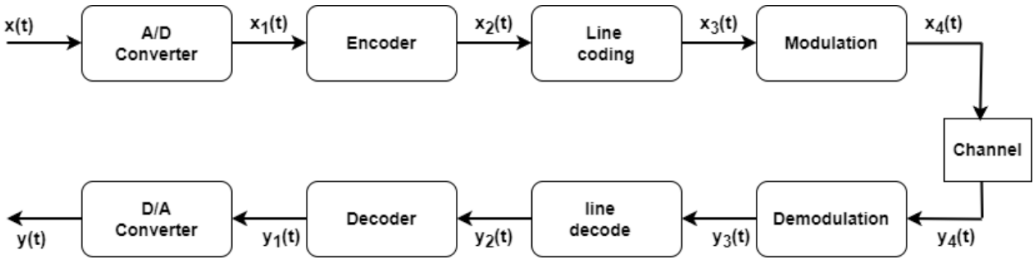
COMMUNICATION THEORY COURSE PROJECT*

Name: Shaik Affan Adeeb
Roll No: 2022102054
ECE, IIIT Hyderabad
shaik.affan@students.iiit.ac.in

Name: V.V.S.R Chetan Krishna
Roll No: 2022102047
ECE, IIIT Hyderabad
chetan.vellanki@students.iiit.ac.in

Introduction :

In this project, we will simulate our own communication model in MATLAB, using the concepts and techniques covered in the entire course. Below is the block diagram that describes the steps we will follow in our simulation:



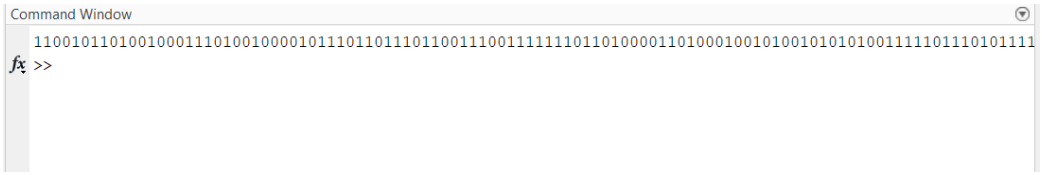
System model

A/D converter :

This block reads an audio file ('project.wav') and converts it into a binary bit stream. The audio samples are first normalized, then converted to uint16 format, and finally represented as a binary sequence.

```
%% A/D converter :
[y, Fs] = audioread('project.wav');
downsampling_factor = 4;
y_downsampled = resample(y, 1, downsampling_factor);
audio_normalized=y_downsampled/max(abs(y_downsampled));
audio_normalized_reshaped = audio_normalized(:);
audio_uint16 = typecast(audio_normalized_reshaped, '
   uint16');
audio_binary_temp = dec2bin(audio_uint16, 16);
audio_binary = audio_binary_temp';
bits = audio_binary(:)';
```

Listing 1: Code for A/D conversion



Output $X_1(t)$ of A/D conversion (still many bits are there in output window i have shown some of them)

Encoding :

We have to implement Quadrature phase-shift keying (QPSK) type 2 :
The binary bit stream is mapped to QPSK symbols based on the two-bit combinations (00, 01, 11, 10). Each symbol represents two bits and is represented as a complex number with in-phase and quadrature components. The QPSK constellation is plotted to visualize the symbols.

SNR Calculation

The signal-to-noise ratio (SNR) is typically represented as the ratio of the energy per bit (E_b) to the noise power spectral density (N_0), expressed in decibels (dB):

$$\text{SNR (dB)} = 10 \log_{10} \left(\frac{E_b}{N_0} \right) \tag{1}$$

where E_b is the energy per bit, and N_0 is the noise power spectral density.

BER Calculation for QPSK in AWGN

The bit error rate (BER) for QPSK in an AWGN channel is given by:

$$\text{BER} = Q \left(\sqrt{2 \frac{E_b}{N_0}} \right) \tag{2}$$

where $Q(x)$ is the Gaussian Q-function, defined as:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{u^2}{2}} du \tag{3}$$

The Q-function represents the tail probability of the standard normal distribution.

For QPSK, each symbol carries two bits of information. The factor of $\sqrt{2}$ in the BER formula is due to the fact that QPSK can be viewed as two

independent BPSK (Binary Phase Shift Keying) signals, one in the I (in-phase) channel and one in the Q (quadrature) channel. The probability of a bit error is the probability of making a decision error in either the I or Q channel, hence the factor of $\frac{1}{2}$ in the BER formula.

Derivation of the BER Formula

Let's consider the I channel (similar derivation applies to the Q channel). The received signal in the I channel is $r = s + n$, where s is the transmitted BPSK signal ($\pm\sqrt{E_b}$), and n is the AWGN with zero mean and variance $N_0/2$.

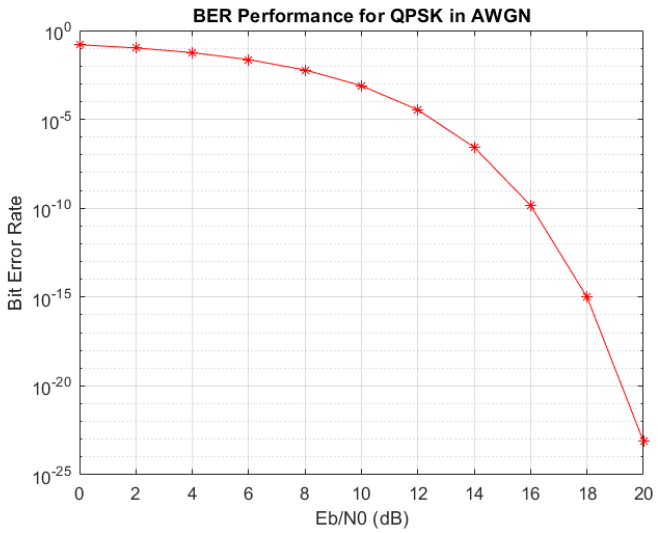
The probability of a bit error is $P(r < 0|s = \sqrt{E_b}) + P(r > 0|s = -\sqrt{E_b})$. Using the Gaussian distribution of the noise, we get:

$$P(r < 0|s = \sqrt{E_b}) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{4}$$

$$P(r > 0|s = -\sqrt{E_b}) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{5}$$

Since the I and Q channels are independent, the total BER is:

$$\text{BER} = \frac{1}{2}\left[Q\left(\sqrt{\frac{2E_b}{N_0}}\right) + Q\left(\sqrt{\frac{2E_b}{N_0}}\right)\right] = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \tag{6}$$



Power Efficiency

The power efficiency of a modulation scheme is typically measured by the E_b/N_0 value required to achieve a target BER performance. A lower required E_b/N_0 means higher power efficiency, as the same performance can be achieved with less energy per bit or less transmit power.

For QPSK in AWGN, the E_b/N_0 required to achieve a BER of 10^{-6} is approximately 9.8 dB.

