Communication Lab II

(ELC3940)

Experiment No.: 05

Object:

Generate an eye pattern of the bit sequence obtained in Exp. 1 for a noiseless channel which has an ideal lowpass frequency characteristics of bandwidth 20 kHz. Determine (a) distortion (b) time jitter (c) noise margin (d) best sampling time, and (e) sensitivity to timing error.

G. No: **GL3136**

S. No: **A3EL-02**

F. No: **19ELB056**

Name: **Maha Zakir Khan**

Date of performing experiment: 15|02|2022

Date of report submission: 04|03|2022

Software Used:

MATLAB®, Release 2021a (R2021a), a programming platform designed specifically to analyze and design systems and products. The heart of MATLAB is the MATLAB language, a matrix-based language, it provides vast library of mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration and solving ordinary differential equations.

Procedure:

<u>Step 1:</u> Polar NRZ Line coding:

Take speech signal and convert it into PCM encoded binary bitstream with bit-rate Rb, and perform polar-NRZ signaling to generate N length bit sequence:

$$x[n]\ ,\ Where\ n\ = 1{,}5{,}9, \ldots, N$$

<u>Step 2:</u> Design FIR Lowpass filter:

Use following parameters for designing the Lowpass filter $h[n]$:

Cut-off frequency $fc = 20Khz$

Sampling Rate $fs = 10 \times Rb$

Filter order $N = 300$

Stopband Attenuation $A_s = 65dB$

Passband Ripple $R_P\ =\ 0.2$

Step 3: Pass x[n] through the filter h[n] to get filtered signal q[n]:

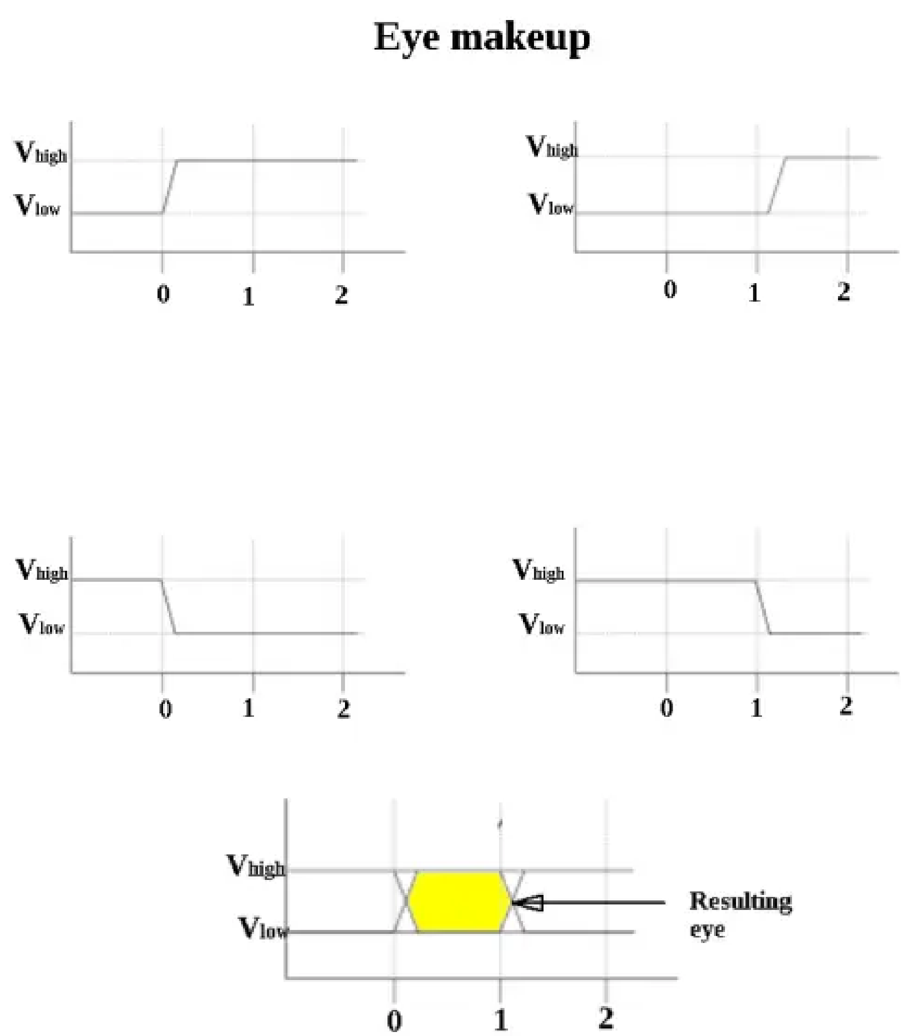
Perform convolution on $x[n]$:

