COMMUNICATION THEORY PROJECT REPORT

QPSK(QUADRATURE PHASE SHIFT KEYING)

TYPE 1 QPSK:-

TEAM MEMBERS:-

1) DIKSHANT GURUDUTT(2022102038)

2) MANAS SACHIN DESHMUKH(2022102040)

THE BLOCK DIAGRAM OF THE SYSTEM TO BE IMPLEMENTED:-

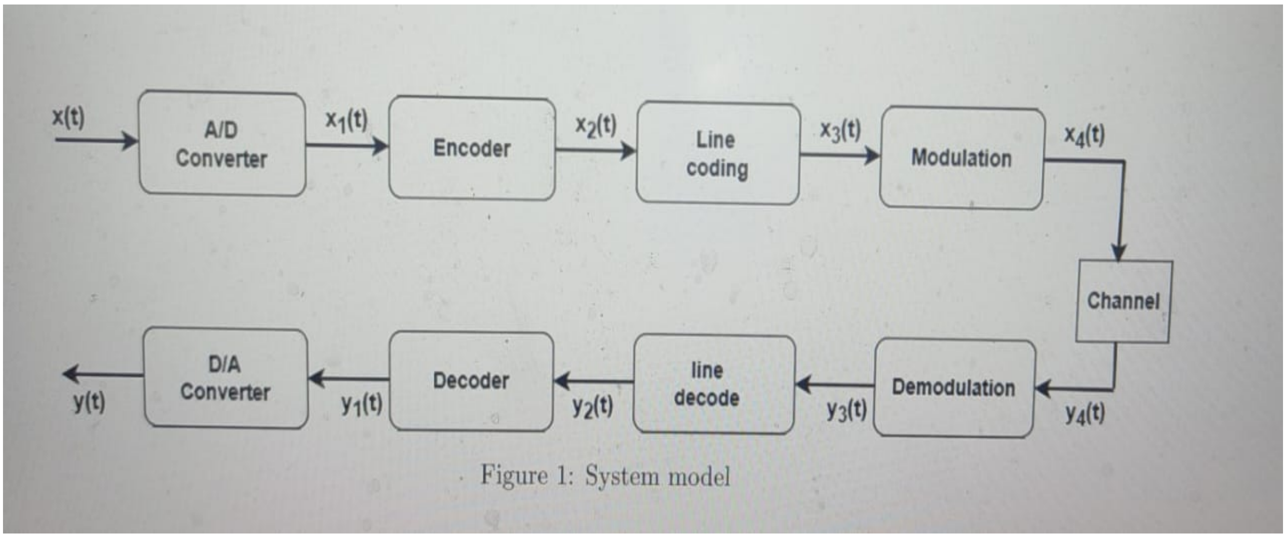


Figure 1: System model

1) A/D CONVERTER :-

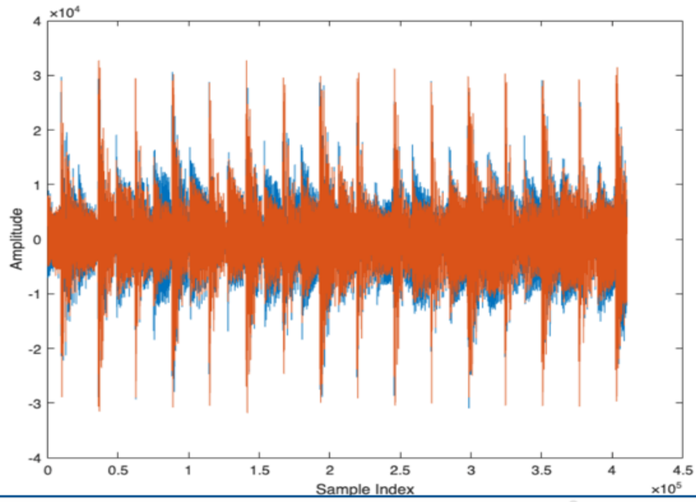
Provided audio signal is converted to binary in this block. A/D convertor converts continuous-time analog signals into discrete-time digital signals. This conversion is necessary because digital systems can only process digital data. Also digital conversion can be transmitted more efficiently by compressing the data, allowing efficient multiplexing to transfer over a single channel reducing the bandwidth.

A/D Converter

```
[audio, fs] = audioread('project.wav');
audio_normalized = int16(audio * 32767);
audio_binary = dec2bin(typecast(audio_normalized(:), 'uint16'), 16);
% binary_vector = audio_binary(:)';
total_16bit_values = length(audio_binary) / 16;
disp(['total 16-bit values: ', num2str(total_16bit_values)]);
```

```
total 16-bit values: 51328
```

```
figure;
plot(audio_normalized);
xlabel('Sample Index');
ylabel('Amplitude');
```



This conversion(A/D CONVERSION) is essential for processing analog signals in digital systems, such as computers, digital signal processors, and communication devices. A/D converters sample the analog signal at regular intervals and quantize the amplitude of each sample into digital values. The accuracy and resolution of an A/D converter are crucial factors that determine the fidelity of the digital representation of the original analog signal.

Code and It’s explanation:

```
clc;
clearvars;
close all;
[wavdata, Fs] = audioread('project.wav');
% quantise the interval -1 to 1 in 128 levels and take closest quantised
% level for wavdata(1) to wavdata(410624) and scale by 128 and store it in
% bits_vector as a 8 bit number bits_vector=zeros(410624*8,1)
 wavnew=zeros(410624,1);
 wavedata_cpy=wavdata;
 for kk=1:410624
     wavnew(kk)=wavdata(kk);
end
temp=wavnew+1;
symbols_tx=round(temp*64);
snew=dec2bin(symbols_tx) - '0';
tri=snew';
bits_fuf=zeros(410624*8,1);
for gv=1:length(tri)
    bits_fuf(gv)=tri(gv);
end
```

We store the audio samples in wavdata and sampling freq in Fs. Then convert audio to mono by selecting only 1 column from wavdata variable. The audio is then shifted to range [0,2] by adding 1. And is quantized into 128 levels by multiplying by 64 and rounding to the nearest integer. quantized values are converted to their binary representation using the 'dec2bin' function. Binary representation is transposed and stored in the 'bits_fuf' vector, which contains the quantized audio data in binary form.

<u>2)ENCODER FOR QPSK TYPE 1:-</u>

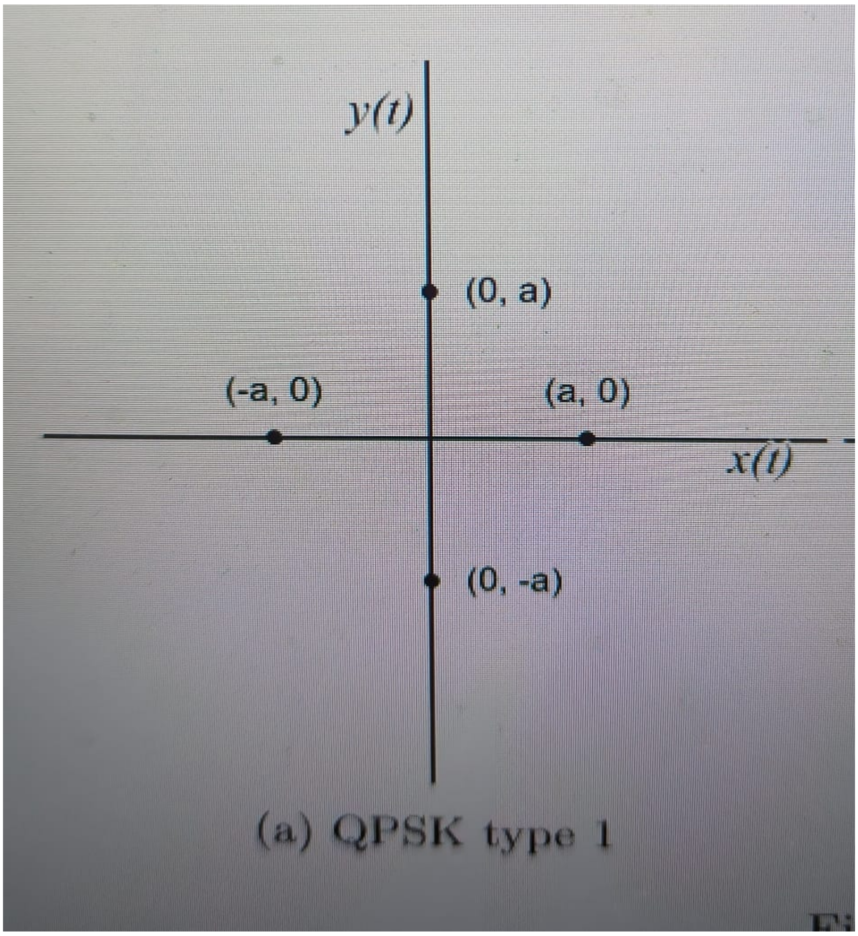
After analog to digital conversion we have a string of bits in the digital form now in QPSK TYPE 1 we will map two consecutive bits to one symbol. So we will have four symbols representing the 4 combinations of set of 2 bits. We have used a function called qpskmap to map the bits to the symbols.

