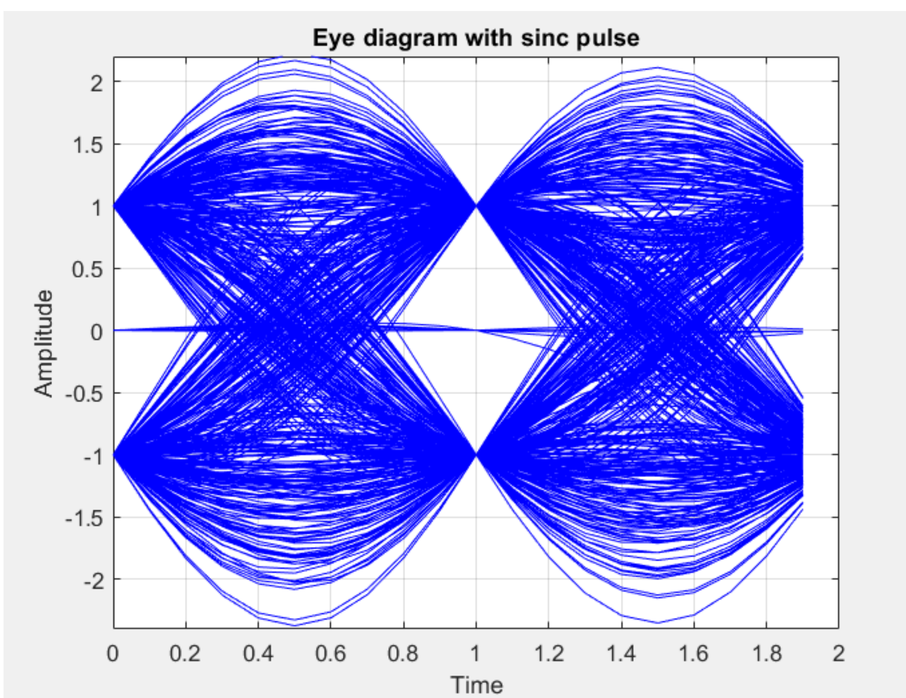
   - Raised cosine pulse shaping filter with gamma = 0.5



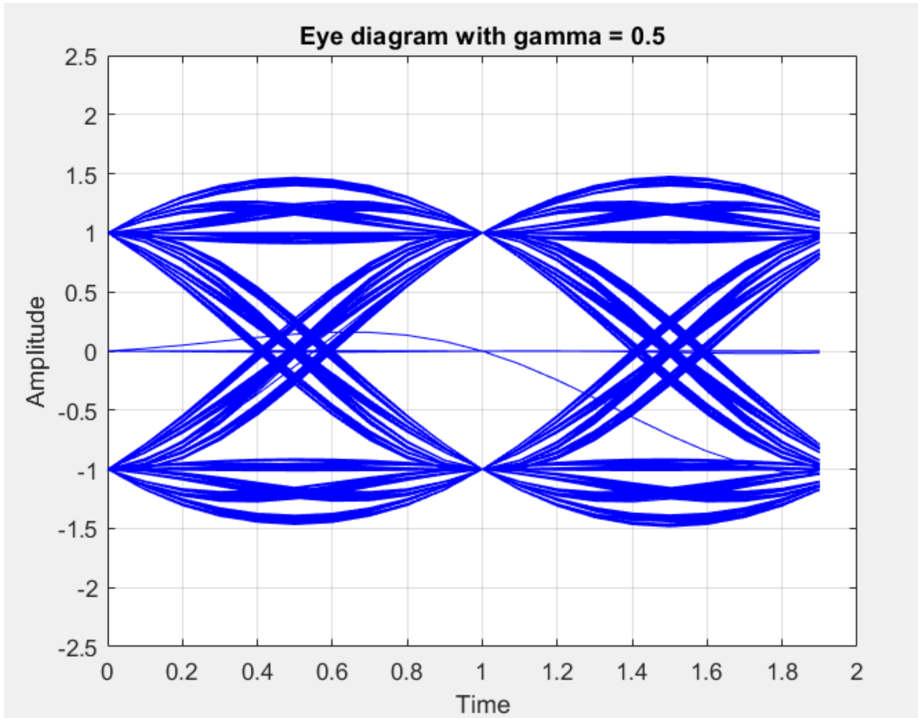
- Raised cosine pulse shaping filter with gamma = 1