$$q[n] = h[n] \Theta x[n] , \quad Where\, n \ = 1,5,9, \dots , N$$

Taking DFT:

$$Q[\omega] \ = \ H[\omega] \times X[\omega] \, , Where\, \omega \ = \ Normalized\, frequency\, (in\, \pi\, rad \,/\, sample\,)$$

Step 4: Plot Eye Diagram of the bit sequence before and after filtering:



Extract signal from: $\frac{nTb}{2}$ to $\frac{(n+1)Tb}{2}$, $Where\ n\ = 1,5,9, \dots, N$

Eye Diagram before filtering $x_e[e]$:

$$x_e[e] = x[n],\ for\ n = \frac{4i+1}{2}\ \ to\ \ \frac{4(i+1)+1}{2},\ e\ = \frac{1Tb}{2}\ \ to\ \ \frac{5Tb}{2};$$

$$Where\ i\ = 0,1,2, \dots, i \leq \frac{2N-5}{4}$$

Eye Diagram after filtering $q_e[e]$:

$$q_e[e] = q[n],\ for\ n = \frac{4i+1}{2}\ \ to\ \ \frac{4(i+1)+1}{2},\ e\ = \frac{1Tb}{2}\ \ to\ \ \frac{5Tb}{2};$$

$$Where\ i = 0,1,2 \dots\dots, i \leq \frac{2N-5}{4}$$

Note: I extracted and overlapped the signal array onto 1/2Tb to 5/2Tb, we can also extract from 0Tb to 1Tb or 0Tb to 2Tb, whichever is comfortable to plot and easier to observe the eye of.

Program:

Using .wav file ‘Hello5’ to read the speech signal in MATLAB and performed PCM encoding using the function PcmEncoder from previous *Experiment-01*

Note: I used a randomly generated binary bit sequence as an alternative to the speech signal encoded values.

```
info = audioinfo('C:\Users\Maha Khan\Downloads\Hello5.wav');
[signal , f] = audioread('C:\Users\Maha Khan\Downloads\Hello5.wav'); % Voice signal with dual channel
                                              % and fs as the sampling frequency
[X ,Rb] = PcmEncoder(signal,info);            % Calling Function to pcm encode the signal
Tb = 1/Rb;  % Tb - Time period of pulse , Rb - Bit-rate
Lx = length(X);

 X = randi([0 1],1,Lx); % Using a random generated binary sequence for better plot results

A = 1;       % Voltage level
Ns = 20;     % Number of samples
nseq = 0:0.01:Ns-0.01;
```

Encoding Pcm sequence with Polar-NRZ

```
%Using Polar NRZ signalling to generate our bit sequence
polar = [Lx];
c = 1;
for i = 1:Lx % Encoding signal to polar NRZ format
    if X(i) == 0
        polar(i)= (-A);
    else
        polar(i) = (A);
    end
end
Lsequence = repelem(polar,100);      % Up-sampling polar sequence to 100 times Bit-rate
sequence = Lsequence(1:20*100);      % Extracting only 20 bits out of the complete sequence for plotting
```

Lowpass filter design

```
% Designing lowpass fir filter
fs = Rb*10;  % Taking Sampling Rate as 10 times Bitrate
fc = 20000;  % Taking Cut-off frequency as 20Khz

lowpass  = designfilt('lowpassfir','FilterOrder',300, 'CutoffFrequency',fc,... % Lowpass filter parameters
                 'PassbandRipple',0.2,'StopbandAttenuation',65,'SampleRate',fs);
```