QPSK MAP FUNCTION:-(NOTE FROM CONSTELLATION DIAGRAM a=1)

```
function ans_bit=qpskmap(input_vec)
         if(input_vec(1)==0 && input_vec(2)==0)
             ans_bit=1;
         end
           if(input_vec(1)==0 && input_vec(2)==1)
             ans_bit=-1;
           end
             if(input_vec(1)==1 && input_vec(2)==0)
             ans_bit=1i;
             end
               if(input_vec(1)==1 && input_vec(2)==1)
             ans_bit=-1i;
               end
         end
```

Here we are mapping bits(00) to symbol 1(along positive x axis) , bits(01) to symbol(-1 along negative x axis), bits(10) to symbol 1j(to indicate along positive y axis) and -1j(to indicate along negative y axis)

REQUIRED/DESIRED CONSTELLATION DIAGRAM:-



(a) QPSK type 1

This mapping ensures proper gray coding satisfying all its criteria. In communication theory, Gray coding refers to a specific binary code sequence where consecutive numbers differ by only one bit. This coding technique is particularly valuable in digital communication systems, especially for applications where errors can occur due to noise or interference. Gray coding is designed to minimize errors during signal transmission. This important technique ensures minimum ISI. QPSK is widely used in Wi-Fi, LTE, satellite communications, and digital radio broadcasting.

3) LINE CODING FOR QPSK TYPE 1:-

It’s the process where the sequence of encoded symbols is transformed into a time-domain waveform suitable for transmission over the communication channel.
The output of this block is according to $x_3(t) = \sum_k a_k \, p(t - kT_b)$ where p(t) is a pulse of duration T_b (symbol duration).
The equation $x_3(t)$ represents the summation of time shifted and amplitude scaled versions of pulse shape p(t), where each pulse is weighted by the corresponding QPSK symbol a_k and a scaling factor X_k. Nyquist achieves zero ISi by choosing a pulse shape that has a nonzero amplitude at its center (say t = 0) and zero amplitudes at t = ±nTb (n = 1, 2, 3, . . .), where Tb is the separation between successive transmitted pulses.
USED RAISED COSINE PULSE FOR LINE CODING TO ENSURING MINIMUM ISI OR NOISE/INTERFERENCE.

$$P(f) = \begin{cases} 1, & |f| < \frac{R_b}{2} - f_x \\ \frac{1}{2}\left[1 - \sin \pi \left(\frac{f - R_b/2}{2f_x}\right)\right] & \left|f - \frac{R_b}{2}\right| < f_x \\ 0, & |f| > \frac{R_b}{2} + f_x \end{cases}$$

This characteristic of above equation is known in the literature as the raised-cosine characteristic. We can make several important observations about the raised-cosine pulse. First, the bandwidth of this pulse is Rb Hz and has a value Rb(1/Tb bit time) at t = 0 and is zero not only at all the remaining signaling instants but also at points midway between all the signaling instants. Second, it decays rapidly, as 1 / t^3 . As a result, the raised-cosine pulse is relatively insensitive to deviations of Rb, sampling rate, timing jitter, and so on. Furthermore, the pulse-generating filter with transfer function Plf) is closely realizable as defined above. So raised cosine pulse helps us in getting minimum ISI and is the most optimum method.

Rectangular pulse is also used which does not satisfy Nyquist first criteria but is clearly identifiable

COMPARISON:-(BETWEEN TWO PULSES FOR LINE CODING)

- **Pulse Shape:**
  - **Rectangular Pulse:** Rectangular pulse line coding uses rectangular pulses as the basic signal elements. These pulses have sharp transitions between 0 and 1.
  - **Raised Cosine:** Raised cosine line coding employs pulses that are shaped like a raised cosine function. These pulses have smoother transitions between 0 and 1, which helps in reducing inter-symbol interference.
- **Bandwidth:**
  - **Rectangular Pulse:** Rectangular pulses have a wider bandwidth compared to raised cosine pulses. This is due to the abrupt transitions, which introduce higher frequency components in the signal.
  - **Raised Cosine:** Raised cosine pulses have a narrower bandwidth compared to rectangular pulses. The smooth transition helps in reducing the high-frequency components, resulting in better spectral efficiency.
- **Inter-Symbol Interference (ISI):**

- **Rectangular Pulse:** Rectangular pulses are more prone to ISI because of their sharp transitions. ISI occurs when the pulses interfere with each other, leading to errors in the received signal.
- **Raised Cosine:** Raised cosine pulses are designed to minimize ISI by having smoother transitions. This makes them more robust to channel impairments and reduces the likelihood of errors.