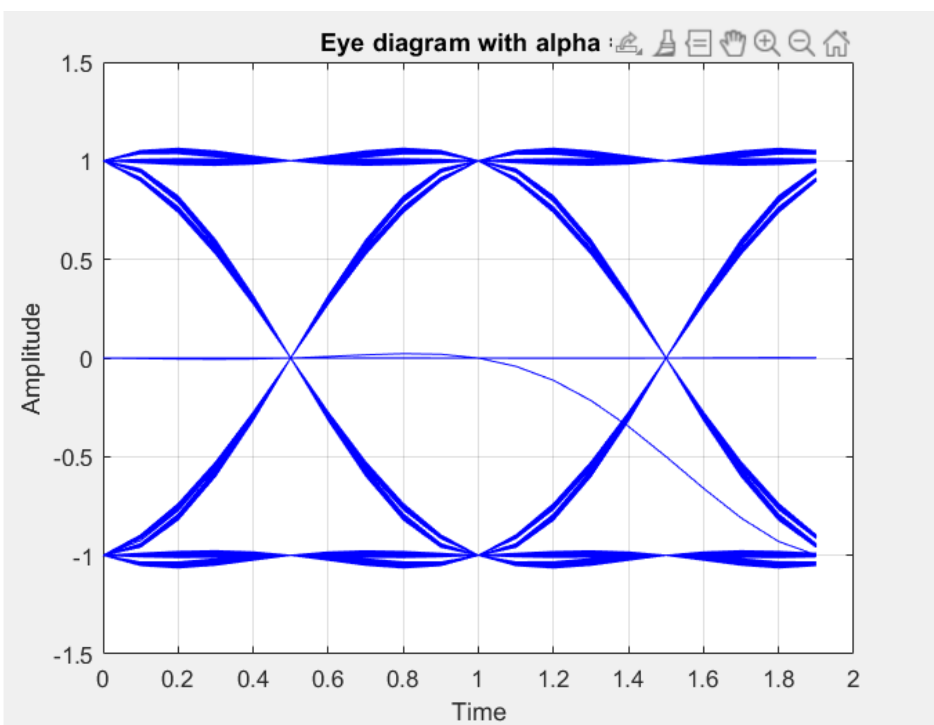
3. Eye diagram of the transmit signal,
    - Sinc pulse shaping filter