This value can be found by substituting the target BER value (10^{-6}) into the BER formula and solving for E_b/N_0:

$$Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = 10^{-6} \tag{7}$$

$$\sqrt{\frac{2E_b}{N_0}} = Q^{-1}(10^{-6}) \approx 4.77 \text{ (Using the inverse Q-function table)} \tag{8}$$

$$\frac{E_b}{N_0} \approx \left(\frac{4.77}{\sqrt{2}}\right)^2 \approx 9.8 \text{ dB} \tag{9}$$

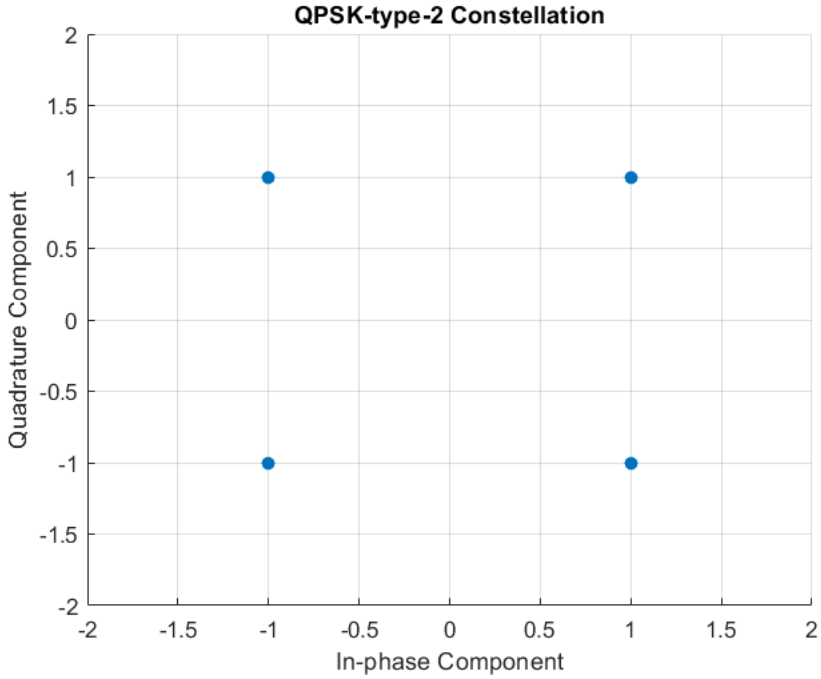
The power efficiency of QPSK can be compared with other modulation schemes by comparing their respective E_b/N_0 values required to achieve the same target BER.

Furthur details about SNR,BER and power efficiency have been included at last after completing all the blocks.

```
%% Encoder :
num_bits=length(bits); a=1;
symbols=zeros(1,num_bits/2); m=1;
for k = 1:2:num_bits-1
    if((bits(k)==bits(k+1)) && bits(k)=='0')
        symbols(m)=a+a*1j;
    end
    if(bits(k)=='0' && bits(k+1)=='1')
        symbols(m)=a-a*1j;
    end
    if((bits(k)==bits(k+1)) && bits(k)=='1')
        symbols(m)=-a-a*1j;
    end
    if(bits(k)=='1' && bits(k+1)=='0')
        symbols(m)=-a+a*1j;
```

```
    end
    m=m+1;
end
```

Listing 2: Code for Encoding



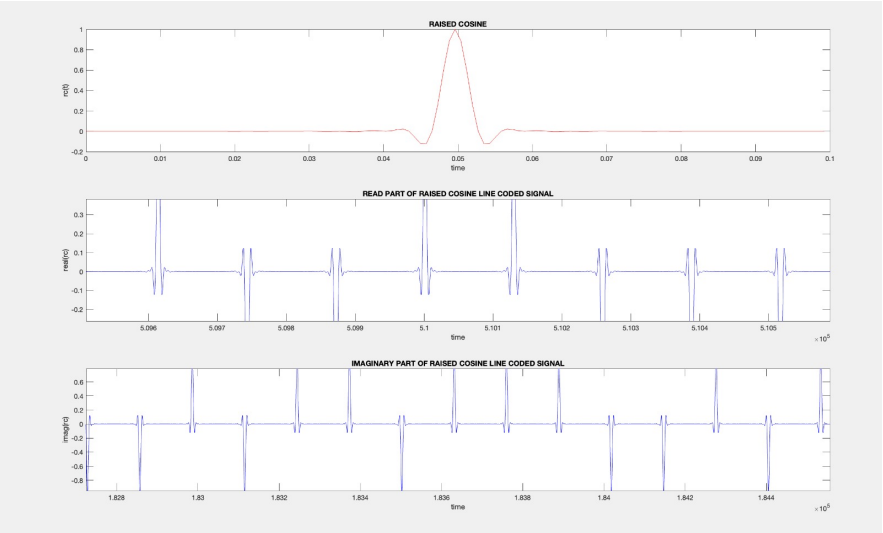
Output $X_2(t)$ of Encoder

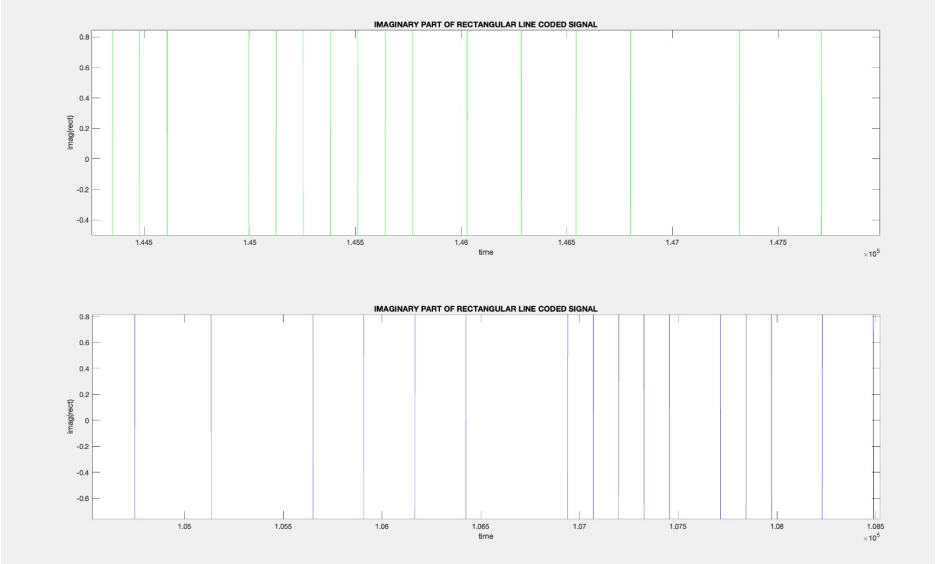
```
Columns 5,86,261 through 5,86,265

-1.0000 + 1.0000i  -1.0000 - 1.0000i   1.0000 + 1.0000i   1.0000 + 1.0000i   1.0000 + 1.0000i

Columns 5,86,266 through 5,86,270

-1.0000 - 1.0000i  -1.0000 - 1.0000i  -1.0000 - 1.0000i  -1.0000 + 1.0000i  -1.0000 + 1.0000i

Columns 5,86,271 through 5,86,275
```

Output $X_2(t)$ of Encoder (still many symbols are there in output window i have shown some of them)

Line coding :

Two pulse shaping techniques are used: raised cosine and rectangular pulses. The raised cosine pulse is generated with parameters like excess bandwidth (alpha), oversampling factor (m), and pulse length (n). The rectangular pulse is a simple NRZ (Non-Return-to-Zero) waveform. These pulses are upsampled and convolved with the QPSK symbols to generate the line-coded signals. Below are plots of $\mathbf{X_3(t)}$ for raised cosine and rectangular p(t) pulse.