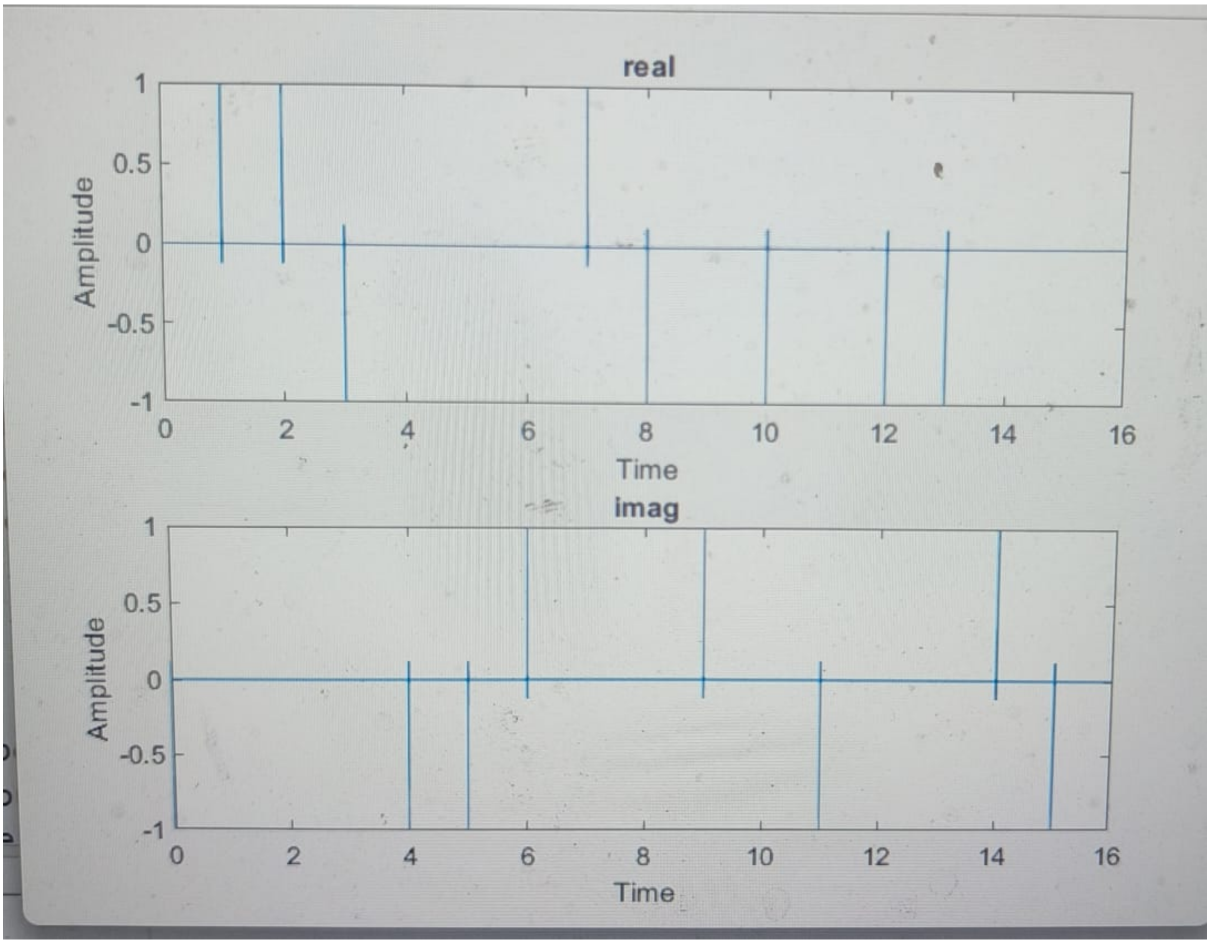
- **Complexity:**

  - **Rectangular Pulse:** Rectangular pulse line coding is simpler to implement compared to raised cosine coding. It involves straightforward shaping of pulses without additional filtering.
  - **Raised Cosine:** Raised cosine line coding requires more complex signal processing due to the shaping filter involved. This adds some complexity to both the transmitter and receiver.

- **Efficiency:**

  - **Rectangular Pulse:** Rectangular pulse line coding is less efficient in terms of bandwidth utilization compared to raised cosine coding, especially in bandwidth-limited scenarios.
  - **Raised Cosine:** Raised cosine line coding is more efficient in utilizing the available bandwidth, making it suitable for communication systems where bandwidth conservation is critical

