

Laboratory #3

Digital Bandpass Transmission

SYSC 4600 Digital Communications

Department of Systems and Computer Engineering
Faculty of Engineering
Carleton University

© November 2016

1 Purpose and Objectives

The purpose of this laboratory experiment is to provide familiarity with some of the key building blocks in digital bandpass transmission systems.

The influence of the transmitter and/or the receiver filtering operations on the received signal is examined both in the time-domain, and in the frequency domain. The lab provides some insight into the differences between two of the popular Modulation methods: Binary Phase Shift Keying (BPSK) and Quadrature Phase Shift Keying (QPSK).

Impact of receiver filtering on the received signal is also examined with emphasis on qualitative effect on Intersymbol Interference (ISI). The construction of matched filters and their impact on the received signal are also examined.

Please note that this lab is designed more as a tutorial to provide students with a visual hand on insight into the many basic concepts of baseband transmission. You are encouraged to invent ways of using it on your own.

2 Background

A simplified block diagram of a baseband communication system is shown below:

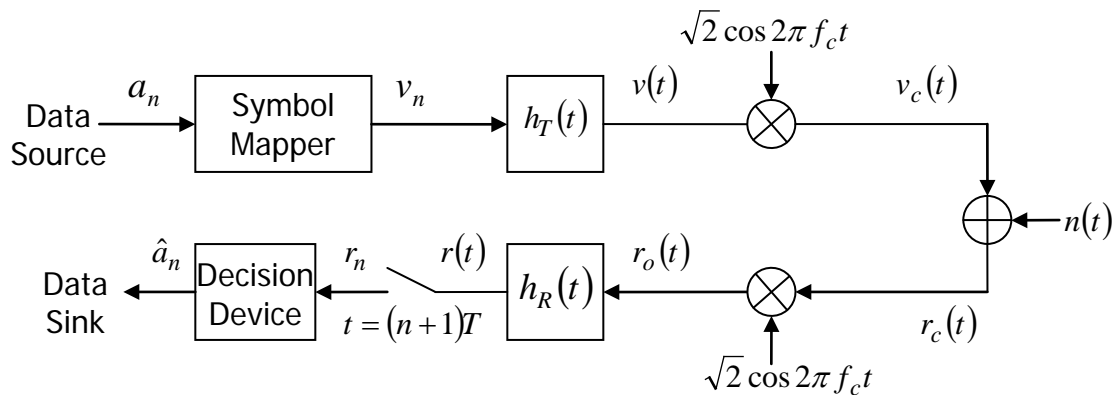


Figure 1: Simplified block diagram of a baseband BPSK communication system.

As shown in Fig. 1, the symbol mapper converts the message bits into real-valued amplitudes, v_n . The transmit filter uses the amplitudes from the symbol mapper, $\underline{v} = [v_0 \ v_1 \ v_2 \ \cdots \ v_{N_a-1}]$, to generate an analogue pulse train,

$$v(t) = \sum_{n=0}^{N_a-1} v_n h_T(t - nT),$$

where T is the pulse duration (also known as the symbol duration or symbol period), and $h_T(t)$ is the pulse shape. Then, the baseband pulse train, $v(t)$, is multiplied by a sinusoidal carrier to generate a BPSK modulated signal, $v_c(t)$, which is transmitted over the AWGN channel. At the receiver, a coherent demodulator extracts the data by first down converting the received BPSK signal, $r_c(t)$, to baseband, giving $r_o(t)$. The baseband signal is applied to a receive filter (with impulse response $h_R(t)$), and the filter output, $r(t)$, is sampled at the symbol rate (one sample every T seconds). The samples are then passed to a decision device.

Part I of this laboratory will consider the effect of the choice of the transmit filter on the power spectral density of the transmitted signal.

Let us assume that a polar NRZ line code with a rectangular pulse shape time-limited from 0 to T is used in a baseband communication system, over an ideal noiseless channel and assume no filtering at the receiver (i.e., $h_R(t) = \delta(t)$), then the output of the receiver filter $r(t)$ to a_0 spans in time from 0 to T . When a sequence of bits is transmitted, then the output to the second bit will span the time from T to $2T$, and thus will not interfere with the output to the first bit. Therefore in that case there is no ISI, and a decision can be made on \hat{a}_0 by sampling the output of the receiver filter at any time between 0 and T . Furthermore, because the channel was assumed to be noiseless, no decision error will be made.