Modulation :

```
%% Modulation :
fc = 1e6;  % Carrier frequency (Hz)
Ac = 2;    % Carrier amplitude
wc = 2 * pi * fc;  % Carrier angular frequency
% Calculate the total time duration based on symbol
   duration (Tb) and
% number of symbolsq
% Modulate using Rectangular Pulse
x4_rect_I = Ac * real(tx_rect) .* cos(wc *
   time_axis_line_coded);
x4_rect_Q = Ac * imag(tx_rect) .* sin(wc *
   time_axis_line_coded);
x4_rect = x4_rect_I + x4_rect_Q;
% Modulate using Raised Cosine Pulse (assuming you have
   defined x3_raised_cosine)
x4_raised_cosine_I = Ac * real(tx_raised_cosine) .* cos(
   wc * time_axis_line_coded);
x4_raised_cosine_Q = Ac * imag(tx_raised_cosine) .* sin(
   wc * time_axis_line_coded);
x4_raised_cosine = x4_raised_cosine_I +
   x4_raised_cosine_Q;
```

Listing 3: Code for Modulation

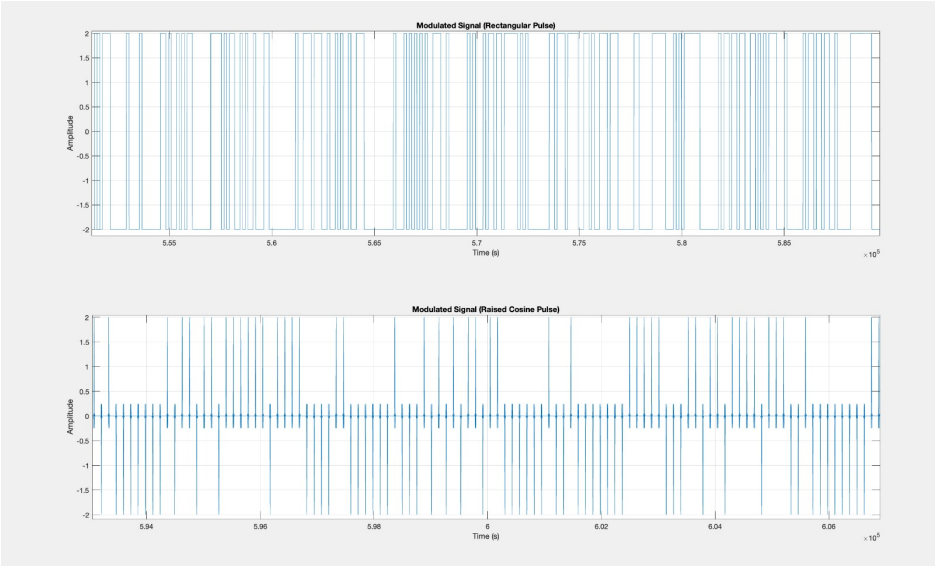
We have this type of modulation(as mentioned in code) technique due to following reasons:

- Orthogonality: The cosine and sine waveforms are orthogonal to each other, meaning that their inner product over one period is zero. This orthogonality property allows the in-phase and quadrature components to be transmitted simultaneously without interfering with each other.
- Efficient Use of Bandwidth: By modulating the in-phase and quadrature components onto the same carrier frequency but with a 90-degree phase shift (cosine and sine), the available bandwidth can be utilized more efficiently compared to transmitting the components separately.
- Complex Signal Representation: The QPSK signal is inherently a complex signal, with the in-phase (real) and quadrature (imaginary) components representing the two bits encoded in each symbol. The quadrature modulation technique provides a way to represent and transmit this complex signal over a real-valued channel.

- Demodulation and Detection: At the receiver end, the same quadrature demodulation technique can be used to recover the in-phase and quadrature components separately. This involves multiplying the received signal with cosine and sine carriers and applying low-pass filters to extract the respective components.

- Compatibility with IQ Modulator/Demodulator Hardware: The quadrature modulation approach aligns well with the hardware architecture of IQ modulators and demodulators, which are commonly used in practical communication systems to efficiently generate and process complex modulated signals.

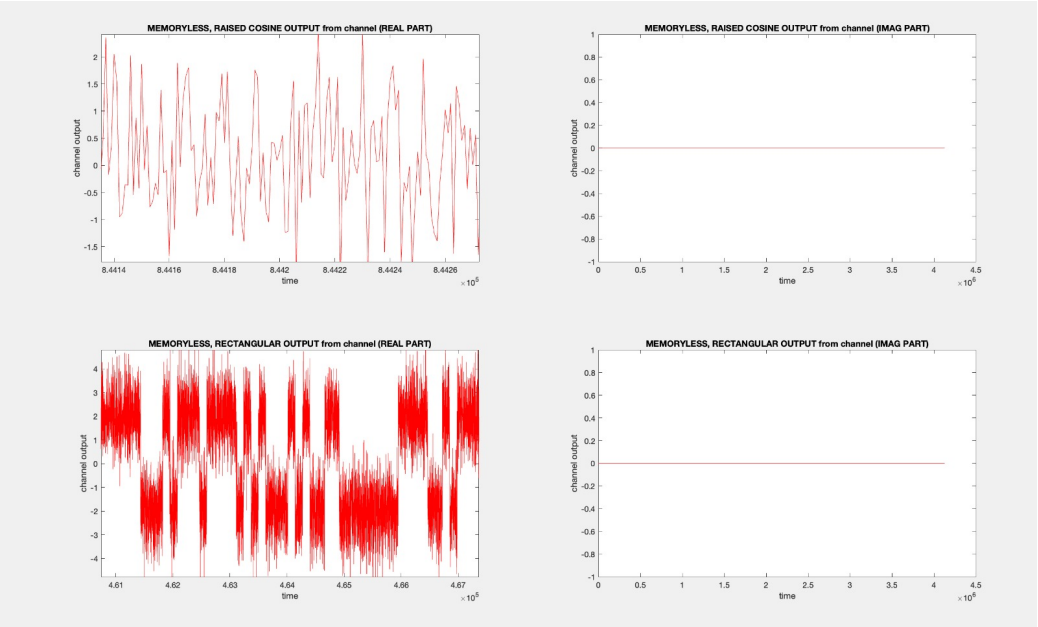
While this quadrature modulation technique is widely used for QPSK and other complex modulation schemes, it is not the only possible implementation. Alternative methods, such as using a single carrier and transmitting the in-phase and quadrature components sequentially, or using different carrier frequencies for each component, are also possible but may have different trade-offs in terms of bandwidth efficiency, hardware complexity, and synchronization requirements. The choice of the quadrature modulation technique in this simulation is likely due to its simplicity, efficiency, and compatibility with common communication system architectures and hardware implementations.



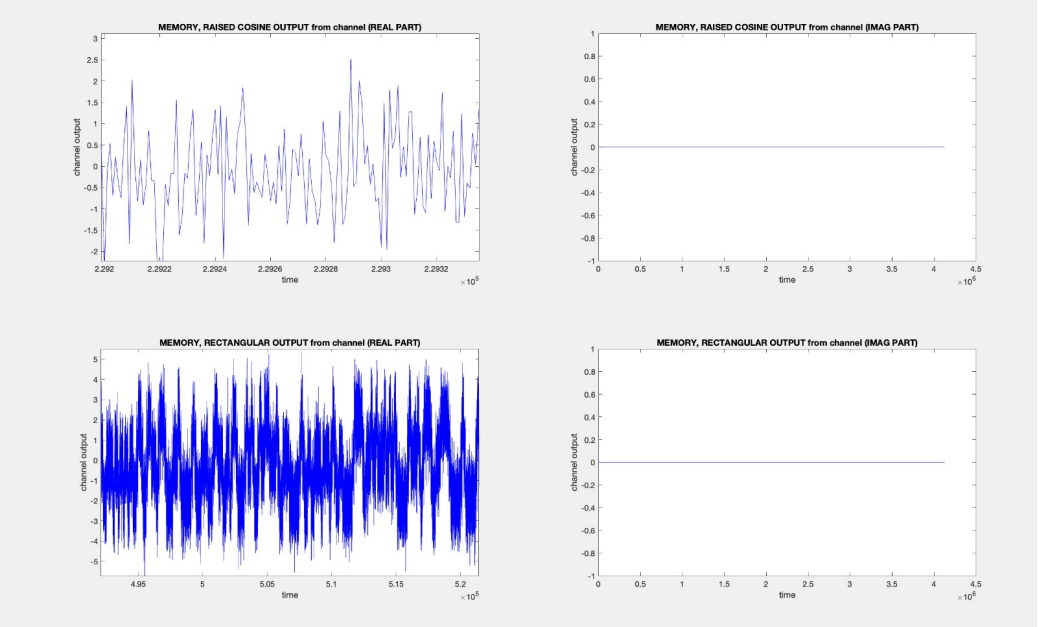
$X_4(t) \implies$ Modulated signals (rectangular pulse and raised cosine pulse)

Channel and Demodulation :

Two channel models are considered: memoryless AWGN (Additive White Gaussian Noise) and a channel with memory. In the memoryless channel, white Gaussian noise is added to the modulated signals. In the channel with memory, the signal is passed through a filter with an impulse response characterized by parameters 'a' and 'b'. The received signals after the channels are plotted.