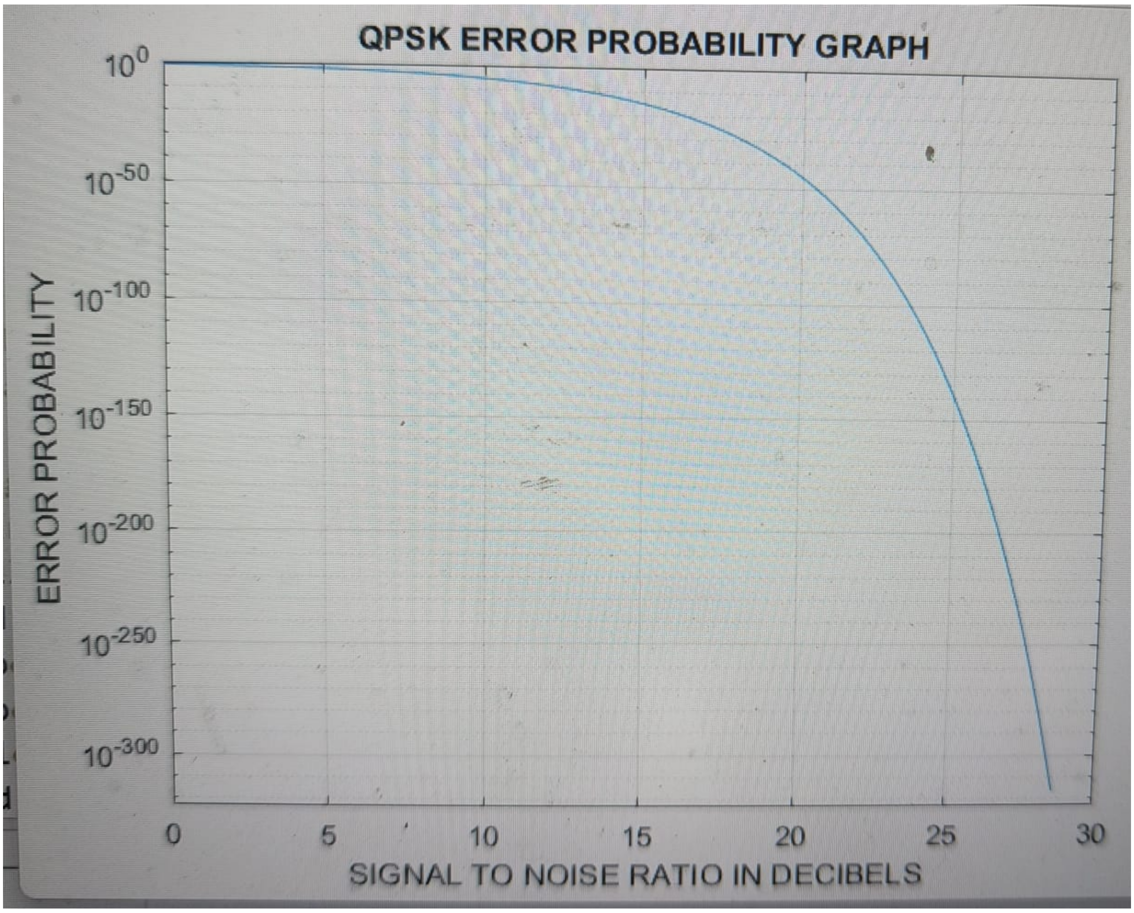
SAMPLE OUTPUT FOR LINE CODING:-



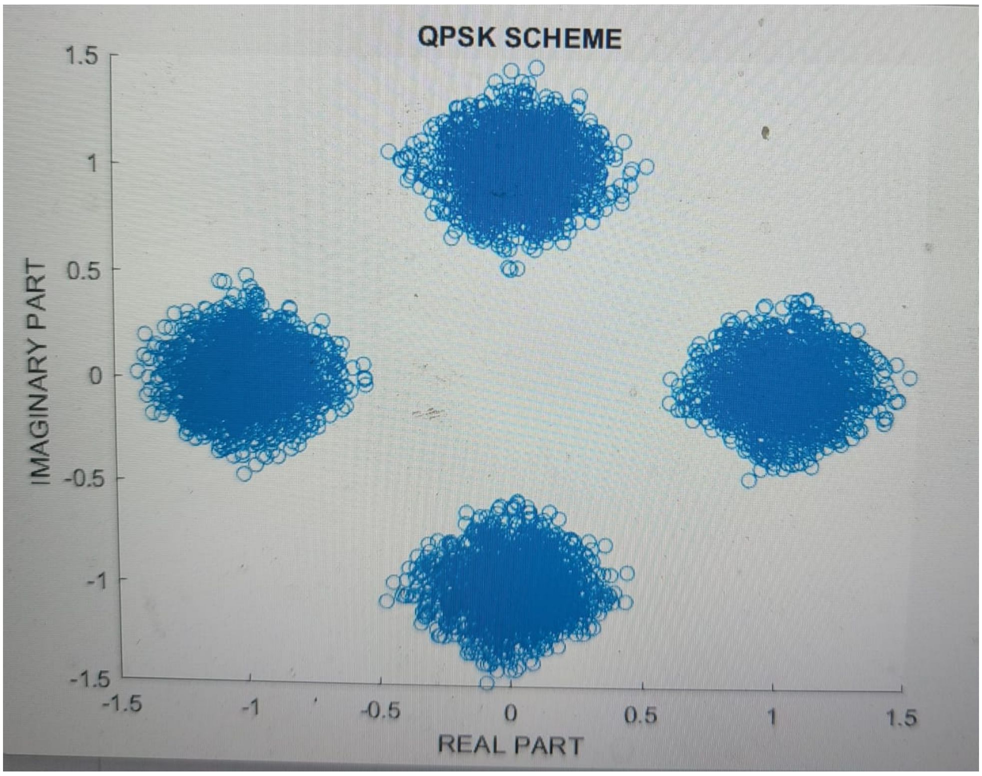
<u>OUTPUT FOR REAL AND IMAGINARY PLOT OF LINE CODING OUTPUT FOR 16 SYMBOLS</u>

<u>(RAISED COSINE PULSE)</u>

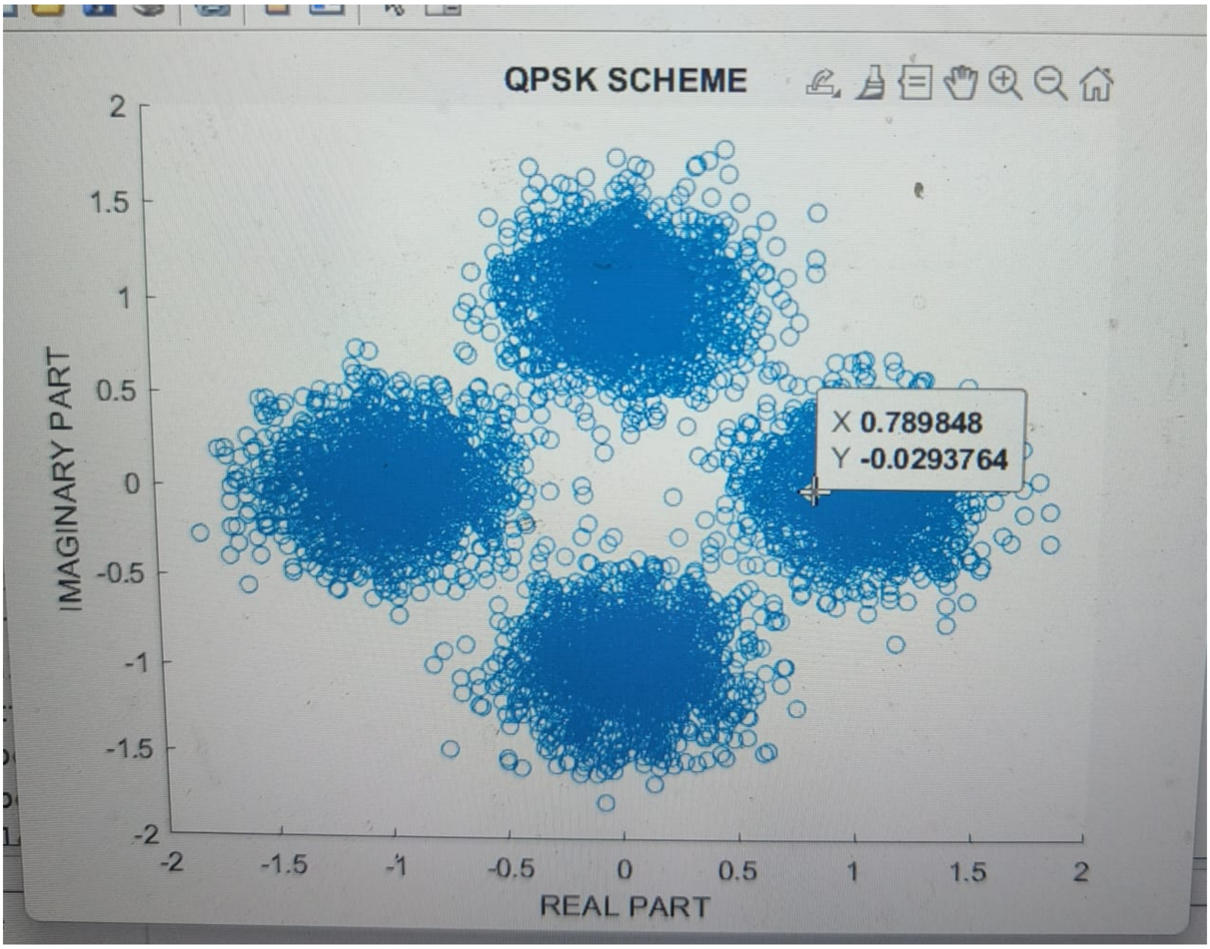
<u>ERROR PLOT FOR QPSK SCHEME:-</u>



SIGNAL CONSTELLATION AT INPUT FOR LESS NOISE(VAR=0.1)



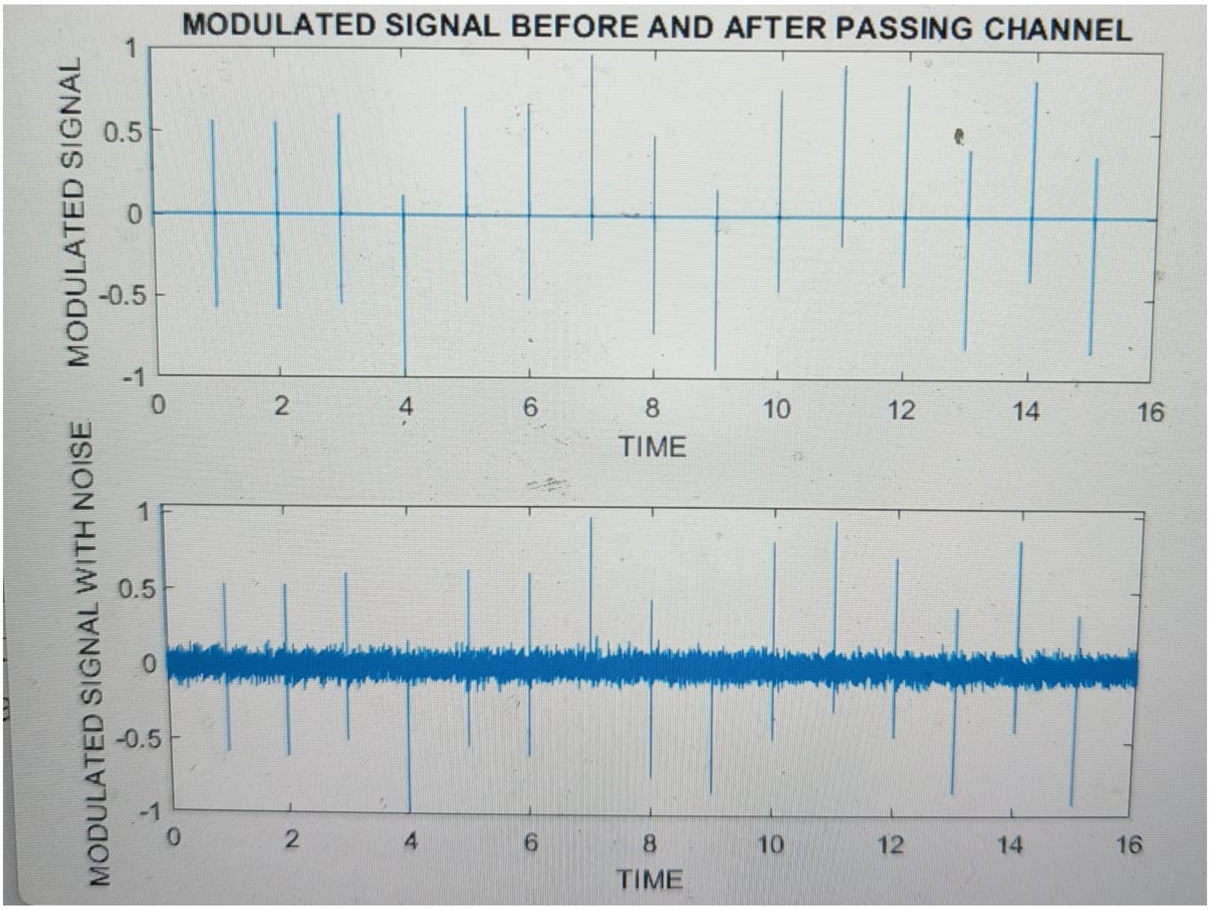
SIGNAL CONSTELLATION DIAGRAM AT INPUT SIDE FOR HIGHER NOIE VARIANCE(0.6):-



CLEARLY NOISE AND ISI INCREASES AND THE PROBABILITY OF ERROR WILL INCREASE DUE TO INTERFERENCE OF SYMBOLS AMONGST EACH OTHER.