- Raised cosine pulse shaping filter with gamma = 0.5



- Raised cosine pulse shaping filter with gamma = 1

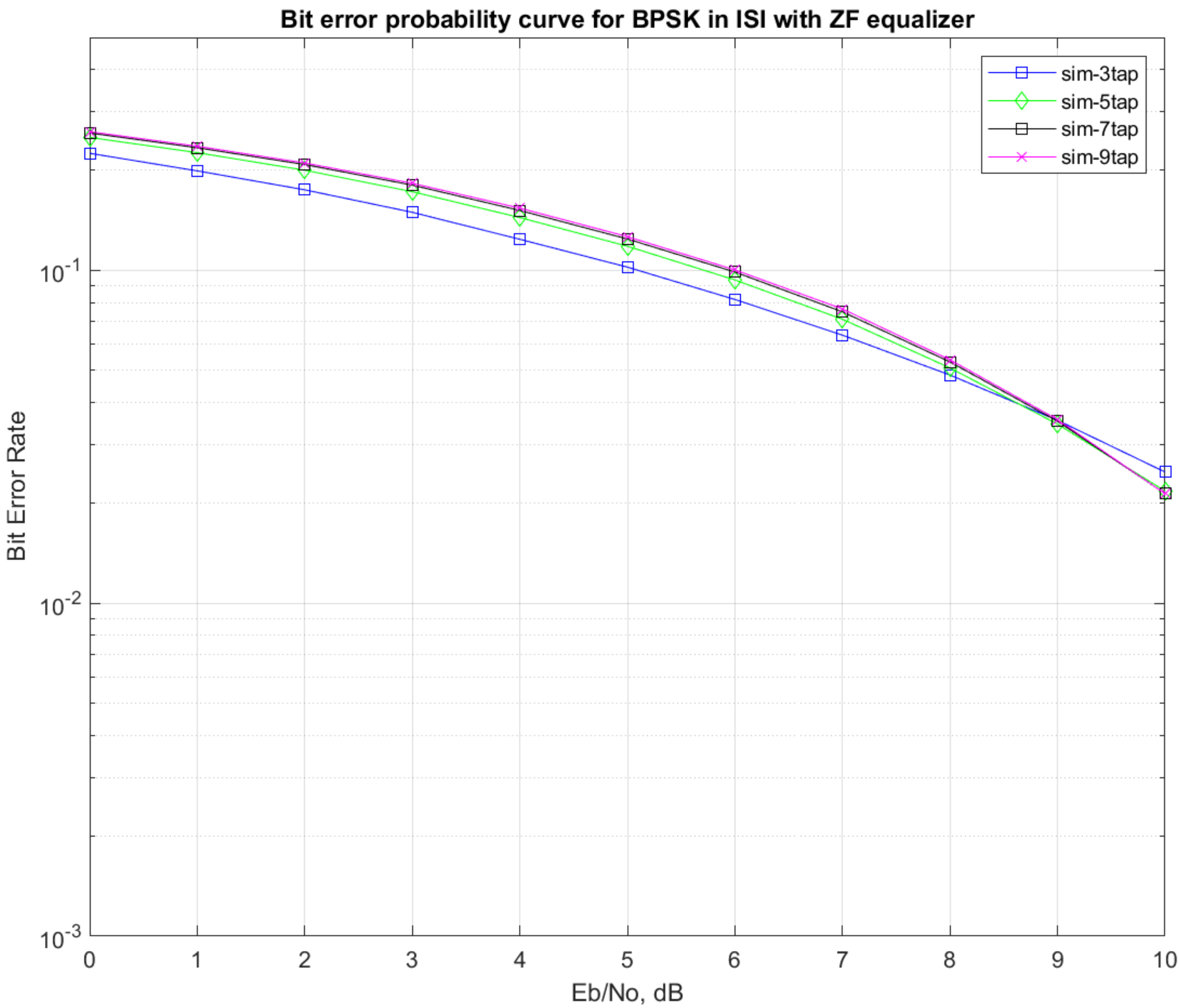


5.