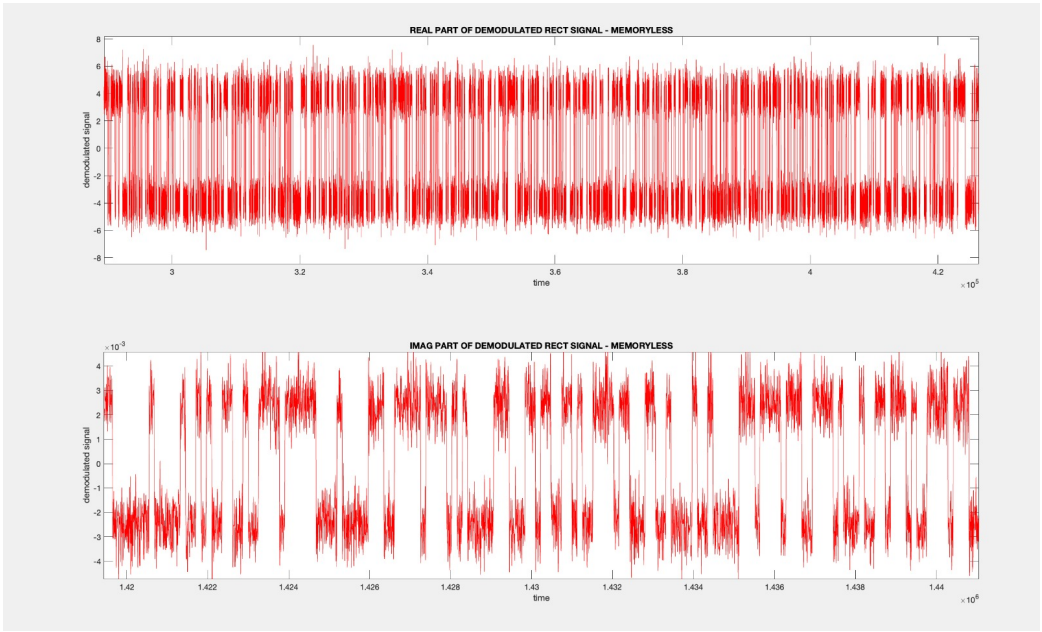
Memoryless AWGN channel output $y_4(t)$



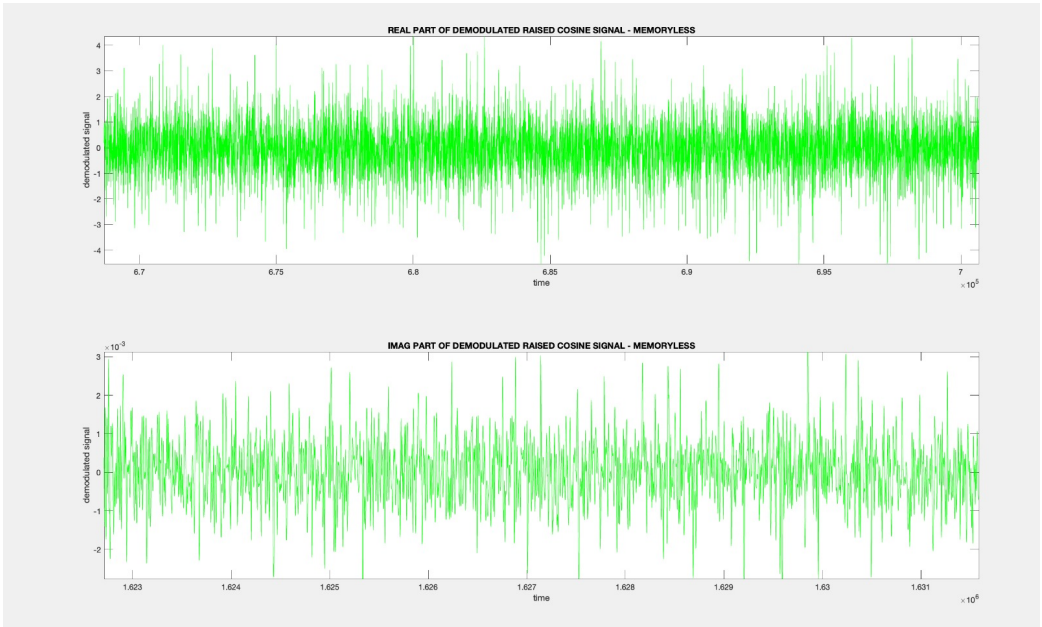
AWGN channel with memory output $y_4(t)$

Demodulation,line decode and decoder :

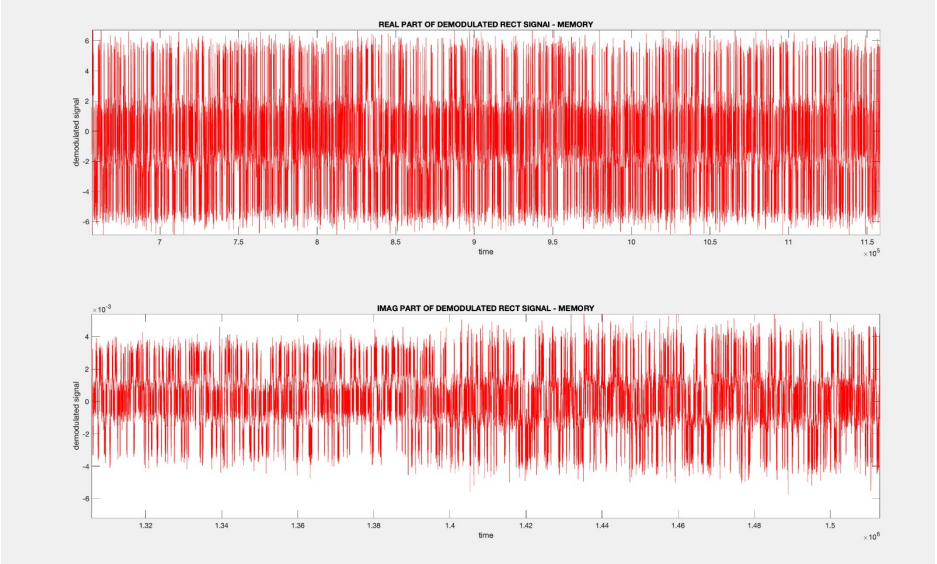
The received signals from both channels (memoryless and memory) are demodulated by multiplying with the carrier frequency and applying a low-pass filter to retrieve the baseband signals. The real and imaginary parts of the demodulated signals are plotted.