4) MODULATION AND PASSING THROUGH CHANNEL:-

For modulation we multiply the real part of the transmit filter output by cosine wave so that symbols 1 and -1(along real x axis) are transmitted separately compared to symbols j and -j(along imaginary y axis). Then the imaginary part of the transmit filter output(line coding output) is multiplied by sine wave for symbols along y axis(j and -j) then these both signals(note that both the signals are real since there is no j term) so after multiplying real and imaginary part by cos(2*pi*fc*t) and sin(2*pi*fc*t) we add them up and transmit this signal to the noisy channel. Here t (time vector) is defined as number of symbols* Ts(symbol time) equally spaced between length of transmitter filter output. This modulated output is passed through the noisy channel and AWGN noise of some variance is added to it in type 1 channel in the type 2 channel it is convoluted with h(t) as given below(Which ensures CHANNEL WITH MEMORY) DUE TO DIRAC DELTA TERM WITH PREVIOUS TIME b*Tb

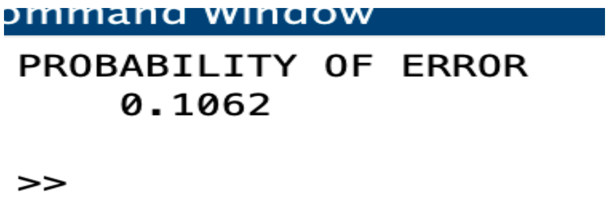


FOR 16 SYMBOLS(ABOVE PLOT)

In a communication system, the noise characteristics greatly influence the system's performance, particularly in terms of error rates. Let's break down the comparison between Additive White Gaussian Noise (AWGN) channels and memory channels, and why error rates tend to be higher in memory channels.

1. **AWGN (Memoryless) Channel**:
    - In an AWGN channel, noise is characterized by being independent and identically distributed (i.i.d.) at each time instant and at each frequency.
    - The noise in AWGN channels does not depend on past transmissions or received symbols, hence it's called memoryless.
    - Due to its simplicity and statistical properties, AWGN channels are often used as models for various communication systems.
    - Error rates in AWGN channels are typically easier to predict and analyze due to their memoryless nature.
2. **Memory Channel**:
    - In contrast to AWGN channels, memory channels exhibit dependencies between transmitted and received symbols over time.
    - These dependencies arise due to various factors such as intersymbol interference (ISI), multipath propagation, fading effects, etc.
    - Memory channels can be modeled as finite impulse response (FIR) or infinite impulse response (IIR) systems, where the output depends not only on the current input but also on past inputs and noise.
    - Due to the presence of memory, the channel introduces intersymbol interference (ISI), which can significantly degrade the performance of the communication system.

- The complexity of characterizing and compensating for these memory effects makes the analysis and design of communication systems over memory channels more challenging compared to memoryless AWGN channels.
- Error rates tend to be higher in memory channels because the receiver needs to not only detect the current symbol correctly but also mitigate the effects of past symbols that might interfere with the current one. This makes the detection and decoding process more complex and error-prone



DUE TO MEMORY NATURE THE DELAY TERM ISI INCREASES AND ERROR INCREASES.

5)**<u>DEMODULATION:-</u>**

We have modulated the signal by multiplying the real part by cos(2*pi*fc*t) and the imaginary part with sin(2*pi*fc*t) now for demodulation we will first multiply the modulated noisy signal with cos(2*pi*fc*t) then pass the resultant signal through a LOW PASS filter to get the real part. We will also multiply the modulated noisy signal with sin(2*pi*fc*t) to get another resultant signal and we will pass it through low pass filter to get the imaginary part.

(NOTE: WE ARE USING a*cos^2(x)=a/2+a*cos(2*x)/2

And b*sin^2(x)=b/2-b*cos(2*x)/2

) The higher frequency term vanishes when passed through low pass filter

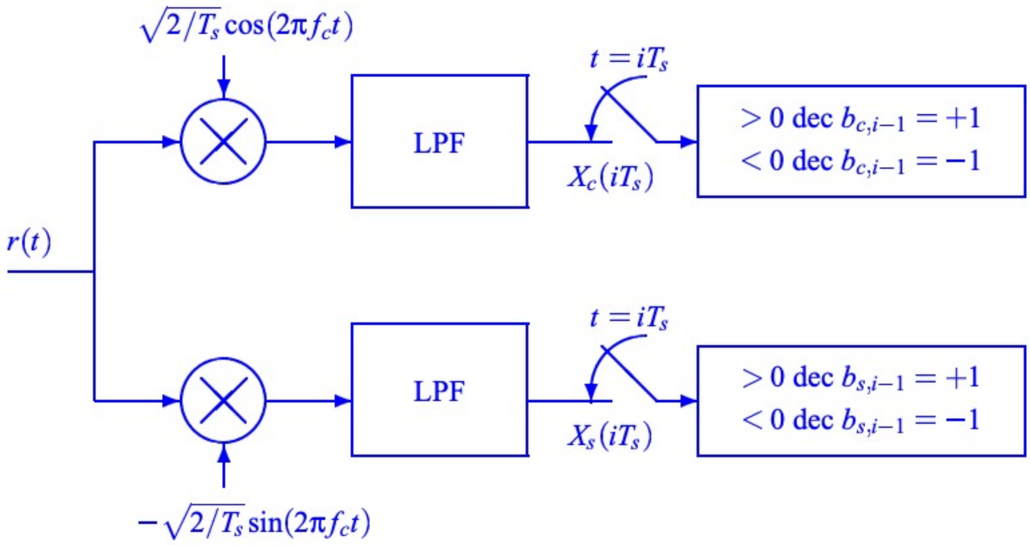
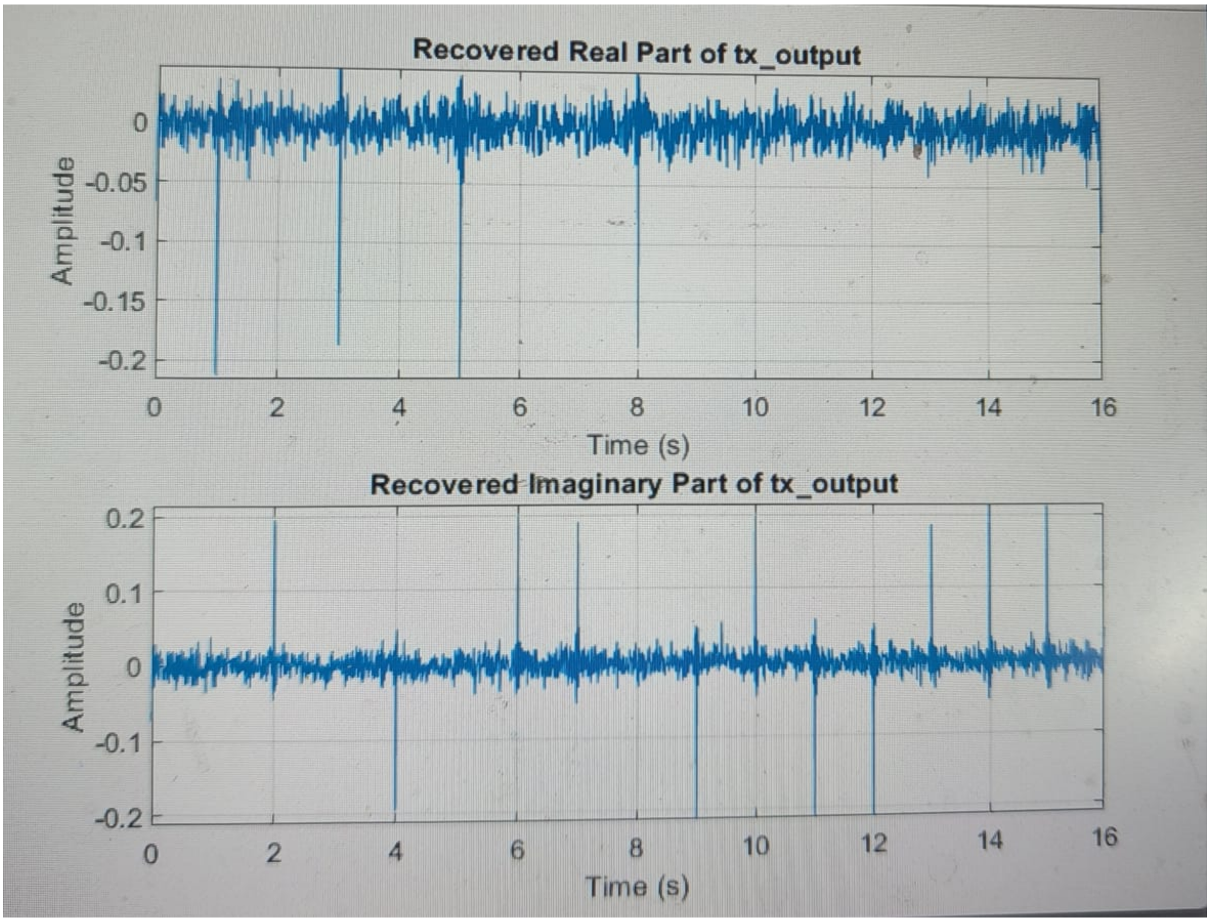


Figure 47: QPSK Demodulator

<u>DEMODULATED OUTPUT FROM MATLAB:-</u>



<u>6)LINE DECODING:-</u>

Line decoding is being done by using a matched filter to get back the best estimate of the original symbols transmitted via transmit filter. Since we are implementing the matched filter our receiver filter RX=conj(flipud(TX)) to ensure best estimate of transmitted symbols. We will convolute with RX

GENERAL POINTS ON LINE DECODING FACTORS TO BE CONSIDERED IN DESIGN:-

- **Channel Characteristics**:
  - Understand the characteristics of the communication channel, including noise level, bandwidth, distortion, and any other impairments.