Plotting Generated Filter Response

```
        % Impulse-Response of Filter
        figure('NumberTitle', 'off', 'Name', 'Frequency Response of Lowpass filter with Cut-off Frequency 20Khz');
          subplot(2,1,1);
            [h, w] = freqz(lowpass);
            magnitude = db(abs(h));
            plot(w/pi,magnitude,'LineWidth',1.2);                       % Magnitude Response
            title('Magnitude Response');
            xlabel('Normalized Frequency (\times\pi rad/sample)');
            ylabel('Magnitude (dB)');
            yline(-65,'--');
            txt = ['Stopband Attenuation: ' num2str(65) ' dB '];
            text(0.8,-60,txt,'FontSize',10,'Color','k')

          subplot(2,1,2);
            phase = unwrap(angle(h));
            plot(w/pi,rad2deg(phase),'LineWidth',1.2,'Color','m');      % Phase Response
            title('Phase Response');
            xlabel('Normalized Frequency (\times\pi rad/sample)');
            ylabel('Phase (degrees)');
            sgtitle('Frequency Response of Lowpass filter H[\omega] with Cut-off Frequency 20Khz');
```

Using filtfilt to generate zero-phase response filtered sequence

```
    filteredAll = filtfilt(lowpass,Lsequence);  % Filtering up-sampled sequence using the lowpass filter
    filtered = filteredAll(1:20*100);           % Extracting first 20 bits of the filtered signal
```