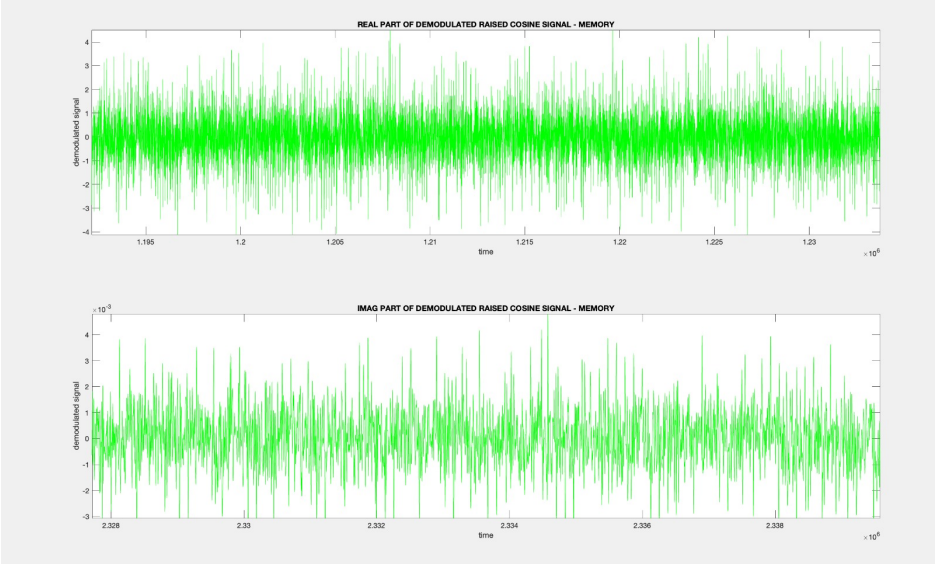
Demodulated rect memory less channel output $y_3(t)$



Demodulated raised cosine memory less channel output $y_3(t)$



Demodulated rect, channel with memory output $y_3(t)$



Demodulated raised cosine, channel with memory output $y_3(t)$

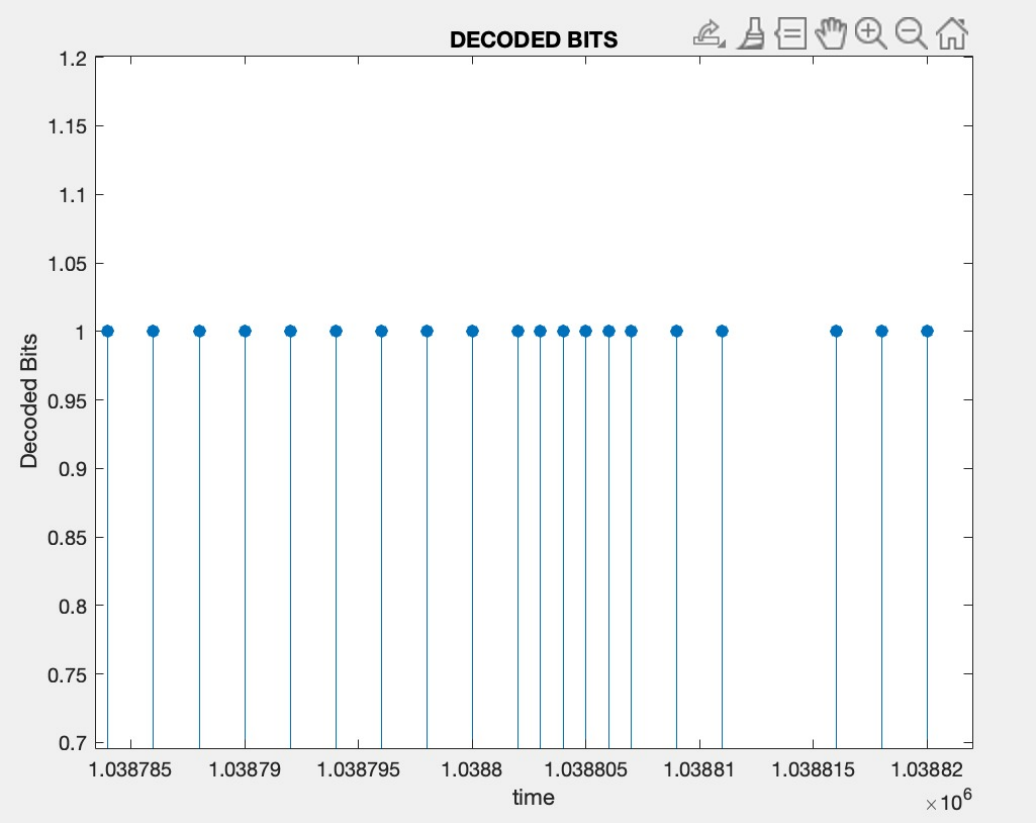
Line Decode :

```
   1     1     1     1     1     1

Columns 51,76,263 through 51,76,269

   1     1     1     1     1     1

Columns 51,76,270 through 51,76,276

   1     1     1     1    -1    -1    -

Columns 51,76,277 through 51,76,283

  -1    -1    -1    -1    -1    -1    -

Columns 51,76,284 through 51,76,290

  -1    -1    -1    -1    -1    -1    -

Columns 51,76,291 through 51,76,297

  -1    -1    -1    -1     1     1

Columns 51,76,298 through 51,76,304

   1     1     1     1     1     1

Columns 51,76,305 through 51,76,311

  -1    -1    -1    -1    -1    -1    -

Columns 51,76,312 through 51,76,318

  -1    -1    -1    -1    -1    -1    -

Columns 51,76,319 through 51,76,325
```

Line decode output $y_2(t)$

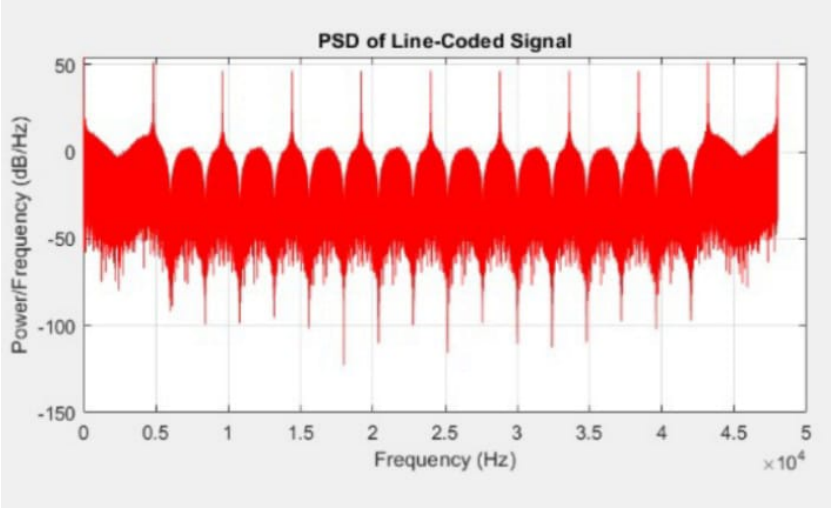
(we have shown only some of them as far as possible in terminal.)

Decoder :