- For example, in a noisy channel, error-correction techniques like forward error correction (FEC) may be necessary to improve reliability.

- **Modulation Scheme**:
  - Different modulation schemes (e.g., Amplitude Shift Keying, Phase Shift Keying, Quadrature Amplitude Modulation) have different decoding complexities and error rate performances.
  - Choose a modulation scheme suitable for the channel conditions and bandwidth constraints.

- **Error Correction Coding**:
  - Employ error correction coding techniques if the channel is prone to errors. This includes techniques like Reed-Solomon codes, convolutional codes, or turbo codes.
  - Determine the appropriate code rate and coding scheme based on the desired error performance and decoding complexity trade-offs.

- **Decoding Algorithm**:
  - Select a decoding algorithm suitable for the modulation scheme and error correction coding used.
  - Common decoding algorithms include maximum likelihood (ML) decoding, maximum a posteriori (MAP) decoding, Viterbi decoding, and belief propagation.

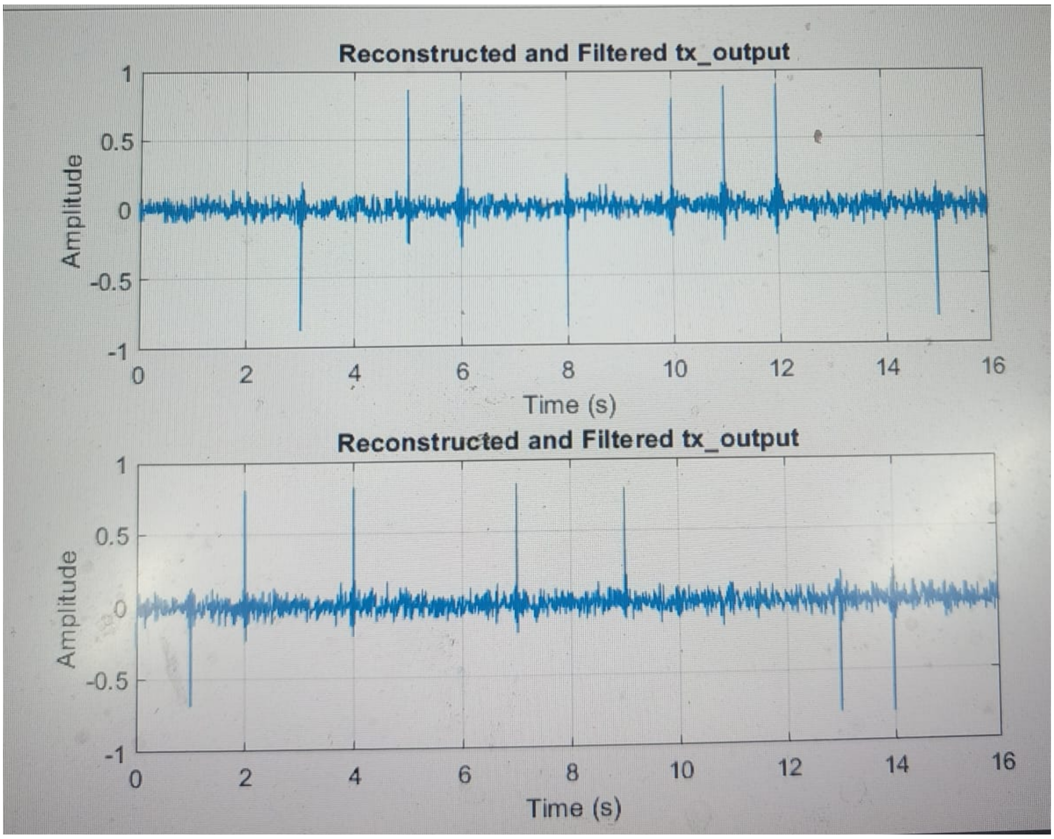
- **Computational Complexity**:
  - Consider the computational complexity of the decoding algorithm, especially in real-time systems or systems with limited processing capabilities.

- Balance between decoding performance and computational complexity to meet system requirements.

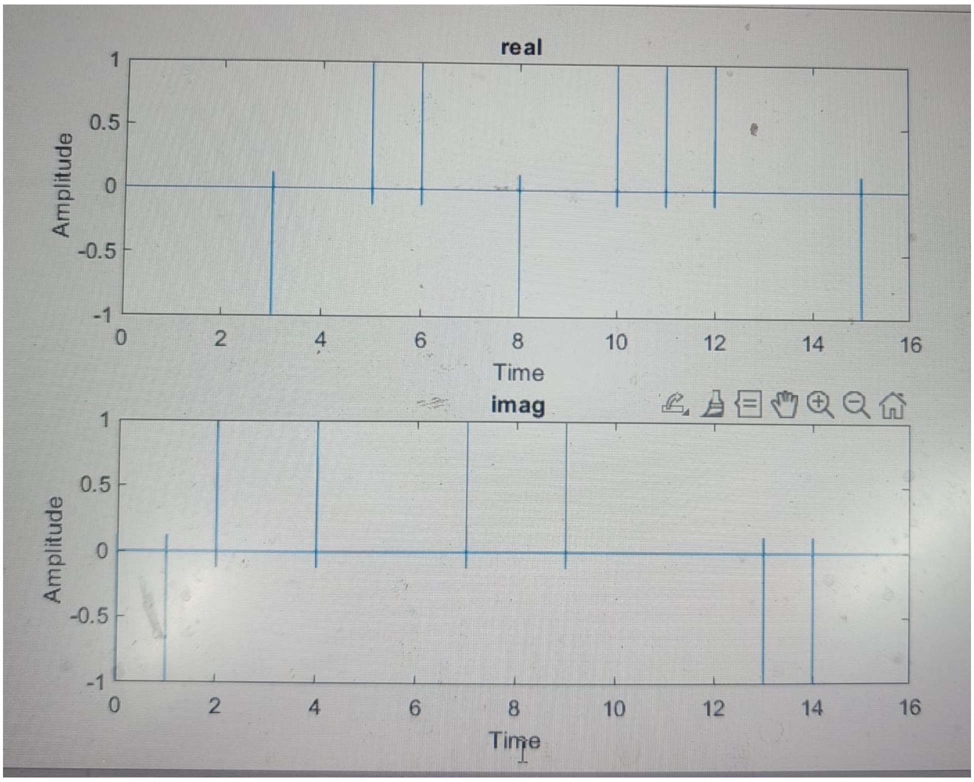
- **Latency**:
  - Minimize decoding latency, especially in real-time communication systems where delay is critical.
  - Choose decoding algorithms and techniques that offer low-latency processing while maintaining acceptable error rates.