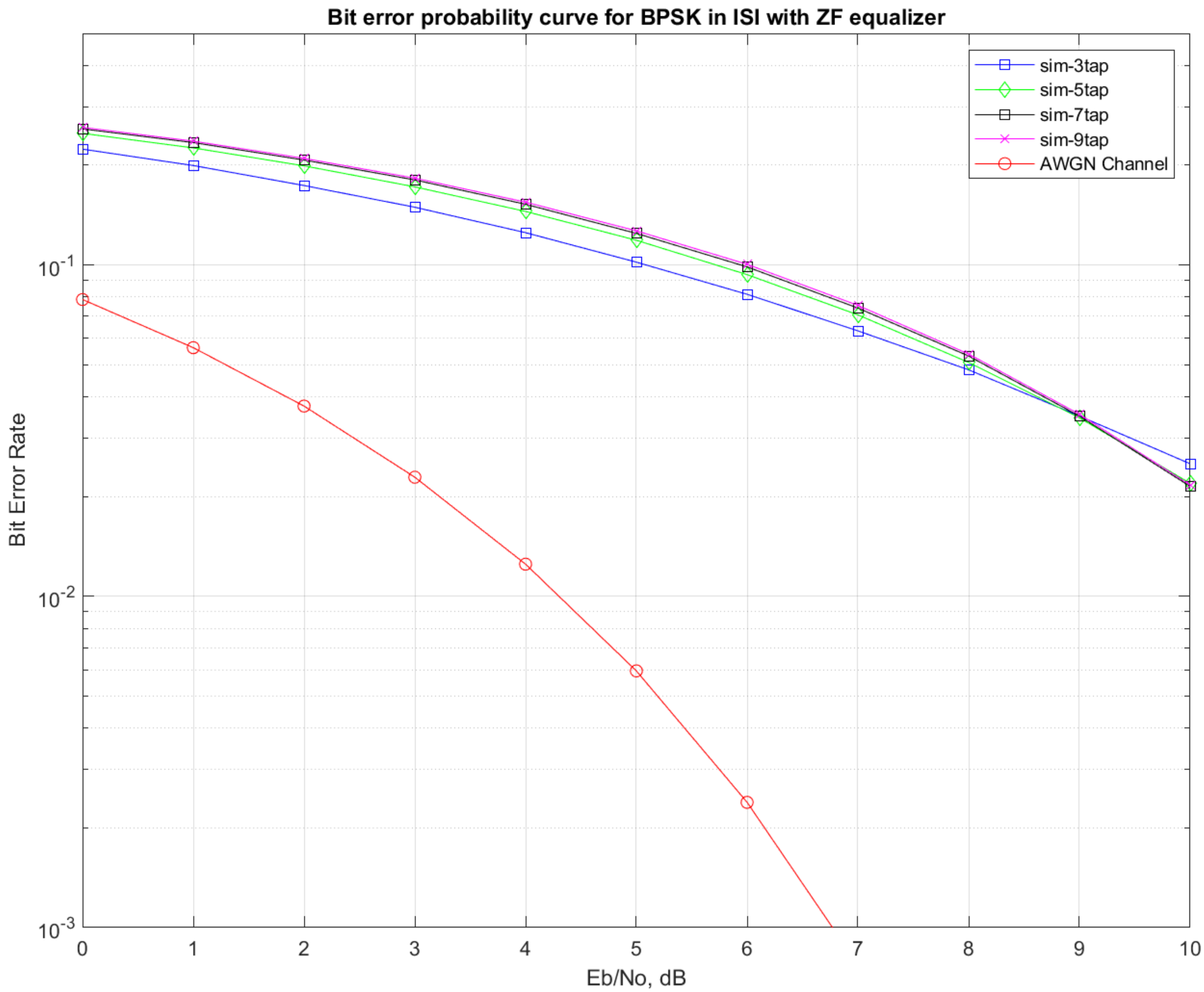
Characteristic	Sinc filter	Raised cosine filter (gamma = 0.5)	Raised cosine filter (gamma = 1)	Conclusion
Noise immunity	When noise is present, the eye height significantly decreases. Low SNR at the sampling. High Bit error occurrence at sampling.	Eye height reduced, but better than with sinc filter. Sampling error probability increased.	Eye height reduced, but the max height among the three filters.	Height of the eye is a direct interpretation of the noise immunity. Larger the height better the immunity. Noise immunity decreases from raised cosine filter with gamma 1 to gamma 0.5 to sinc filter.
Sampling time	Has the lowest range to the error free sampling. Less robust to sampling errors. Very high jitter as the time variation at the zero crossing is high.	Range and jitter are better than sinc but worse than raised cosine filter with high gamma.	Range and jitter both are best in this.	Sampling error probability increases from raised cosine filter with gamma 1 to gamma 0.5 to sinc filters.
Synchronization errors	Highest slope of the three.	Slope varies between a range. Consists of the lowest slope among three as well. Most robust to synchronization errors.	Has an average slope. It makes this more robust than sinc pulse but more errors than raised cosine with gamma = 0.5	Levels of synchronization errors decreases from sinc to raised cosine with gamma 1 to raised cosine with gamma 0.5.

Task 3

9. The BER for with all tap settings and $Eb\ /N0$ values in the same figure.



10. The BER for an additive white Gaussian noise (AWGN) channel in the same figure.



11.

In the AWGN channel, the received signal is corrupted by Gaussian noise, and the receiver makes hard decisions based on the received signal to decode the transmitted data. The theoretical BER of the AWGN channel can be accurately predicted using mathematical formulas.

However, in a multipath channel, the transmitted signal reaches the receiver through multiple paths with different delays and amplitudes. This causes intersymbol interference (ISI), where the symbols from adjacent symbols overlap in the received signal. The ZF equalizer is designed to combat ISI by trying to eliminate the interference caused by the multipath channel.

The ZF equalizer works by applying an inverse filter to the received signal to cancel out the effect of the channel. However, the ZF equalizer is based on ideal assumptions and requires perfect knowledge of the channel impulse response. In practice*, there are limitations and imperfections in estimating the channel characteristics (Channel Estimation), which can lead to errors in equalization.*

The design of the ZF equalizer assumes that the channel is invertible and has no noise. However*, in real-world scenarios, the channel is not perfectly invertible, and there is always some level of noise present*. These imperfections in the equalization process can result in residual ISI and residual noise, leading to errors in the decoded symbols.

Therefore, **the BER of the ZF equalized multipath channel may be higher compared to the AWGN channel due to the presence of residual ISI and noise. The ZF equalizer's performance is affected by the accuracy of channel estimation and the level of noise present in the channel.**

12.

Using **binary orthogonal signaling scheme instead of BPSK can potentially increase the BER (Bit Error Rate)** by improving symbol discrimination.

APPENDICES

Each task was completed using MATLAB R2021a.

Task 1 & Task 2

```
Sclear all; close all; clc;
%Defining the system Parameters
BitLength = 10^3; % No of Bits Transmitted
SampleFreq = 10; % sampling frequency in Hz
SNR_dB = 10;
NoisePower = 1./(10.^(0.1*SNR_dB)); % Noise Power (Eb = 1 in BPSK)
time = -SampleFreq:1/SampleFreq:SampleFreq; % Time Array

%% Generating the BPSK Signal
% Mapping 0 -> -1 and 1 -> 1 (0 phase and 180 phase)
BPSKSignal = 2*(rand(1,BitLength)>0.5)-1;
t = 0:1/SampleFreq:99/SampleFreq;
stem(t, BPSKSignal(1:100)); xlabel('Time'); ylabel('Amplitude');
title('BPSK Impulse Train');
axis([0 10 -1.2 1.2]); grid on;

%% Upsampling the transmit sequence without noise
BPSK_Upsample = [BPSKSignal;zeros(SampleFreq-1,length(BPSKSignal))];
%Upsampling the BPSK to match the sampling frequency
BPSK_U = BPSK_Upsample(:).';
figure;
stem(t, BPSK_U(1:100)); xlabel('Time'); ylabel('Amplitude');
title('Upsampled BPSK Impulse Train');
axis([0 10 -1.2 1.2]); grid on;

%% sinc pulse shaping filter
Sinc_Num = sin(pi*time); % numerator of the sinc function
Sinc_Den = (pi*time); % denominator of the sinc function
Sinc_DenZero = find(abs(Sinc_Den) < 10^-10); % Finding the t=0 position
Sinc_Filt = Sinc_Num./Sinc_Den;
Sinc_Filt(Sinc_DenZero) = 1; % Defining the t=0 value
figure;
plot(time, Sinc_Filt);
title('Sinc Pulse shape');
xlabel('Time'); ylabel('Amplitude');
axis([-SampleFreq SampleFreq -0.5 1.2]); grid on

Conv_sincpulse = conv(BPSK_U, Sinc_Filt);
Conv_sincpulse = Conv_sincpulse(1:10000);
Conv_sincpulse_reshape = reshape(Conv_sincpulse, SampleFreq*2,
BitLength*SampleFreq/20).';

transmit_signal_sinc = Conv_sincpulse(1,100:999);
tt=(-0.1:0.1:89.8);
plot(tt,transmit_signal_sinc);
title('Transmit signal with sinc pulse');

figure;
```

```
plot(0:1/SampleFreq:1.99, real(Conv_sincpulse_reshape).', 'b');
title('Eye diagram with sinc pulse');
xlabel('Time'); ylabel('Amplitude');
axis([0 2 -2.4 2.2]);
grid on

%% Raised cosine pulse shaping filter (gamma = 0.5)
roll_off = 0.5;
cos_Num = cos(roll_off*pi*time);
cos_Den = (1 - (2 * roll_off * time).^2);
cos_DenZero = abs(cos_Den)<10^-10;
RaisedCosine = cos_Num./cos_Den;
RaisedCosine(cos_DenZero) = pi/4;
RC_gamma5 = Sinc_Filt.*RaisedCosine; % Getting the complete raised cosine