In practice, the channel introduces some additive noise, and therefore the receiver includes a receiver filter before sampling to improve performance. The purpose of the receiver filter is to filter part of the noise to reduce the number of decision errors. As shown in the lecture notes, the filter that will maximize the signal-to-noise ratio at the output of the receiver filter at a given sampling time (and hence will reduce the number of decision errors over an AWGN channel) is the matched filter, i.e, a filter with impulse response $h_R(t) = h_T(T - t)$ assuming a sampling time equal to T .

A second effect will occur involving the principle of duality between time and frequency, namely absolutely time-limited waveforms cannot be absolutely band-limited and absolutely band-limited waveforms cannot be absolutely time-limited. In practice, the channel will have a finite bandwidth; therefore the transmitter will have to include filters to ensure that the bandwidth of the transmitted signal after filtering is smaller than the channel bandwidth. Due to the duality between time and frequency, considering a transmitter filter of finite bandwidth means that the filter impulse response has infinite time duration. Practically, $H_T(f)$ will be almost band-limited and $h_T(t)$ will be almost time-limited, but of duration that may span more than one bit duration (more than T). This means that the output to one bit will span from 0 to more than T , and thus will

interfere with the output to the second bit of the transmitted sequence. In other words ISI will occur. In that case, the output of the receiver filter at the sampling time used to detect \hat{a}_n is composed of a term proportional to \hat{a}_0 , a noise term and a term that represents the contribution of the past bits, called ISI. The value of the ISI term will depend on channel impulse response ($h_c(t)$), the choice of $h_T(t)$ and $h_R(t)$, the choice of sampling time as well as the values of the past transmitted bits. Note that in a system model it is also possible to include all the filtering effect of the channel in the transmitter filter $h_T(t)$ and assume for simplicity that $h_c(t) = \delta(t)$; this approach is considered in this lab. If the composite pulse shape $h(t) = h_T(t) * h_c(t) * h_R(t)$ satisfies Nyquist criterion, like raised cosine or sinc pulse shapes for example, then the ISI term equals to zero. Note that this requires to know the channel impulse response exactly, so in practice even though $h_T(t)$ and $h_R(t)$ are designed so that Nyquist criterion is satisfied, there is always some residual ISI due to the imperfect estimation of the channel impulse response. In one example in this lab, $h_T(t)$ and $h_R(t)$ will be squared root raised cosine pulse shapes yielding a raised cosine composite pulse $h(t) = h_T(t) * h_R(t)$.

Choice of sampling time:

Let us assume that it is known that $a_0 = 1$ (or equivalently $v_0 = 1$ for polar NRZ) was transmitted, then the best sampling time would be the one such that the value of the output of the receiver filter at the sampling time (including the ISI and noise term) is the closest to 1, in other words is maximum. Let us assume that it is known that $a_0 = 0$ (or equivalently $v_0 = -1$ for polar NRZ) was transmitted, then the best sampling time would be the one such that the value of the output of the receiver filter at the sampling time (including the ISI and noise term) is the closest to -1, in other words is minimum. Of course in practice, the value of a_0 is not known beforehand, so the sampling time should be chosen taking into account all possible cases. Since the ISI term depends on past transmitted bits, all possible transmitted sequences should be considered. The eye diagram is a convenient way to find the best sampling time as well as how sensitive a communication system is to noise given a choice of transmitter filter and receiver filter. The eye diagram at the output of the receiver filter is obtained by plotting on the same graph the output of the receiver filter assuming all possible combination of transmitted bits (i.e., all possible transmitted sequences). By looking at the eye diagram, the best sampling time that will be suitable for both cases can be deduced. Usually, the best sampling time is at a point where the eye is opened at the widest.