- **Adaptability**:
  - Design decoding algorithms that can adapt to varying channel conditions, such as changing noise levels or signal-to-noise ratios.
  - Implement adaptive modulation and coding techniques to dynamically adjust the modulation scheme and coding rate based on channel conditions

OUTPUT OF LINE DECODING PART FROM MATLAB FOR QPSK TYPE 1:-



<u>THIS IS FOR THE 16 SYMBOLS AS FOLLOWS (SMALLER NUMBER FOR CLARITY AND BETTER VISUALIZATION(fc=100):-</u>



7) DECODER:-

Now the modulated signal is passed through a noisy channel so and then after that it is demodulated and then convoluted with received filter RX so even real symbols have some imaginary part(due to noise/demodulation but very less) and vice versa so for decoding to symbols we will take maximum of absolute value of real and imaginary part if it is real and the real part is positive then symbol is 1 ,if it is real and real part is negative then the symbol is -1.

If maximum is imaginary and imaginary part is negative then symbol is -1j and the imaginary part is positive then the symbol is 1j.

CODE TO IMPLEMENT ABOVE DECODING

```
for yy=1:length(tx_output_filtered2)
if abs(real(tx_output_filtered2(yy)))>abs(imag(tx_output_filtered2(yy))) &
real(tx_output_filtered2(yy))>0
    dec_sym(rb)=1;
    rb=rb+1;
end
if abs(real(tx_output_filtered2(yy)))>abs(imag(tx_output_filtered2(yy))) &
real(tx_output_filtered2(yy))<0
    dec_sym(rb)=-1;
    rb=rb+1;
end
if abs(real(tx_output_filtered2(yy)))<abs(imag(tx_output_filtered2(yy))) &
imag(tx_output_filtered2(yy))>0
    dec_sym(rb)=1j;
    rb=rb+1;
end
if abs(real(tx_output_filtered2(yy)))<abs(imag(tx_output_filtered2(yy))) &
imag(tx_output_filtered2(yy))<0
dec_sym(rb)=-1j;
    rb=rb+1;
end
end
```

D/A CONVERTER

```
binary_matrix = reshape(binary_vector, [], 16); % Reshape back to mat
audio_integers = bin2dec(binary_matrix); % Convert binary to decimal
audio_reconstructed = typecast(uint16(audio_integers), 'int16'); % Ty
audio_reconstructed_normalized = double(audio_reconstructed) / 32767;
```

For this we reshapes the binary_vector into a matrix with 16 columns. The [] in the first argument of reshape tells MATLAB to automatically determine the number of rows based on the length of binary_vector and the specified number of columns (16).
Then convert the binary representation stored in binary_matrix to decimal values using the bin2dec function. The bin2dec function interprets each row of the binary_matrix as a binary number and converts it to its corresponding decimal value. The resulting audio_integers is a matrix of the same size as binary_matrix, but with decimal values instead of binary. The audio_integers are then reinterpreted as signed 16 bit integers. Finally normalizing the audio samples to the range [-1, 1]. Since the maximum value is 32767 (2^15 - 1), and the minimum value is -32768. By dividing the audio samples by 32767, they are scaled to the range [-1, 1], which is a common representation for audio signals.

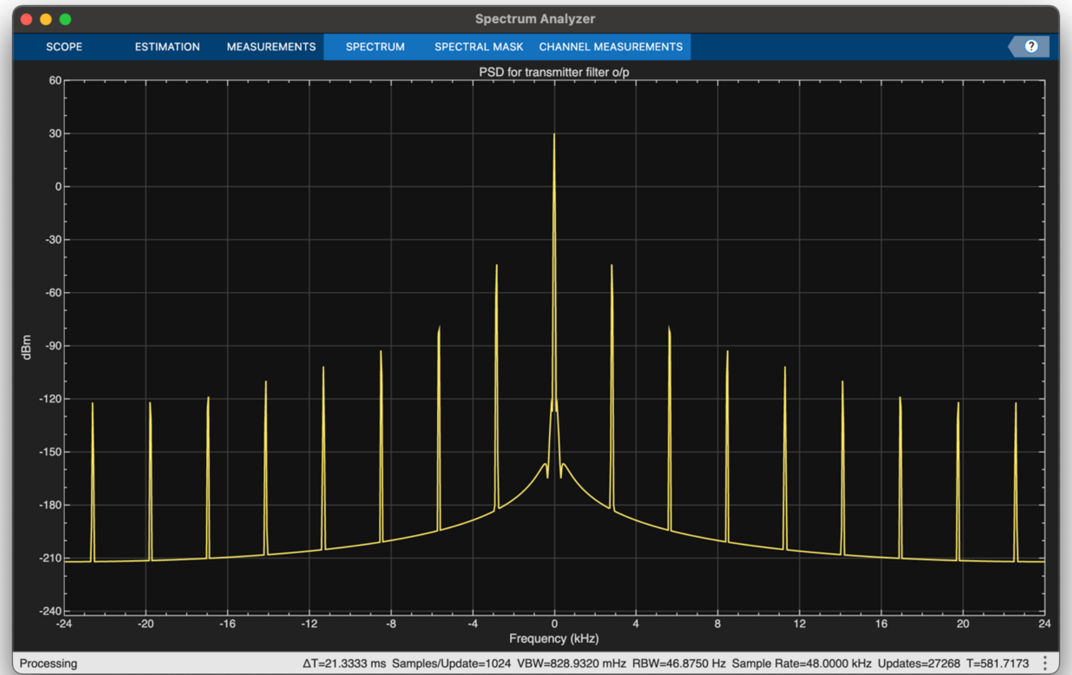
PSD PLOTS

The Power Spectral Density (PSD) is a measure of the distribution of signal power over different frequencies. It describes how the signal power is distributed across the frequency spectrum. PSD of QPSK is important in determining bandwidth requirements, adjacent channel interference, and spectral efficiency.

For plotting PSD we have used spectrumAnalyzer function from dsp library.

PSD after transmitter filter

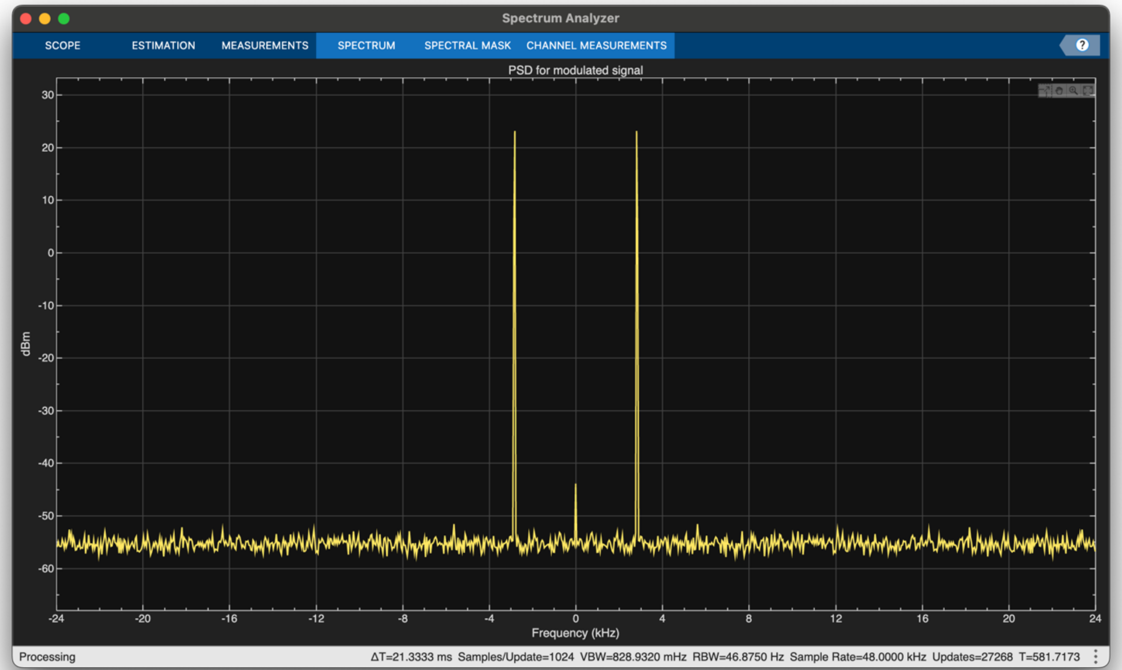
The shape of the PSD at this stage will depend on the characteristics of the pulse-shaping filter. A raised-cosine filter, will result in a PSD with a main lobe and side lobes, with the side lobes decaying at a rate determined by the roll-off factor of the filter. The main lobe have a width proportional to the symbol rate of the QPSK signal.



