To study eye diagram using matlab

```
% EXPT NO 10 - SAMPLING clc; clear all; close all;

% Input parameters N = input('Enter the duration (in seconds): '); fm = input('Enter message signal frequency in Hz (fm): ');

% Time vector and message signal t = 0:0.001:N; % finer time step for smooth signal x = 8 * sin(2 * pi * fm * t);

% Plot original message signal subplot(3,2,1); plot(t, x, 'LineWidth', 1.5); xlabel('Time (s)'); ylabel('Amplitude'); title('Original Message Signal'); grid on;

% Sampling frequency input fs = input('Enter sampling frequency in Hz (fs): ');

% Sampling signal q = 0:1/fs:N; p = ones(1, length(q)); % sampling pulses (all ones) subplot(3,2,3); stem(q, p, 'filled'); xlabel('Time (s)'); ylabel('Amplitude'); title('Sampling Signal'); grid on;

% Sampled signal n1 = 0:1/fs:N; x1 = 8 * sin(2 * pi * fm * n1); z = x1 .* p;

subplot(3,2,5); stem(n1, z, 'filled'); xlabel('Time (s)'); ylabel('Amplitude'); title('Sampled Signal'); grid on;

% Spectrum of original message signal xf1 = fft(x); mag = abs(xf1); subplot(3,2,2); plot(linspace(0, fs, length(mag)), mag); xlabel('Frequency (Hz)'); ylabel('Magnitude'); title('Spectrum of Message Signal'); grid on;

% Spectrum of sampling signal pf1 = fft(p); magp = abs(pf1); subplot(3,2,4); plot(linspace(0, fs, length(magp)), magp); xlabel('Frequency (Hz)'); ylabel('Magnitude'); title('Spectrum of Sampling Signal'); grid on;



% Spectrum of sampled signal zf1 = fft(z, length(t)); % zero-padded to match length magz = abs(zf1); subplot(3,2,6); plot(linspace(0, fs, length(magz)), magz); xlabel('Frequency (Hz)'); ylabel('Magnitude'); title('Spectrum of Sampled Signal'); grid on;
```

```
% EXPT NO 11 - ISI and Eye Diagram

clc;

clear all;

close all;


%% Parameters

M = 2;            % Modulation order (BPSK)

nsamp = 4;        % Oversampling rate

filtorder = 40;     % Filter order

rolloff = 0.25;     % Roll-off factor

delay = filtorder / (nsamp * 2);  % Group delay
```

```
%% Raised Cosine Filter Design
rrcFilter = rcosdesign(rolloff, filtorder/nsamp, nsamp, 'sqrt'); % SRRC filter

% Plot impulse response
figure;
impz(rrcFilter);
title('Impulse Response of Square Root Raised Cosine (SRRC) Filter');
grid on;

%% Transmitted Signal
% Generate random binary data
data = randi([0 1], 1, 100);

% BPSK modulation
txSymbols = 2*data - 1;   % Map 0 -> -1, 1 -> +1

% Upsample and apply SRRC filter
ytx = upfirdn(txSymbols, rrcFilter, nsamp);

% Add AWGN noise
ytx_noisy = awgn(ytx, 40, 'measured');

% Plot transmitted and noisy signal
figure;
subplot(2,1,1);
plot(ytx(1:200));
title('Transmitted Signal (Filtered)');
xlabel('Samples'); ylabel('Amplitude'); grid on;

subplot(2,1,2);
plot(ytx_noisy(1:200));
```

```
title('Received Signal with Noise');

xlabel('Samples'); ylabel('Amplitude'); grid on;
```

```
% EXPT NO 12 - SCRAMBLER AND DESCRAMBLER clc; clear all; close all;

%% SCRAMBLER

% Example input data (bytes) Scrambler_input = [80 255 16 9 48 255 80 0 25 0 145];

s = 20255; % Initial seed (scrambler state) rand_data = zeros(size(Scrambler_input)); % Store scrambled output

% Scrambling process for j = 1:length(Scrambler_input) for i = 1:8 msb = bitxor(bitget(s,1), bitget(s,2)); % XOR feedback s = bitshift(s,-1); s = bitset(s,15,msb); t = bitxor(bitget(Scrambler_input(j), 9 - i), msb); rand_data(j) = bitset(rand_data(j), 9 - i, t); end end

scrambler_out = rand_data;

disp('--- SCRAMBLER OUTPUT ---'); disp(scrambler_out);

%% DESCRAMBLER

s = 20255; % Re-initialize descrambler with same seed descrambler_in = zeros(size(scrambler_out));

% Descrambling process for j = 1:length(scrambler_out) for i = 1:8 msb = bitxor(bitget(s,1), bitget(s,2)); s = bitshift(s,-1); s = bitset(s,15,msb); t = bitxor(bitget(scrambler_out(j), 9 - i), msb); descrambler_in(j) = bitset(descrambler_in(j), 9 - i, t); end end

descrambler_out = descrambler_in;

disp('--- DESCRAMBLER OUTPUT ---'); disp(descrambler_out);

%% Verification if isequal(Scrambler_input, descrambler_out) disp('SUCCESS ✅: Original and Descrambled data match.'); else disp('ERROR ❌: Data mismatch!'); end
```