A typical eye diagram is as follows:

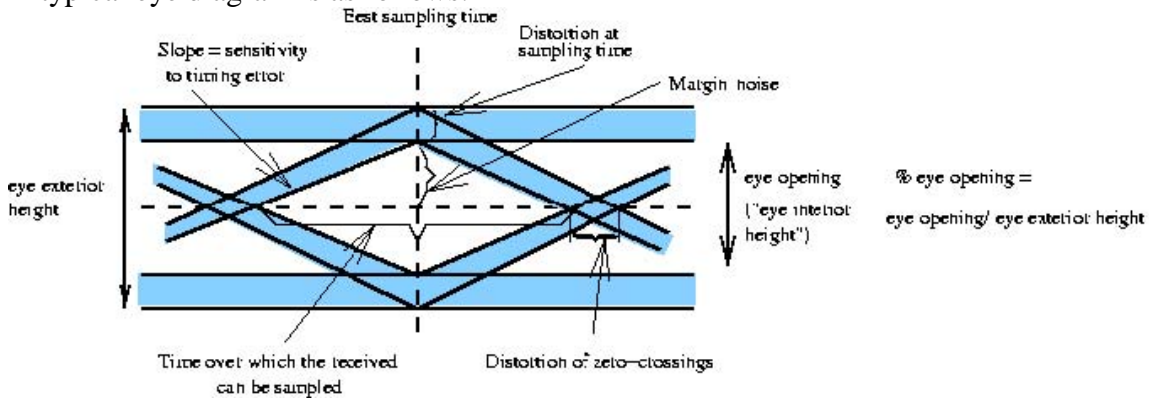


Figure 2: Eye diagram.

The width represents the time interval over which the received signal can be sampled without error from ISI.

The interior region of the eye diagram is called the eye opening (“eye interior height”). The % eye opening is defined as the eye opening (“eye interior height”) divided by the eye exterior height.

The height of the eye opening compared to the center of the eye diagram, at a specified sampling time defines the noise margin of the system, that is how much noise can be added before a pulse that would be classified as corresponding to ‘1’ (by the detector) is so much distorted (due to the additive noise) that it “changes sign” (i.e., would be classified as corresponding to ‘-1’ by the detector). The rate of closure of the eye as the sampling time is varied (i.e., slope of the eye) determines the sensitivity of the eye to timing errors.

Usually the eye diagram is drawn by considering the output of the receiver filter so that the best sampling time for detection can be deduced. Note that another eye diagram can also be constructed by considering the output of the transmitter filter. This would correspond to considering a detector that makes decision at the output of the transmitter (looking at the effect of the transmitter filter alone).

Part II of this laboratory will investigate the effect of the transmitter filter ($h_T(t)$) and the receiver filter ($h_R(t)$) on the communication system by looking at the eye diagrams at their outputs as well as comparing the two eye diagrams.

3 Prelaboratory

1. Find and plot the theoretical power spectral density (PSD) of a BPSK signal with a square root raised-cosine (SRRC) pulse shape in Watts/Hz and in dB/Hz (assume a roll-off factor of 50%, a carrier frequency of 100 Hz, and data rate of 63 bit/sec). Note that the magnitude response of a SRRC filter is $|H_{\text{srrc}}(f)| = |H_{\text{rc}}(f)|^{1/2}$.
2. Using the course notes, find the theoretical bit error rate of BPSK detected coherently as a function of E_b/N_0 . Using Matlab, plot the obtained BER expression as a function of E_b/N_0 expressed in dB. Use semilogy to plot your BER so that log scale is used for the y axis and a linear scale is used for the x axis. Your plot should display E_b/N_0 from 0 to 10 dB and should look similar to the coherent detection BPSK curve given in the textbook.
3. Find and plot the theoretical power spectral density of a QPSK signal with a SRRC pulse shape in Watts/Hz and in dB/Hz (assume a roll-off factor of 50%, a carrier frequency of 100 Hz, and data rate of 126 bit/sec).

4 Matlab Models

Figure 3 shows a version of Fig. 2 that is more simulation oriented.

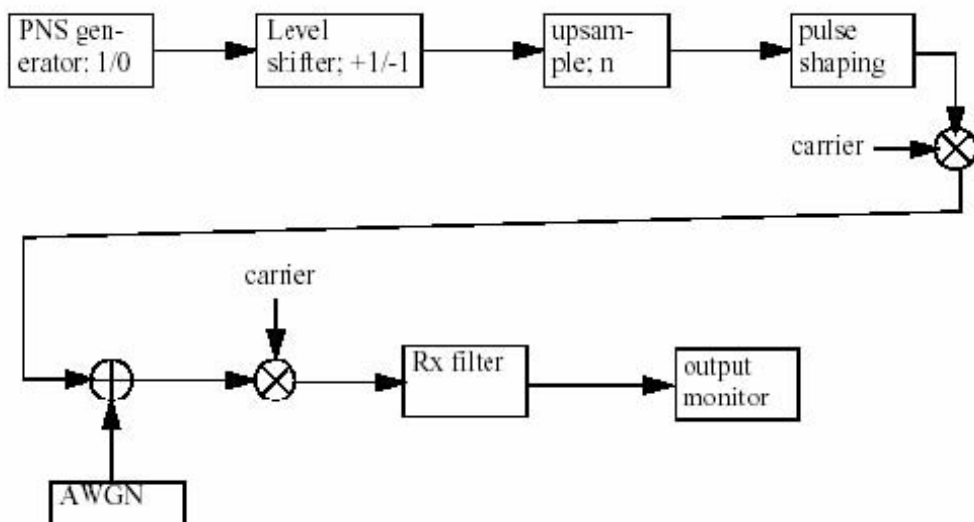


Figure 3: Simulation procedure.

Here is how this simulation model would work:

- Generate a random binary sequence (1's and 0's).

- Use a level shifter to change the sequence values from binary to +1 and -1
- Upsample the binary sequence by a factor of 16. This is achieved by inserting 15 zeros between consecutive bits.
- The upsampled sequence is applied to an FIR filter that does the required pulse shaping.
- The output of the pulse shaping filter is applied to a double-side-band modulator (i.e. a multiplier) for carrier modulation.
- The output of the multiplier is passed to an AWGN channel.
- The channel output is applied to a demodulator (another multiplier), and the resulting signal is applied to a receive filter which could be a matched filter.

The simulation model has been implemented in simulink using standard blocks from simulink, DSP blockset, and the Communications blockset. A number of custom blocks, made for the purpose of this lab have been incorporated in the simulation models. The custom blocks are various implementations for the pulse shaping filter and the Rx filter. All custom blocks are in the simulink model titled “pulse_shapes”. The contents of the model are shown in Fig. 4:

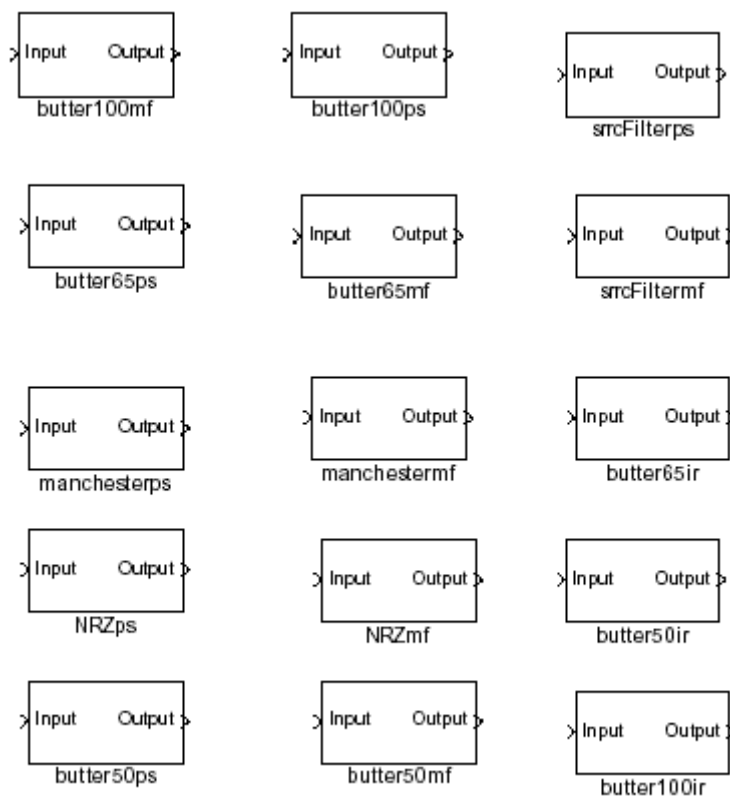


Figure 4: Custom pulse shaping and Rx filter blocks.

The functions of the various blocks are explained in Table 1.