pulse
figure;
plot(time, RC_gamma5);
title('Raised Cosine Pulse shape gamma = 0.5');
xlabel('Time'); ylabel('Amplitude');
axis([-SampleFreq SampleFreq -0.5 1.2]); grid on

Conv_RCgamma5 = conv(BPSK_U,RC_gamma5);
Conv_RCgamma5 = Conv_RCgamma5(1:10000);
Conv_RCgamma5_reshape =
reshape(Conv_RCgamma5,SampleFreq*2,BitLength*SampleFreq/20).';

transmit_signal_RCgamma5 = Conv_RCgamma5(1,100:999);
tt=(-0.1:0.1:89.8);
plot(tt,transmit_signal_RCgamma5);
title('Transmit signal with gamma = 0.5');

figure;
plot(0:1/SampleFreq:1.99, Conv_RCgamma5_reshape.','b');
title('Eye diagram with gamma = 0.5');
xlabel('Time'); ylabel('Amplitude');
axis([0 2 -2.5 2.5]);
grid on

%% Raised cosine pulse shaping filter (gamma = 1)
roll_off = 1;
cos_Num = cos(roll_off * pi * time);
cos_Den = (1-(2 * roll_off * time).^2);
cos_DenZero = find(abs(cos_Den)<10^-20);
RaisedCosine = cos_Num./cos_Den;
RaisedCosine(cos_DenZero) = pi/4;
RC_gamma1 = Sinc_Filt.*RaisedCosine; % Getting the complete raised cosine
pulse
figure;
plot(time, RC_gamma1);
title('Raised Cosine Pulse shape gamma = 1');
xlabel('Time'); ylabel('Amplitude');
axis([-SampleFreq SampleFreq -0.5 1.2]); grid on

Conv_RCgamma1 = conv(BPSK_U,RC_gamma1);
Conv_RCgamma1 = Conv_RCgamma1(1:10000);
```

```
Conv_RCgamma1_reshape =
reshape(Conv_RCgamma1,SampleFreq*2,BitLength*SampleFreq/20).';

transmit_signal_RCgamma1 = Conv_RCgamma1(1,100:999);
tt=(-0.1:0.1:89.8);
plot(tt,transmit_signal_RCgamma1);
title('Transmit signal with gamma = 1');

figure;
plot(0:1/SampleFreq:1.99, Conv_RCgamma1_reshape.','b');
title('Eye diagram with alpha =1');
xlabel('Time'); ylabel('Amplitude');
axis([0 2 -1.5 1.5 ]);
grid on


%% Noise Array Generation based on SNR = 10dB
Noise1D = normrnd (0 , sqrt(NoisePower/2), [1, BitLength]);
AWGN_TX = BPSKSignal + Noise1D;
figure;
stem(t, AWGN_TX(1:100)); xlabel('Time'); ylabel('Amplitude');
title('BPSK with noise Impulse Train');
axis([0 10 -1.5 1.5]); grid on;


%% upsampling the transmit sequence with Noise
AWGNTx_Upsample = [AWGN_TX;zeros(SampleFreq-1,length(BPSKSignal))];
AWGNTx_U = AWGNTx_Upsample(:);
figure;
stem(t, AWGNTx_U(1:100)); xlabel('Time'); ylabel('Amplitude');
title('Upsampled BPSK with noise Impulse Train');
axis([0 10 -1.5 1.5]); grid on;

%% sinc with noise
Conv_sincnoise = conv(AWGNTx_U,Sinc_Filt);
Conv_sincnoise = Conv_sincnoise(1:10000);
Conv_sincnoise_reshape = reshape(Conv_sincnoise, SampleFreq*2,
BitLength*SampleFreq/20).';
figure;
plot(0:1/SampleFreq:1.99, Conv_sincnoise_reshape.', 'b');
title('Eye diagram with sinc pulse');
xlabel('Time'); ylabel('Amplitude');
axis([0 2 -2.4 2.2]);
grid on

%% raised cosine with noise (gamma = 0.5)
Conv_RC5noise = conv(AWGNTx_U,RC_gamma5);
Conv_RC5noise = Conv_RC5noise(1:10000);
Conv_RC5noise_reshape =
reshape(Conv_RC5noise,SampleFreq*2,BitLength*SampleFreq/20).';
figure;
plot(0:1/SampleFreq:1.99, Conv_RC5noise_reshape.', 'b');
title('Eye diagram with gamma = 0.5 noisy');
xlabel('Time'); ylabel('Amplitude');
axis([0 2 -2.4 2.2]);
```

```
grid on

%% raised cosine with noise (gamma = 1)
Conv_R1noise = conv(AWGNTx_U,RC_gamma1);
Conv_R1noise = Conv_R1noise(1:10000);
Conv_R1noise_reshape =
reshape(Conv_R1noise,SampleFreq*2,BitLength*SampleFreq/20).';
figure;
plot(0:1/SampleFreq:1.99, Conv_R1noise_reshape.', 'b');
title('Eye diagram with gamma = 1.0 noisy');
xlabel('Time'); ylabel('Amplitude');
axis([0 2 -2.4 2.2]);
grid on
```

Task 3

```
%Task 3  Desining a zero-forcing (ZF) equalizer for a 3-tap multipath channel

sequenceLength = 10^6; % Length of the binary sequence
ebN0dB = 0:10; % multiple Eb/N0 values
tapCount = 4;
errorCount = zeros(tapCount, length(ebN0dB)); % Initialize error count

for n_i = 1:length(ebN0dB)

    % Transmitter
    Ik = rand(1, sequenceLength) > 0.5; % Generating a random binary sequence
    ak = 2 * Ik - 1; % BPSK modulation 0 -> -1; 1 -> +1

    % Channel model, multipath channel
    nTap = 3;
    channelImpulseResponse = [0.3 0.9 0.4];

    channelOutput = conv(ak, channelImpulseResponse);
    noise = 1/sqrt(2) * [randn(1, sequenceLength +
length(channelImpulseResponse) - 1) + 1j * randn(1, sequenceLength +
length(channelImpulseResponse) - 1)]; % White Gaussian noise, 0dB variance

    % Noise addition
    y = channelOutput + 10^(-ebN0dB(n_i) / 20) * noise; % Additive white
Gaussian noise

    for kk = 1:tapCount
        chaImpulseResponseLen = length(channelImpulseResponse);
        hM = toeplitz([channelImpulseResponse(2:end), zeros(1, 2 * kk + 1 -
chaImpulseResponseLen + 1)], [channelImpulseResponse(2:-1:1), zeros(1, 2 * kk
+ 1 - chaImpulseResponseLen + 1)]);

        d = zeros(1, 2 * kk + 1);
        d(kk + 1) = 1;

        c = inv(hM) * d.';
```

```
        % Matched filter
        filteredSignal = conv(y, c);
        filteredSignal = filteredSignal(kk + 2:end);
        filteredSignal = conv(filteredSignal, ones(1, 1)); % Convolution
        sampledSignal = filteredSignal(1:1:sequenceLength); % Sampling at
time T

        % Receiver - hard decision decoding
        decodedSequence = real(sampledSignal) > 0;

        % Counting the errors
        errorCount(kk, n_i) = sum(Ik ~= decodedSequence);
    end
end

simulatedBER = errorCount / sequenceLength; % Simulated BER
theoryBer = 0.5 * erfc(sqrt(10.^(ebN0dB / 10))); % Theoretical BER

% Plot
close all
figure
semilogy(ebN0dB, simulatedBER(1,:), 'bs-'), 'Linewidth',2;
hold on
semilogy(ebN0dB, simulatedBER(2,:), 'gd-'), 'Linewidth',2;
semilogy(ebN0dB, simulatedBER(3,:), 'ks-'), 'Linewidth',2;
semilogy(ebN0dB, simulatedBER(4,:), 'mx-'), 'Linewidth',2;
semilogy(ebN0dB, theoryBer, 'ro-'), 'Linewidth',2;
axis([0 10 10^-3 0.5])
grid on

%legend('sim-3tap', 'sim-5tap', 'sim-7tap', 'sim-9tap');
legend('sim-3tap', 'sim-5tap', 'sim-7tap', 'sim-9tap','AWGN Channel');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('Bit error probability curve for BPSK in ISI with ZF equalizer');
```

END!!