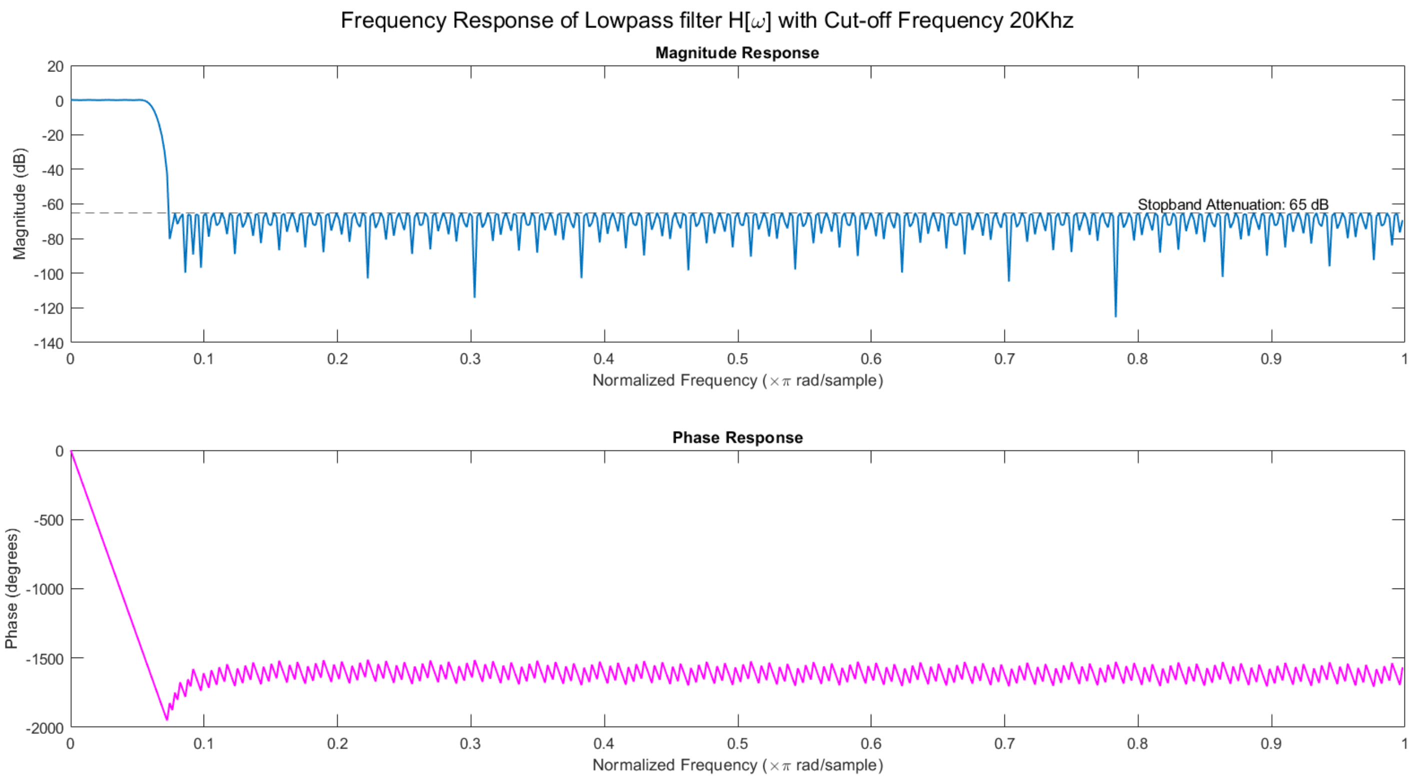
Plots and subplots of the results

```
figure('NumberTitle', 'off', 'Name', 'Eye Diagram of polar-NRZ sequence ');

subplot(4,1,1); % Plotting first 20 bits of Polar-NRZ coded Bit-sequence
    stairs(nseq,sequence,'LineWidth',2,'Color','#0072BD');
    yline(0,':');
    axis([0 20 -1.5 1.5]);
    title('NRZ-Polar Data Bit-Sequence p[n]');
    xlabel('Discrete time n (in Tb) \rightarrow');
    ylabel('Amplitude A in volts');
    xticks([1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20]);
    xticklabels({'\bf1Tb','\bf2Tb','\bf3Tb','\bf4Tb','\bf5Tb',...
                '\bf6Tb','\bf7Tb','\bf8Tb','\bf9Tb','\bf10Tb',...
                '\bf11Tb','\bf12Tb','\bf13Tb','\bf14Tb','\bf15Tb',...
                '\bf16Tb','\bf17Tb','\bf18Tb','\bf19Tb','\bf20Tb'});
    txt = ['Time Period Tb: ' num2str(Tb) ' seconds '];
    text(17,-2.5,txt,'FontSize',10,'FontWeight','bold','Color','r')
subplot(4,1,2);  % Plotting first 20 bits of filtered Bit-sequence
    plot(nseq,filtered,'LineWidth',2,'Color','r');
    yline(0,':');    yline(1,'--');    yline(-1,'--');
    axis([0 20 -1.5 1.5]);
    title('Low-Pass Filtered Bit-Sequence q[n] = p[n]\oplush[n]  (Cut-off frequency fc = 20Khz)');
    xlabel('Discrete time n (in Tb) \rightarrow');
    ylabel('Amplitude A in volts');
    xticks([1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20]);
    xticklabels({'\bf1Tb','\bf2Tb','\bf3Tb','\bf4Tb','\bf5Tb',...
        '\bf6Tb','\bf7Tb','\bf8Tb','\bf9Tb','\bf10Tb',...
        '\bf11Tb','\bf12Tb','\bf13Tb','\bf14Tb','\bf15Tb',...
        '\bf16Tb','\bf17Tb','\bf18Tb','\bf19Tb','\bf20Tb'});

    txt = ['Bit-rate Rb: ' num2str(Rb) ' bits/second'];
    text(17,-2.5,txt,'FontSize',10,'FontWeight','bold','Color','r')
subplot(2,2,3);  % Eye-diagram of original bit sequence
        for i = 1:200:2000
            stairs(1:200,Lsequence(i+50:i+50+199),'LineWidth',2);
            axis([1 200 -1.5 1.5]);
            hold on;
        end

    title('Eye-diagram  of Polar-NRZ Data Bit-Sequence eye p_{e}[e] ');
    xlabel(' e in Tb (from 0.5Tb to 2.5Tb) \rightarrow');
    ylabel('Amplitude A in volts');
    xline([51 151]);xline(101,'--');yline(0,'--')
    xticks([1 51 101 151 200]);
    xticklabels({'\bf0.5Tb','\bf1Tb' ,'\bf1.5Tb','\bf2Tb','\bf2.5Tb'});
    hold off;
```

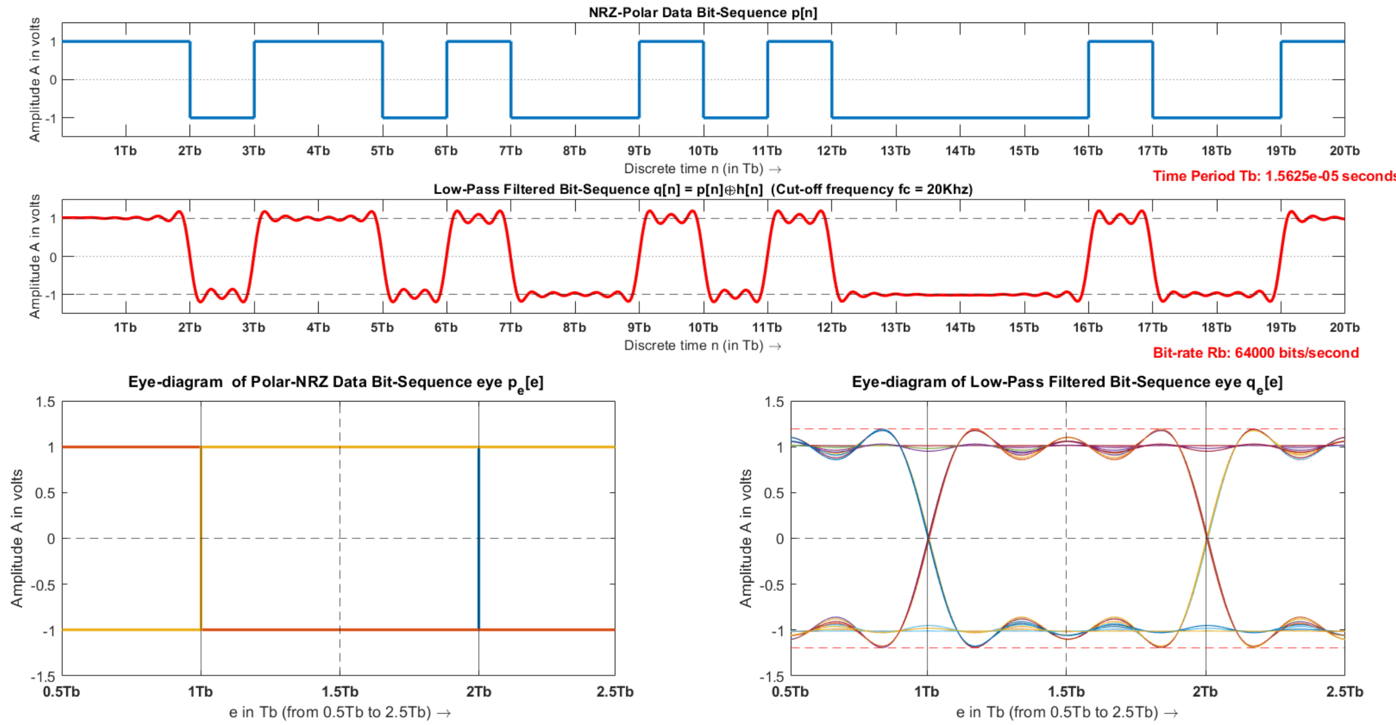
```
subplot(2,2,4); %Eye-diagram of the first 20 filtered bit-sequence
        eye = [];
        ymax = 0; ymin = 0;
        for i = 1:200:100000
            eye = filteredAll(i+50:i+50+199);
            plot(1:200,eye);
                if ymax<max(eye)
                ymax = max(eye);
                end
                if ymin > min(eye)
                ymin = min(eye);
                end

            axis([1 200 -1.5 1.5]);
            hold on;
        end

    title('Eye-diagram of Low-Pass Filtered Bit-Sequence eye q_{e}[e]');
    xlabel('e in Tb (from 0.5Tb to 2.5Tb) \rightarrow');
    ylabel('Amplitude A in volts');
    xline([50 150]);xline(100,'--');yline(0,'--'); yline([ymax ymin],'--','Color','r');
    xticks([1 50 100 150 200]);
    xticklabels({'\bf0.5Tb','\bf1Tb' ,'\bf1.5Tb','\bf2Tb','\bf2.5Tb'});

    hold off;
```

Observation:

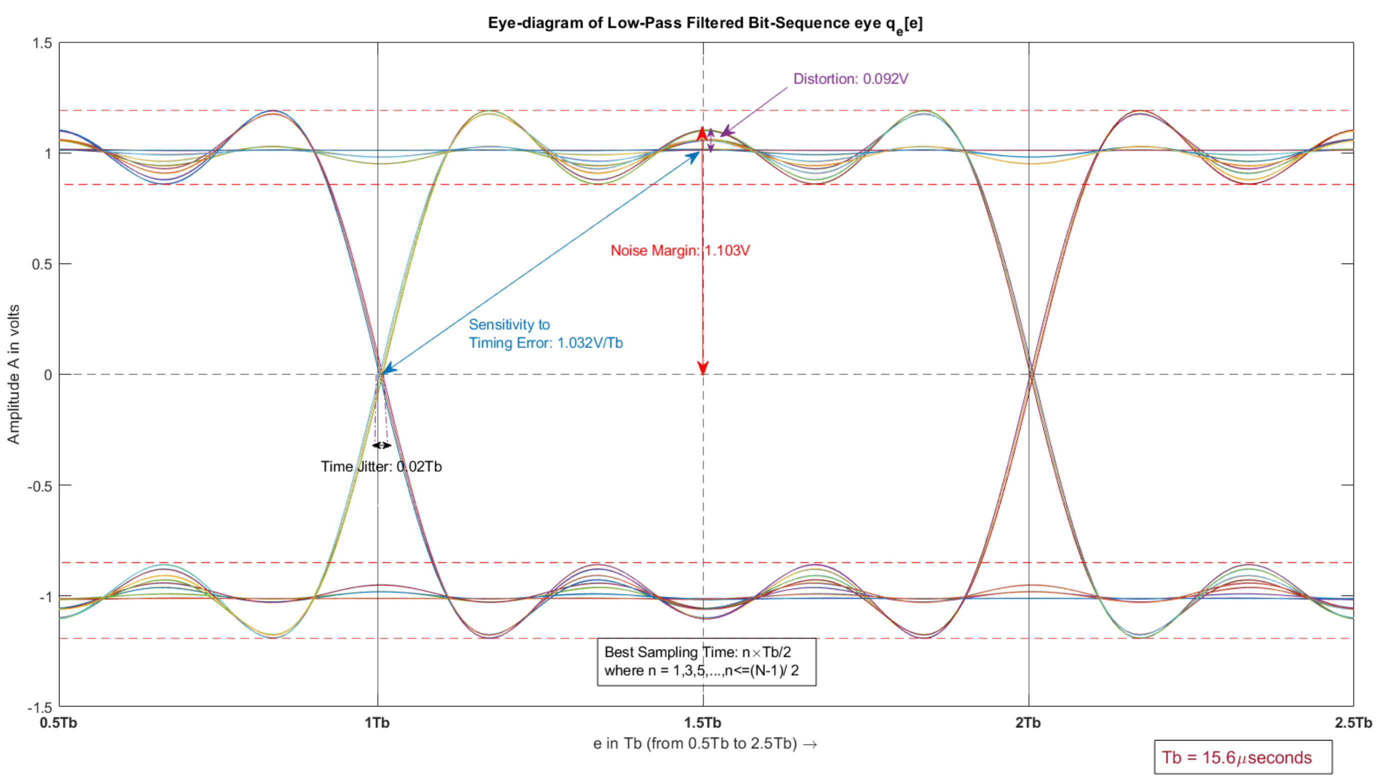
Frequency Response of FIR Lowpass filter:



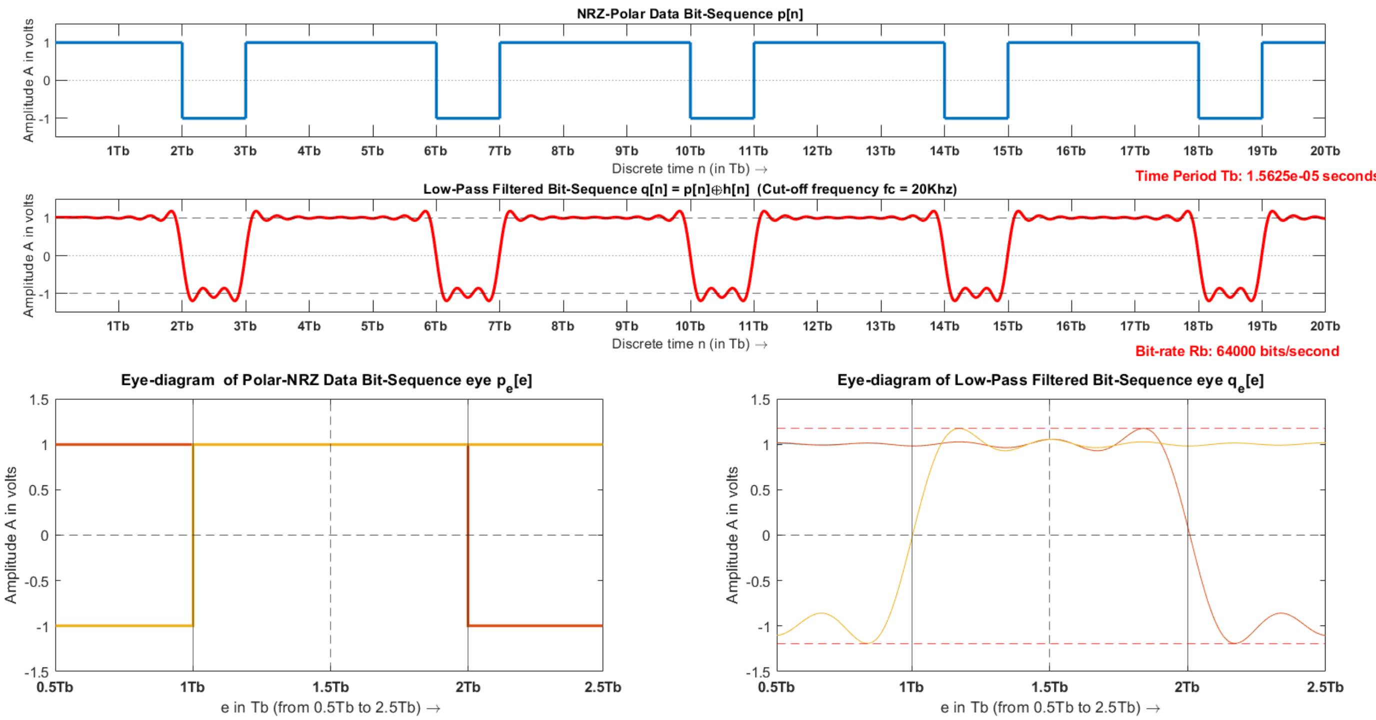
(1) For Randomly Generated Binary Bit-sequence:



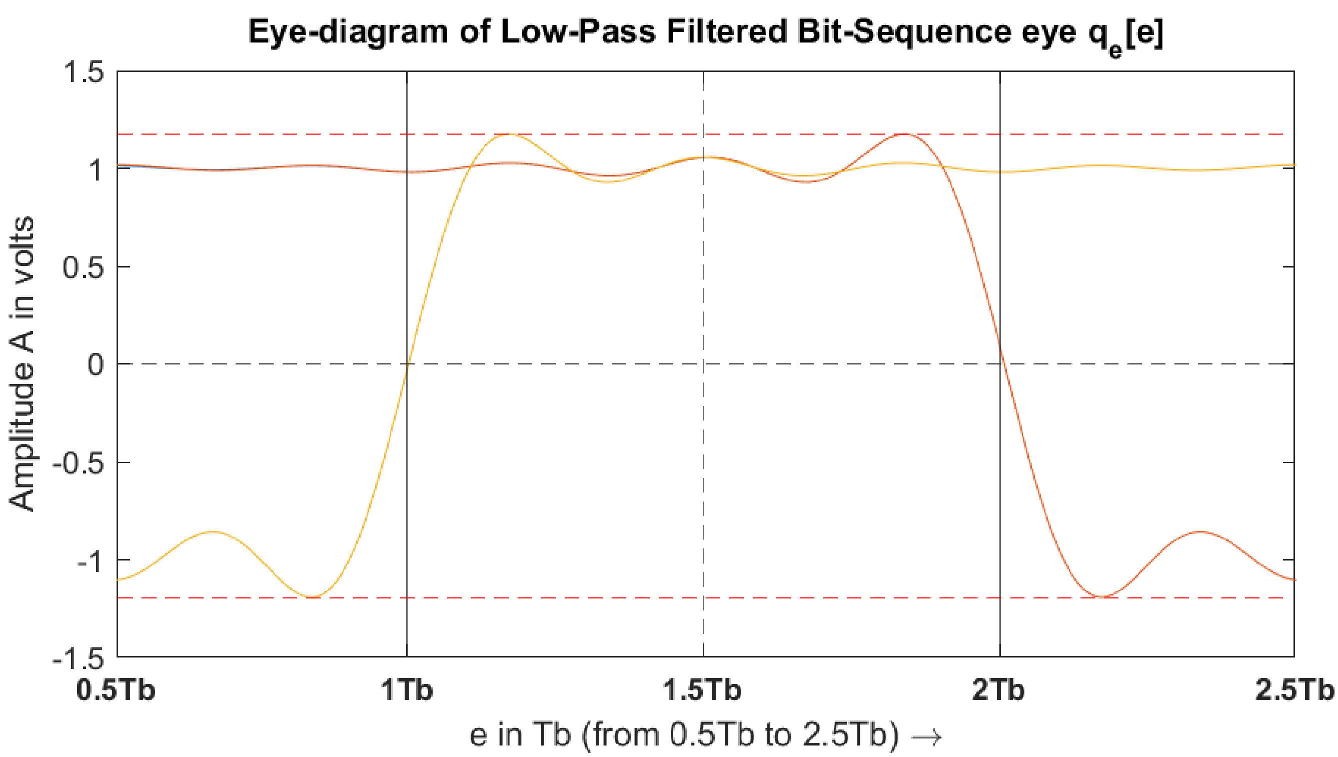
Eye Diagram:



(2) For Speech Signal Encoded Bit-sequence:



<u>Eye Diagram:</u>



Result:

(a) distortion: 0.092 Volts

(b) time jitter: 0.02Tb

(c) noise margin: 1.103 Volts

(d) best sampling time: 0.5Tb,1.5Tb,...,(N-1)*0.5Tb

(e) sensitivity to timing error: 1.032 Volts/Tb

Discussion:

In this Experiment, I used only 500 bits of the filtered sequence to generate the eye-diagram to reduce computation. I also used randomly generated data bit sequence to plot the eye diagram and measure the results as the diagram was not complete in case of using speech encoded signal. The filtered bit sequence has ripples of peak-to peak amplitude of around 0.4 Volt, an artifact from windowing function. I used FIR filter to design my lowpass filter and kept the order of filter as high as possible to make the magnitude response as ideal as possible, also the filtfilt function used is a zero-phase filter which also gives ideal lowpass filter phase response.