Table 1:

block	function	block	function
NRZps	NRZ pulse shape, no Tx filter	NRZmf	filter matched to NRZps
Manchesterps	Manchester coded pulse, no Tx filter	Manchestermf	Filter matched to Manchesterps
butter100ps	NRZ pulse shape convolved with a 4th order Butterworth filter with a BW equal to the data rate	butter100mf	filter matched to butter100ps
butter65ps	NRZ pulse shape convolved with a 4th order Butterworth filter with a BW equal to 65% of the data rate	butter65mf	filter matched to butter65ps
butter50ps	NRZ pulse shape convolved with a 4th order Butterworth filter with a BW equal to 50% of the data rate	butter50mf	filter matched to butter50ps
srrcfilterps	square-root raised cosine pulse shape, roll-off factor=50%	srrcfiltermf	filter matched to srrcfilterps
butter100ir	butterworth filter, 4th order, cutoff-frequency equal to 100% of the bit rate		
butter65ir	butterworth filter, 4th order, cutoff-frequency equal to 65% of the bit rate		
butter50ir	butterworth filter, 4th order, cutoff-frequency equal to 50% of the bit rate		

Figures 5 and 6 show two BPSK models meant for spectral analysis/simulation and for BER simulation, respectively. Figure 7 shows the QPSK simulation model.

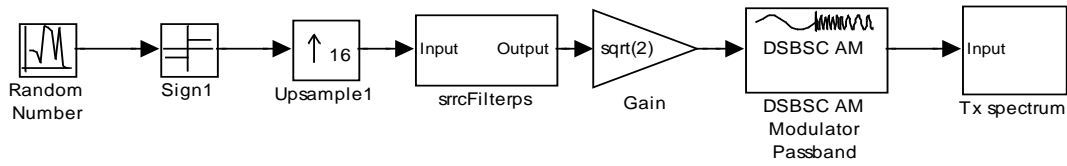


Figure 5: BPSK simulation model.

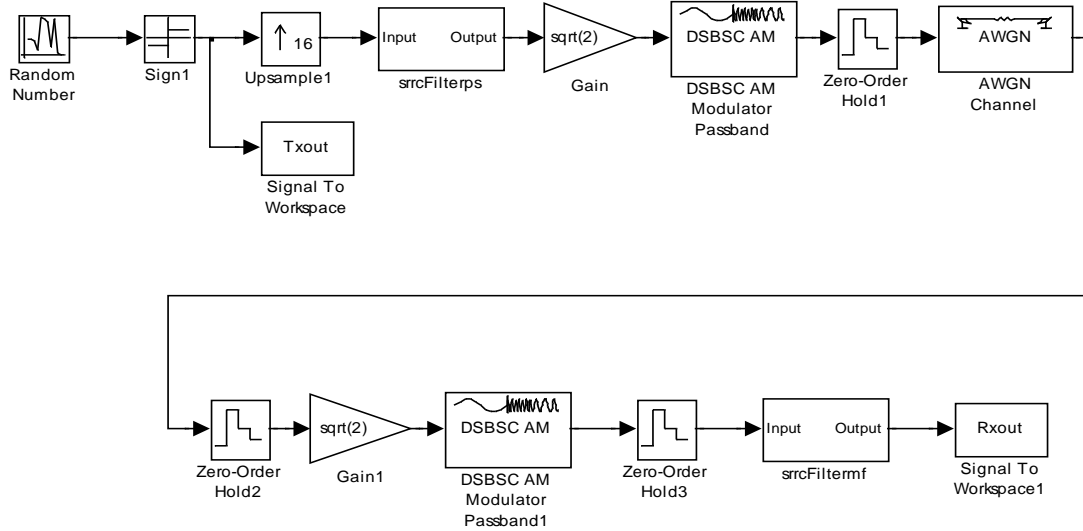


Figure 6: BPSK BER simulation model.

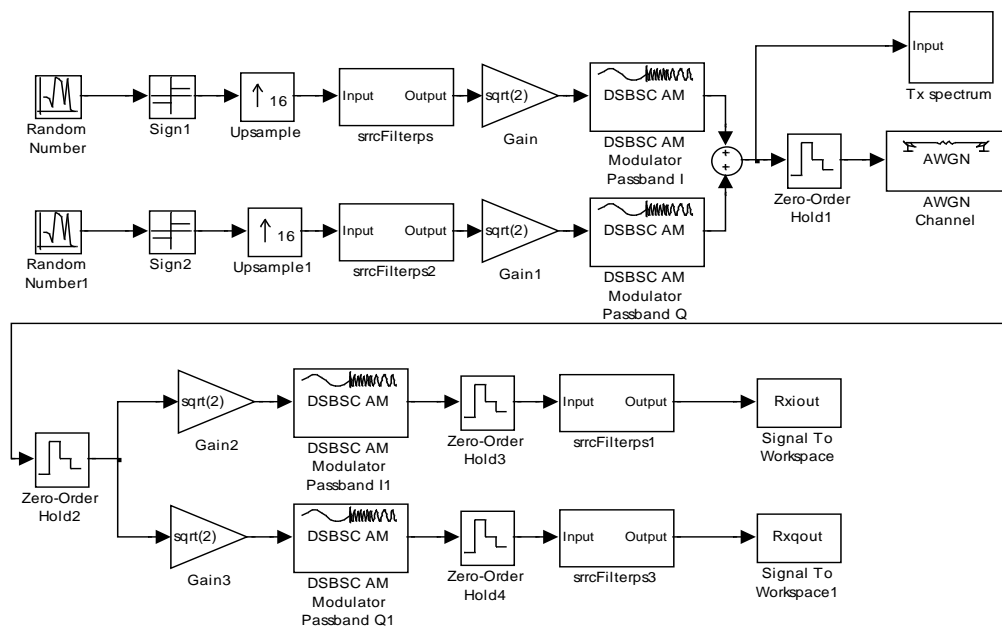


Figure 7: QPSK simulation model.

5 Laboratory

Create the folder M:\lab3

Download all Matlab files for Lab 3 to the folder M:\lab3, from the course website:

<http://www.sce.carleton.ca/courses/sysc-4600/f12/labs.shtml>

Start Matlab

Change the current directory to M:\lab3

Start simulink

Part I: BPSK Spectral Analysis

Open the model titled “**BPSK_PSD.mdl**” which should appear as shown in Fig. 5.

1) Run the simulation. Print the spectrum of the transmitted signal. Compare the result you obtained with the theoretical spectrum obtained in your prelab. Base the comparison on the relative dB values only.

2) Open the model titled “pulse_shapes” which contains the custom blocks. This should appear as shown in Fig. 5. Replace the pulse shaping filter by the block “NRZps” and run the simulation. Print the spectrum of the transmitted. Compare the spectrum of the transmitted you obtained in this part with the one you obtained using SRRC filter in Step 1.

Part II: BPSK BER Simulation

Open the model titled “**BPSK_BER.mdl**” which should appear as shown in Fig. 6.

1) Double click the AWGN channel model. Set the desired Eb/N0 to 50 dB. Leave all other parameters unchanged. Run the simulation for 100 sec.

Once the simulation is completed you would have two variables named “Txout” and “Rxout” in your work space.

Plot and print the received eye-diagrams using the command:

```
>> eyed(Rxout)
```

2) Double click the AWGN channel model. Set the desired Eb/N0 to 0 dB. Leave all other parameters unchanged. Run the simulation for 100 sec.

Now use the variables Txout and Rxout to compute the expected BER. For this step use the command:

```
>> BER(Txout, Rxout)
```

3) Repeat Step 2 for different values of E_b/N_0 : 2, 4, 6, 8, and 10 dB.

Plot the BER results obtained from simulation. Compare the simulation and theoretical results obtained in your prelab.

4) Set the desired E_b/N_0 to 0 dB. Double click the block labeled “DSB-SC AM passband1”. Set the carrier phase to $\pi/8$. Run the simulation for 100 sec. Use the Matlab function “BER” to compute the BER based on the workspace variables “Txout” and “Rxout” as you did in the previous two steps.

5) Repeat Step 4 for different values of E_b/N_0 : 2, 4, 6, 8, and 10 dB.

Plot the BER results obtained from simulation. Compare the BER curve you obtain in this step with the one you obtained in Step 3. Find an explanation for this result.

Part III: Qualitative Experiment on QPSK

Open the model titled “**QPSK.mdl**” which should appear as shown in Fig. 7.

1) Run the simulation. Print the spectrum of the transmitted signal. Is the experimental plot consistent with the analytical model of the PSD obtained from your prelab? Explain.

2) Double click the AWGN channel model. Set the desired E_b/N_0 to 10 dB. Leave all other parameters unchanged. Run the simulation for 100 sec. Examine and print the eye diagram and signal constellation for the QPSK system.

Use the command:

```
>>eyed_qpsk(Rxiout,Rxqout)
```

3) Repeat Step 2 for different values of E_b/N_0 : 20, 30, and 40 dB. Explain what you observe.