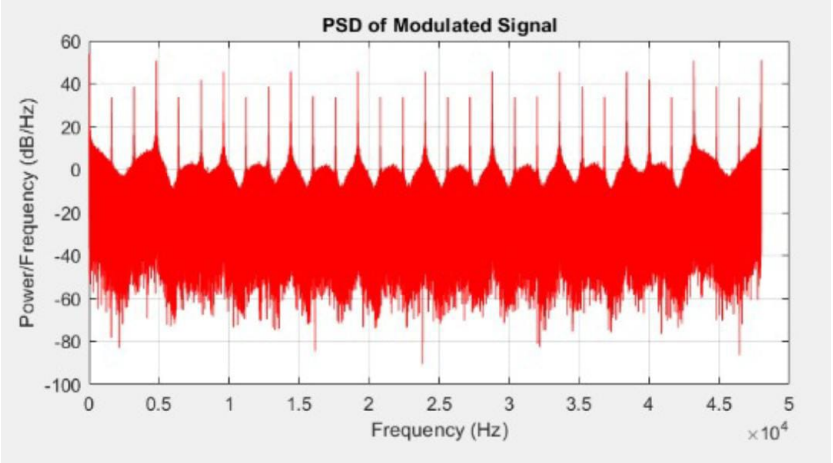


Decoder output $y_1(t)$

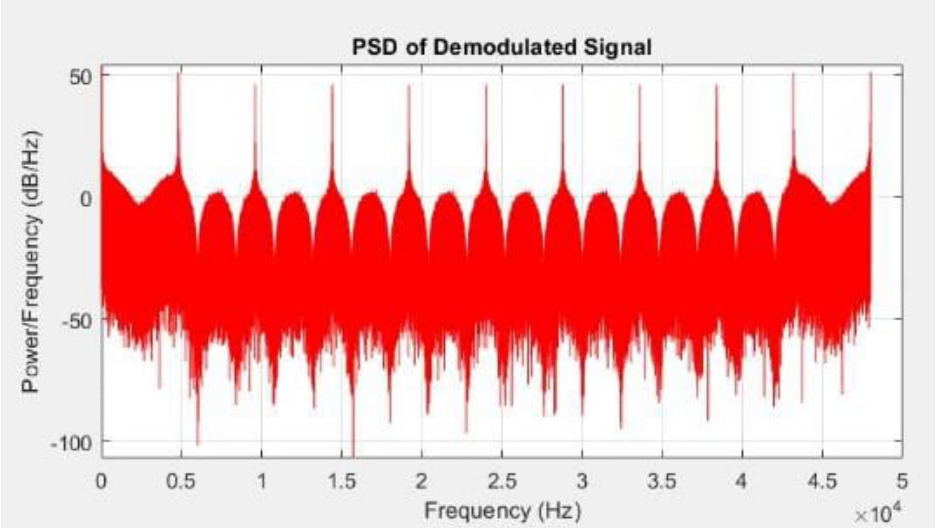
PSD's :



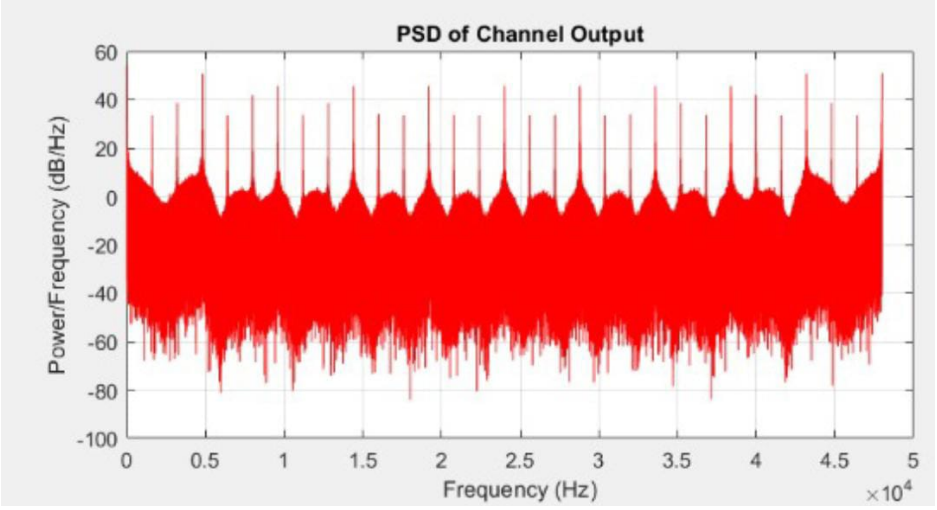
PSD of Line coded signal $X_3(t)$



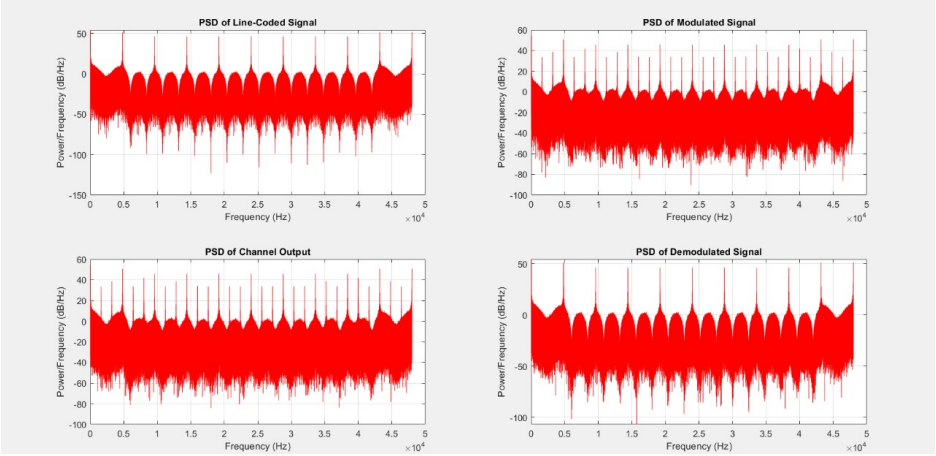
PSD of Modulated signal $X_4(t)$



PSD of DeModulated signal $y_3(t)$

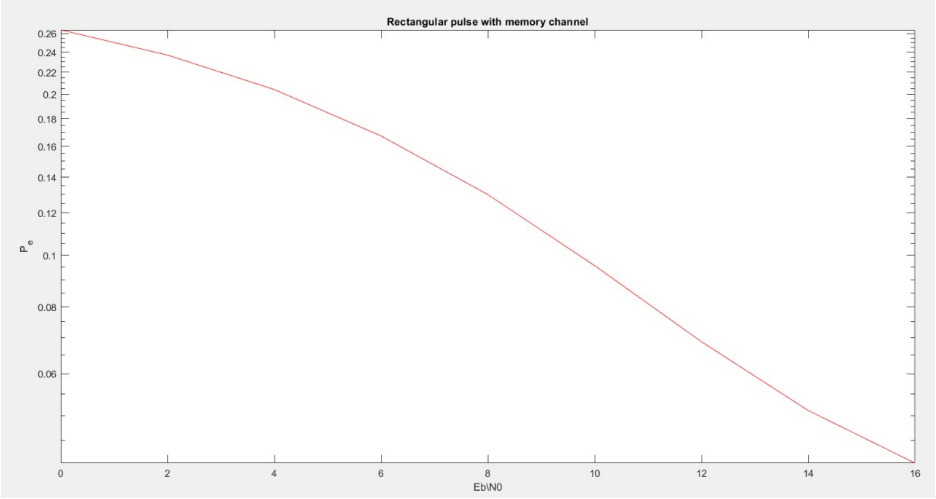


PSD of Channel output $y_4(t)$

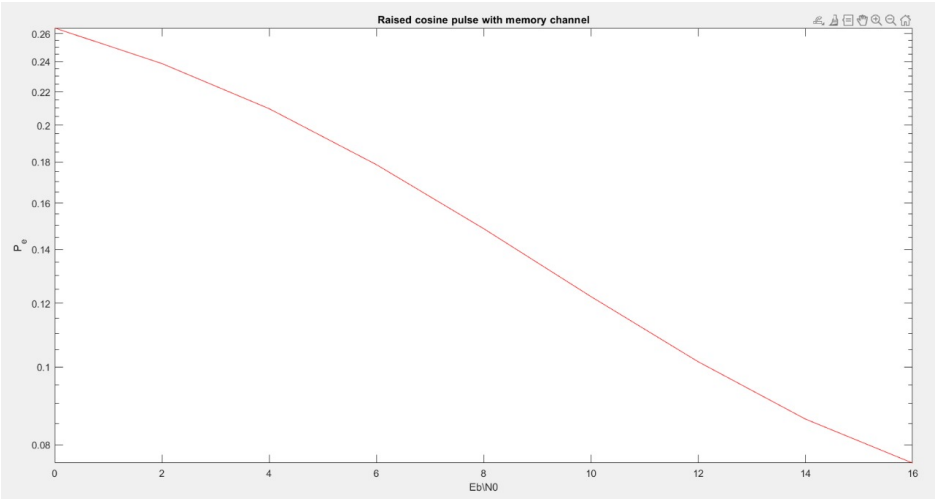


PSD's of all outputs which are expected

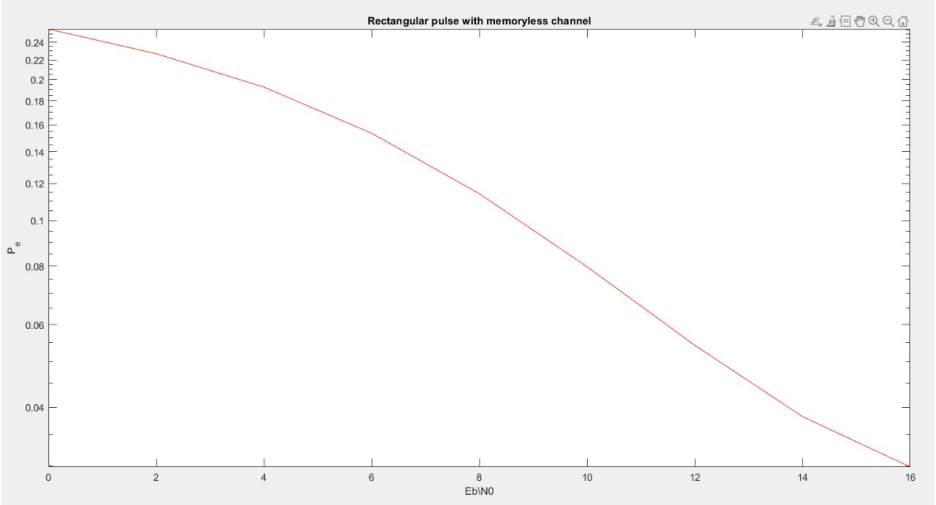
Probability of error Pe vs SNR :



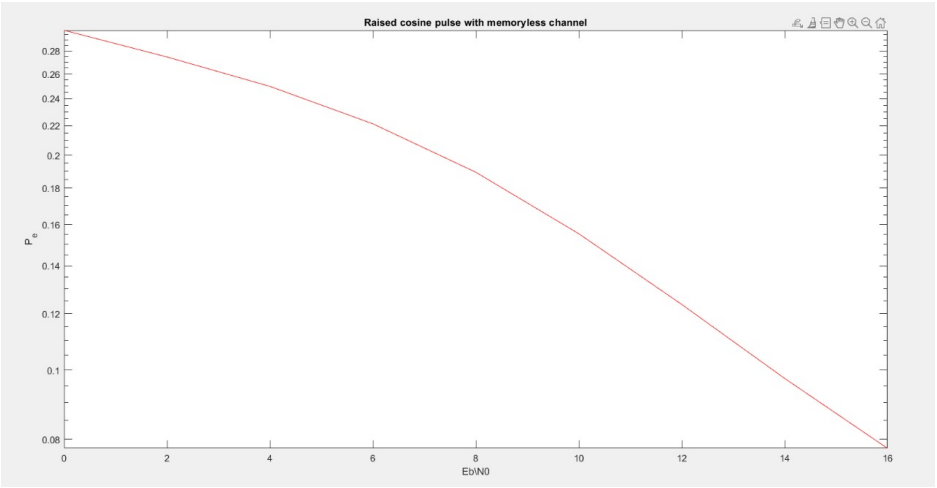
Pe vs SNR, Rectangular pulse with memory channel



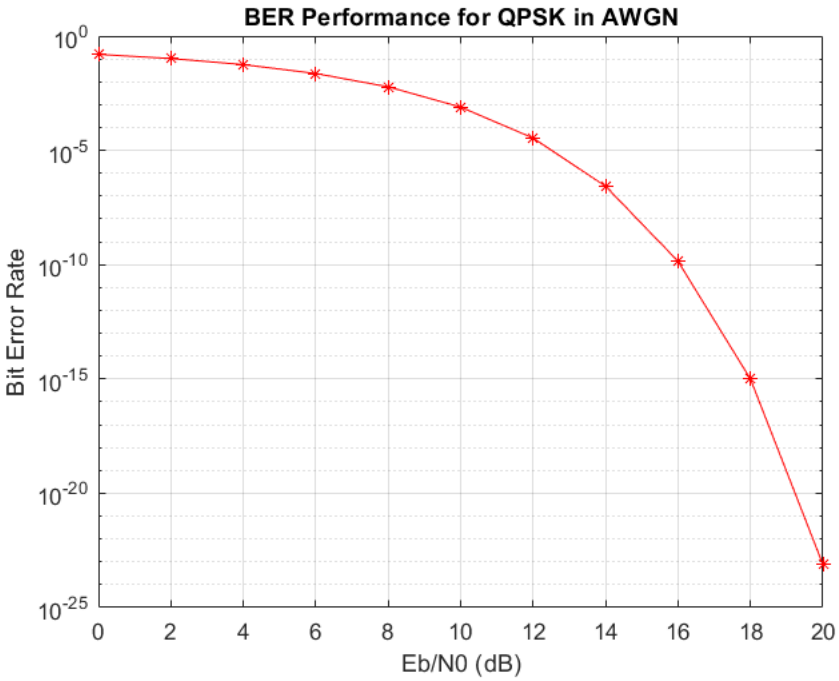
Pe vs SNR, rasised cosine pulse with memory channel



Pe vs SNR, Rectangular pulse with memoryless channel

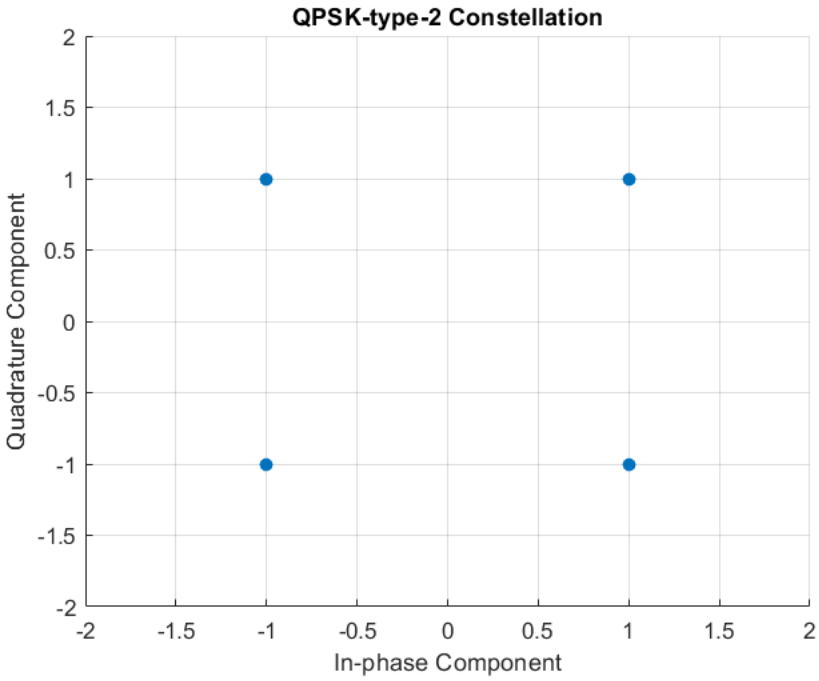


Pe vs SNR, rasised cosine pulse with memoryless channel

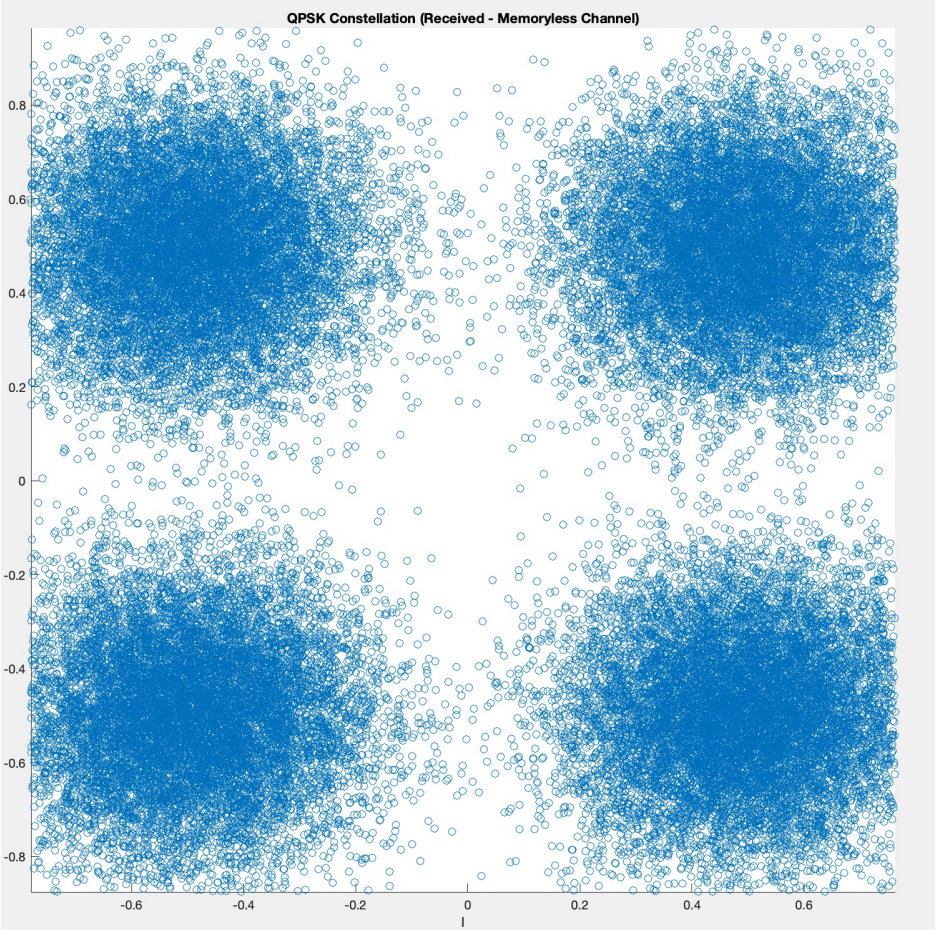


BER performance for QPSK in AWGN

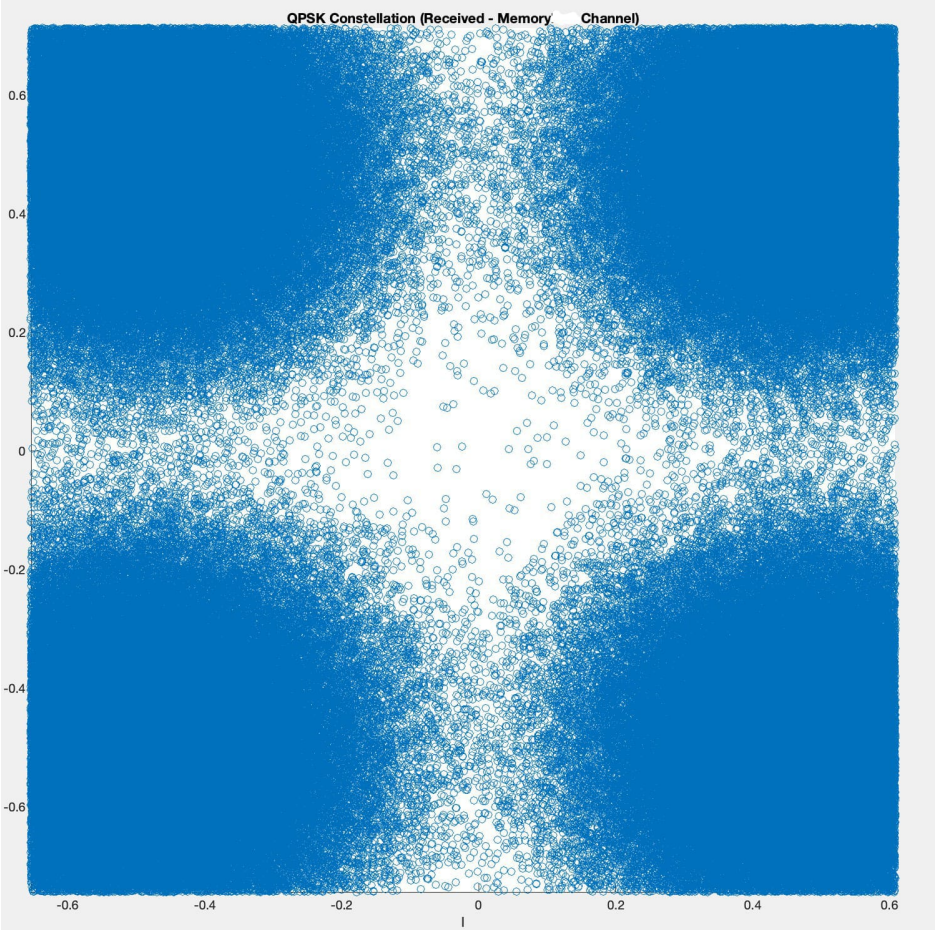
Input and Output constellation:



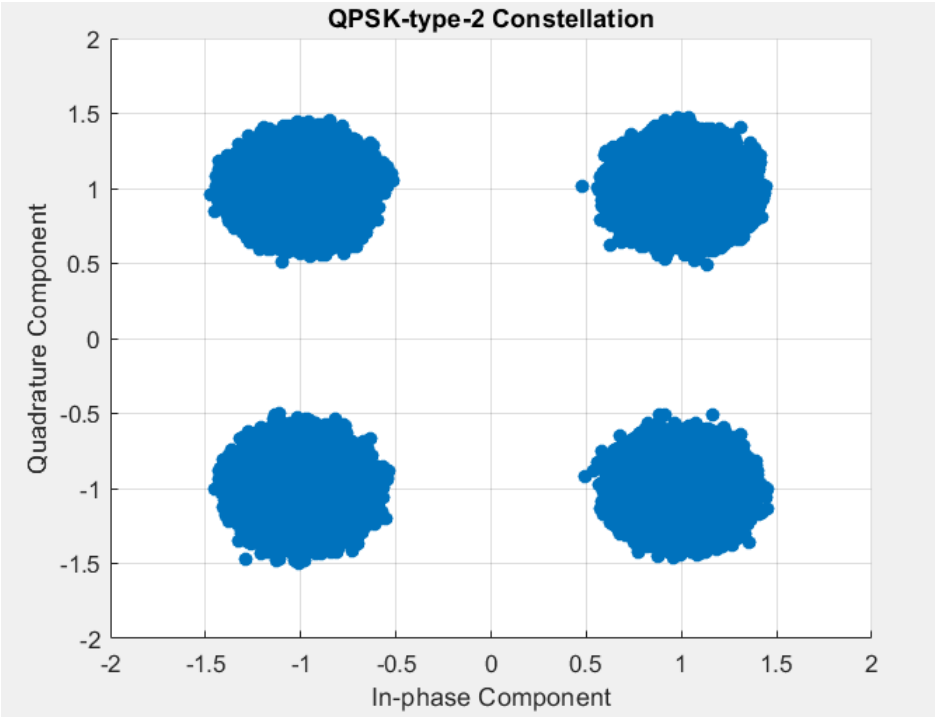
Input constellation



Recieved memory less channel constellation



Recieved with memory channel constellation



Recieved with memory channel constellation (aother view)