PSD plot for transmitter filter

PSD after modulation

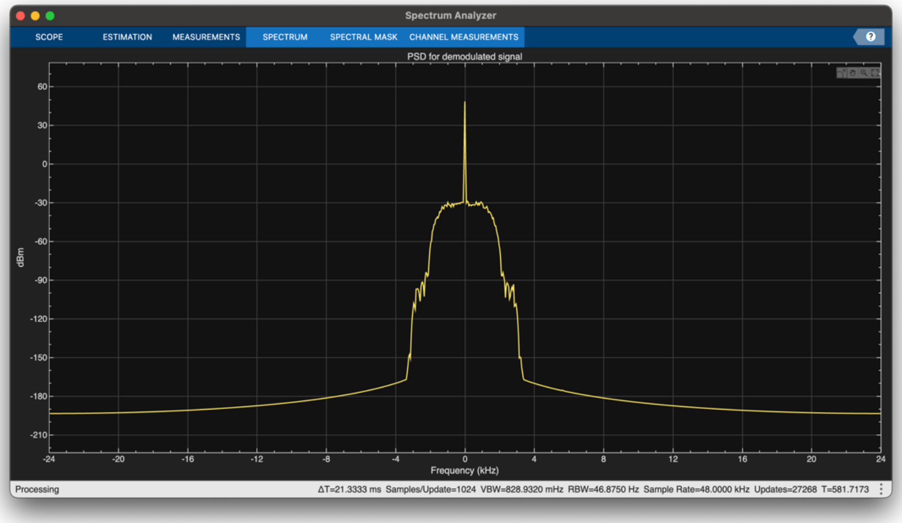
The shape of the PSD will remain similar to the baseband PSD, but it will now be centred at the carrier frequency. The main lobe and side lobes will be present, but their positions will be shifted by the carrier frequency.



PSD plot for modulated signal

PSD after demodulation

At this stage, the PSD may exhibit some additional distortions or noise due to channel impairments, such as multipath fading, interference, or noise.



PSD plot for demodulated signal

QPSK

QPSK is a digital modulation technique used in various communication systems, such as wireless communications, satellite communications, and digital broadcasting. It is a form of Phase Shift Keying (PSK) modulation, where the phase of the carrier signal is modulated to represent the digital data.

In QPSK, the phase of the carrier signal is shifted to one of four possible values, each separated by 90 degrees (or π/2 radians). These four phases represent the four possible combinations of two binary digits (bits): 00, 01, 11, and 10. This means that each phase shift in the QPSK signal represents two bits of information.

The modulation process in QPSK can be understood as a combination of two independent binary Phase Shift Keying (BPSK) signals, one in the in-phase (I) component and the other in the quadrature (Q) component. These two components are modulated onto the carrier signal, creating a constellation of four possible signal points in the I-Q plane.

The four possible phase shifts in QPSK are 0, π/2, π, and 3π/2 radians, corresponding to the binary values 00, 01, 11, and 10, respectively. At the receiver, the demodulator measures the phase of the received signal and maps it back to the corresponding pair of bits.

QPSK is widely used in various communication standards and systems, such as Wi-Fi, cellular networks (CDMA, WCDMA, LTE), satellite communications, and digital video broadcasting (DVB). It strikes a balance between bandwidth efficiency, power efficiency, and implementation complexity.

Advantages:

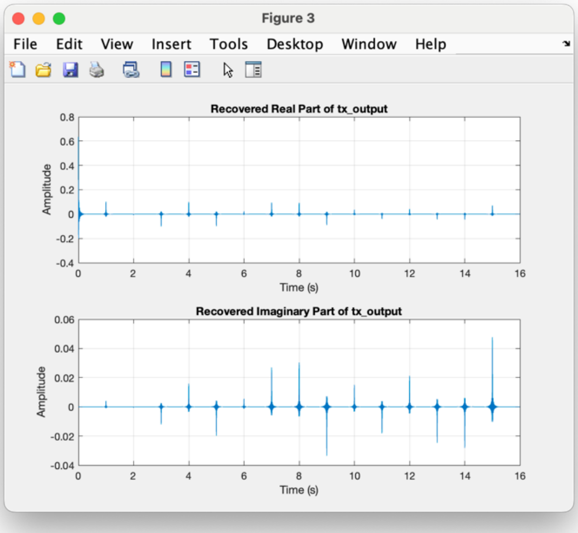
- *Bandwidth efficiency*: QPSK transmits two bits of information per symbol, which means it has a higher bandwidth efficiency compared to BPSK.
- *Power efficiency:* QPSK is more power-efficient than higher-order modulation schemes like 8-PSK or 16-QAM, as it requires less signal-to-noise ratio (SNR) to achieve the same bit error rate (BER) performance.

Limitations:

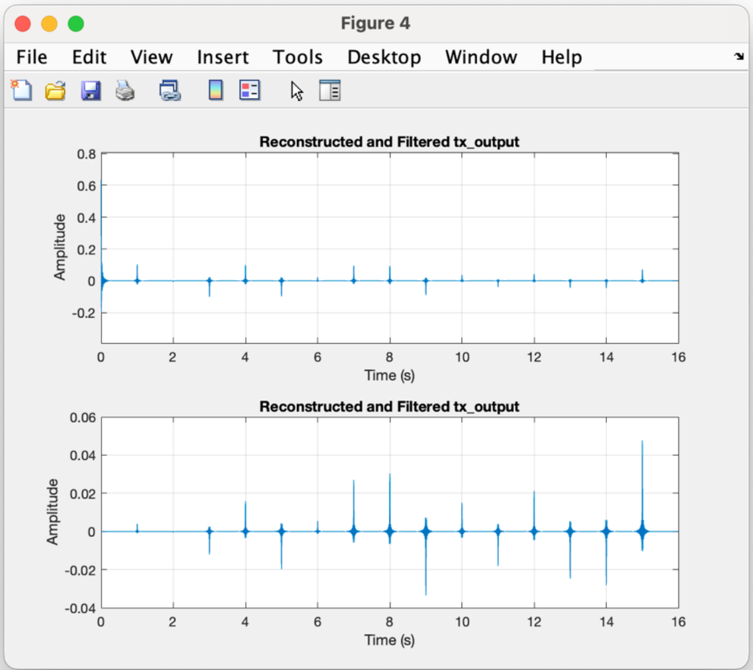
- Susceptibility to phase noise: QPSK is more sensitive to phase noise compared to BPSK, as phase errors can cause symbol rotation and lead to incorrect decisions at the receiver.

LINE CODING USING RECTANGULAR FUNCTION

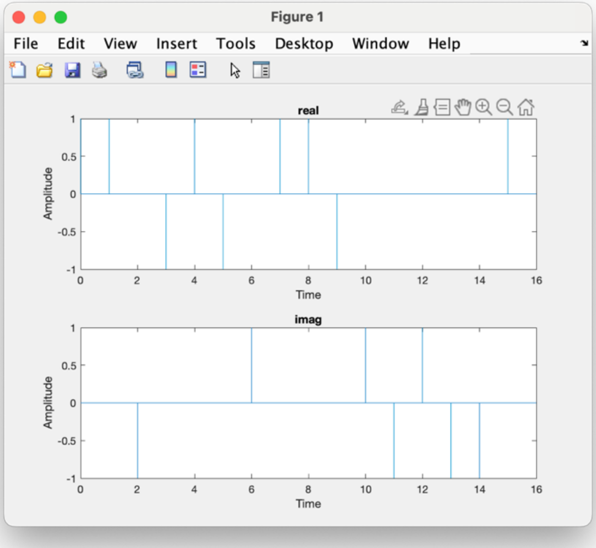
DEMODULATED OUTPUT FROM MATLAB



OUTPUT OF LINE DECODING PART



THIS IS FOR THE 16 SYMBOLS AS FOLLOWS (SMALLER NUMBER FOR CLARITY AND BETTER VISUALIZATION(fc=100)



PROBABILITY OF ERROR:

```
Command Window
PROBABILITY OF ERROR
    0.0830
```

Plot of Symbols before modulation: