Lab Manual
of
“ EC 302: Wireless and Mobile Communication ”
For B.Tech III
Year 2021- 2022

Submitted By

Name: **SANJIT ANAND** Division: **A**
Roll Number: **U19EC008** Year: **3rd**

Dr. (Mrs.) Shweta N. Shah
Assistant Professor, ECED.

S V National Institute of Technology, Surat
Department of Electronics Engineering

Jan-May 2022

S V National Institute of Technology, Surat

Department of Electronics Engineering

CERTIFICATE

In context of practical work assigned, this is to certify that Mr./Ms. ____________SANJIT ANAND__________________studying in B.Tech III year, DIV: ___A___ and bearing class Roll No. U19EC008_ has satisfactory completed the practical work offered in subject named " Wireless and Mobile Communication" at Electronics Engineering Department, S V National Institute of Technology, Surat.

Date: **18 April 2022** Signature of Concerned Teacher

INDEX

SANJIT ANAND U19EC008

17-01-2022 PRACTICAL ASSIGNMENT - 1

WIRELESS AND MOBILE COMMUNICATION

AIM:

To Simulate M-PSK and M-QAM Modulation Techniques with the help of MATLAB software where M = 4, 8, 16, 32, 64. Also plot the constellation diagram for each M.

APPARATUS:

MATLAB Software

THEORY:

Phase Shift Keying

Phase Shift Keying (PSK) is the digital modulation technique in which the phase of the carrier signal is changed by varying the sine and cosine inputs at a particular time. PSK technique is widely used for wireless LANs, bio-metric, contactless operations, along with RFID and Bluetooth communications.
M represents a digit that corresponds to the number of conditions, levels, or combinations possible for a given number of binary variables. The number of bits necessary to produce a given number of conditions is expressed mathematically as N=log2M.
Some prominent features of M-ary PSK are –

- The envelope is constant with more phase possibilities.
- This method was used during the early days of space communication.
- Better performance than ASK and FSK.
- Minimal phase estimation error at the receiver.
- The bandwidth efficiency of M-ary PSK decreases and the power efficiency increases with the increase in M.

Representation of M-ary PSK

$$S_i(t) = \sqrt{\frac{2E}{T}}\cos(w_o t + \phi_i t) \qquad 0 \leq t \leq T \quad and \quad i = 1, 2 \ldots M$$

$$\phi_i\,(t) = \frac{2\pi i}{M} \quad where \quad i = 1, 2, 3 \ldots \ldots M$$

Quadrature Phase Shift Keying (QPSK)

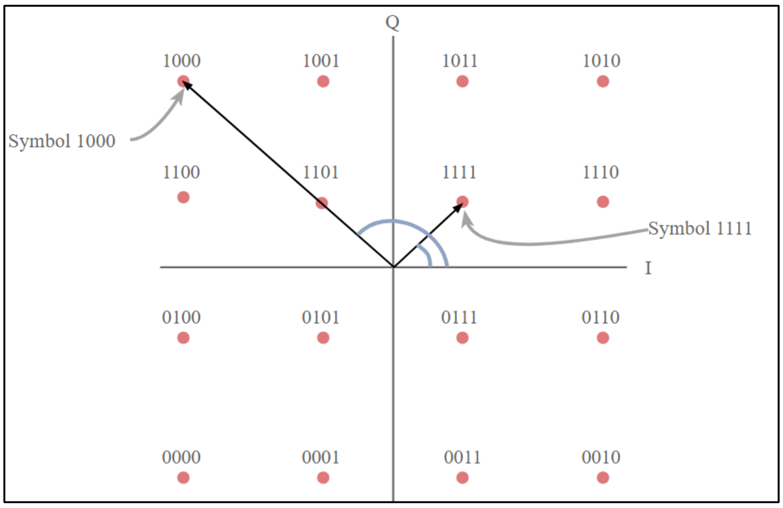
ASK, FSK and BPSK transmit one bit per symbol and hence carrier is assumed to have one of the two possible states to transmit 1 or 0.

Quadrature Phase Shift Keying (QPSK) is a form of Phase Shift Keying which transmits two bits per symbol.
Since it transmits two bits per symbol there are four possible combinations and thus there is four different phases.

- For **π**/4 QPSK, the four different phases are 45, 135, 225, 315.
- QPSK symbols are not represented by 0 or 1 but it is represented as 00, 01, 10 and 11.
- QPSK carry twice as much information as ordinary PSK using the same bandwidth.
- QPSK is used for satellite transmission of MPEG2 video, cable modems, videoconferencing, cellular phone systems, and other forms of digital communication over an RF carrier.

Quadrature Amplitude Modulation (QAM)

- Quadrature Amplitude Modulation, QAM utilizes both amplitude and phase components to provide a form of modulation that is able to provide high levels of spectrum usage efficiency.
- QAM, quadrature amplitude modulation has been used for some analogue transmissions including AM stereo transmissions, but it is for data applications where it has come into its own. It is able to provide a highly effective form of modulation for data and as such it is used in everything from cellular phones to Wi-Fi and almost every other form of high speed data communications system.
- Quadrature Amplitude Modulation, QAM is a signal in which two carriers shifted in phase by 90 degrees (i.e. sine and cosine) are modulated and combined. As a result of their 90° phase difference they are in quadrature and this gives rise to the name.
- Often one signal is called the In-phase or "I" signal, and the other is the quadrature or "Q" signal.
- The term M as in M-QAM indicates how many bits are transmitted per time interval or symbol for each unique amplitude/phase combination. The simplest form of QAM is 2-QAM, more commonly called QPSK or quadrature phase shift keying.



16 – QAM Constellation Diagram

Phase Modulation

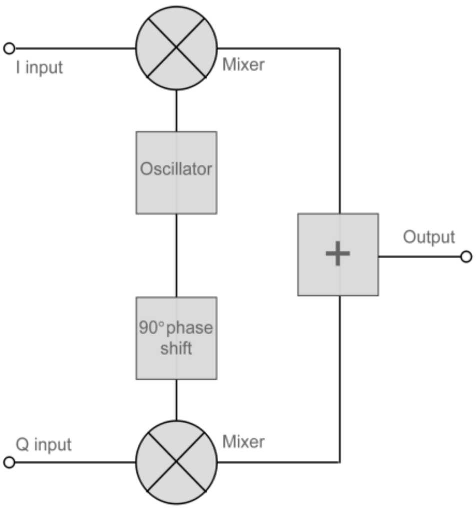
- Represents bits by changing the angle of a wave.
- As seen above, QPSK can have sixteen different phase changes as sixteen different angles.
- It is the angle of the constellation point.

QAM Modulator

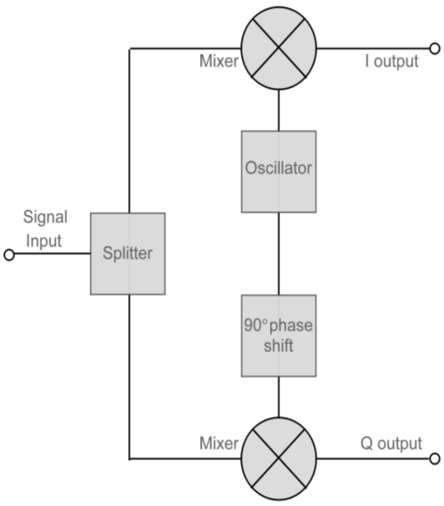
- The QAM modulator essentially follows the idea that can be seen from the basic QAM theory where there are two carrier signals with a phase shift of 90° between them.
- These are then amplitude modulated with the two data streams known as I or In-phase and the Q or quadrature data streams. These are generated in the baseband processing area.
- The two resultant signals are summed and then processed as required in the RF signal chain, typically converting them in frequency to the required final frequency and amplifying them as required.

QAM demodulator

- The QAM demodulator is very much the reverse of the QAM modulator. The signals enter the system, they are split and each side is applied to a mixer. One half has the in-phase local oscillator applied and the other half has the quadrature oscillator signal applied.
- The basic modulator assumes that the two quadrature signals remain exactly in quadrature.



QAM Modulator



QAM Demodulator

PROCEDURE:

Step 1: Read the image using the imread command and display the image.

Step 2: Convert the image matrix to Column matrix

Step 2: Convert the column matrix to binary column matrix

Step 3: Modulate the obtained binary column matrix using inbuilt functions for different order of modulation **(qammod /pskmod).**

Step 4: Demodulate received input using inbuilt functions **(qamdemod /pskdemod)**.

Step 5: Reshape the demodulated matrix into matrix having 8 columns, after that convert this matrix to decimal using uint8 and bi2dec.

Step 6: Reshape the output matrix to 256 x 256, display output image and plot the constellation diagram.

MATLAB CODE:

1. M-QAM

```
% Lab01 - Part 1 M-QAM
% U19EC008

clc;
clear all;
close all;

%% Input
M = input('Value of modulation order: ');
id = imread('cameraman.tif');
figure('name','Transmitted U19EC008');
imshow(id);

%% Input matrix to 1-D Row Vector
ida = id(:);

%% Decimal to Binary conversion
ib = de2bi(ida);
ib = ib(:);
x = mod(length(ib),log2(M));
pp = log2(M)-x;
ib5 = [ib.' zeros(1,pp)];

%% Reshaping column matrix
ib2 = reshape(ib5,[],log2(M));
ib3 = bi2de(ib2);
```

```
%% Modulation
qm = qammod(ib3,M);
scatterplot(qm);

%% Demodulation
qmde = qamdemod(qm,M);
scatterplot(qmde);
de = de2bi(qmde);

%% Double to uint8 converion
r = uint8(de);
r = r(:);

%% Reshaping demodulated output
temp = r(1:length(r)-pp,:);
t = reshape(temp,[],8);

%% Binary to Decimal conversion
k = bi2de(t);

%% Reshaping into matrix of the size of image
p = reshape(k,256,256);

%% Displaying output
figure('name','Received U19EC008');
imshow(p);
```

2. M-PSK

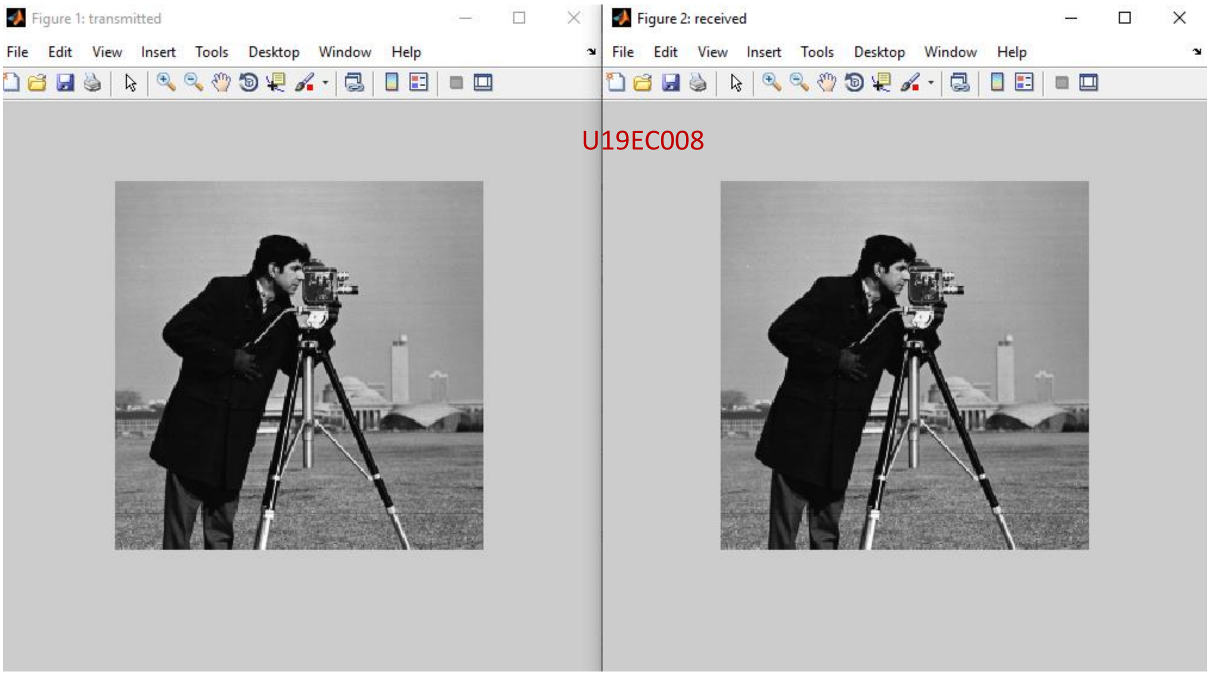
```
% Lab01 - Part 2 M-PSK
% U19EC008

clc;
clear all;
close all;

%% Input
M = input('Value of modulation order: ');
id = imread('cameraman.tif');
figure('name','transmitted U19EC008');
imshow(id);

%% Input matrix to 1-D Row Vector
ida = id(:);

%% Decimal to Binary conversion
ib = de2bi(ida);
ibd = double(ib);
```

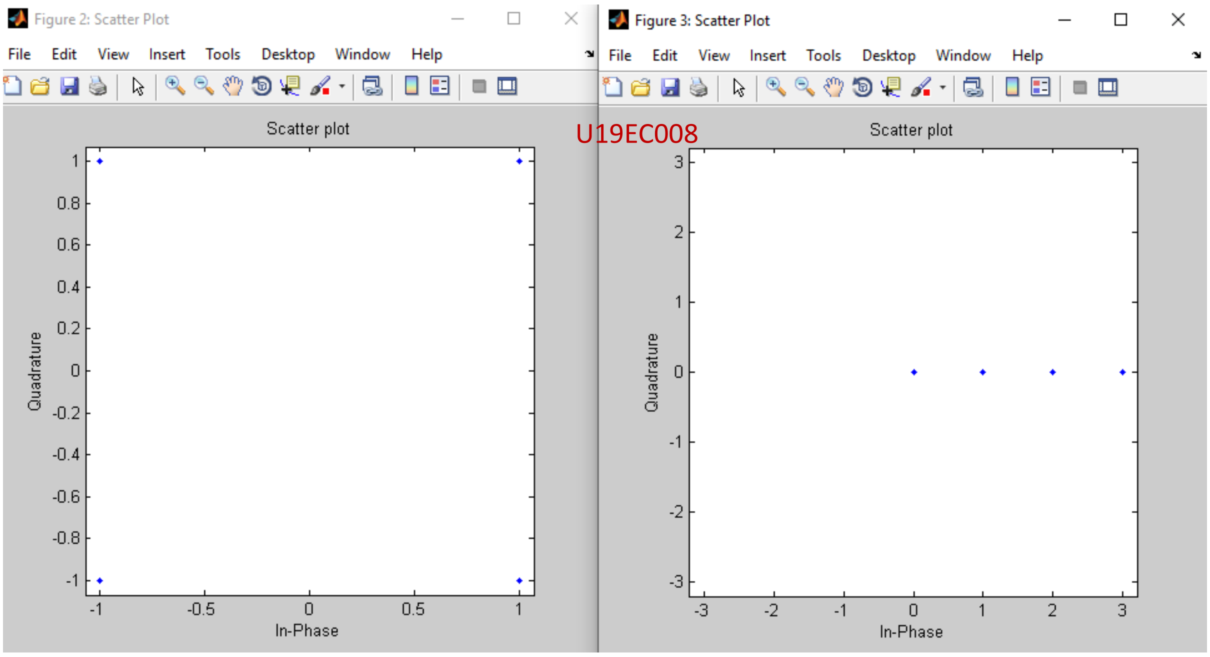
```matlab
%% Reshaping column matrix
ib = ibd(:);
x = mod(length(ib),log2(M));
pp=log2(M)-x;
ib5 = [ib.' zeros(1,pp)];

%% Reshaping column matrix
ib2 = reshape(ib5,[],log2(M));
ib3 = bi2de(ib2);

%% Modulation
pm = pskmod(ib3,M);

%% Demodulation
scatterplot(pm);
pdm = pskdemod(pm,M);
scatterplot(pdm);

%% Reshaping demodulated output
de = de2bi(pdm);

%% Double to uint8 converion
r = uint8(de);
r = r(:);

%% Reshaping demodulated output
temp = r(1:length(r)-pp,:);
t = reshape(temp,[],8);

%% Binary to Decimal conversion
k = bi2de(t);

%% Reshaping into matrix of the size of image
p = reshape(k,256,256);

%% Displaying output
figure('name','received U19EC008');
imshow(p);
```

OUTPUT:

1. M-QAM Output:

For M = 4 (QAM)





For M = 8 (QAM)

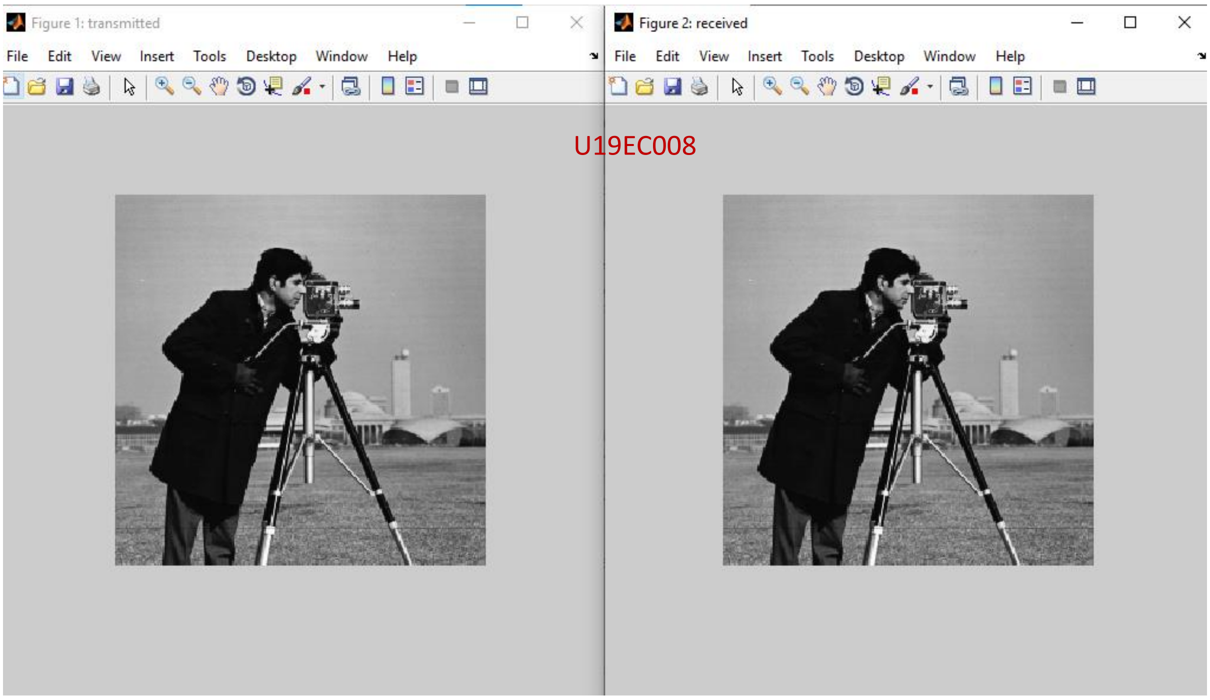


Figure 1: transmitted
File Edit View Insert Tools Desktop Window Help
Figure 2: received
File Edit View Insert Tools Desktop Window Help
U19EC008

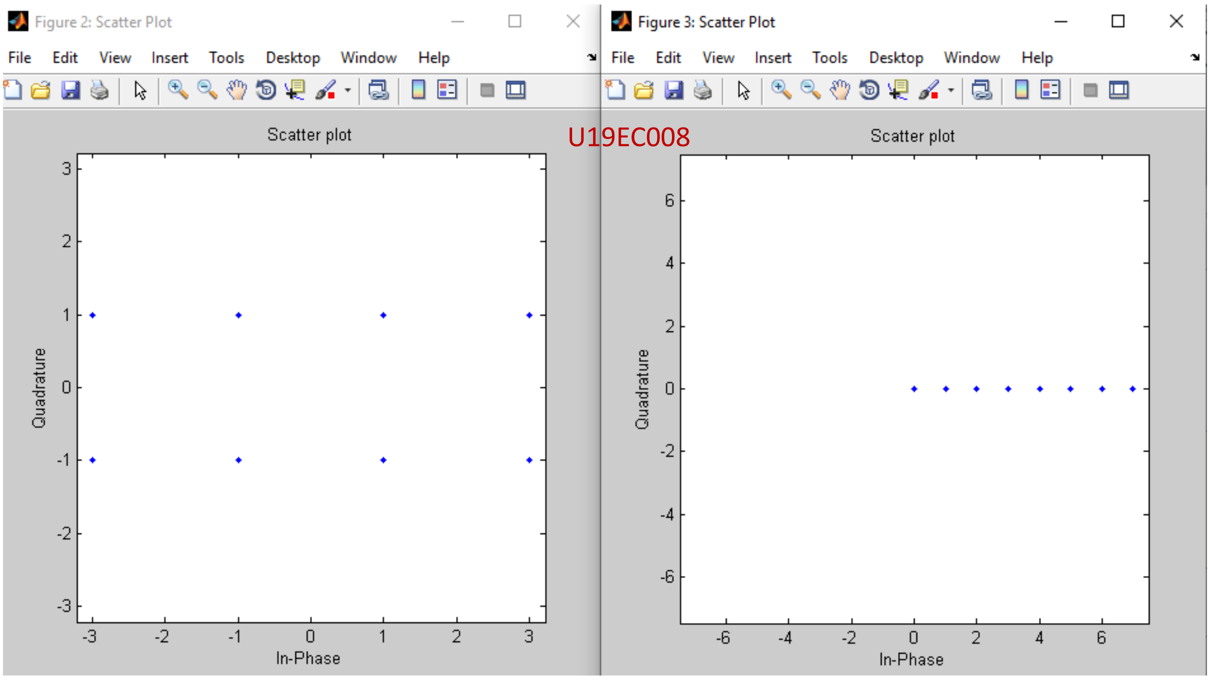


Figure 2: Scatter Plot
File Edit View Insert Tools Desktop Window Help
Scatter plot
Quadrature
In-Phase
U19EC008
Figure 3: Scatter Plot
File Edit View Insert Tools Desktop Window Help
Scatter plot
Quadrature
In-Phase

For M = 16 (QAM)

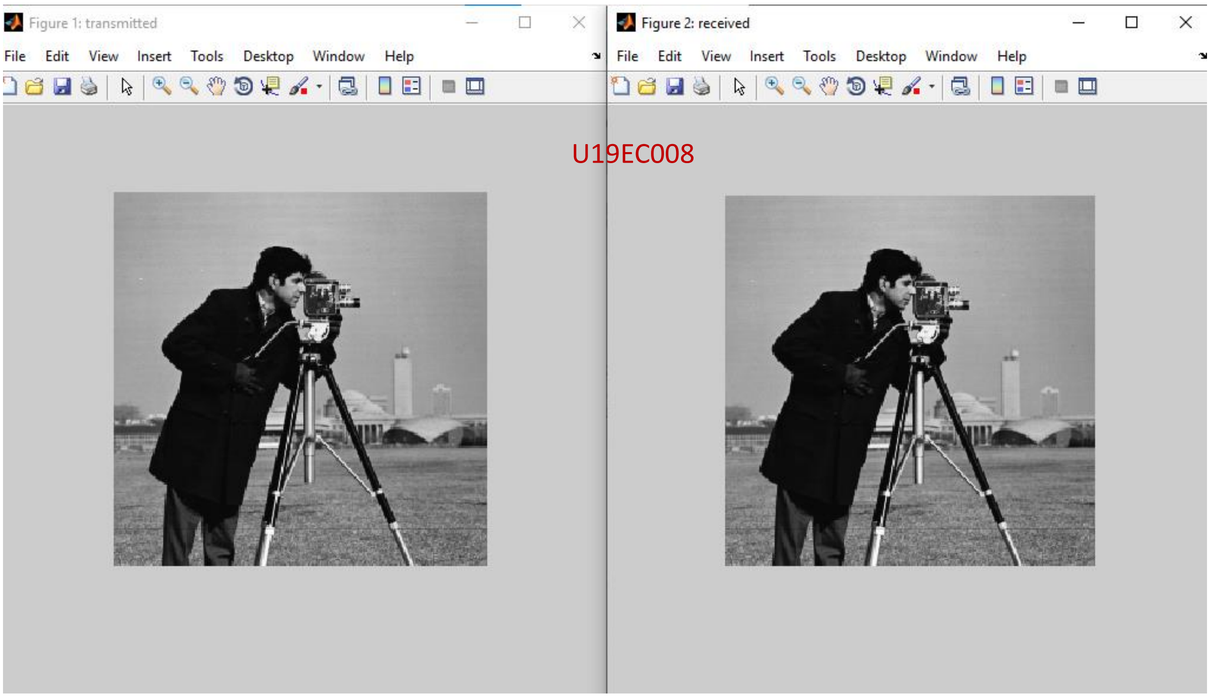


Figure 1: transmitted
Figure 2: received
U19EC008

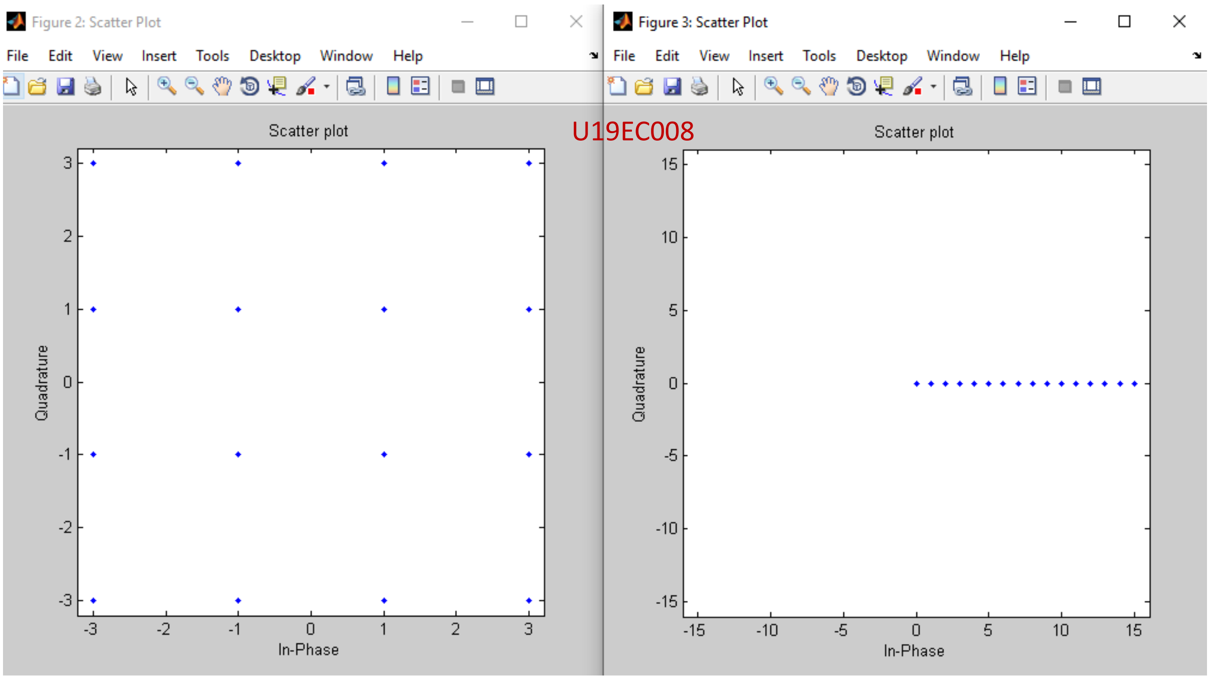


Figure 2: Scatter Plot
Scatter plot
Quadrature
In-Phase
Figure 3: Scatter Plot
U19EC008
Scatter plot
Quadrature
In-Phase

For M = 32 (QAM)

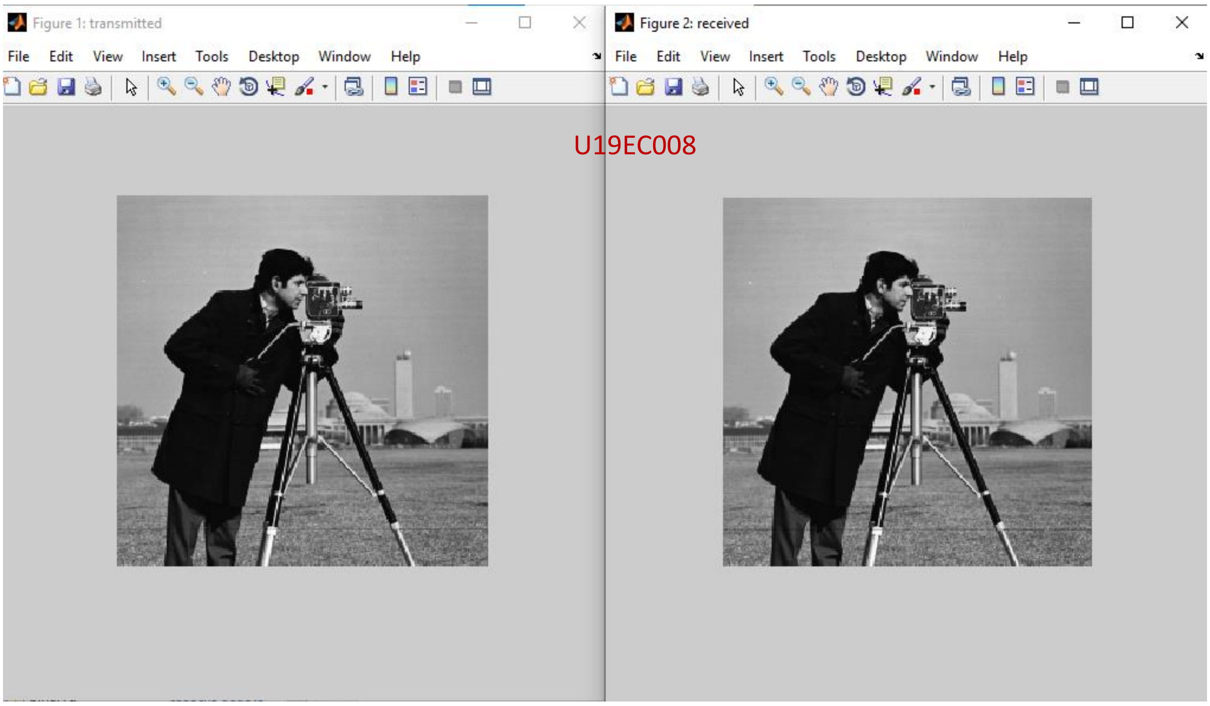


Figure 1: transmitted
Figure 2: received
U19EC008

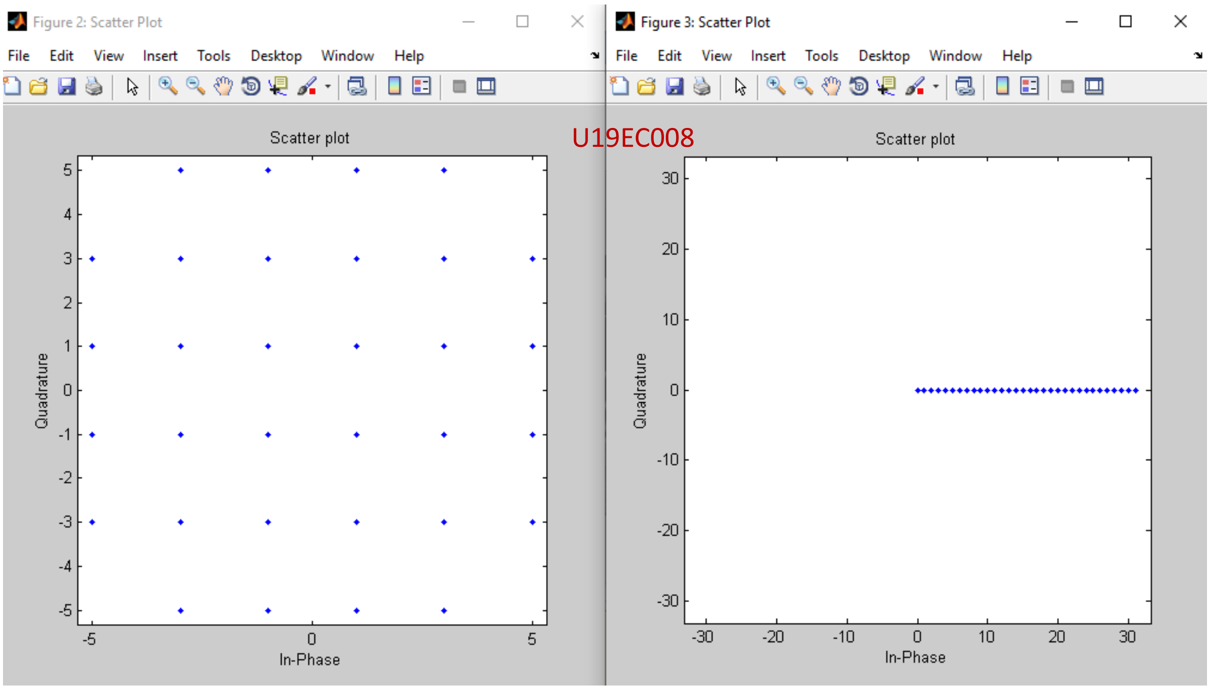


Figure 2: Scatter Plot
Figure 3: Scatter Plot
U19EC008
Scatter plot
Quadrature
In-Phase
Scatter plot
Quadrature
In-Phase

For M = 64 (QAM)


Figure 1: transmitted
Figure 2: received
U19EC008

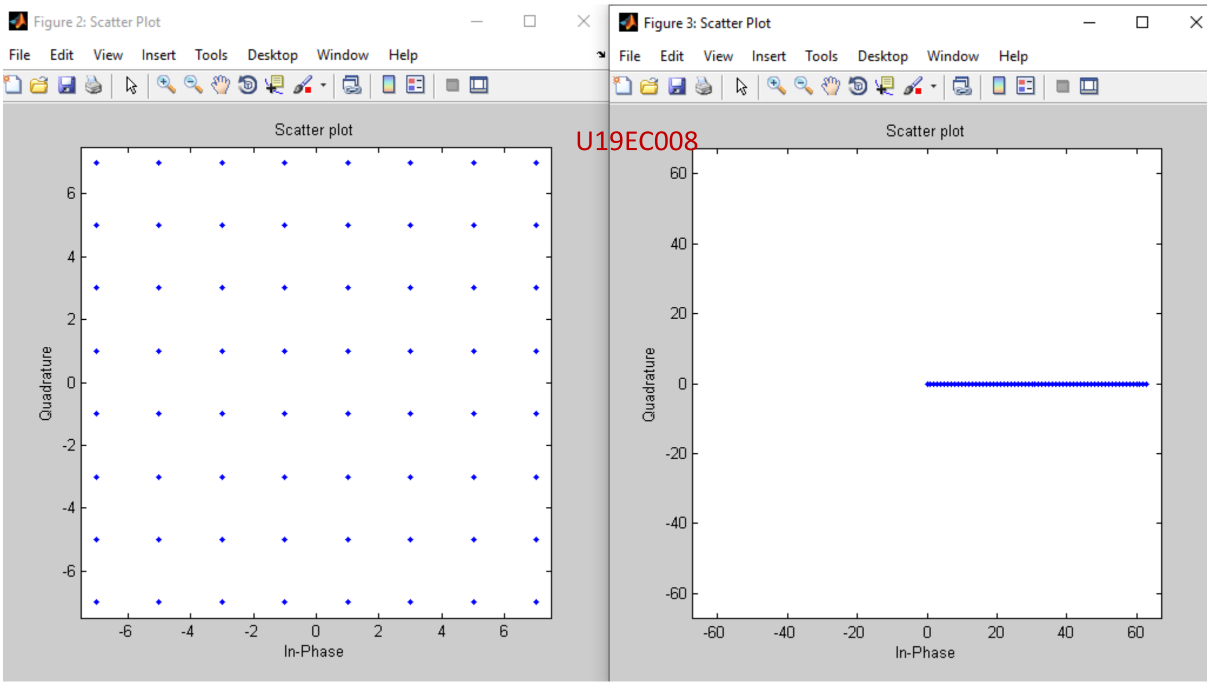
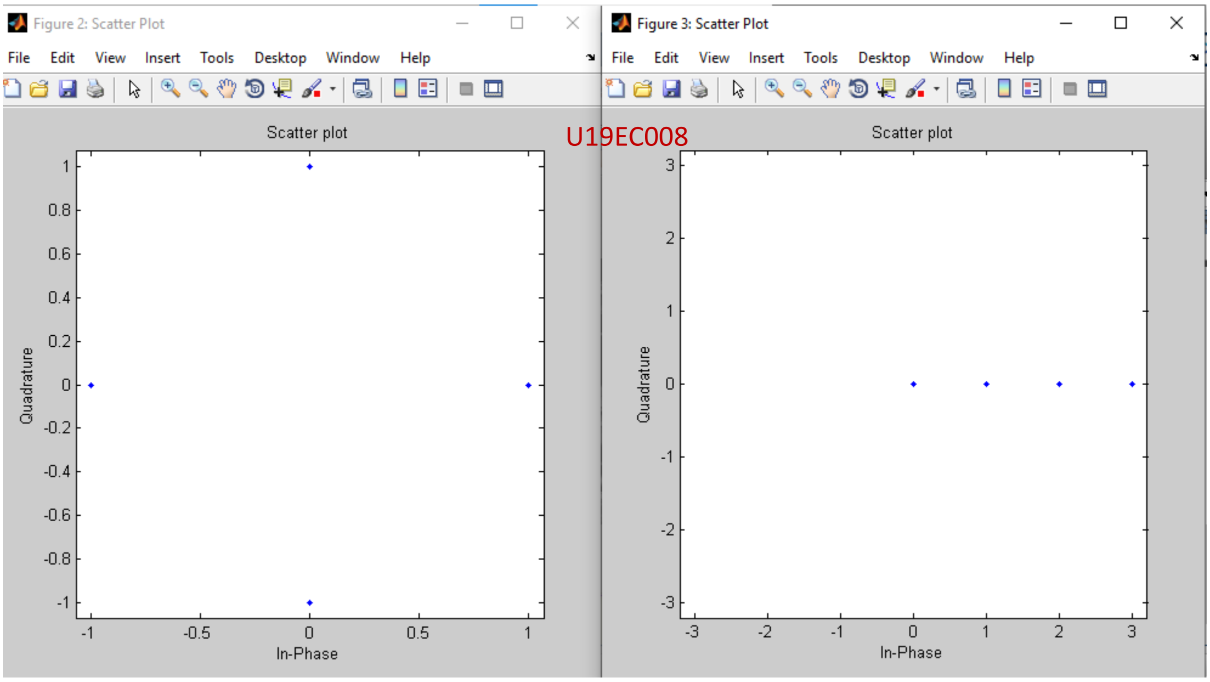


Figure 2: Scatter Plot
Scatter plot
Quadrature
In-Phase
Figure 3: Scatter Plot
U19EC008
Scatter plot
Quadrature
In-Phase

2. M-PSK Output:

For M = 4 (PSK)



For M = 8 (PSK)

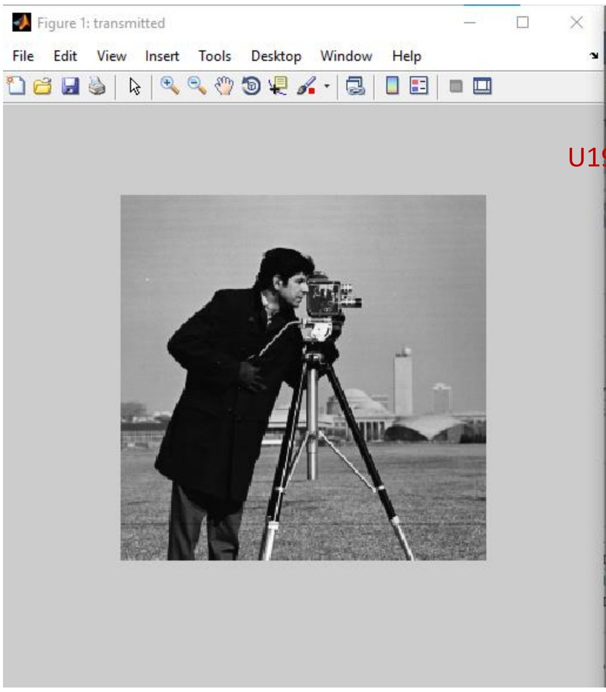


Figure 1: transmitted
File Edit View Insert Tools Desktop Window Help

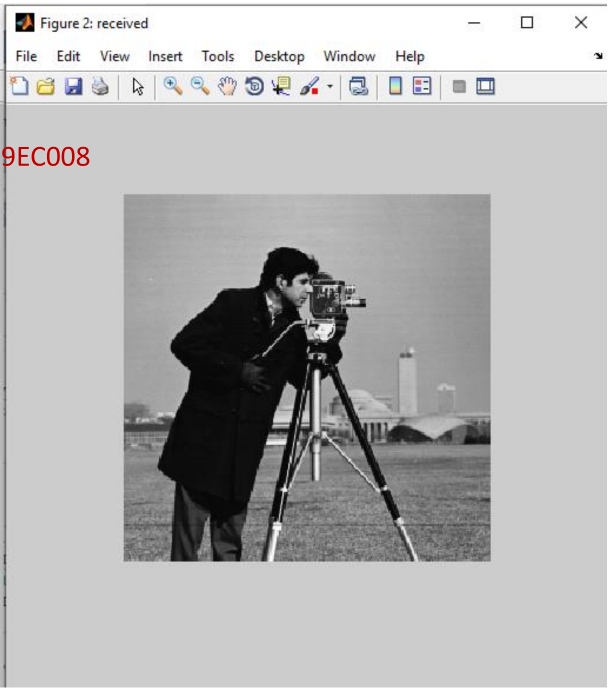


Figure 2: received
File Edit View Insert Tools Desktop Window Help
U19EC008

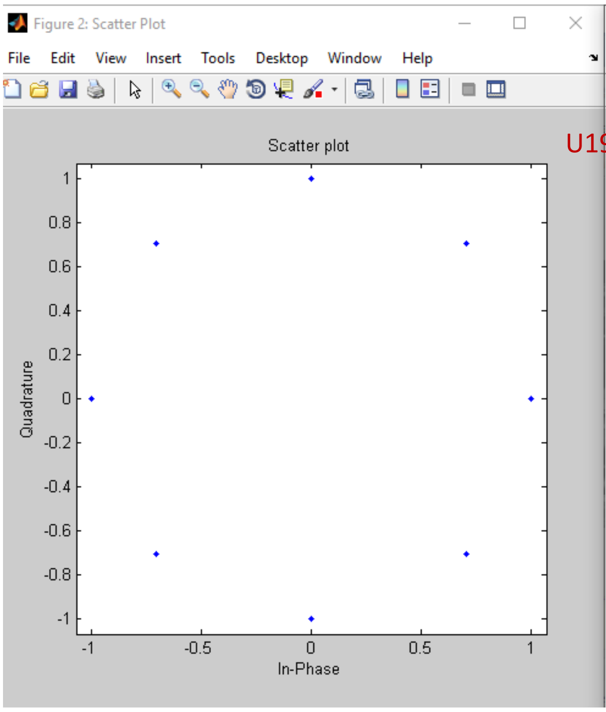


Figure 2: Scatter Plot
File Edit View Insert Tools Desktop Window Help
Scatter plot
Quadrature
In-Phase
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.5
0
0.5
1

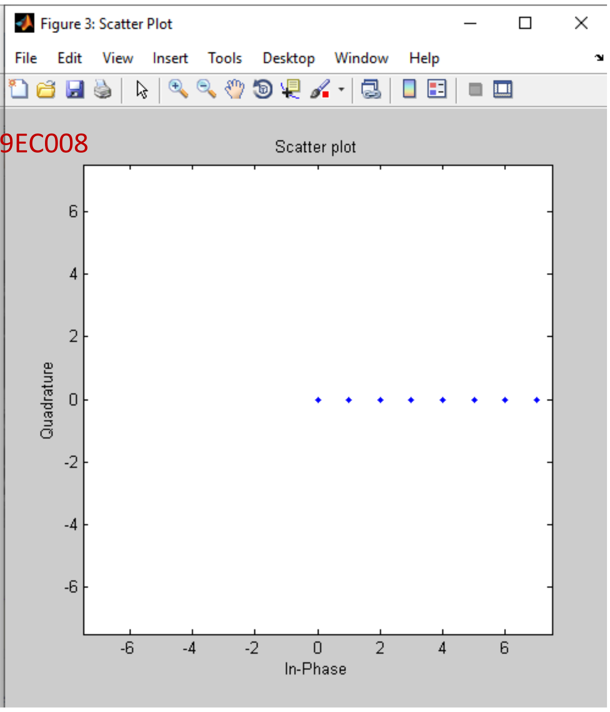


Figure 3: Scatter Plot
File Edit View Insert Tools Desktop Window Help
U19EC008
Scatter plot
Quadrature
In-Phase
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

For M = 16 (PSK)

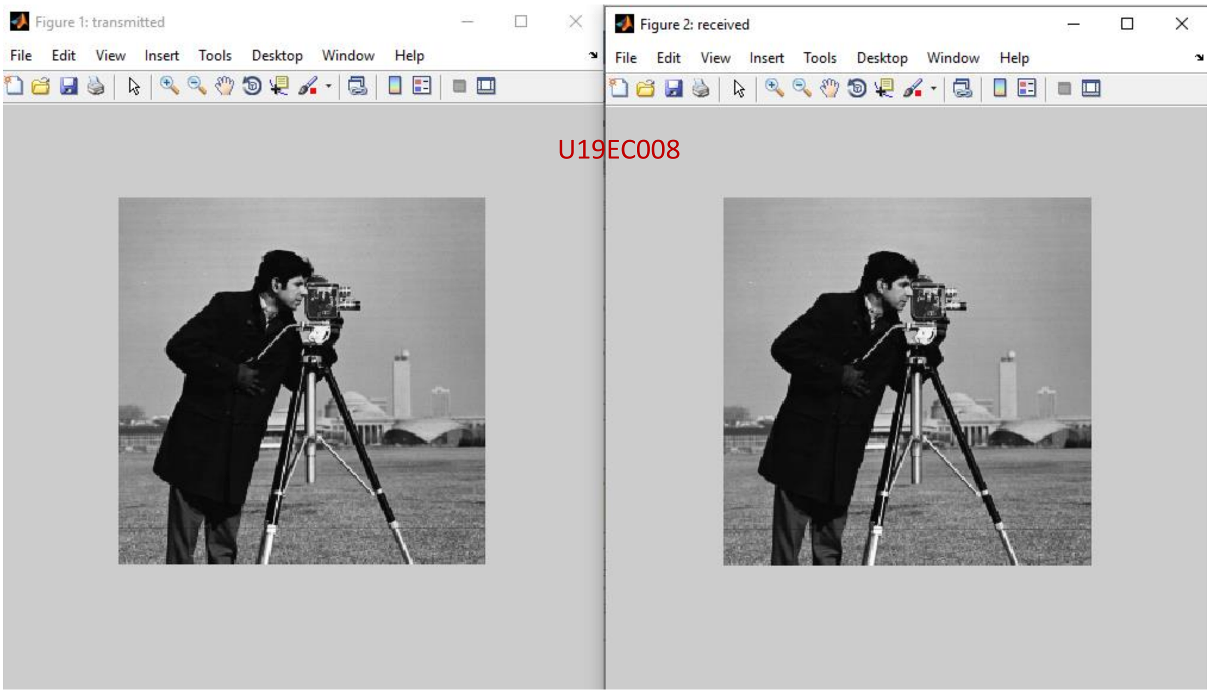


Figure 1: transmitted
File Edit View Insert Tools Desktop Window Help
U19EC008
Figure 2: received
File Edit View Insert Tools Desktop Window Help

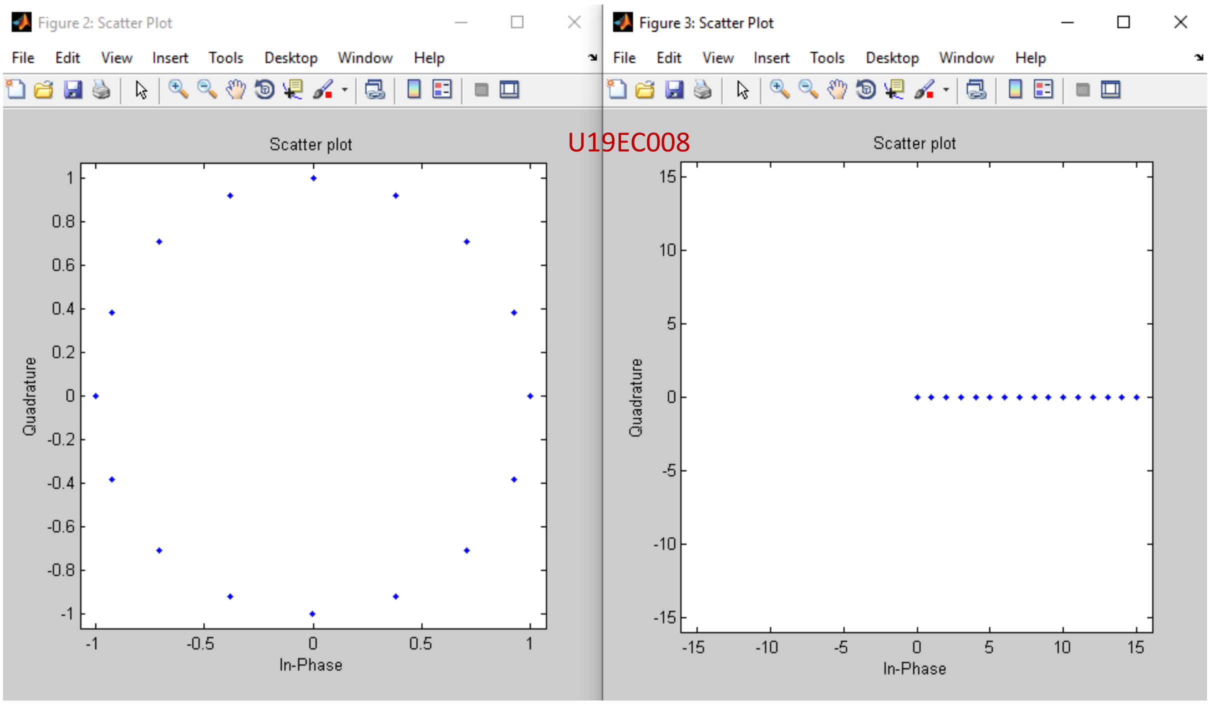


Figure 2: Scatter Plot
File Edit View Insert Tools Desktop Window Help
Scatter plot
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Quadrature
-1
-0.5
0
0.5
1
In-Phase
U19EC008
Figure 3: Scatter Plot
File Edit View Insert Tools Desktop Window Help
Scatter plot
15
10
5
0
-5
-10
-15
Quadrature
-15
-10
-5
0
5
10
15
In-Phase

For M = 32 (PSK)


Figure 1: transmitted
File Edit View Insert Tools Desktop Window Help
Figure 2: received
File Edit View Insert Tools Desktop Window Help
U19EC008

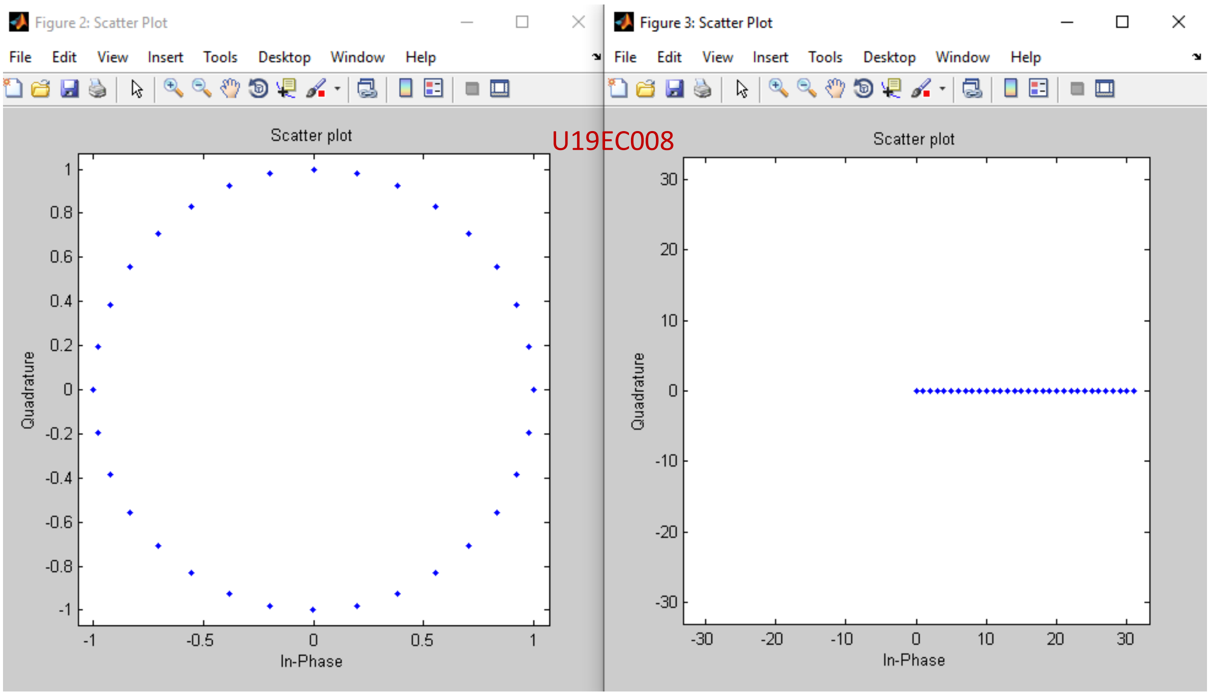
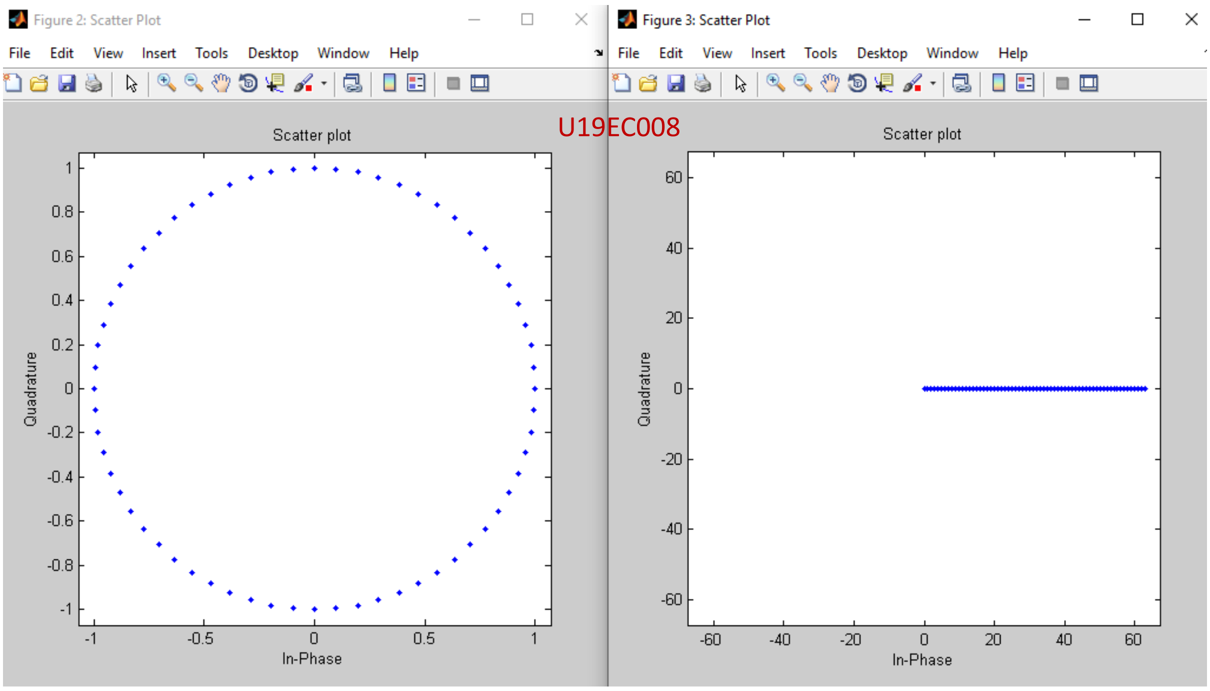


Figure 2: Scatter Plot
File Edit View Insert Tools Desktop Window Help
Scatter plot
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
Quadrature
-1
-0.5
0
0.5
1
In-Phase
U19EC008
Figure 3: Scatter Plot
File Edit View Insert Tools Desktop Window Help
Scatter plot
30
20
10
0
-10
-20
-30
Quadrature
-30
-20
-10
0
10
20
30
In-Phase

For M = 64 (PSK)



CONCLUSION:

In this practical, we have implemented and simulated M-QAM and M-PSK Modulation techniques for various values of M (4, 8, 16, 32 and 64) using MATLAB software and hence plotted the constellation diagrams. We have modulated and demodulated a gray scale image using M-QAM and M-PSK.

SANJIT ANAND U19EC008

24-01-2022 PRACTICAL ASSIGNMENT - 2

WIRELESS AND MOBILE COMMUNICATION

AIM:

To study and observe the effect of multipath at different time instants.

- To observe transmitted signal and received signal after multipath in time domain and frequency domain.
- To obtain the transfer function and impulse response of the time varying channel for various time instants.

APPARATUS:

MATLAB Software

THEORY:

Multipath Model

Consider the signal $e^{j*2\pi*f_0*t}$ transmitted from the transmitter and the corresponding received signal after subjected to multipath transmission is represented as follows:

$$y_e(t) = \sum_{j=1}^{j=J} \beta_j(t) e^{i*2*\pi*f_0*(t-\tau_j(t))}$$

Where,

- j is the total number of multipath,
- β_j (t) is the attenuation in the j^{th} path
- τ_j (t) is the time delay in the j^{th} path

The transfer function of the multipath channel at f0,

$$H(f_0, t) = \sum_{j=1}^{j=J} \beta_j(t) e^{-i*2*\pi*f_0*\tau_j(t)}$$

Similarly, it can be interpreted as the transfer function of the time varying channel for any value of f:

$$H(f, t) = \sum_{j=1}^{j=J} \beta_j(t) e^{-i*2*\pi*f*\tau_j(t)}$$

Thus, the impulse response of the time varying channel is obtained as follows:

$$h(\tau, t) = \sum_{j=1}^{j=J} \beta_j(t)\delta(\tau - \tau_j(t))$$

Here, the response of time varying multipath channel to the input signal cos(2π*fo*t) is given by,

$$y(t) = \Re(\sum_{j=1}^{j=J} \beta_j(t)e^{2*\pi*f_0*(t-\tau_j(t))})$$

PROCEDURE:

Step1: Initialize the values of transmitted signal and number of paths as per given in the aim.

Step2: Using two for loops, for every time instant between 0 and 1 with step size 0.01, introduce the phase change in transmitted signal using rand function and for delay multiply rand by time.

Step3: Plot the transmitted signal, received signal, spectrum of transmitted signal and spectrum of received signal.

Step4: Using the formulas given in the theory, compute the values of transfer function and impulse response of the channels at different time instants.

Step5: Plot the values of the above computed transfer function and impulse response for the channel at different time instances.

MATLAB CODE:

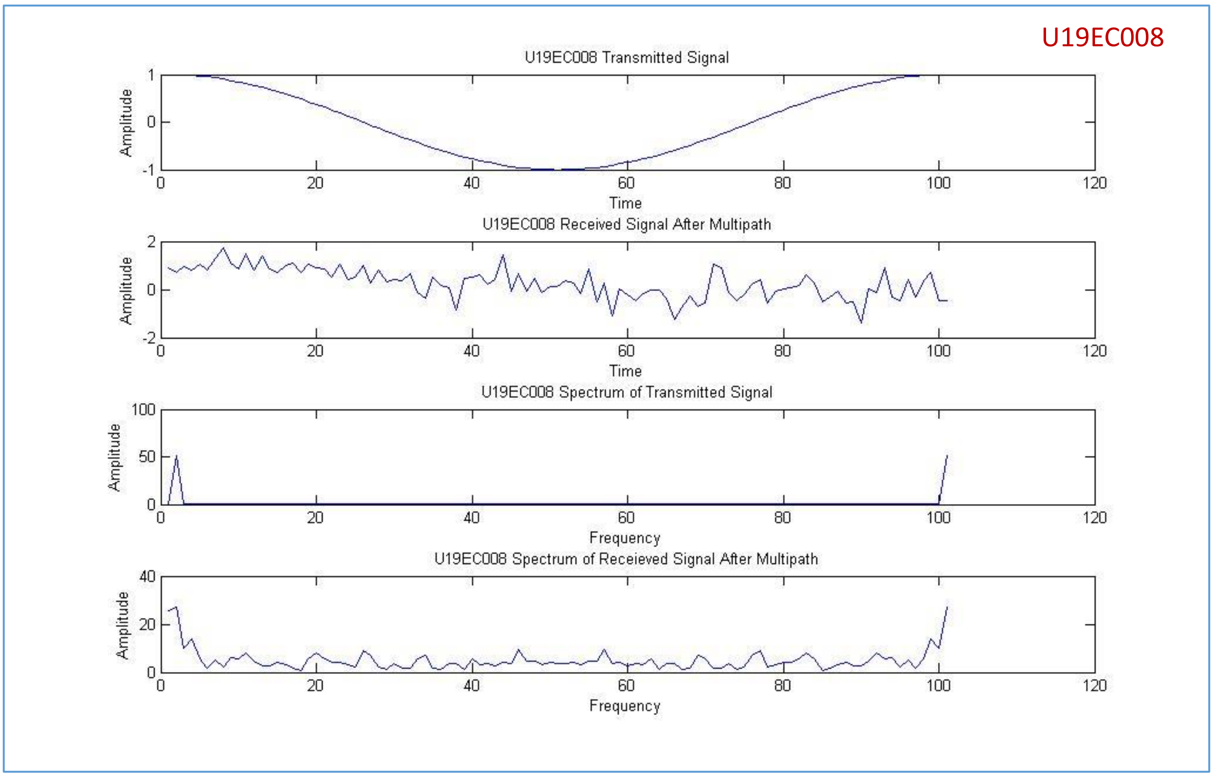
```
% LAB 02 U19EC008
clc;
clear all;
close all;

% Transmitted Signal Frequency
f = 1;

% Number of paths
nop = 2;

% Received Signal
rxsignal = [];
t = 0:0.01:1;
```

```
% Transmitted Signal
txsignal = cos(2*pi*f*t);

z = 1;

for t = 0:0.01:1
    temp = 0;
    for p = 1:1:nop
        beta(p) = rand;
        delay(p) = rand*t;
        temp = temp + beta(p)*exp(1i*2*pi*f*(t-delay(p)));
    end
    BETA{z} = beta;
    DELAY{z} = delay;
    beta = 0;
    delay = 0;
    rxsignal = [rxsignal temp];
    z = z+1;
end

save CONSTANTS BETA DELAY

% OUTPUTS
figure(1);
subplot(4, 1, 1);
plot(txsignal);
title('U19EC008 Transmitted Signal');
ylabel('Amplitude');
xlabel('Time');

subplot(4, 1, 2);
plot(real(rxsignal));
title('U19EC008 Received Signal After Multipath');
ylabel('Amplitude');
xlabel('Time');

subplot(4, 1, 3);
plot(abs(fft(txsignal)));
title('U19EC008 Spectrum of Transmitted Signal');
ylabel('Amplitude');
xlabel('Frequency');

subplot(4, 1, 4);
plot(abs(fft(real(rxsignal))));
title('U19EC008 Spectrum of Receieved Signal After Multipath');
ylabel('Amplitude');
xlabel('Frequency');

hold

% TRANSFER FUNCTION PART
load CONSTANTS

fs = 100;
u = 1;
```

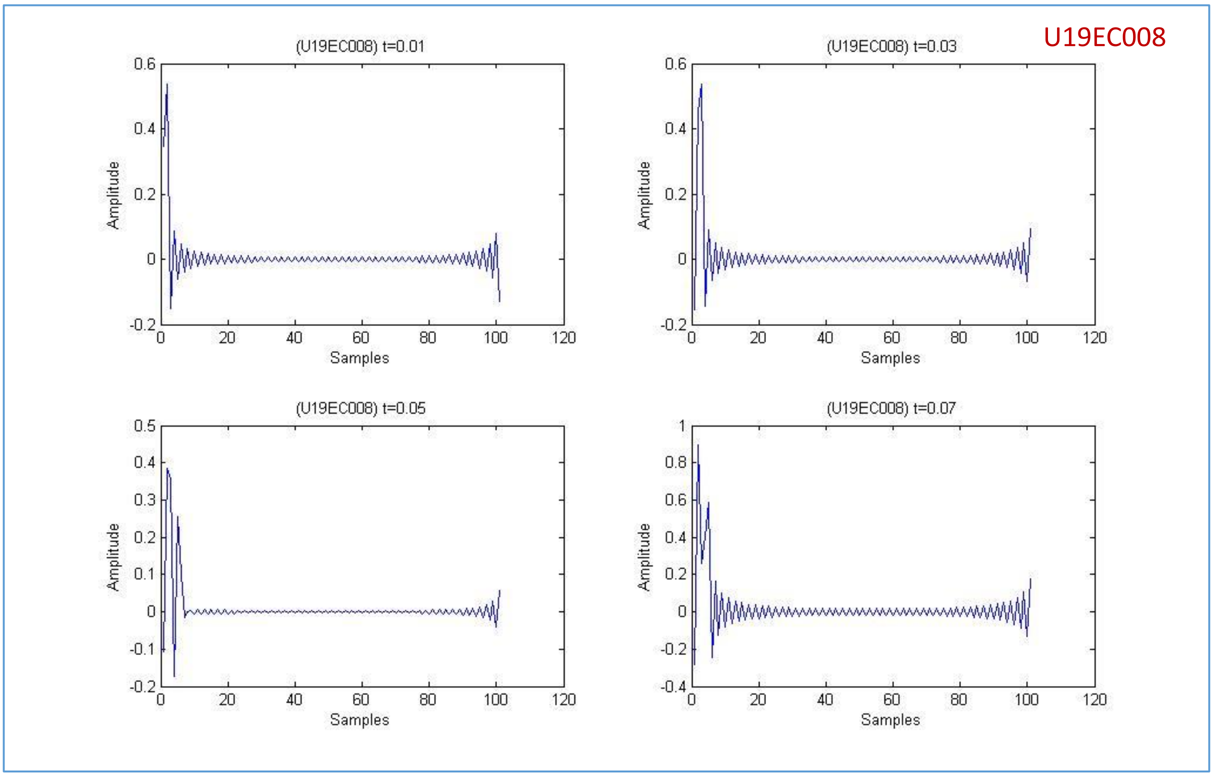
```
for f=0:fs/101:(50*fs)/101
    rxsignal=[];
    temp = 0;
    z = 1;

    for t = 0:0.01:1
        temp = 0;
        for p=1:1:nop
            temp = temp + BETA{z}(p)*exp(1i*2*pi*f*(t-DELAY{z}(p)));
        end
        rxsignal = [rxsignal temp];
        z = z+1;
    end

    t = 0:0.01:1;
    tv_TF_f{u} = rxsignal.*exp(-1i*2*pi*f*t);
    u = u + 1;

end

TEMP = cell2mat(tv_TF_f');

for i=1:1:101
    u = TEMP(:,i);
    u1=[u;transpose(u(length(u):-1:2)')];
    tv_TF_t{i} = ifft(u1);
end

TF_mat = abs(cell2mat(tv_TF_f'));
IR_mat = cell2mat(tv_TF_t);

s = [2:2:8];

for i=1:1:4

    figure(2);
    subplot (2,2,i);
    plot(IR_mat(1:1:101,s(i)));
    title(strcat('(U19EC008) t=',num2str((s(i)-1)/100)));
    xlabel('Samples');
    ylabel('Amplitude');
    figure(3);
    subplot(2,2,i);
    plot(TF_mat(:,s(i)));
    title(strcat('(U19EC008) t=',num2str((s(i)-1)/100)));
    xlabel('Samples');
    ylabel('Amplitude');

end
```

OUTPUT:

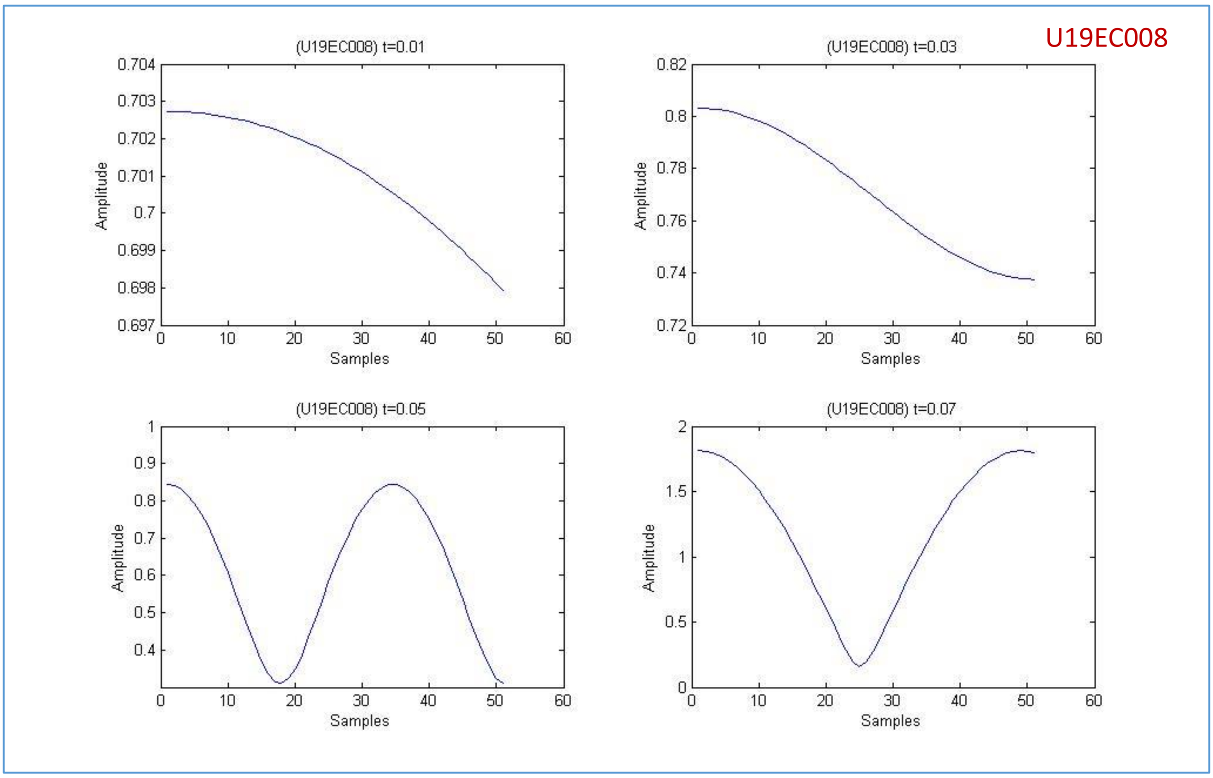
Transmitted and Received Signals



Impulse Response of channel at different time instants



Transfer Function of channel at different time instants



CONCLUSION:

In this practical, we have observed the effect of multipath at various time instants. We have observed the transmitted signal, received signal, spectrum of transmitted signal and spectrum of received signal. We have also obtained the transfer function, impulse response of the channel at different time instants.
We can conclude that as we increase the number of multipath, we get more fluctuations and thus we get more distorted output.

SANJIT ANAND U19EC008

31-01-2022 PRACTICAL ASSIGNMENT - 3

WIRELESS AND MOBILE COMMUNICATION

AIM:

To Simulate M PSK and M QAM Modulation Techniques using AWGN channel considering input as an Image with the help of MATLAB software Plot SNR v/s BER where M= 4 8 16 32 64 and constellation as well

APPARATUS:

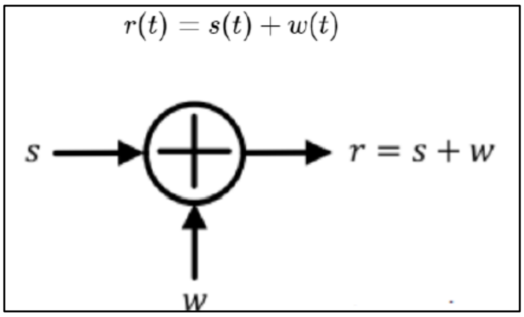
MATLAB Software

THEORY:

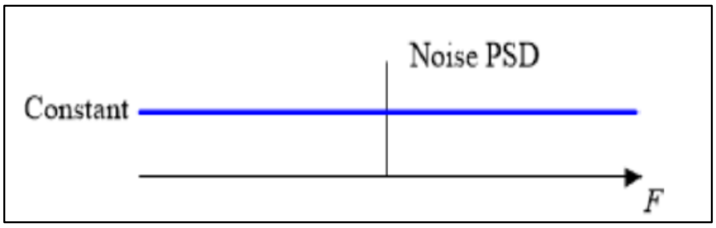
Additive White Gaussian Noise (AWGN)

- **AWGN -** A basic noise model used to mimic the effect of many random processes that occur in nature
- Channel produces Additive White Gaussian Noise (AWGN).

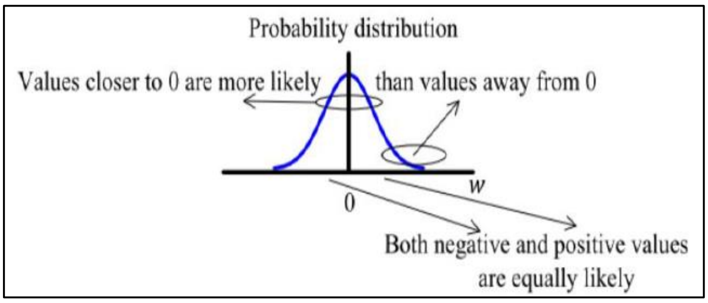
Additive: The received signal equals the transmit signal plus some noise, where the noise is statistically independent of the signal.



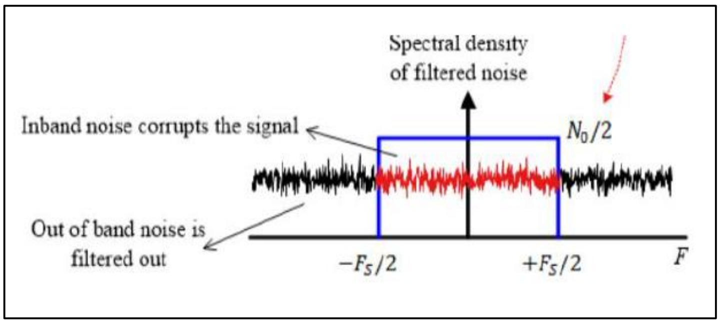
White: It refers that the noise has the same power distribution at every frequency OR it has uniform power across the frequency band for the information system.
It is an analogy to the colour white which has uniform emissions at all frequencies in the visible spectrum if I focused a beam of light for each colour on the visible spectrum onto a single spot that combination would result in a beam of white light.
As a consequence, the Power Spectral Density (of white noise is constant for all frequencies ranging from to as shown in figure below



Gaussian: Gaussian distribution, or a normal distribution, has an average of zero in the time domain, and is represented as a bell shaped curve.
The probability distribution of the noise samples is Gaussian with a zero mean. The values close to zero have a higher chance of occurrence while the values far away from zero are less likely to appear.



In reality, the ideal flat spectrum from to is true for frequencies of interest in wireless communications (a few kHz to hundreds of GHz) but not for higher frequencies.



Signal-to-Noise Ratio: The SNR or S/N is a measure used in science and engineering that compares the level of a desired signal to the level of background noise.
It is defined as the ratio of signal power to the noise power, often expressed in decibels.
A ratio higher than 1 1 (greater than 0 dB) indicates more signal than noise.
SNR, bandwidth, and channel capacity of a communication channel are connected by the Shannon Hartley theorem.

$$\mathrm{SNR_{dB}} = 10\log_{10}\left(\frac{P_{\mathrm{signal}}}{P_{\mathrm{noise}}}\right).$$

Shannon-Hartley theorem: It states the channel capacity bits per second OR information rate of data that can be communicated at low error rate using an average received signal power through communication channel subject to additive white Gaussian noise (AWGN) of power.

$$C = B\log_2\left(1+\frac{S}{N}\right)$$

It is the related to signal to noise ratio (SNR) or the carrier to noise ratio (CNR) (expressed as a linear power ratio, not as logarithmic decibels)

- **5 dB - 10 dB:** is below the minimum level to establish a connection, due to the noise level being nearly indistinguishable from the desired signal (useful information**)**
- **25 dB - 40 dB:** is deemed to be good
- **41 dB or higher:** is considered to be excellent

PROCEDURE:

Step1: Input type of modulation and order of modulation technique

Step2: Converting colored image into grayscale image

Step3: Convert 2D pixel matrix into 1D Row vector

Step4: Convert 1D vector to binary and reshape to column vector

Step5: Padding 0 if required Modulate (M-ary PSK or QAM)

Step6: Add AWGN because of Channel Demodulation

Step7: Convert demodulated binary output to decimal and then to 2D matrix

Step8: Display reconstructed image and compare with original grayscale

Step9: Plot SNR Vs BER Graph END

MATLAB CODE:

```
% LAB 03 - U19EC008

clc;
clear all;
close all;

% Read Image
img=imread('cameraman.tif');
```

```
mxM = 6;
mxSNR = 40;
BERQAM = zeros(mxM,mxSNR/2);
BERPSK = zeros(mxM,mxSNR/2);
QAMstr = 'QAM (M = %d) U19EC008';
PSKstr = 'PSK (M = %d) U19EC008';
t = 1:2:mxSNR;

figure(1);
subplot(mxM/3,3,1);
imshow(img);
title('Original U19EC008');

figure(2);
subplot(mxM/3,3,1);
imshow(img);
title('Original U19EC008');

col = ['b','g','r','y','m','k']

for m  = 2:mxM
    Mod_Ord = 2^m;
    SymSize = log2(Mod_Ord);
    z = rem(length(img),SymSize);

    if z ~= 0;
        img = [img; zeros(SymSize - z, prod(size(img))/length
(img))];
    end

    img_bin = de2bi(img);
    img_rsp = reshape(img_bin, 8*length(img_bin)/SymSize, SymSize);
    img_dec = bi2de(img_rsp);
    yQAM = qammod(img_dec, Mod_Ord);
    yPSK = pskmod(double(img_dec), Mod_Ord);

    for s = 1:2:mxSNR
        nQAM = awgn(yQAM,s);
        nPSK = awgn(yPSK,s);
        zQAM = qamdemod(nQAM, Mod_Ord);
        zPSK = pskdemod(nPSK, Mod_Ord);
        [a,b] = biterr(img_dec,zQAM);
        BERQAM(m,(s+1)/2) = 100*b;
        [c,d] = biterr(img_dec,zPSK);
        BERPSK(m,(s+1)/2) = 100*d;
    end

    QAM_dec = de2bi(zQAM);
    QAM_rsp = reshape(QAM_dec, size(img_bin));
    QAMM = bi2de(QAM_rsp);
    QAMM = uint8(reshape(QAMM,size(img)));

    PSK_dec = uint8(de2bi(zPSK));
    PSK_rsp = reshape(PSK_dec, size(img_bin));
    PSKK = bi2de(PSK_rsp);
```

```
    PSKK = uint8(reshape(PSKK,size(img)));

    scatterplot(nQAM);
    suptitle('U19EC008')
    scatterplot(nPSK);
    suptitle('U19EC008')

    figure(1);
    subplot(mxM/3,3,m);
    imshow(QAMM);
    title(sprintf(QAMstr,Mod_Ord));

    figure(2);
    subplot(mxM/3,3,m);
    imshow(PSKK);
    title(sprintf(PSKstr,Mod_Ord));

    figure(mxM+1);
    subplot(211);

plot(t,BERQAM(m,:),'linewidth',2,'DisplayName',sprintf(QAMstr,Mod_Or
d), 'color', col(m-1));
    title('BER vs SNR (QAM) U19EC008');
    xlabel('Signal to Noise Ratio');
    ylabel('Bit Error Rate(%)');
    legend;

    hold on;

    subplot(212);

plot(t,BERPSK(m,:),'linewidth',2,'DisplayName',sprintf(PSKstr,Mod_Or
d), 'color', col(m-1));
    title('BER vs SNR (PSK) U19EC008');
    xlabel('Signal to Noise Ratio');
    ylabel('Bit Error Rate(%)');
    legend;
    hold on;

end
```

OUTPUT:

(BER Performance of System)
Bit Error Rate vs Signal to Noise Ratio (QAM and PSK)

U19EC008

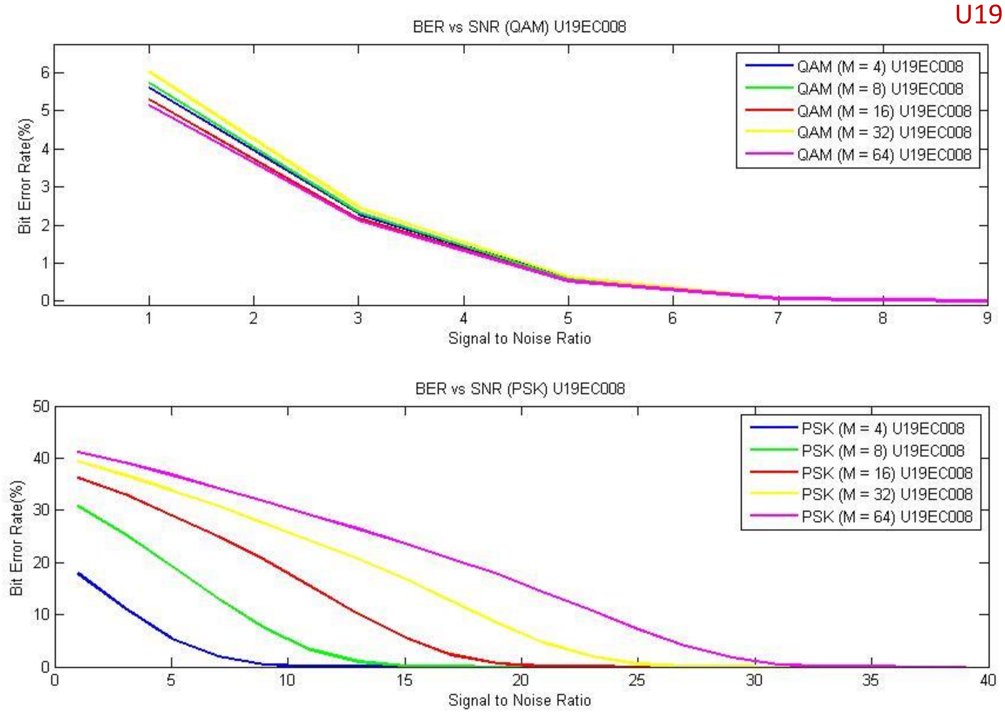


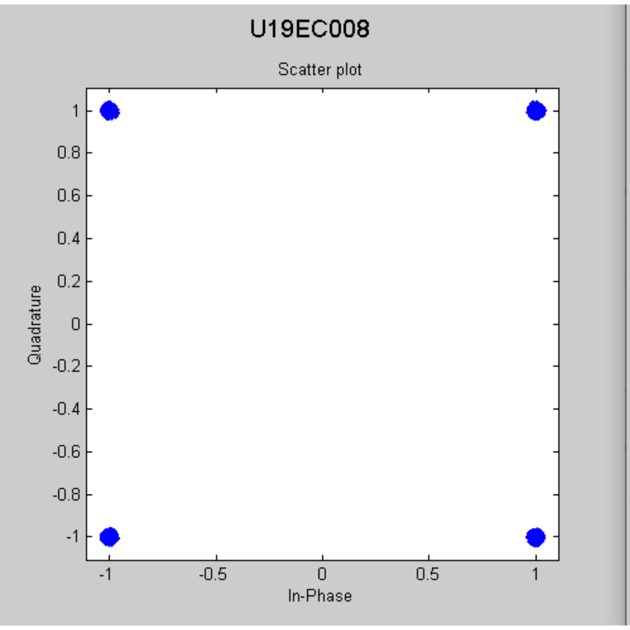
Image Output using QAM [Original, M=4, 8, 16, 32, 64]

U19EC008

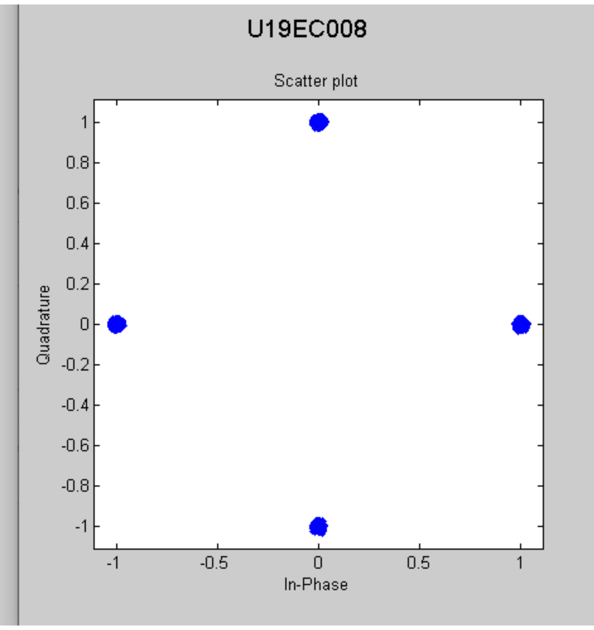
Image Output using PSK [Original, M=4, 8, 16, 32, 64]

Constellation Diagrams (Scatterplots):

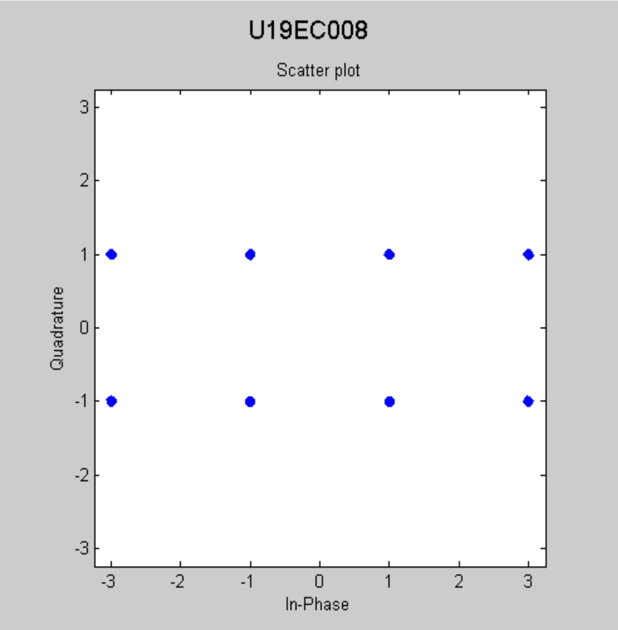
QAM (M= 4)



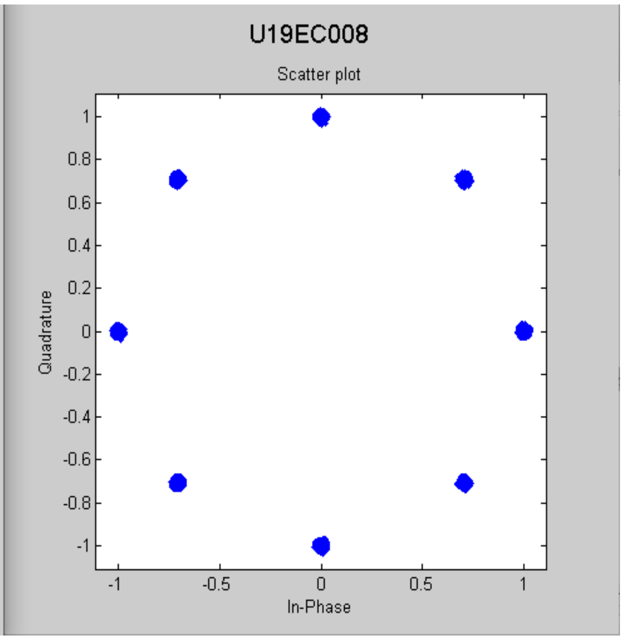
PSK (M=4)



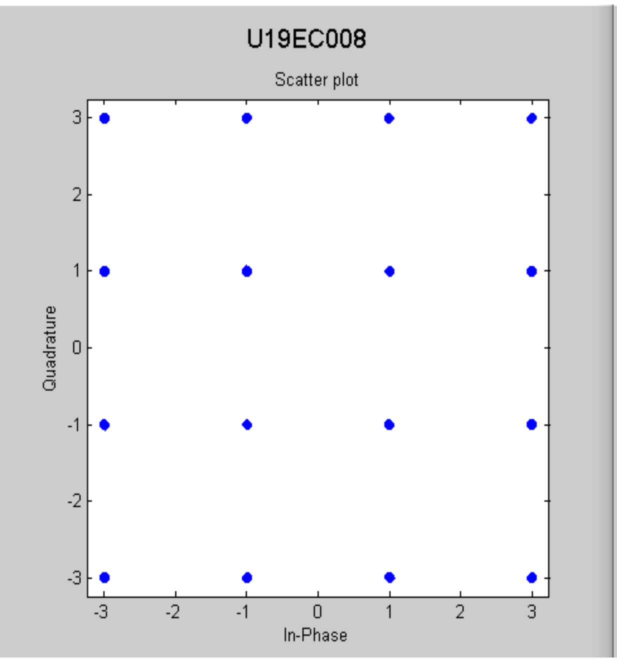
QAM (M= 8)



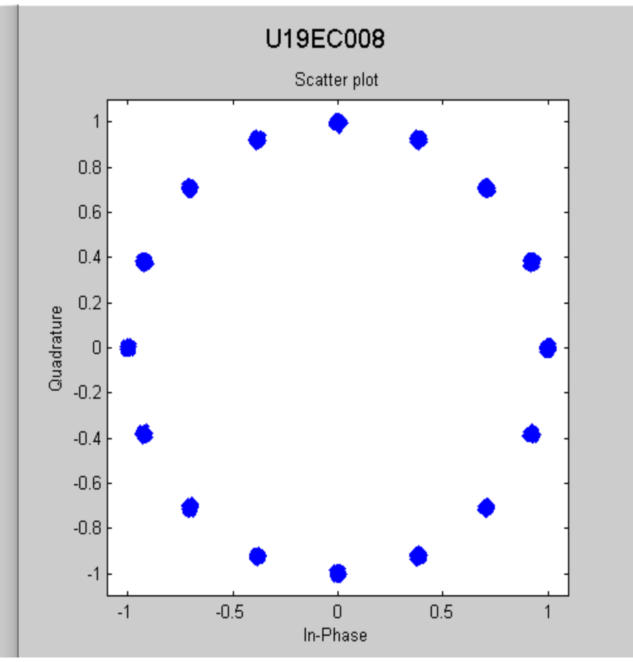
PSK (M=8)



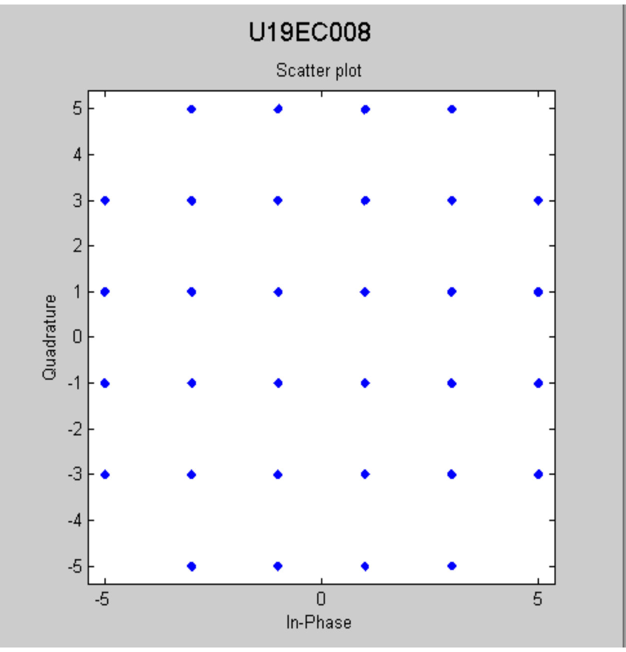
QAM (M= 16)



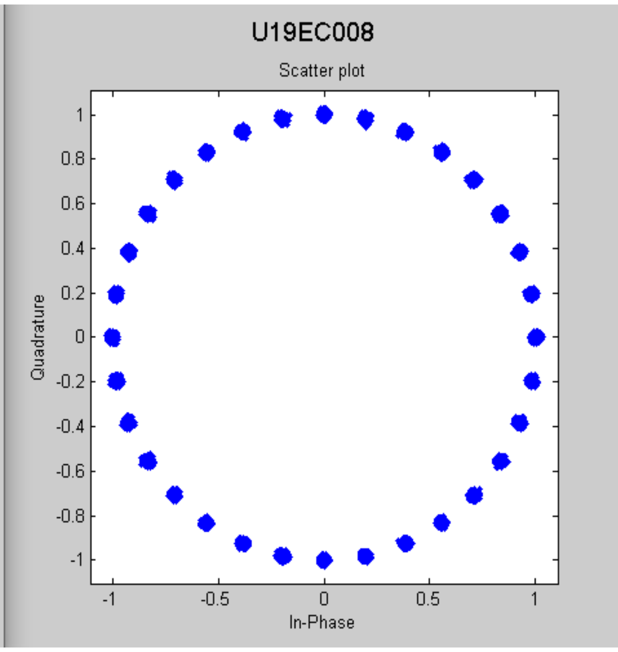
PSK (M=16)



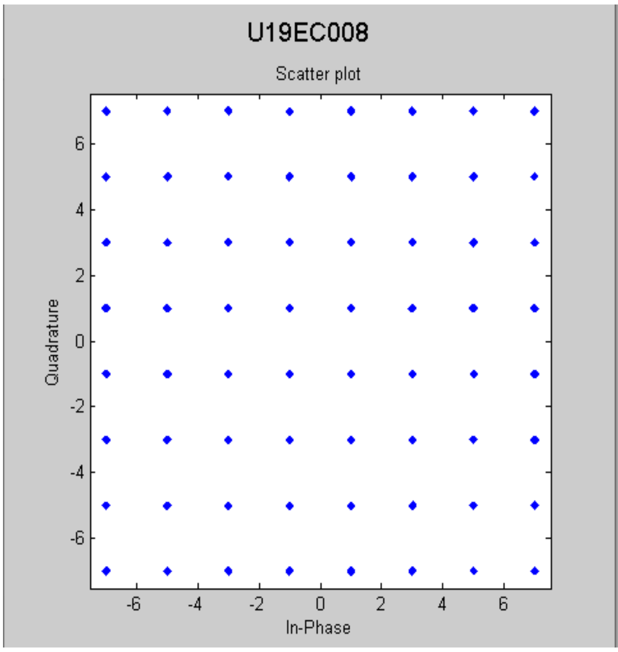
QAM (M= 32)



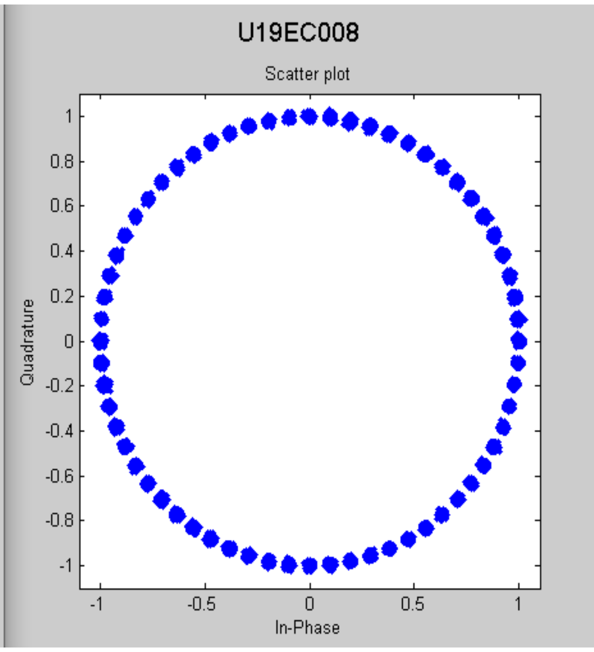
PSK (M=32)



QAM (M= 64)



PSK (M=64)



CONCLUSION:

In this practical, we have successfully transmitted the input source image using different orders of PSK modulation technique and calculated the Bit Error Rates for different SNR values for each of them. We can conclude that as the values of N increases, BER increases and the value of SNR increases BER decreases.

SANJIT ANAND U19EC008

07-02-2022 PRACTICAL ASSIGNMENT - 4

WIRELESS AND MOBILE COMMUNICATION

AIM:

To study and observe the effect of Doppler spread and delay spread for fast fading and slow fading channel and calculate the coherent bandwidth in MATLAB.

APPARATUS:

MATLAB Software

THEORY:

Doppler Spread

• Doppler shift is the random changes in a channel introduced as a result of a mobile user's mobility.
• Doppler spread has the effect of shifting or spreading the frequency components of a signal
• Types of fading on the basis of doppler spread are fast fading and slow fading.
• Fast fading: Channel impulse response changes rapidly within the symbol duration.
• Slow fading: Channel impulse response changes at a rate much slower than the transmitted symbol bandwidth.
• Doppler spread is expressed in the following formula. As mentioned, doppler spread is defined as maximum doppler shift (fm).

Doppler spread:-

$$f_m = \frac{v}{\lambda}$$

v = velocity of moving vehicle
λ = wavelength = c/f
f = frequency of carrier
c = speed of electromagnetic wave in free space (3×10^8 m/s)

Coherence Time:

• The coherence time of the channel is the inverse of the Doppler spread.

• It is the measure of the speed at which channel characteristics change.

• The coherence time is the time over which a propagating wave may be considered coherent. In other words, it is the time interval within which its phase is, on average, predictable.

Coherence time :-

$$T_c \approx \frac{1}{2\pi f_m}$$

where:

Maximum doppler spread,

$$f_m = \frac{v}{\lambda}$$

In Doppler spread, how fast the transfer function of the time-varying channel changes with time for a fixed frequency is to be studied. Doppler spread and the coherence time are used for the same. It is due to the different Doppler shift frequencies associated with the multiple propagation paths when there is relative motion between the transmitter and the receiver

Delay Spread

Delay spread is a measure of the multipath profile of a mobile communications channel. It is generally defined as the difference between the time of arrival of the earliest component (e.g., the line-of-sight wave if there exists) and the time of arrival of the latest multipath component.

Coherence bandwidth Bc is a statistical measure of the range of frequencies over which the channel can be considered flat (i.e., it passes all spectral components with approximately equal gain and linear phase). All frequency components of the transmitted signal within the coherence bandwidth will fade simultaneously. The coherence bandwidth is inversely proportional to the delay spread, and we thus have the following:

$$B_c = \frac{1}{\sigma_\tau}$$

In delay spread, how fast the transfer function of the time-varying channel changes with frequency at a particular time instant is to be studied. It happens because different propagation paths have different time delays.

PROCEDURE:

- **Step1:** For **Doppler Spread Fast fading**, consider the signal cos $2\pi fot$, where fo = 1 MHz. Also let the rate at which the delay (τj) is changing with time be randomly chosen as TAU_J = [0.62 1.84 0.86 0.37] for fast fading.

- **Step2**: Hence the corresponding Doppler shift for the frequency fo in the corresponding paths is obtained as DJ = $-fo$+TAUJ and the actual shift in the frequency is given as fshift = |DJ + fo| = [0.38 0.84 0.14 0.63]

- **Step3**: Attenuation in individual paths, BETA(β) = [0.23 0.17 0.23 0.44]

- **Step4**: Thus the received signal is represented as $\sum_{j=1}^{4} \beta(j) \cos(2\pi(fshift(j))t)$

- **Step5**: Plot the received signal and the corresponding spectrum

- For **Doppler Spread Slow fading**, take Tauj as [0.0042 0.0098 0.0030 0.007] and Beta as [0.2691 0.4228 0.5479 0.9427];

- For **Delay spread**, take Tau as [0.9143 -0.0292 0.6006 -0.7162] and Beta as [0.9575 0.9649 0.1576 0.9706] and vary frequency at particular time instant.

MATLAB CODE:

1. Doppler Spread (Fast Fading)

```
% LAB04 U19EC008
% Doppler Spread Fast Spreading

clc;
clear all;
close all;

Tau0 =0;
% f is in MHz and sampling time is in microseconds
f=1;
% Number of Multipaths
nop = 4;

% Rate at which the delay is changing
Tauj = [0.62 1.84 0.86 0.37];
% Attenuation of Individual Paths
BETA = [0.23 0.17 0.23 0.44];

% Actual shift in the frequency
fshift=[];
```

```
for j=1:1:nop
    fshift(j)=abs(-f+Tauj(j));
end

% Received Signal
rxsignal =[];
%  time varying transfer function at f
tvtf = [];

t = 0:(0.01):100;

% Transmitted Signal
txsignal = cos(2*pi*f*t);

for t =0:(0.01):100
    temp = 0;
    tf = 0;
    for p=1:1:nop
        temp = temp+ BETA(p)*cos(2*pi*fshift(p)*t);  %1.11
        tf = tf+BETA(p)*exp(-1i*2*pi*f*Tau0)*exp(-1i*2*pi*f*Tauj(p)*t);
% 1.9
    end
    rxsignal = [rxsignal temp];
    tvtf = [tvtf tf];
end

% PLOTS
figure(1)
subplot(2,2,1)
plot(txsignal)
axis([1 1000 -2 2]);
title('U19EC008 Transmitted signal');
xlabel('time (microseconds)');
ylabel('Amplitude');
grid on;

subplot(2,2,2)
plot(real(rxsignal),'r')
axis([1 1000 -2 2]);
title('U19EC008 Received signal');
xlabel('time (microseconds)');
ylabel('Amplitude');
grid on;

subplot(2,2,3)
fre = (0:1:length(rxsignal)-1)/100;
plot(fre, abs(fft(txsignal)));
axis([0 2 0 6000]);
title('U19EC008 Spectrum of transmitted signal');
xlabel('Frequency (MHz)');
ylabel('Amplitude');
grid on;

subplot(2,2,4)
plot(fre, abs(fft(real(rxsignal))), 'r');
axis([0 2 0 2500]);
title('U19EC008 Corresponding Spectrum of Received signal');
xlabel('Frequency (MHz)');
```

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


figure(2);
subplot(2,1,1)
plot(abs(tvtf));
axis([0 1000 0 2]);
title('U19EC008 Time varying transfer function (Magnitude)');
xlabel('time (microseconds)');
ylabel('Magnitude');
grid on;

subplot(2,1,2)
plot(phase(tvtf));
axis([0 1000 -25 0]);
title('U19EC008 Time varying transfer function (Phase)');
xlabel('time (microseconds)');
ylabel('Phase');
grid on;
```

2. Doppler Spread (Slow Fading)

```
% LAB04 U19EC008
% Doppler Spread Slow Spreading

clc;
clear all;
close all;

Tau0 =0;
% f is in MHz and sampling time is in microseconds
f=1;
% Number of Multipaths
nop = 4;

% Rate at which the delay is changing
Tauj = [0.0042 0.0098 0.0030 0.0070];
% Attenuation of Individual Paths
BETA = [0.2961 0.4228 0.5479 0.9427];

% Actual shift in the frequency
fshift=[];
for j=1:1:nop
    fshift(j)=abs(f-Tauj(j));
end

% Received Signal
rxsignal =[];
%  time varying transfer function at f
tvtf = [];

t = 0:(0.01):100;

% Transmitted Signal
```

```
txsignal = cos(2*pi*f*t);

for t =0:(0.01):100
    temp = 0;
    tf = 0;
    for p=1:1:nop
        temp = temp+ BETA(p)*cos(2*pi*fshift(p)*t);  %1.11
        tf = tf+BETA(p)*exp(-1i*2*pi*f*Tau0)*exp(-
1i*2*pi*f*Tauj(p)*t); % 1.9
    end
    rxsignal = [rxsignal temp];
    tvtf = [tvtf tf];
end

% PLOTS

figure(1)
subplot(2,2,1)
plot(txsignal)
xlim([0 10000])
title('U19EC008 Transmitted signal');
xlabel('time (microseconds)');
ylabel('Amplitude');
grid on;

subplot(2,2,2)
fre = (0:1:length(rxsignal)-1)/100;
plot(real(rxsignal),'r')
xlim([0 10000])
title('U19EC008 Received signal');
xlabel('time (microseconds)');
ylabel('Amplitude');
grid on;

subplot(2,2,3)
plot(fre, abs(fft(txsignal)));
xlim([0 2])
ylim([0 7000])
title('U19EC008 Spectrum of transmitted signal');
xlabel('Frequency (MHz)');
ylabel('Amplitude');
grid on;

subplot(2,2,4)
plot(fre, abs(fft(real(rxsignal))), 'r');
xlim([0 2])
ylim([0 7000])
title('U19EC008 Corresponding Spectrum of Received signal');
xlabel('Frequency (MHz)');
ylabel('Amplitude');
grid on;

figure(2);
subplot(2,1,1)
plot(abs(tvtf));
xlim([0 10000])
```

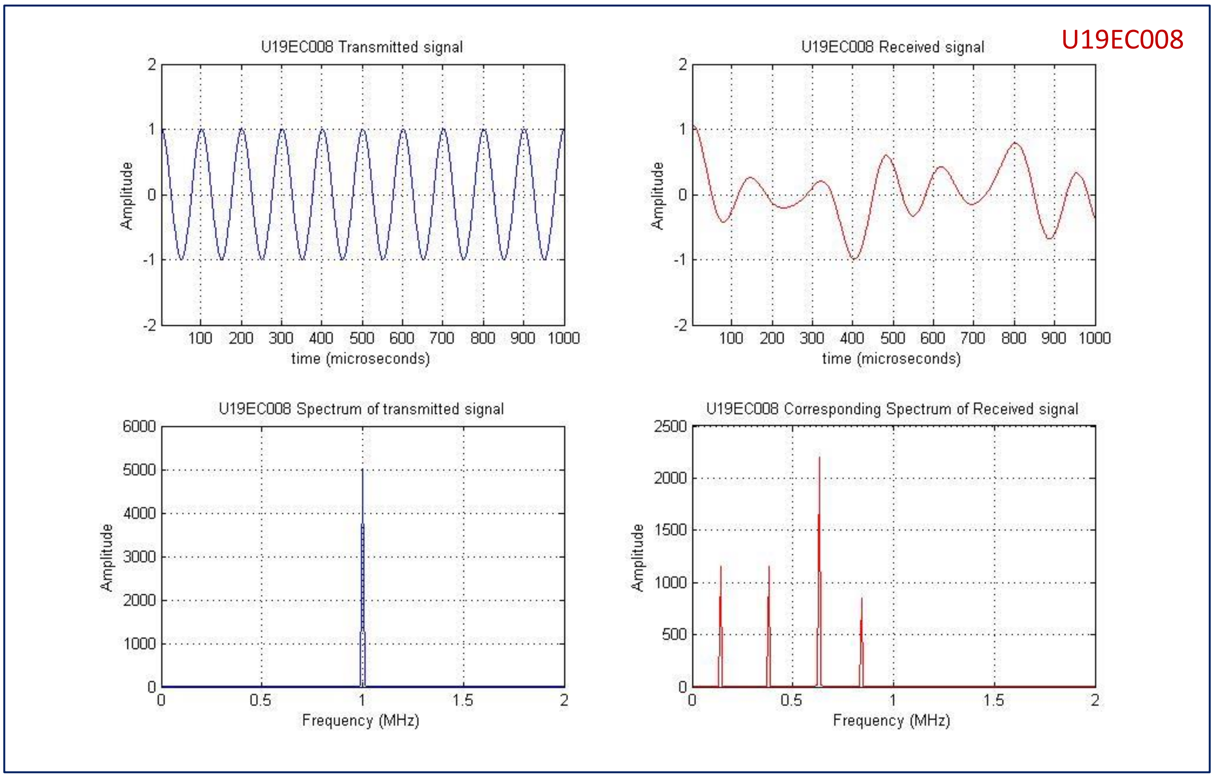
```
title('U19EC008 Time varying transfer function (Magnitude)');
xlabel('time (microseconds)');
ylabel('Magnitude');
grid on;

subplot(2,1,2)
plot(phase(tvtf));
xlim([0 10000])
title('U19EC008 Time varying transfer function (Phase)');
xlabel('time (microseconds)');
ylabel('Phase');
grid on;
```

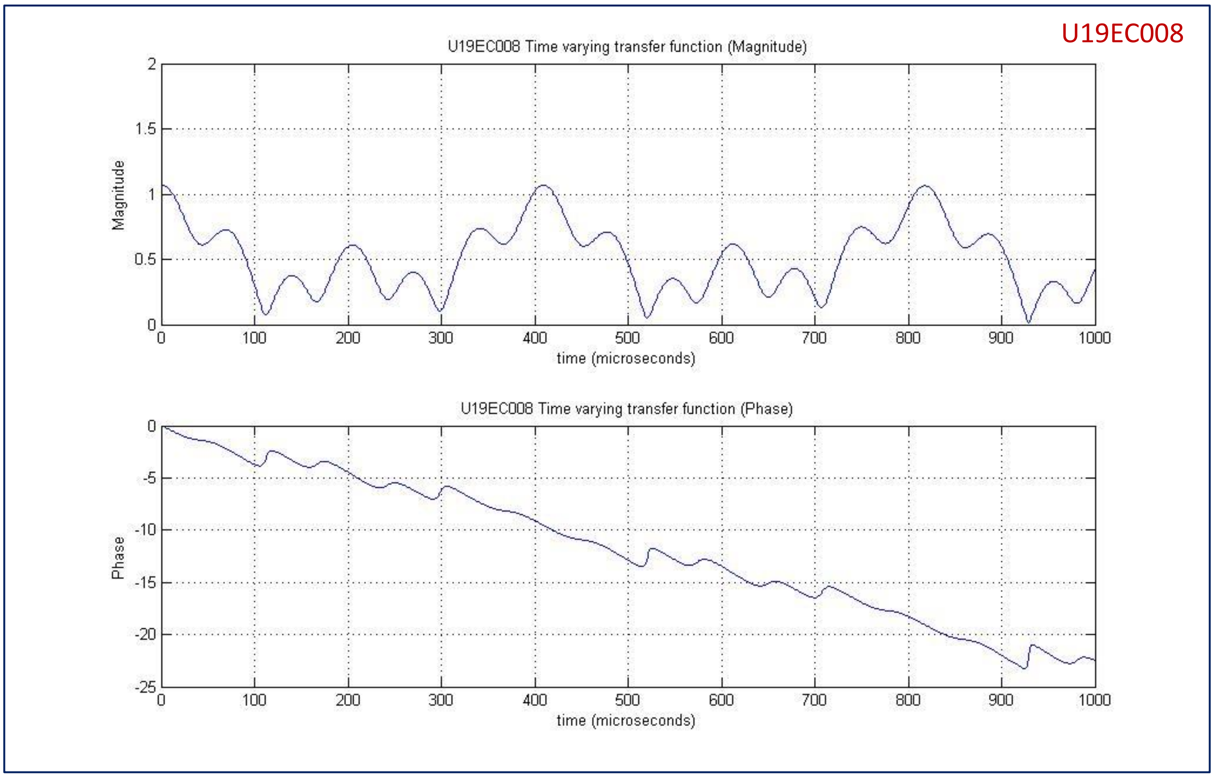
3. Delay Spread

```
% LAB4 - U19EC008
% DELAY SPREAD

clc;
clear all;
close all;

TAU0=0;
% Time Instant
t0=1;
% Number of Multipaths
nop=4;
% Attenuation of Individual Paths
BETA=[0.9575 0.9649 0.1576 0.9706];
% Rate at which the delay is changing
TAUJ=[0.9143 -0.0292 0.6006 -0.7162];

%  time varying transfer function at t0
tvtf_t0=[];
for f=0:(1/1000):1
    temp1=0;
    for p=1:1:nop
        temp1=temp1+BETA(p)*exp(-1i*2*pi*f*TAU0)*exp(-
1i*2*pi*f*TAUJ(p)*t0);
    end
    tvtf_t0=[tvtf_t0 temp1];
end
figure;
plot((0:(1/1000):1)*1000,abs(tvtf_t0));
title('U19EC008 Time Varying Transfer Function computed at the time
instant t0=1us')
xlim([0 1000])
xlabel('Frequency (KHz)');
ylabel('Amplitude');
grid on;
```

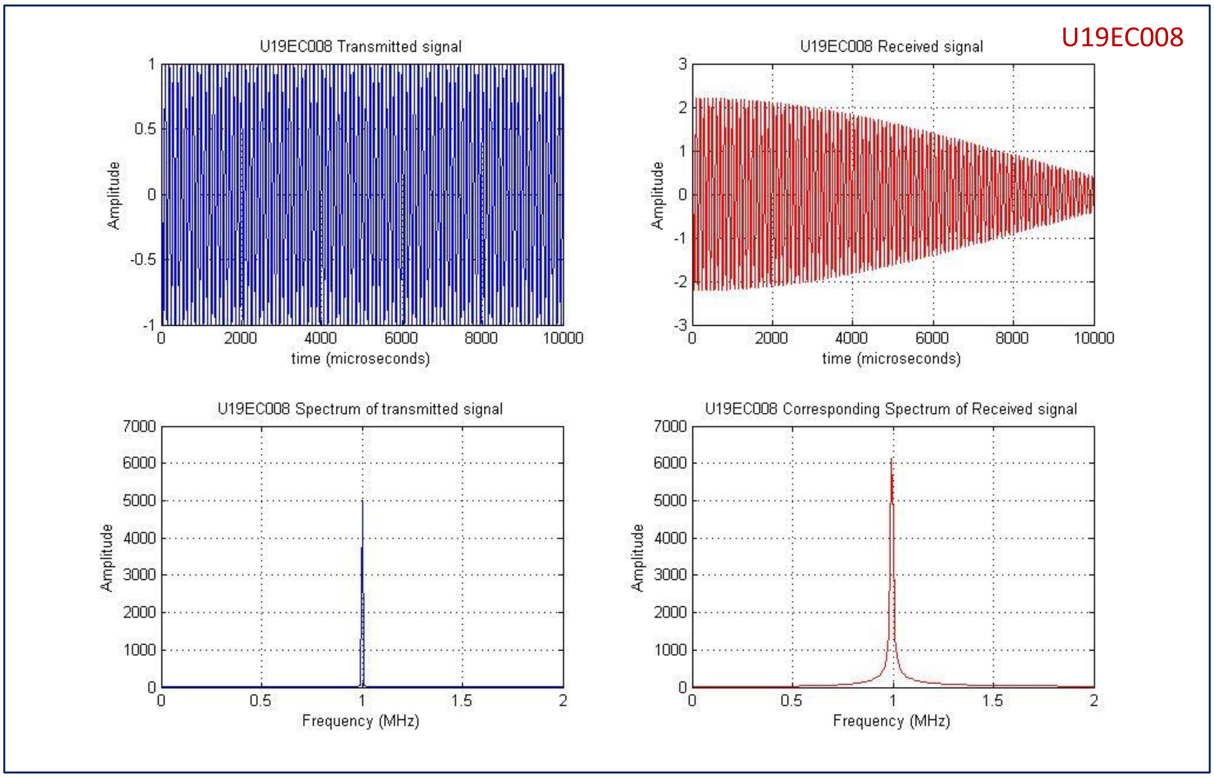
OUTPUT:

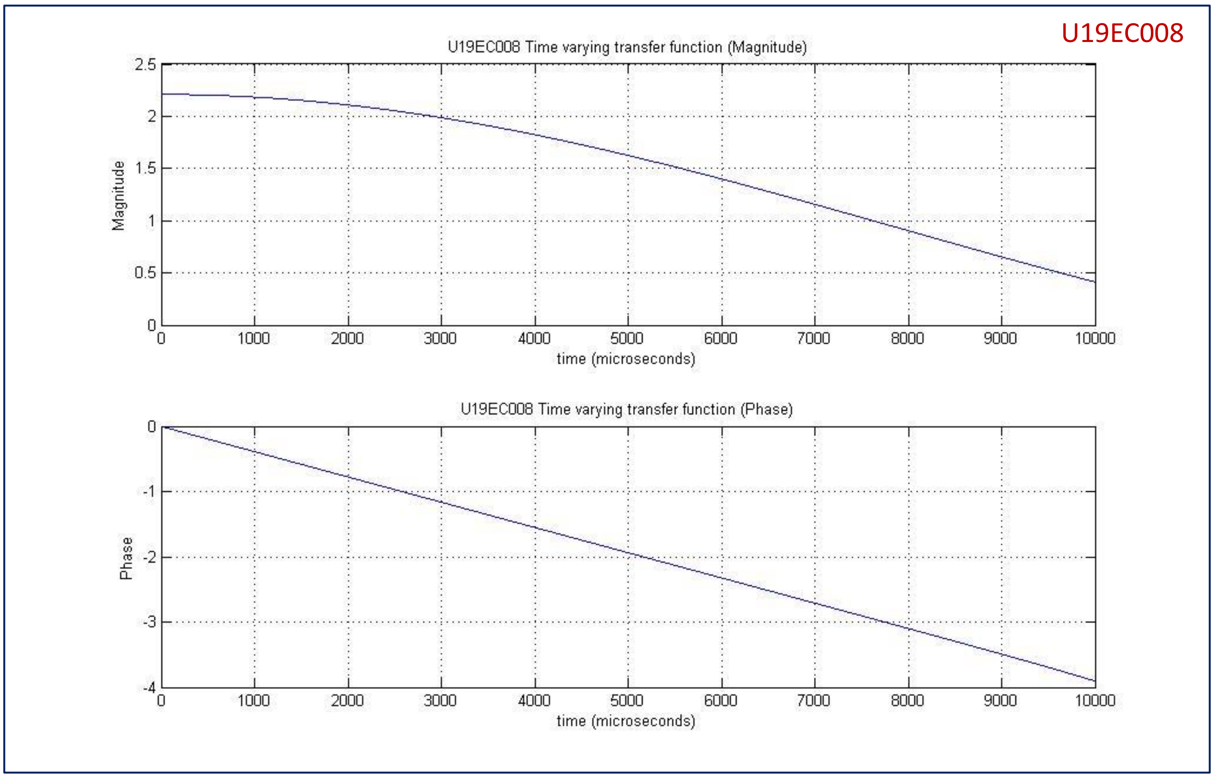
Doppler Spread (Fast Fading)



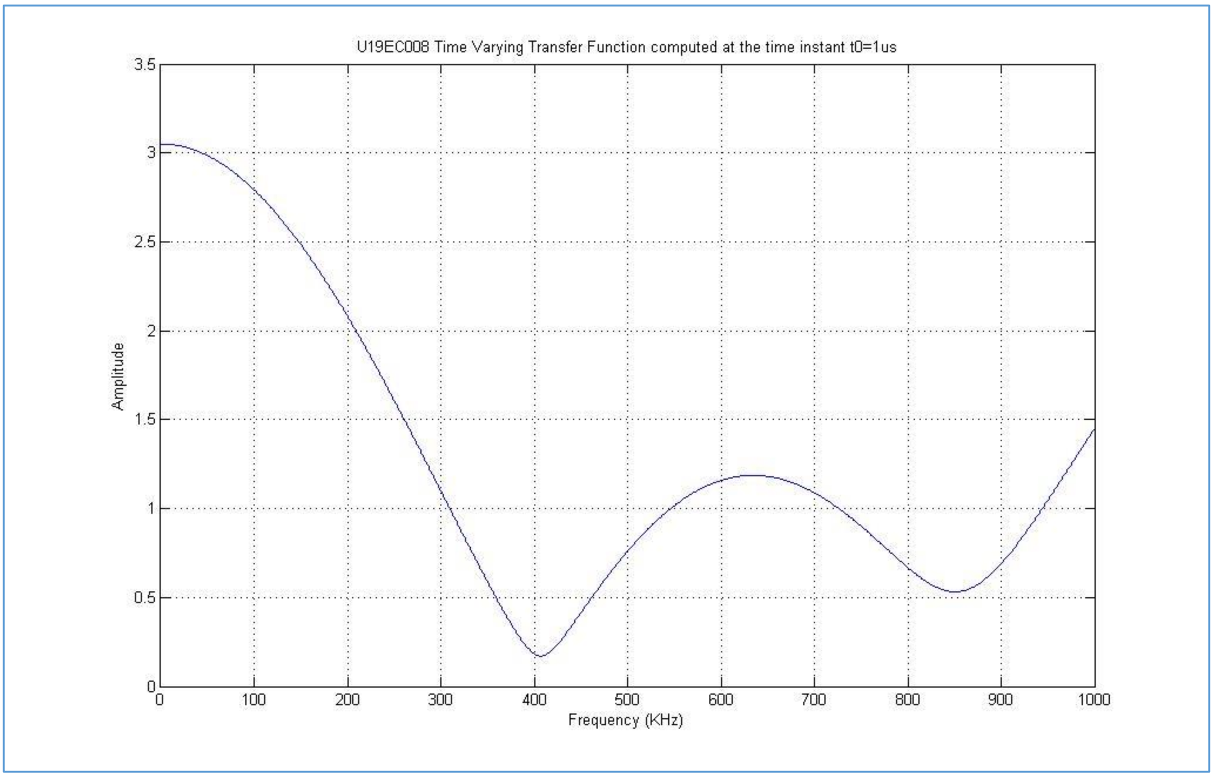


Doppler Spread (Slow Fading)





Delay Spread



CALCULATIONS:

Coherence Time = 1/2D = **73us**
Delay Spread L = **1.6306**
Coherence Frequency = 1/2L= **306KHz**

CONCLUSION:

In this experiment, we observed the effect of Delay spread and Doppler spread for fast fading and slow fading and verified the coherent bandwidth using MATLAB. In doppler spread, how fast the transfer function of the time-varying channel changes with time for a fixed frequency is to be studied. Doppler spread and the coherence time are used for the same. In delay spread, how fast the transfer function of the time-varying channel changes with frequency at a particular time instant is to be studied.

SANJIT ANAND U19EC008

21-02-2022 PRACTICAL ASSIGNMENT - 5

WIRELESS AND MOBILE COMMUNICATION

AIM:

To study CDMA spreading/despreading techniques and apply it on the Communication link in MATLAB.

APPARATUS:

MATLAB Software

THEORY:

Multiple Access Techniques

Multiple access techniques are used to allow a large number of mobile users to share the allocated spectrum in the most efficient manner.

- As the spectrum is limited, so the sharing is required to increase the capacity of cell or over a geographical area by allowing the available bandwidth to be used at the same time by different users.
- And this must be done in a way such that the quality of service doesn't degrade within the existing users. –A cellular system divides any given area into cells where a mobile unit in each cell communicates with a base station.
- The main aim in the cellular system design is to be able to increase the capacity of the channel i.e. to handle as many calls as possible in a given bandwidth with a sufficient level of quality of service.

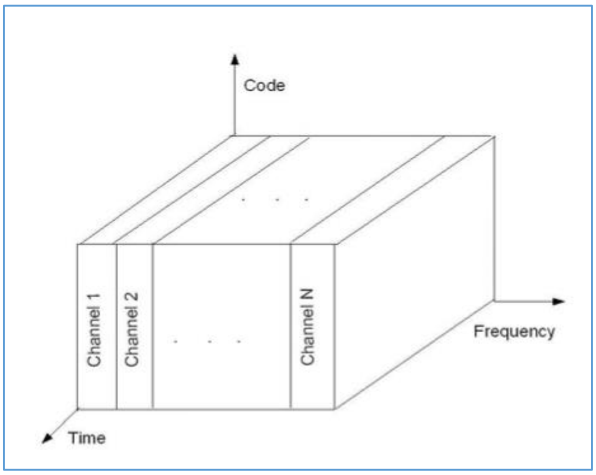
These includes mainly the following:

1. Frequency division multiple-access (FDMA)

2. Time division multiple-access (TDMA)

3. Code division multiple-access (CDMA)

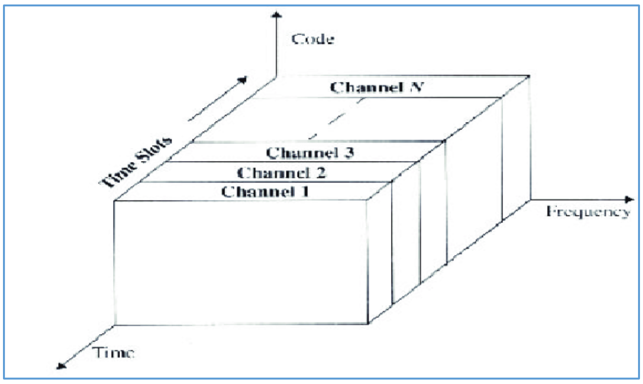
1. Frequency division multiple-access (FDMA)



•Each individual user is assigned a pair of frequencies while making or receiving a call as shown in Figure.

•One frequency is used for downlink and one pair for uplink. This is called frequency division duplexing (FDD).

2. Time division multiple-access (TDMA)

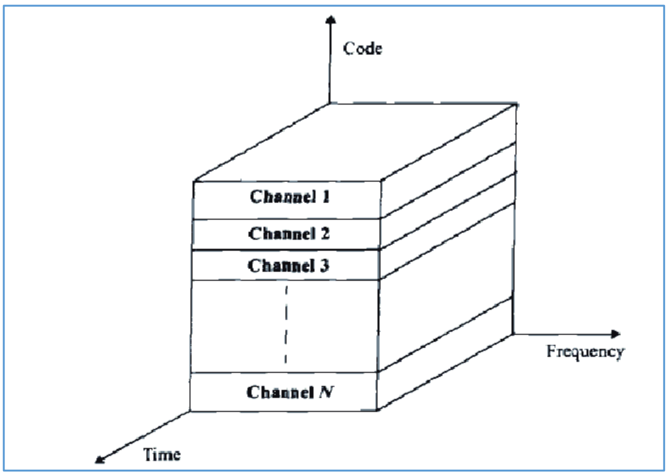


•In digital systems, continuous transmission is not required because users do not use the allotted bandwidth all the time. In such cases, TDMA is a complimentary access technique to FDMA.

•Global Systems for Mobile communications (GSM) uses the TDMA technique.

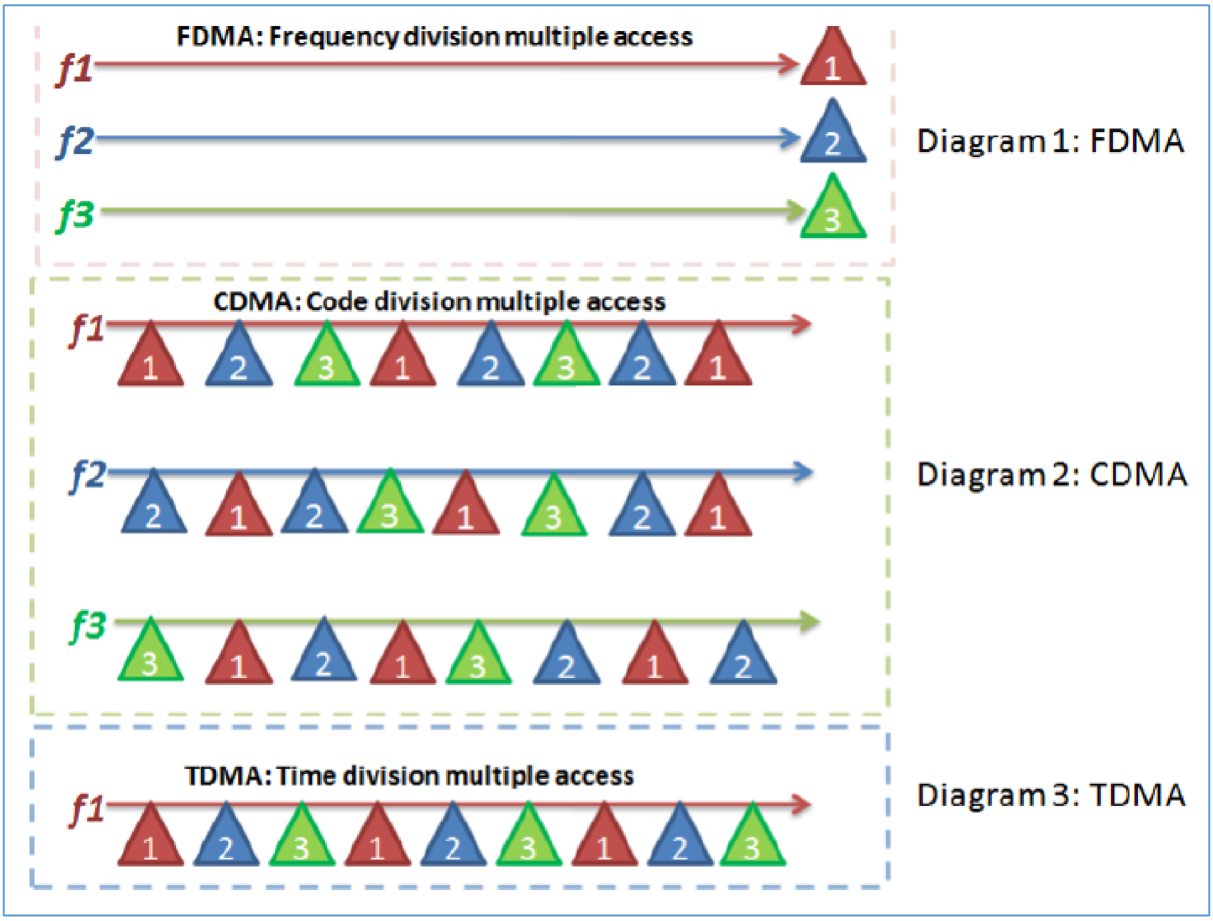
• In TDMA, the entire bandwidth is available to the user but only for a finite period of time. The users are allotted time slots during which they have the entire channel bandwidth at their disposal, as shown in Figure

3. Code division multiple-access (CDMA)



•In CDMA, the same bandwidth is occupied by all the users, however they are all assigned separate codes, which differentiates them from each other shown in Figure

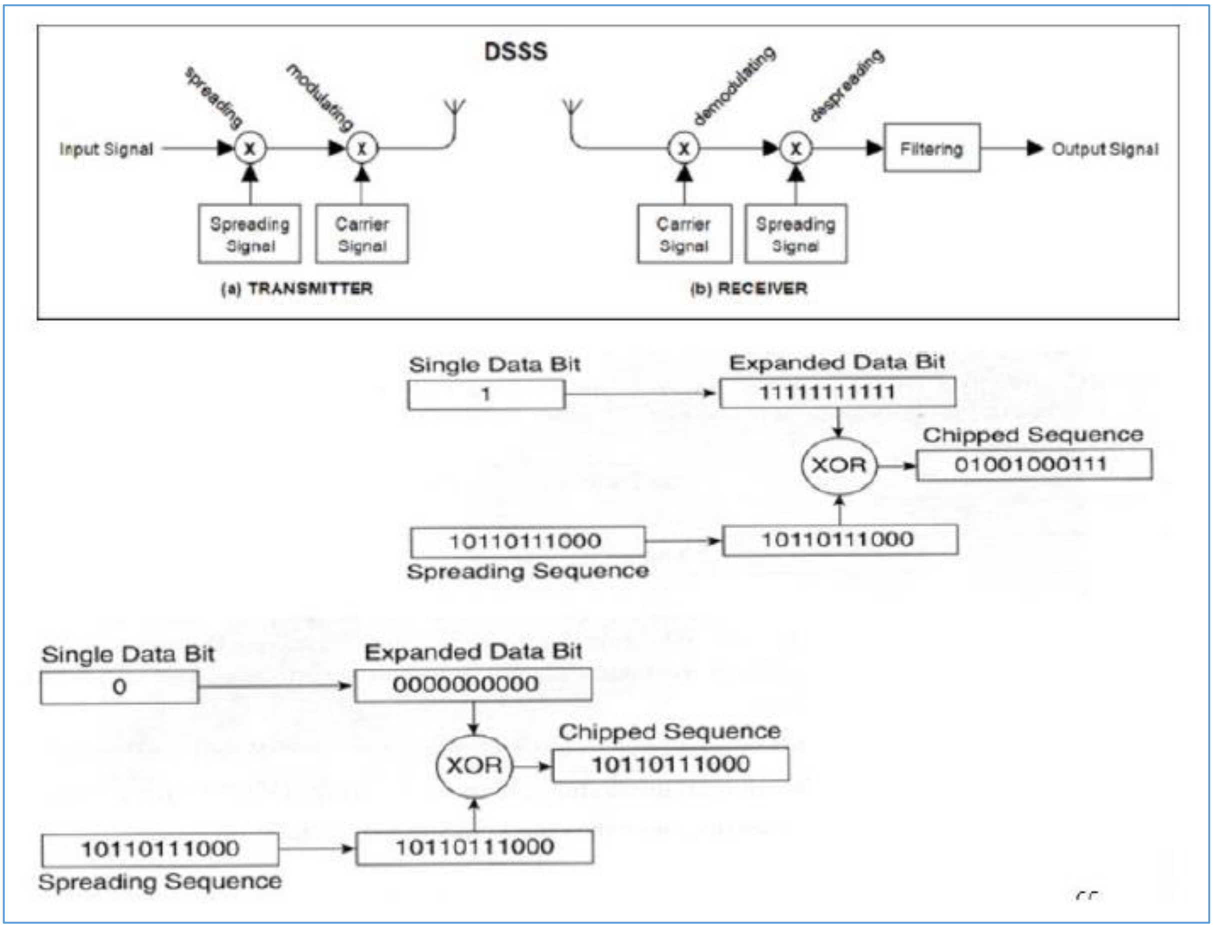
•CDMA utilize a spread spectrum technique in which a spreading signal (which is uncorrelated to the signal and has a large bandwidth) is used to spread the narrow band message signal.

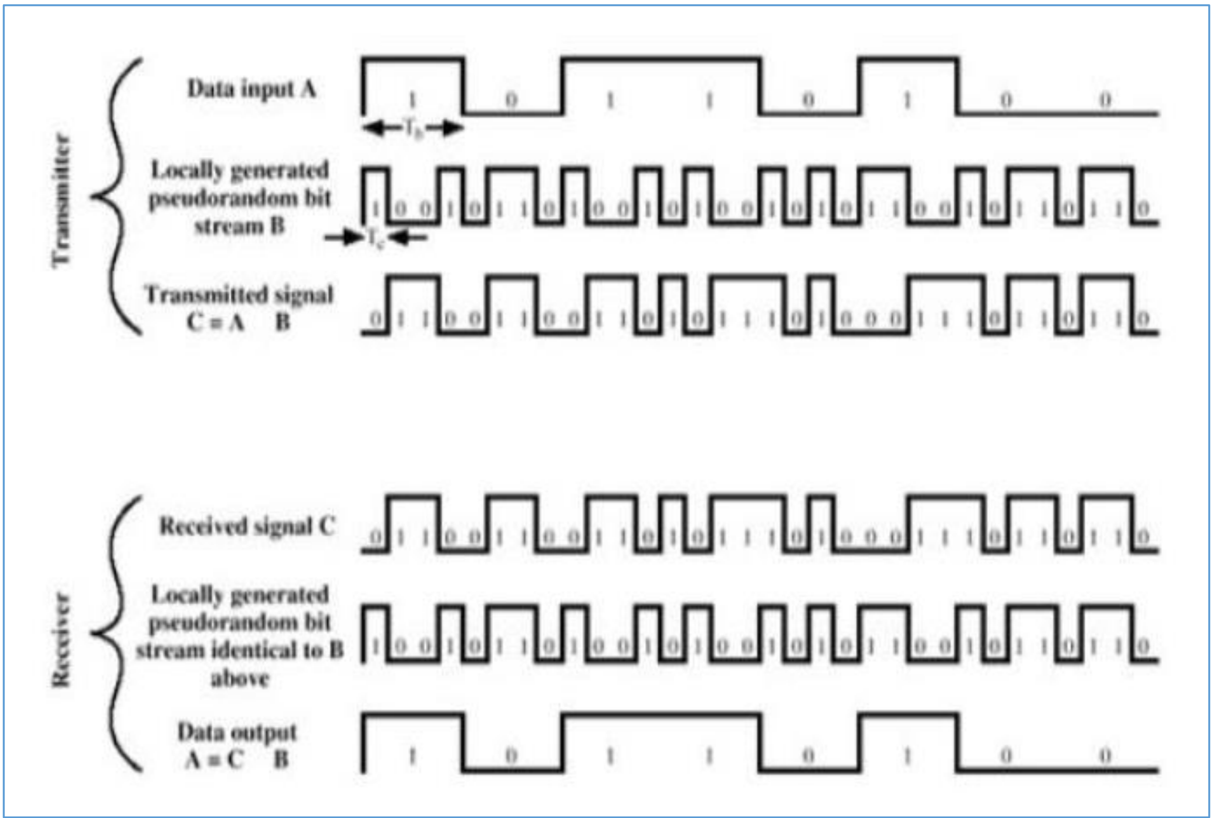


Direct Sequence Spread Spectrum (DSSS)

•In DS-SS, the message signal is multiplied by a Pseudo Random Noise Code.

•Each user is given his own codeword which is orthogonal to the codes of other users and in order to detect the user, the receiver must know the codeword used by the transmitter.





PROCEDURE:

Part 1

- Generate a binary message signal using **rand** function. Take size as (1000,1) initially and later change it as per (1XXX,1), where XXX is your three digit roll No.
- Generate a PN sequence using **commsrc.pn** from communication system tool box of MATLAB.
- Expand the message bit using **repmat**, so as to make it of size [1000, 8]
- Xor(msg, pn) size[1000,8]:
- Convert the vector into row vector: **Spreaded data [8000, 1]**
- Change the binary data into uint8 before modulation
- Choose mod=4, QPSK/QAM
- Add awgn for a single value of SNR
- Demodulate the signal using qamdemod /pskdemod
- Using reshape, convert the serial demodulated data vector to parallel before despreading. [1000, 8]
- Despread the signal by using xor function
  despread_data= xor(demod_data, pn)
- Contract the data to original format by

- msg_rx=round(mean(despread_data,1)); mean(x,1) row vector mean(x,2) column vector
- Example: [1 0 1 1 1 1 1 1] = 7/8 =0.875
- Calculate the BER
  - BER=mean(abs(msg_rx-msg));
  - This is for a single value or SNR.
- Now, vary SNR [-10:1:10]
  - Plot BER vs SNR
  - Output plots [choose msg bits=4]
- Msg signal
- PN sequence
- Spreaded data
- Modulated signal
- Demodulated signal
- Despreaded signal
- Received message signal
- SNR versus BER plot [msg bits=1000]

Part 2

- Choose 3 different message signals of same length using rand function.
- Allocate three PN sequence for each message signal.
- Perform spreading, modulation, demodulation, despreading same as in Part-I for three messages and three
- PN sequence.
- Plot BER vs SNR for three messages.

MATLAB CODE & OUTPUTS:

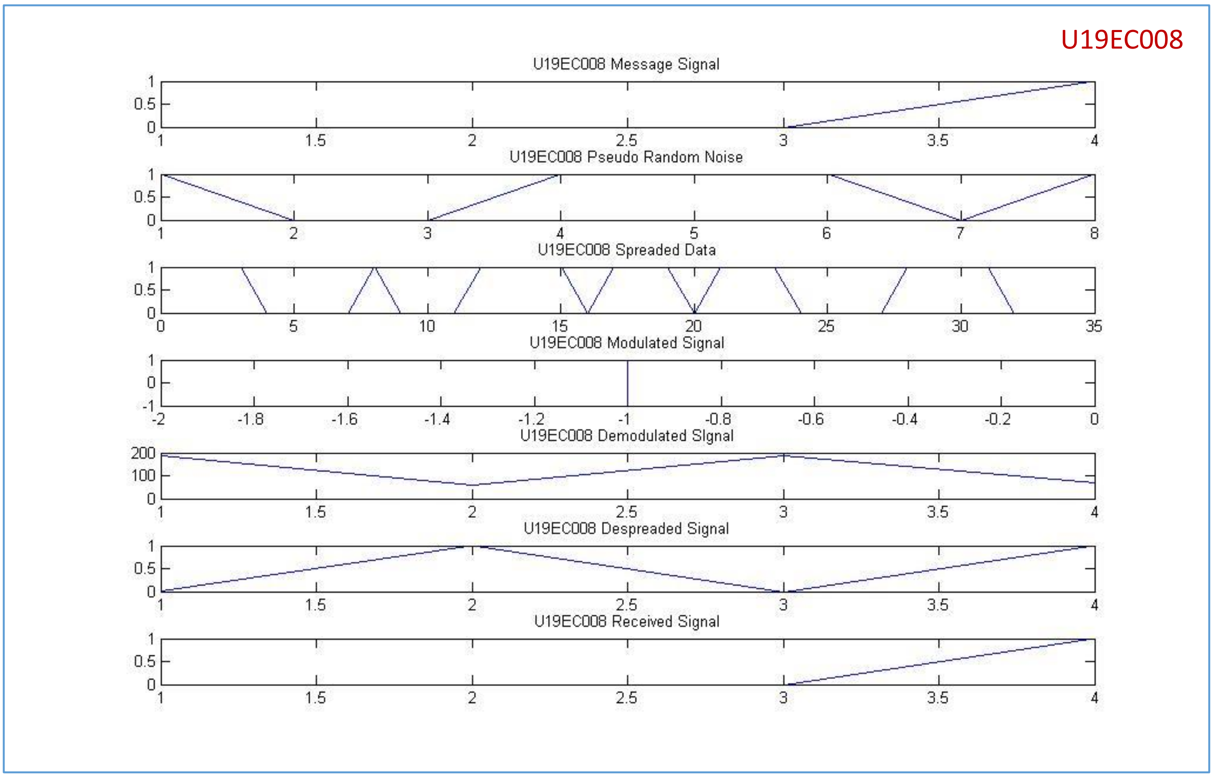
Part 1: a) Modulated and Demodulated Signal

```
% Lab5 U19EC008
% Modulated, Demodulated
clc;
clear all;
close all;

% Generating Transmitted Signal (4x1)
msg = randi([0,1],4,1);

% Repeating message signal
msg_re = repmat(msg,[1,8]);

H = commsrc.pn('GenPoly', [3 2 0], ...
```

```
                'InitialStates',[0 0 1], ...
                'CurrentStates', [0 0 1],...
                'Mask', [0 0 1], ...
                'NumBitsOut', 8);

% Generating PN sequence
pn = generate(H);
rn = pn;
% Repeating PN sequence
pn = repmat(pn', [4, 1]);

% taking xor (Spreaded Data)
x = xor(msg_re,pn);
spread_i = x(:);
spread_i = uint8(spread_i);

snr = 4;
% Modulated Signal
y1 = qammod(spread_i,4);
y2 = awgn(y1,snr);

% Demodulated Signal
y3 = qamdemod(y2,4);
y3 = reshape(y3,[],8);

% Despreaded Signal
despread_d = xor(y3,pn);

% Received Signal
rxsignal = round(mean(despread_d,2));
BER = mean(abs(rxsignal - msg));

% PLOTS
subplot(7,1,1);
plot(msg);
title('U19EC008 Message Signal');

subplot(7,1,2);
plot(rn);
title('U19EC008 Pseudo Random Noise');

subplot(7,1,3);
plot(spread_i);
title('U19EC008 Spreaded Data');

subplot(7,1,4);
plot(y1);
title('U19EC008 Modulated Signal');

subplot(7,1,5);
plot(bi2de(y3));
title('U19EC008 Demodulated SIgnal');

subplot(7,1,6);
plot(despread_d);
title('U19EC008 Despreaded Signal');
```

```
subplot(7,1,7);
plot(rxsignal);
title('U19EC008 Received Signal');
```

Output- Part 1: a) Modulated and Demodulated Signal

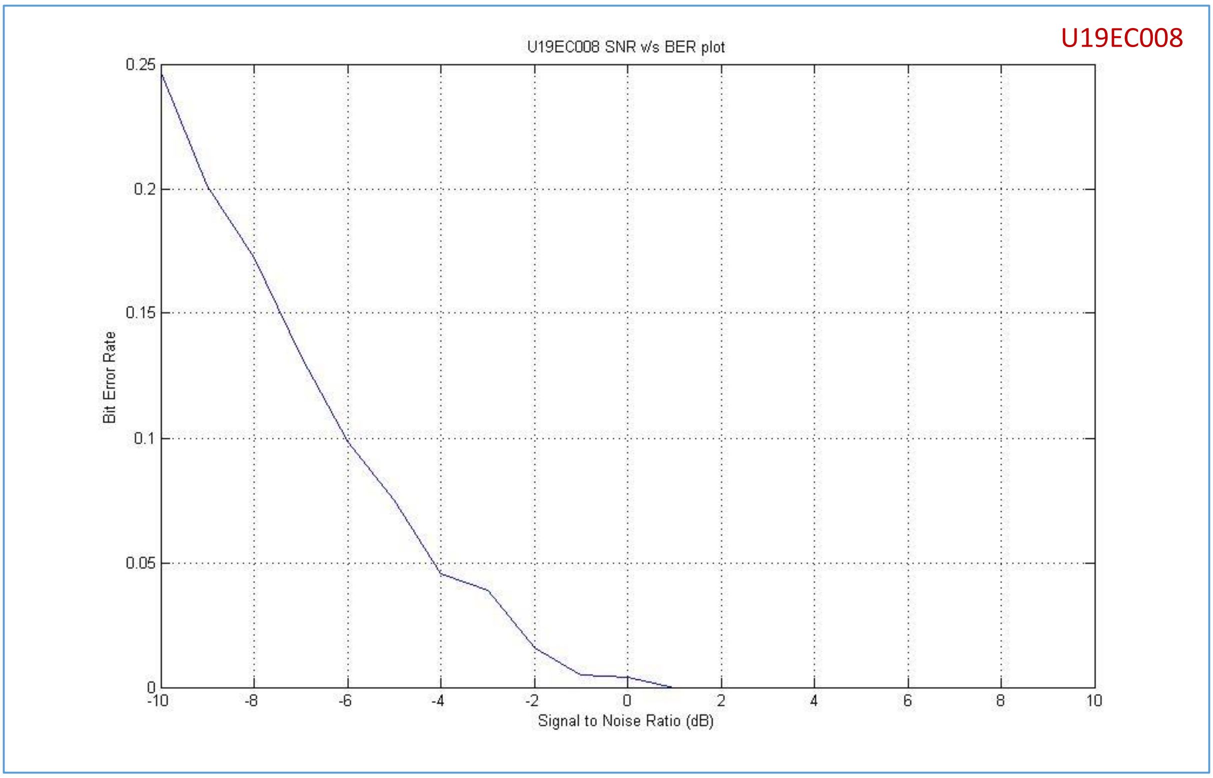


Part 1: b) Signal to Noise Ratio vs Bit Error Rate

```
% Lab5 U19EC008
% Modulated, Demodulated
clc;
clear all;
close all;

% Roll No. = U19EC008
msgbits = 1008;
% Generating Transmitted Signal (1008x1)
msg = randi([0,1],msgbits,1);
% Repeating message signal
msg_re = repmat(msg,[1,8]);

% Generating PN sequence
pn = generate(commsrc.pn('GenPoly', [3 2 0], ...
                         'InitialStates',[0 0 1], ...
                         'CurrentStates', [0 0 1],...
                         'Mask', [0 0 1], ...
                         'NumBitsOut', 8));
pn = reshape(pn,[1,8]);
% Repeating PN sequence
```

```
pn = repmat(pn,msgbits,1);

% taking xor (Spreaded Data)
x = xor(msg_re,pn);
spread_i = x(:);
spread_i = uint8(spread_i);
BER=[];

% For SNR values between -10 to 10
for snr = -10:1:10
    y1 = qammod(spread_i,4);
    y1 = awgn(y1,snr);

    y2 = qamdemod(y1,4);
    y2 = reshape(y2,[msgbits,8]);
    despread_d = xor(y2,pn);
    msg_rx = round(mean(despread_d,2));

    ber = mean(abs(msg_rx - msg));
    BER = [BER ber];
end

% PLOT
snr = -10:1:10;
plot(snr,BER);
title('U19EC008 BER v/s SNR plot');
xlabel('Signal to Noise Ratio (dB)');
ylabel('Bit Error Rate');
grid on;
```

Output - Part 1: b) Signal to Noise Ratio vs Bit Error Rate

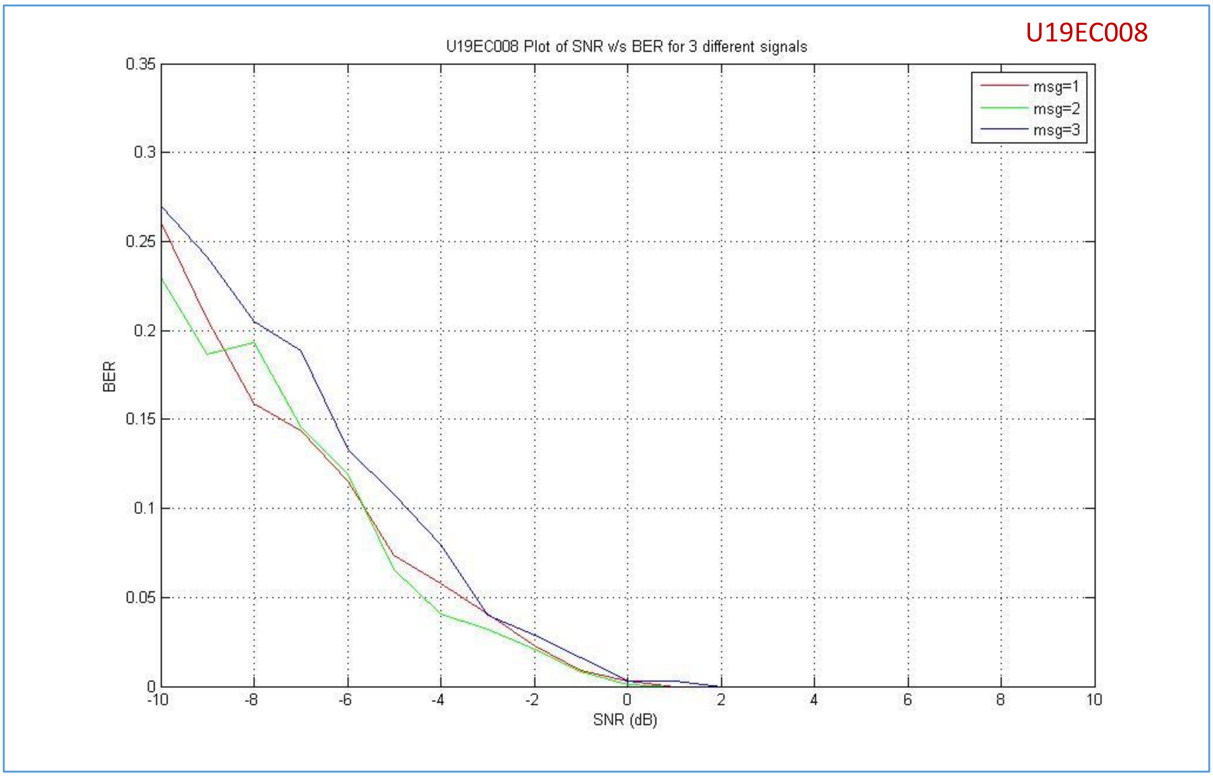


Part 2: SNR vs BER for 3 different signals

```
% Lab5 U19EC008
% SNR vs BER for 3 different Signals
clc;
clear all;
close all;

% Roll Number: U19EC008
msgbits = 1008;
colors = ['r', 'g', 'b'];

for j=1:3
    % Generating Transmitted Signal (1008x1)
    msg = randi([0,1],msgbits,1);
    % Repeating message signal
    msg_re = repmat(msg,[1,8]);

    % Generating PN sequence
    if j==1
        H = commsrc.pn('Genpoly',[3 2 0],...
                       'InitialStates',[0 0 1],...
                       'CurrentStates',[0 0 1],...
                       'Mask',[0 0 1],...
                       'NumBitsOut',8);
        pn = generate(H);
    elseif j==2
        H = commsrc.pn('Genpoly',[4 3 0],...
                       'InitialStates',[0 0 0 1],...
                       'CurrentStates',[0 0 0 1],...
                       'Mask',[0 0 0 1],...
                       'NumBitsOut',8);
        pn = generate(H);
    else
        H = commsrc.pn('Genpoly',[5 3 0],...
                       'InitialStates',[0 0 0 0 1],...
                       'CurrentStates',[0 0 0 0 1],...
                       'Mask',[0 0 0 0 1],...
                       'NumBitsOut',8);
        pn = generate(H);
    end

    pn = reshape(pn,[1,8]);
    % Repeating PN sequence
    pn = repmat(pn,msgbits,1);

    % taking xor (Spreaded Data)
    x = xor(msg_re,pn);
    spread_i = x(:);
    spread_i = uint8(spread_i);

    BER=[];
    for snr = -10:1:10
        y1 = qammod(spread_i,4);
        y1 = awgn(y1,snr);
```

```
        y2 = qamdemod(y1,4);
        y2 = reshape(y2,[msgbits,8]);
        despread_d = xor(y2,pn);
        msg_rx = round(mean(despread_d,2));
        ber = mean(abs(msg_rx - msg));
        BER = [BER ber];
    end
    % PLOTS
    snr = -10:1:10;
    plot(snr,BER,'DisplayName',sprintf('msg=%d',j),
'color',colors(j));
    title('U19EC008 Plot of SNR v/s BER for 3 different signals');
    xlabel('SNR (dB)');
    ylabel('BER');
    grid on;
    legend;
    hold on;
end
```

Output - Part 2: SNR vs BER for 3 different signals



CONCLUSION:

In this experiment, we have implemented CDMA (Code Division Multiple Access) spreading/dispreading techniques and applied it on the communication link using MATLAB and plotted Signal to Noise Ratio (SNR) versus Bit Error Rate (BER) plot. We have also plotted SNR vs BER graph for 3 different signals. We have observed that when the value of SNR is negative, the value of BER is high, but as the value of SNR becomes non-negative, the value of BER tends to zero. We have observed that in DS-SS, the message signal is multiplied by a Pseudo Random Noise Code (PRN). Each user is given his own codeword which is orthogonal to the codes of other users and in order to detect the user, the receiver must know the codeword used by the transmitter.

SANJIT ANAND U19EC008

14-03-2022 PRACTICAL ASSIGNMENT - 6

WIRELESS AND MOBILE COMMUNICATION

AIM:

To Simulate Equalization Techniques using AWGN channel considering input as any random data as well as an Image with the help of MATLAB software. Also compare output of both with and without equalization.

APPARATUS:

MATLAB Software

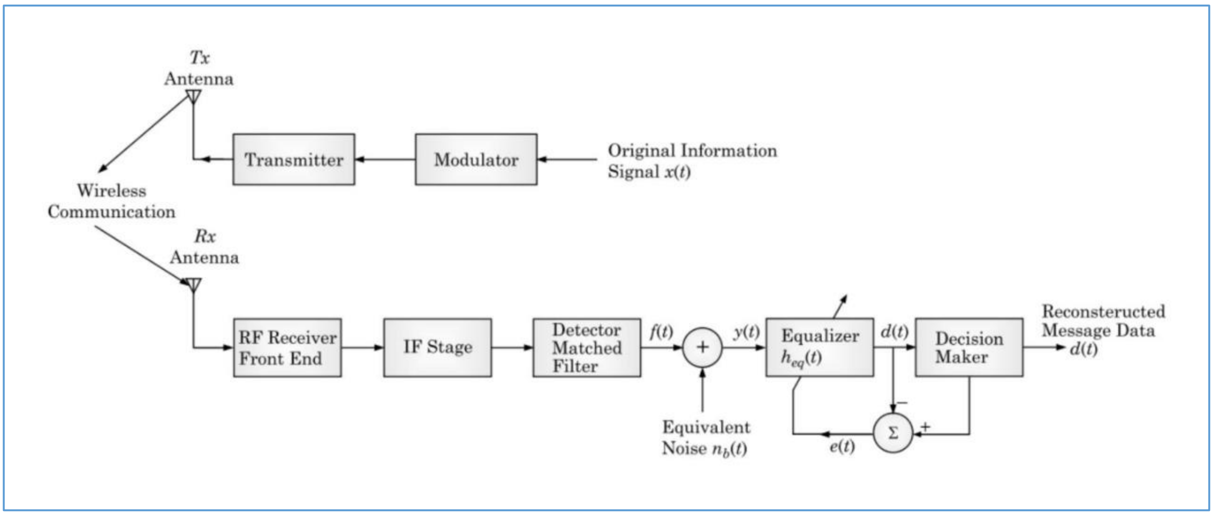
THEORY:

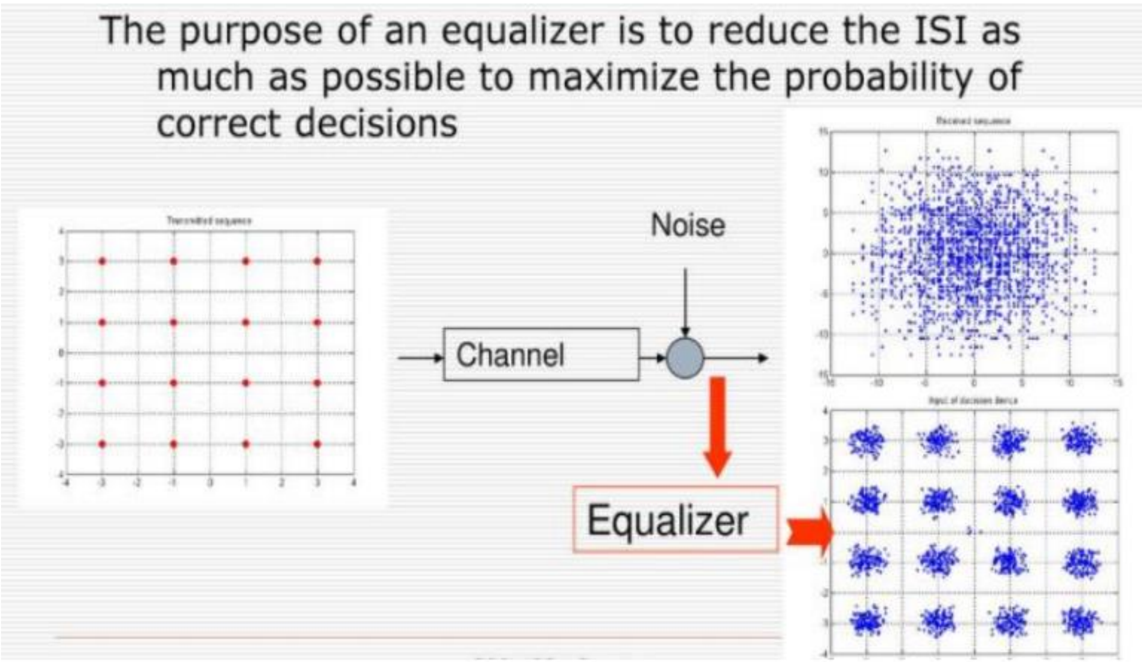
Equalization Techniques

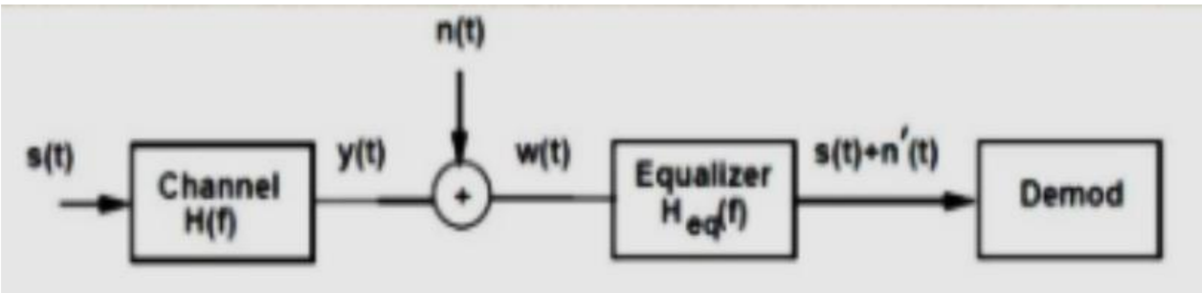
With the growth in wireless communication, there was a requirement to improve the received signal quality because there were a number of channels working at different frequencies which cause signal fading and interference. Thus techniques were evolved to improve received signal of quality. These techniques are 'Equalization and Diversity'. Equalization is used to compensate intersymbol interference (ISI) while Diversity is used to compensate for fading.

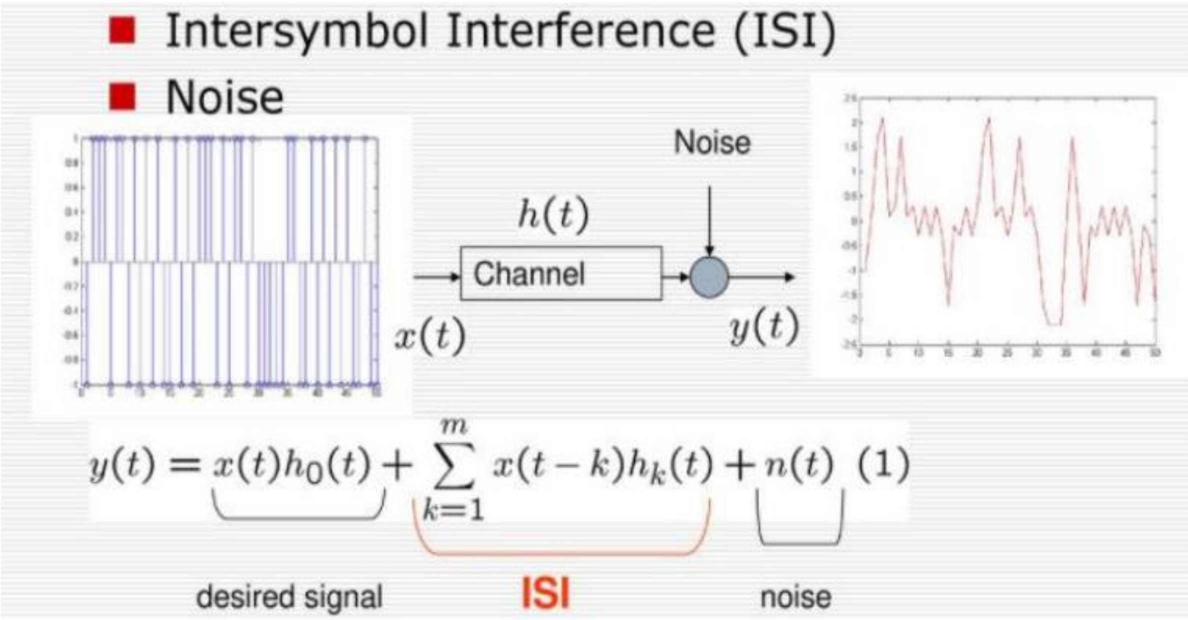
An equalizer is usually implemented at base band or at IF stage in a receiver. Adaptive equalizer works on two operating modes, that are training and tracking. A fixed length training sequence is sent by the transmitter so that the receiver's equalizer may receive the signal for minimum bit error rule (BER). This sequence is a pseudo random binary signal followed by the user data.

Block Diagram of Communication Signal with Equalization









PROCEDURE:

1. Input the random data (Let say 1000 in number) and change into required format.
2. Modulate the data with psk
3. Assume SNR and add AWGN
4. Assume Tau, PdB and Rayleigh channel
5. Realize it with filter function
6. Assume LMS as adaptive algorithm object
7. Construct linear equalizer object (lineareq)
8. Equalize signal using equalizer object (equalize)
9. Demodulate the data
10. Convert it into required format
11. Find BER
12. Compare output for both with and without equalizer (Reapet steps 9-11 for without equalizer)
13. Repeat the above procedure for input as an image

MATLAB CODE & OUTPUTS:

Part 1: Using input as Image

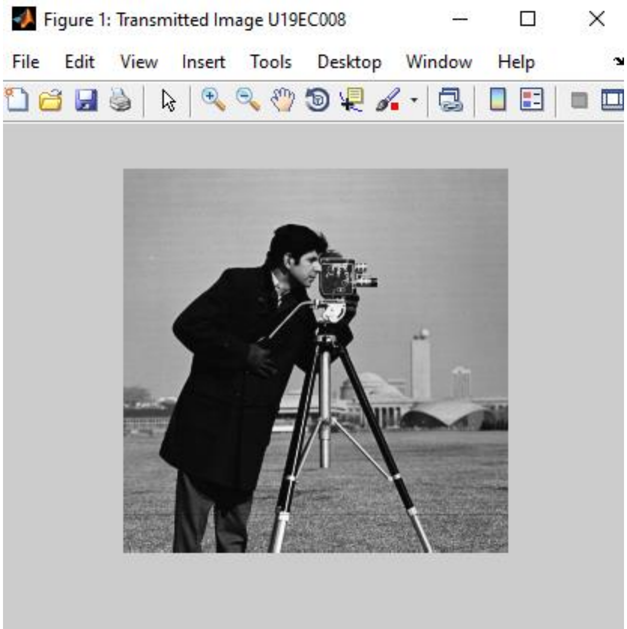
```
% Lab-06 U19EC008
% Equalization, Input as Image

clc;
clear all;
close all;

id = imread('cameraman.tif');
figure('name','Transmitted Image U19EC008');
imshow(id);

ida = id(:);
ib = de2bi(ida);
ib = ib(:);

M = 4;
x = mod(length(ib),log2(M));


Mod=2;
data_vector = reshape(ib, [numel(ib)/Mod Mod]);
msg = bi2de(data_vector);
hMod = comm.QPSKModulator('PhaseOffset',0);

% Modulate using QPSK.
modmsg = step(hMod,msg);

SNR= 40;
modmsg = awgn(modmsg , SNR);
scatterplot(modmsg);
legend('Modulated signal U19EC008');

% Length of training sequence
trainlen = 200;

Tauj = [0.986 0.845 0.237 0.123];
Beta = [-0.1 0 -0.03 0.31];

chan = rayleighchan(1,0,Tauj, Beta);
chanCoeff = chan.AvgPathGaindB + 1i*chan.PathDelays;

% Introduce channel distortion.
filtmsg = filter(chanCoeff,1, modmsg);

% Equalize the received signal.

% Create an equalizer object.
eq1 = lineareq(8, lms(0.01));
release(hMod);

% Set signal constellation.
eq1.SigConst = step(hMod,(0:M-1)')';
```

```
% Equalize.
[symbolest,yd] = equalize(eq1,filtmsg,modmsg(1:trainlen));
scatterplot(filtmsg)
legend('Filtered signal U19EC008')
scatterplot(symbolest)
legend('Equilized signal U19EC008')

% Compute error rates with and without equalization.
hDemod = comm.QPSKDemodulator('PhaseOffset',0);
% Demodulate unequalized signal.
demodmsg_noeq = step(hDemod,filtmsg);
% Demodulate detected signal from equalizer.
demodmsg = step(hDemod,yd);
bidemodmsg = de2bi(demodmsg);

%% OUTPUT IMAGE WITHOUT EQUALIZATION
de = de2bi(demodmsg_noeq);
ber_without_eq = sum(mean(abs(uint8(de) - data_vector)))
%Double to uint8 conversion
r = uint8(de);
r = r(:);

%Reshaping demodulated output
temp = r(1:length(r),:);
t = reshape(temp,[],8);

%Binary to Decimal conversion
k = bi2de(t);
%Reshaping into matrix of the size of image
p = reshape(k,256,256);
%Displaying output
figure('name','Received without Equalisation U19EC008');
imshow(p);

%% OUTPUT IMAGE WITH EQUALIZATION
de = de2bi(demodmsg);
ber_with_eq = sum(mean(abs(uint8(de) - data_vector)))

%Double to uint8 conversion
r = uint8(de);
r = r(:);

%Reshaping demodulated output
temp = r(1:length(r),:);
t = reshape(temp,[],8);

%Binary to Decimal conversion
k = bi2de(t);
%Reshaping into matrix of the size of image
p = reshape(k,256,256);
%Displaying output
figure('name','Received with Equalisation U19EC008');
imshow(p);

disp('Bit Error Rates with Equalisation')
```

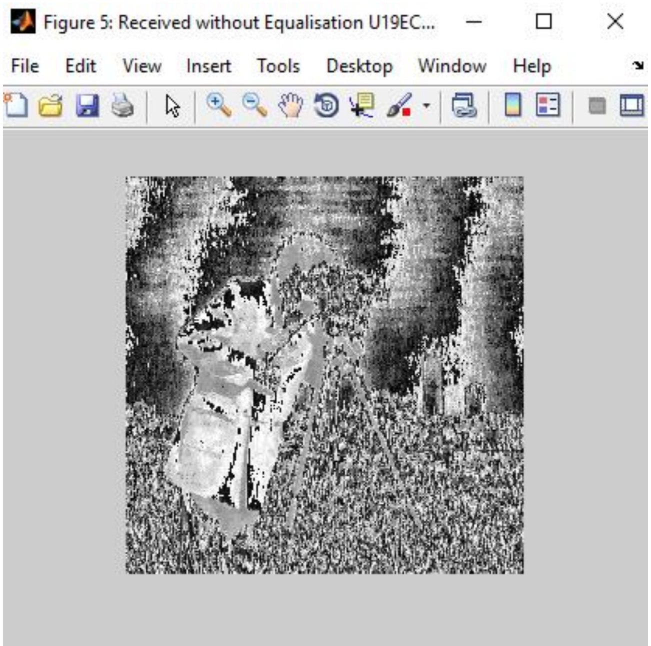
```
disp(ber_with_eq)

disp('Bit Error Rates without Equalisation')
disp(ber_without_eq)
```

Output- Part 1: Using input as Image

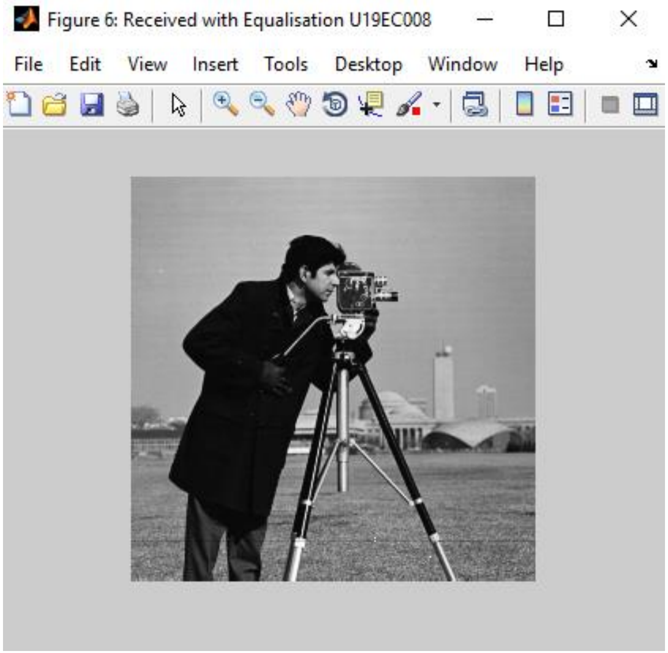
Transmitted Image



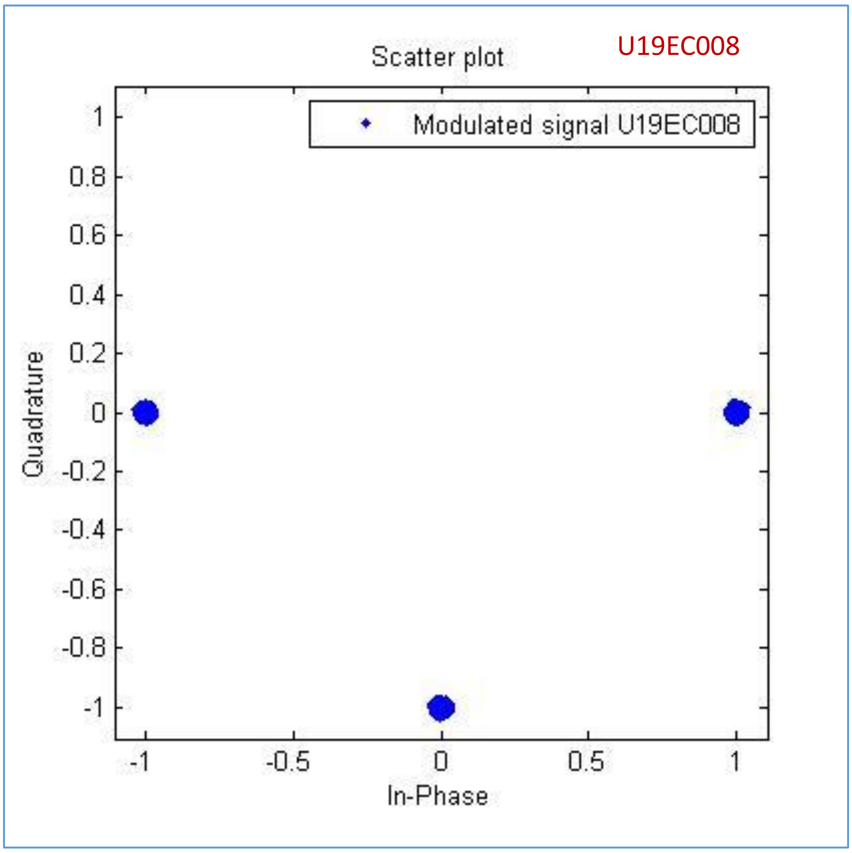
Received Image without Equalisation



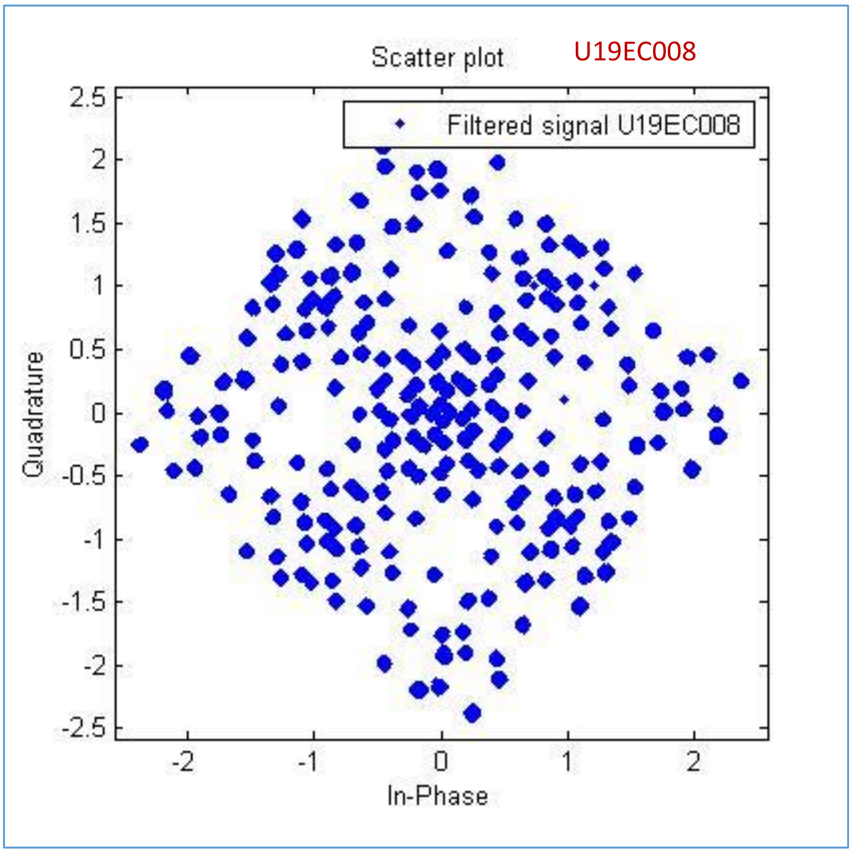
Received Image with Equalisation



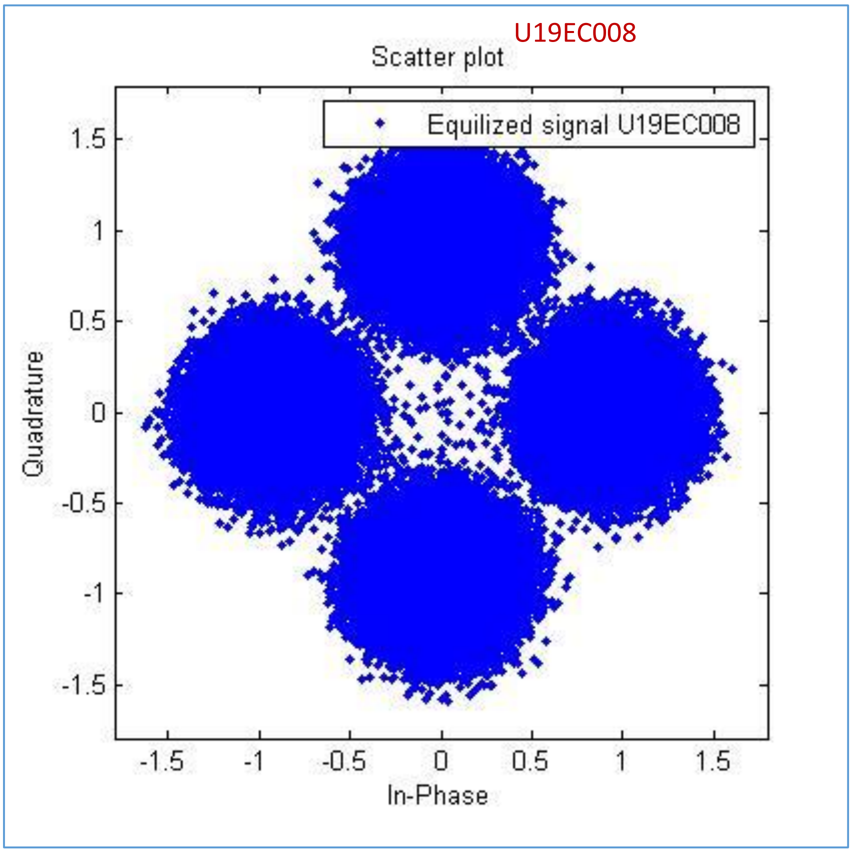
Modulated Signal



Filtered Signal



Equalized Signal



Bit Error Rates (with & without Equalization)

```
Command Window

Bit Error Rates with Equalisation
   2.2125e-04

Bit Error Rates without Equalisation
    0.5738
```

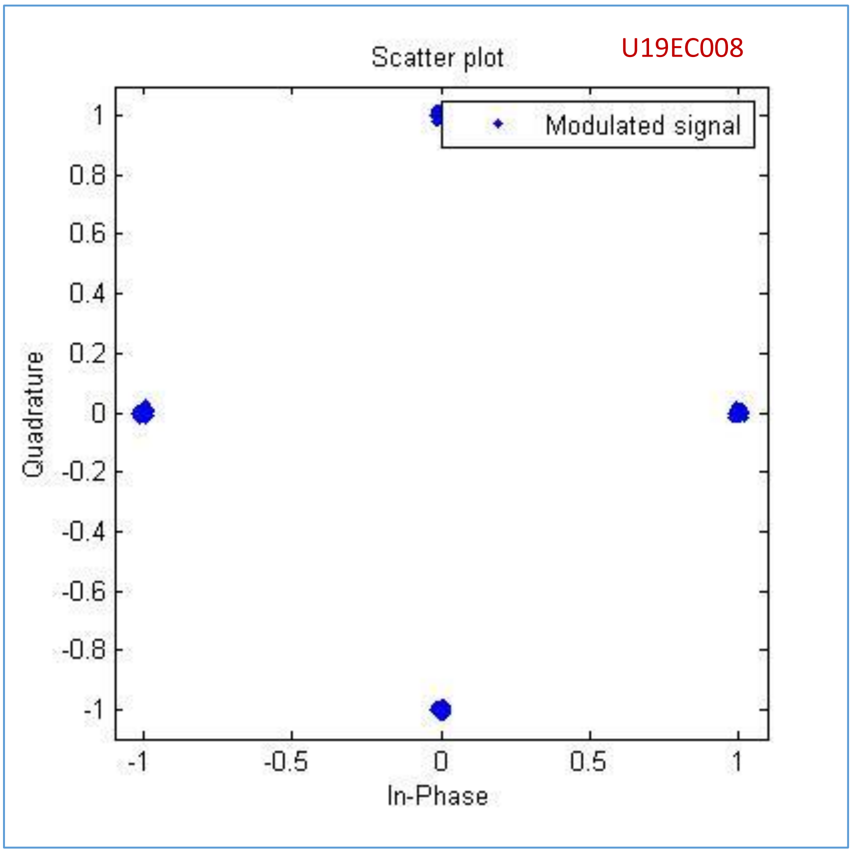
Part 2: Using Random data as input

```
% Lab-06 U19EC008

clc;
clear all;
close all;

M = 4;
% Generating message signal
msg = randi([0 1], 5000, 1);

[n, m] = size(msg);

Mod = 2;
data_vector = reshape(msg, [numel(msg)/Mod Mod]);
msg = bi2de(data_vector);
hMod = comm.QPSKModulator('PhaseOffset',0);

% Modulated signal using QPSK.
k = step(hMod, msg);
SNR = 40;
modmsg = awgn(k,SNR);
scatterplot(modmsg);
legend('Modulated signal');

% Length of training sequence
trainlen = 200;

Tauj = [0.986 0.845 0.237 0.123];
Beta = [-0.1 0 -0.03 0.31];

chan = rayleighchan(1, 0, Tauj, Beta);
chanCoeff = chan.AvgPathGaindB + 1i*chan.PathDelays;

% Introduce channel distortion
filtmsg = filter(chanCoeff, 1, modmsg);

%%% Equalize the received signal

% Create an equalizer object
eq1 = lineareq(8, lms(0.01));
% Set signal constellation
```

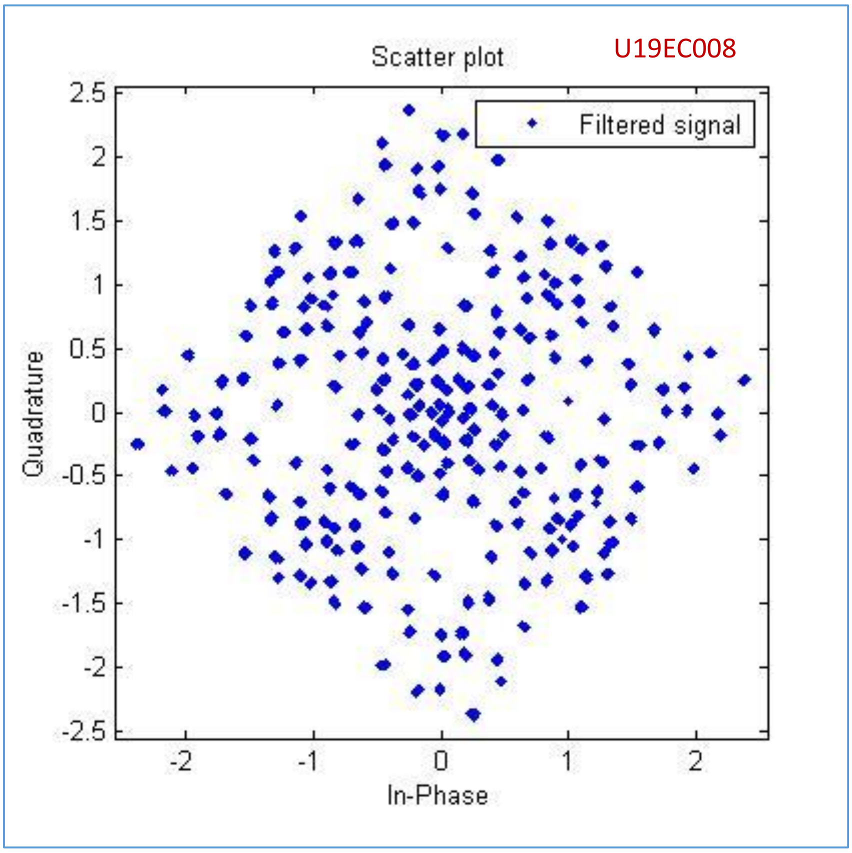
```
eq1.SigConst = step(hMod,(0:M-1)')';
% Equalize.
[symbolest,yd] = equalize(eq1,filtmsg,modmsg(1:trainlen));

scatterplot(filtmsg)
legend('Filtered signal')
scatterplot(symbolest)
legend('Equilized signal')

% Compute error rates with and without equalization.
hDemod = comm.QPSKDemodulator('PhaseOffset',0);

% Demodulate unequalized signal
demodmsg_noeq = step(hDemod,filtmsg);

% Demodulate detected signal from equalizer
demodmsg = step(hDemod,yd);

% Calculating Bit Error Rates

de = de2bi(demodmsg_noeq);
ber_without_eq = sum(mean(abs((de) - data_vector)))

de = de2bi(demodmsg);
ber_with_eq = sum(mean(abs((de) - data_vector)))

disp('Bit Error Rates with Equalisation')
disp(ber_with_eq)

disp('Bit Error Rates without Equalisation')
disp(ber_without_eq)
```

Output - Part 1: Using Random data as input

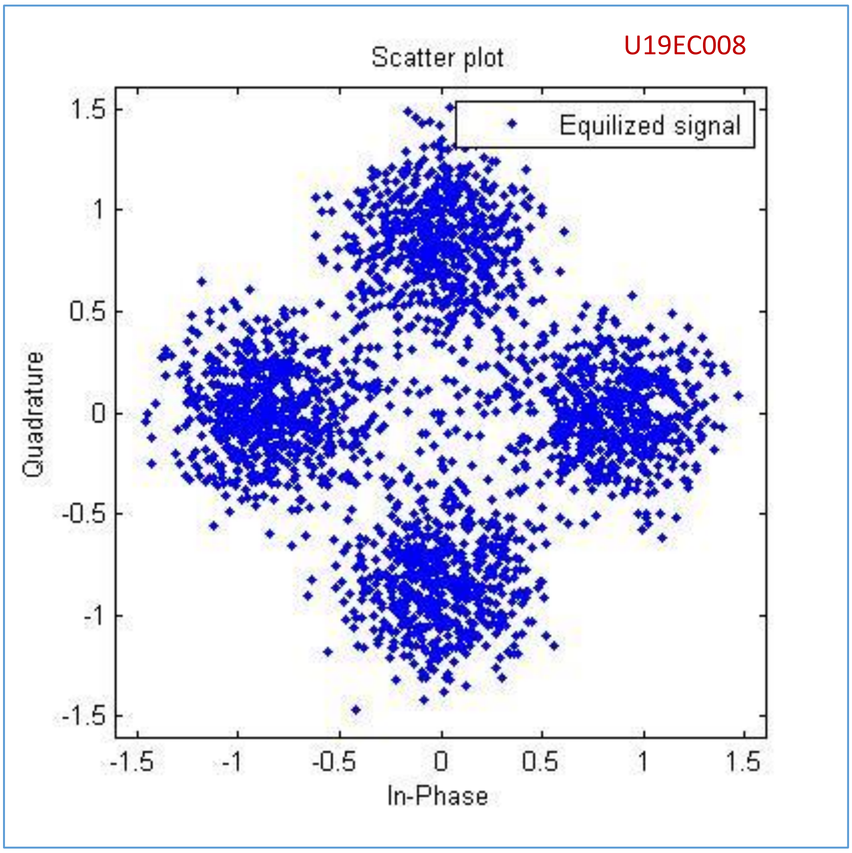
Modulated Signal



Filtered Signal



Equalized Signal



Bit Error Rates (with & without Equalization)

```
Command Window
Bit Error Rates with Equalisation (U19EC008)
    0.0180

Bit Error Rates without Equalisation (U19EC008)
    1.0984

fx >>
```

CONCLUSION:

In this experiment, we have implemented and simulate Equalization Techniques using AWGN channel considering input as any random data as well as an Image with the help of MATLAB software. We have noticed that the equalization helps in reducing noise and Inter Symbol Interference. We have also observed that bit Error Rates observed with equalization is reduced than that of without equalization.

SANJIT ANAND U19EC008

21-03-2022 **PRACTICAL ASSIGNMENT - 7**

WIRELESS AND MOBILE COMMUNICATION

AIM:

To calculate received signal strength as a function of distance separation between transmitter and receiver & understand path loss prediction formula.

Impact of parameters on received signal strength

• Transmitter Power

• Path Loss Exponent

• Carrier Frequency

• Receiver Antenna height

• Transmitter Antenna height

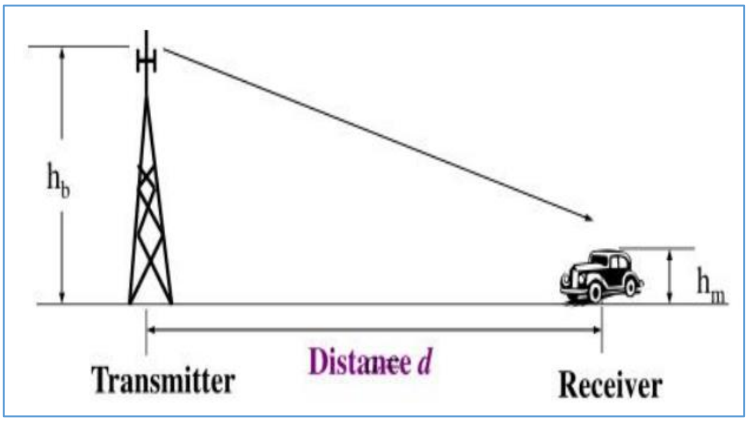
APPARATUS:

Virtual Labs (Based on Java 7)

THEORY:

- The design of a communication system involves selection of values of several parameters.
- Most important parameter is Transmit power.
- In terrestrial mobile communication system, electromagnetic waves propagation is affected by reflection, diffraction and scattering.
- These leads to dynamic variation of signal strength as a function of frequency, distance of separation, antenna height etc.

Free Space Propagation Model



The Friis Free Space Propagation Loss for received power at a distance d is given by:

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}$$

Where

Pt = Transmitter Power.
Pr(d) = Received power at a distance 'd'.
Gt = Transmit antenna power gain.
Gr = Received antenna power gain.
λ = Wavelength.
L ≥ 1 System loss factor not related to propagation, Transmission line, Filter losses, Antenna loss etc.
D= Tx−Rx separation distance.
Pr decrease as square of distance 20 dB/ decade

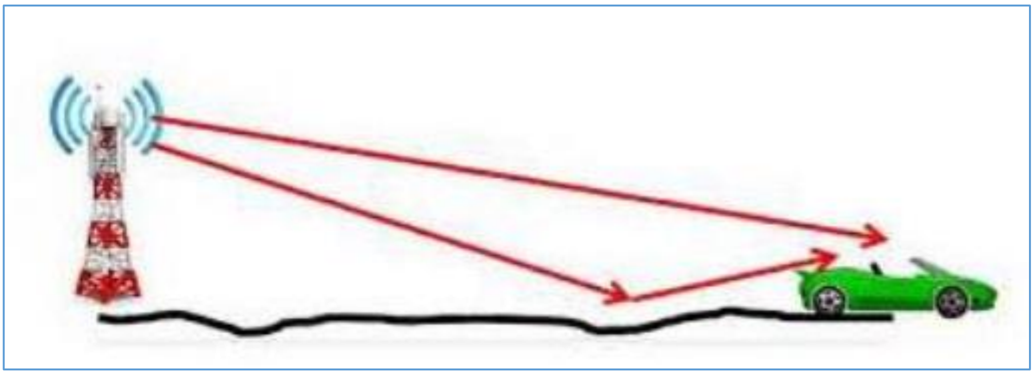
Received Power at a distance d is given by:

$$P_r(d) = P_r(d_0) + 10 * n_p * log\left(\frac{d_0}{d}\right) \quad [Used\ in\ 1A\ and\ 1B]$$

$$P_L(d) = P_L(d_0) + 10 * n_p * log\left(\frac{d}{d_0}\right)$$

Two Ray Propagation Model

A free space propagation model is inaccurate. A useful propagation model (Two Ray Propagation Model) considers both direct path and ground reflected path between transmitter and receiver.



$$P_L(d) = 10 * n_p * log\ (d) + 7.8 - 18\ log(H_{TX})\ - 18\ log(H_{RX}) + 20\ log(f(GHz))$$

Used in 1C, 1D and 1E

Where,

D = Tx – Rx, i.e, Tx and Rx separation distance in metres.

H_{TX} = the transmitter (base station) antenna height in metres.

H_{RX} = the receiver (user terminal) antenna height in metres.

fc = Carrier frequency in GHz.

Environment	Path Loss Exponent
Free space	2
Urban area cellular radio	2.7 to 3.5
Shadowed urban cellular radio	3 to 5
In building line-of-site	1.6 to 1.8
Obstructed in building	4 to 6
Obstructed in factories	2 to 3

PROCEDURE:

1. Open the downloaded JNLP file.
2. Select the experiment from 1A to 1E.
3. Hover the mobile receiver to the desired location (d).
4. Click on take readings which will generate.
5. Calculate the desired value as per the equations.
6. Generate the report after submitting the predicted value to validate with the actual value.

OBSERVATIONS:

For distance d = 100 m

Name: SANJIT U19EC008

REPORT				
1A: Calculation of Received Power	1B: Calculation of Pathloss Exponent	1C: Calculation of Carrier Frequency	1D: Calculation of Receiver Antenna Height	1E: Calculation of BS Antenna Height
Pr(d0): -24.88 dBm	Pr(d0): -18.07 dBm	n: 4.62	fc: 2.0 Ghz	fc: 2.0 Ghz
Dist: 100.0 m	TxPow: 50.0 dBm	TxPow: 50.0 dBm	TxPow: 50.0 dBm	TxPow: 50.0 dBm
d0: 89.0m	Dist: 100.0 m	hTx: 30.0 m	hTx: 30.0 m	n: 3.04
	Pr(d): -27.02 dBm	Dist: 100.0 m	Dist: 100.0 m	Dist: 100.0 m
	d0:61.0m	Pr(d): -30.22 dBm	Pr(d): 12.87 dBm	Pr(d): 1.37 dBm
		hRx: 1.0 m	n: 3.23	hRx: 1.0 m
Pr(Entered):-25.89 dBm	n(Entered):4.17	fc(Entered):2.14 GHz	hRx(Entered):6.558 m	hTx(Entered):27.79 m
Pr(Actual):-25.89 dBm	n(Actual):4.17	fc(Actual):2.14 GHz	hRx(Actual):6.56 m	hTx(Actual):27.79 m

For distance d = 500 m

Name: SANJIT U19EC008

REPORT				
1A: Calculation of Received Power	1B: Calculation of Pathloss Exponent	1C: Calculation of Carrier Frequency	1D: Calculation of Receiver Antenna Height	1E: Calculation of BS Antenna Height
Pr(d0): -25.16 dBm	Pr(d0): -16.91 dBm	n: 3.33	fc: 2.0 Ghz	fc: 2.0 Ghz
Dist: 500.0 m	TxPow: 50.0 dBm	TxPow: 50.0 dBm	TxPow: 50.0 dBm	TxPow: 50.0 dBm
d0: 59.0m	Dist: 500.0 m	hTx: 30.0 m	hTx: 30.0 m	n: 3.67
	Pr(d): -49.82 dBm	Dist: 500.0 m	Dist: 500.0 m	Dist: 500.0 m
	d0:83.0m	Pr(d): -33.34 dBm	Pr(d): -23.22 dBm	Pr(d): -37.73 dBm
		hRx: 1.0 m	n: 3.71	hRx: 1.0 m
Pr(Entered):-43.72 dBm	n(Entered):4.22	fc(Entered):4.1 GHz	hRx(Entered):6.11 m	hTx(Entered):24.93 m
Pr(Actual):-43.72 dBm	n(Actual):4.22	fc(Actual):4.1 GHz	hRx(Actual):6.11 m	hTx(Actual):24.92 m

For distance d = 800 m

Name: SANJIT U19EC008

REPORT				
1A: Calculation of Received Power	1B: Calculation of Pathloss Exponent	1C: Calculation of Carrier Frequency	1D: Calculation of Receiver Antenna Height	1E: Calculation of BS Antenna Height
Pr(d0): -25.5 dBm	Pr(d0): -22.87 dBm	n: 3.72	fc: 2.0 Ghz	fc: 2.0 Ghz
Dist: 800.0 m	TxPow: 50.0 dBm	TxPow: 50.0 dBm	TxPow: 50.0 dBm	TxPow: 50.0 dBm
d0: 63.0m	Dist: 800.0 m	hTx: 30.0 m	hTx: 30.0 m	n: 3.07
	Pr(d): -66.04 dBm	Dist: 800.0 m	Dist: 800.0 m	Dist: 800.0 m
	d0:87.0m	Pr(d): -42.96 dBm	Pr(d): -24.52 dBm	Pr(d): -25.98 dBm
		hRx: 1.0 m	n: 3.32	hRx: 1.0 m
Pr(Entered):-47.57 dBm	n(Entered):4.48	fc(Entered):1.54 GHz	hRx(Entered):3.2 m	hTx(Entered):31.483 m
Pr(Actual):-47.57 dBm	n(Actual):4.48	fc(Actual):1.54 GHz	hRx(Actual):3.2 m	hTx(Actual):31.48 m

For distance d = 1000 m

Name: SANJIT U19EC008

REPORT				
1A: Calculation of Received Power	1B: Calculation of Pathloss Exponent	1C: Calculation of Carrier Frequency	1D: Calculation of Receiver Antenna Height	1E: Calculation of BS Antenna Height
Pr(d0): -20.49 dBm	Pr(d0): -16.88 dBm	n: 3.46	fc: 2.0 Ghz	fc: 2.0 Ghz
Dist: 1000.0 m	TxPow: 50.0 dBm	TxPow: 50.0 dBm	TxPow: 50.0 dBm	TxPow: 50.0 dBm
d0: 98.0m	Dist: 1000.0 m	hTx: 30.0 m	hTx: 30.0 m	n: 4.86
	Pr(d): -59.15 dBm	Dist: 1000.0 m	Dist: 1000.0 m	Dist: 1000.0 m
	d0:62.0m	Pr(d): -48.71 dBm	Pr(d): -29.38 dBm	Pr(d): -82.71 dBm
		hRx: 1.0 m	n: 3.52	hRx: 1.0 m
Pr(Entered):-40.66 dBm	n(Entered):3.5	fc(Entered):4.84 GHz	hRx(Entered):5.59 m	hTx(Entered):31.26 m
Pr(Actual):-40.67 dBm	n(Actual):3.5	fc(Actual):4.84 GHz	hRx(Actual):5.59 m	hTx(Actual):31.27 m

For distance d = 1500 m

Name: SANJIT U19EC008

REPORT				
1A: Calculation of Received Power	1B: Calculation of Pathloss Exponent	1C: Calculation of Carrier Frequency	1D: Calculation of Receiver Antenna Height	1E: Calculation of BS Antenna Height
Pr(d0): -14.94 dBm	Pr(d0): -13.69 dBm	n: 3.84	fc: 2.0 Ghz	fc: 2.0 Ghz
Dist: 1500.0 m	TxPow: 50.0 dBm	TxPow: 50.0 dBm	TxPow: 50.0 dBm	TxPow: 50.0 dBm
d0: 60.0m	Dist: 1500.0 m	hTx: 30.0 m	hTx: 30.0 m	n: 3.58
	Pr(d): -55.21 dBm	Dist: 1500.0 m	Dist: 1500.0 m	Dist: 1500.0 m
	d0:68.0m	Pr(d): -59.32 dBm	Pr(d): -48.26 dBm	Pr(d): -51.87 dBm
		hRx: 1.0 m	n: 3.67	hRx: 1.0 m
Pr(Entered):-42.89 dBm	n(Entered):3.09	fc(Entered):2.03 GHz	hRx(Entered):2.03 m	hTx(Entered):26.62 m
Pr(Actual):-42.9 dBm	n(Actual):3.09	fc(Actual):2.03 GHz	hRx(Actual):2.03 m	hTx(Actual):26.64 m

RESULTS:

PART 1A

Distance (m)	Pr (d) (dBm)	Avg. Pr(d) (dBm)
100	-25.89	-20.344
	-15.04	
	-15.13	
	-26.82	
	-18.84	

PART 1B

Distance (m)	Pr (d) (dBm)	Path Loss Exponent (n)
	-27.02	4.17

100	-24.19	3.45
	-30.77	4.74
	-23.61	4.64
	-19.42	4.27

PART 1C

Distance (m)	Pr (d) (dBm)	Carrier Frequency (Fc) (GHz)
1000	-48.71	4.84
	-59.4	1.16
	-56.92	1.74
	-90.83	3.98
	-36.91	4.42

PART 1D

Distance (m)	Pr (d) (dBm)	Receiver Antenna Height (hRx)(m)
1000	-29.38	5.59
	-62.65	2.15
	-68.25	5.26
	-21.75	6.62
	-36.91	4.42

PART 1E

Distance (m)	Pr (d)	Transmitter Antenna Height (hTx)(m)
1000	-82.71	31.26
	-74.16	22.56
	-60.47	29.1
	-61.09	29.03
	-73.02	34.15

CONCLUSION:

In this experiment, we have simulated the path loss models and calculated path loss components using Visual Lab and verified the calculated and received values. We have calculated the path loss components for various cases in which we are required to find path loss exponent, carrier frequency, transmitter height, and receiver height. We have understood the concept of path loss by using the formulas and by practically finding and validating the values.

SANJIT ANAND U19EC008

21-03-2022 PRACTICAL ASSIGNMENT - 8

WIRELESS AND MOBILE COMMUNICATION

AIM:

To understand the cellular frequency reuse concept by:

1. Finding the co-channel cells for a particular cell.
2. Finding the cell clusters within certain geographic area.

APPARATUS:

Virtual Labs (Based on Java 7)

THEORY:

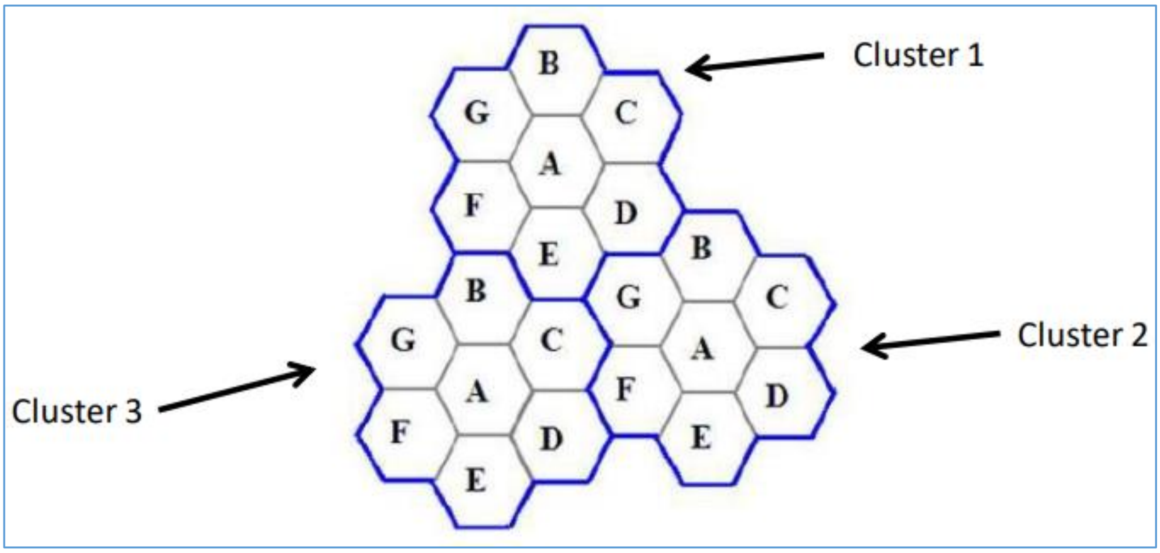
Frequency Reuse

- In mobile communication systems radio resource unit (Channel) is assigned to a user in order to support a call.
- In Mobile Communication System, we have limited spectrum. Thus the number of users who can be supported in a wireless system is highly limited.
- In order to support a large no. of users within a limited spectrum in a region the concept of frequency re-use is used.
- In term of cellular systems, the same frequency can used by two base stations which a sufficient spaced apart. In this way the same frequency gets reused by two or more different base station different users simultaneously.
- Now it is important to select the set of base stations which will use the same set of radio resources/ channel of frequencies or technically the co-channel cells.
- In this context the minimum adjacent set of cells which use different frequencies is called a **cluster**.

Cell Cluster

Considering a cellular system that has a total of S duplex radio channels. If each cell is allocated a group of k channels ($k < S$) and if the S channels are divided among N cells, then,

$$S = kN$$

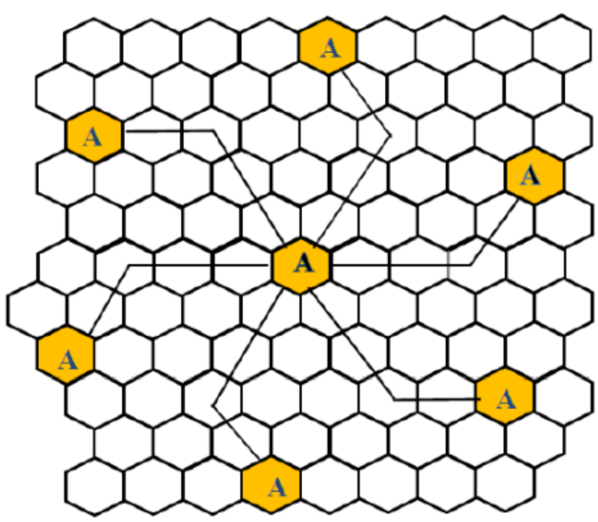


Co-Channel Cells

A larger cluster size causes the ratio between the cell radius and the distance between co-channel cells to decrease reducing co-channel interference. The value of N is a function of how much interference a mobile or base station can tolerate while maintaining a sufficient quality of communications. Since each hexagonal cell has six equidistant neighbours and the line joining the centres of any cell and each of its neighbours are separated by multiples of 60 degrees, only certain cluster sizes and cell layouts are possible. To connect without gaps between adjacent cells, the geometry of hexagons is such that the numbers of cells per cluster, N, can only have values that satisfy,

$$N = i^2 + ij + j^2$$

N = 19 (i.e., i = 3, j = 2)



PROCEDURE:

Part A: Finding co-channel cells for a particular cell

1. Open the downloaded JNLP file of the experiment.
2. Select the values of cell radius, i and j.
3. Click on show cells
4. Mark the correct co-channel cells
5. Generate the report after submitting to validate with the actual cells.

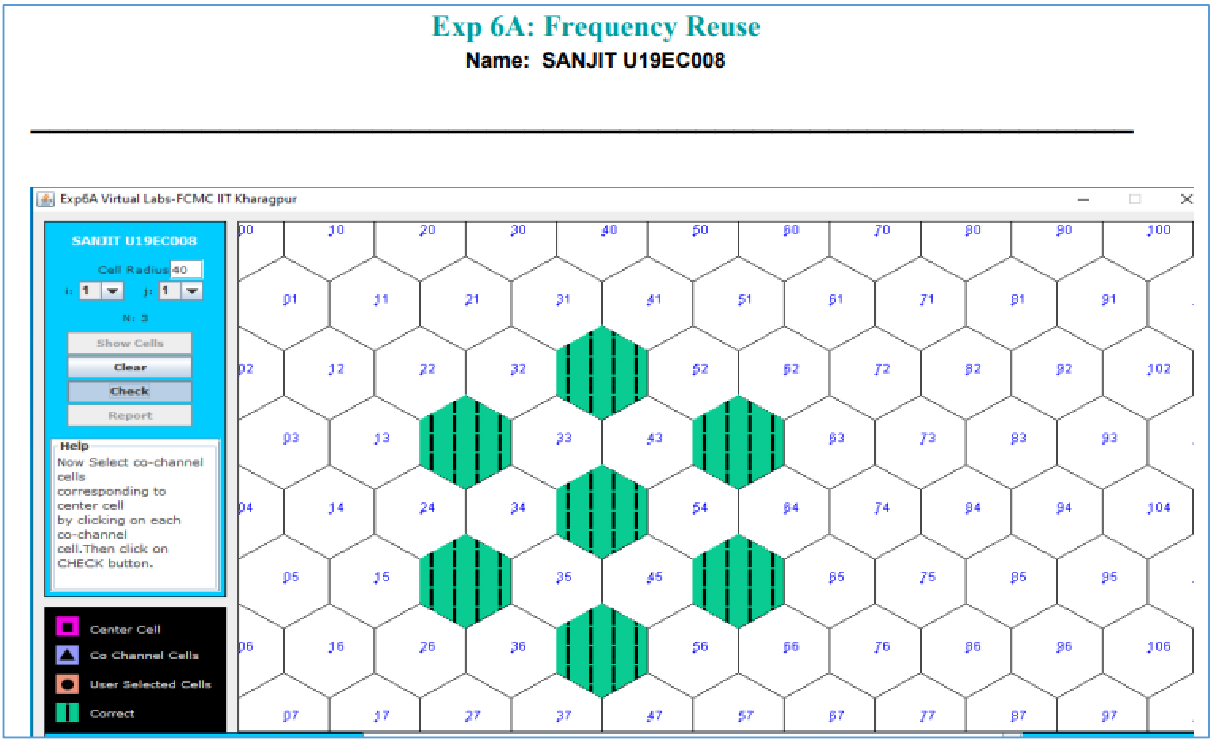
Part B: Finding the cell clusters within certain geographic area.

1. Open the downloaded JNLP file of the experiment.
2. Select the values of cell radius and N(cluster size).
3. Click on show cells
4. Mark the correct cluster cells.
5. Generate the report after submitting to validate with the actual cells.

OBSERVATIONS:

Part A: Finding co-channel cells for a particular cell

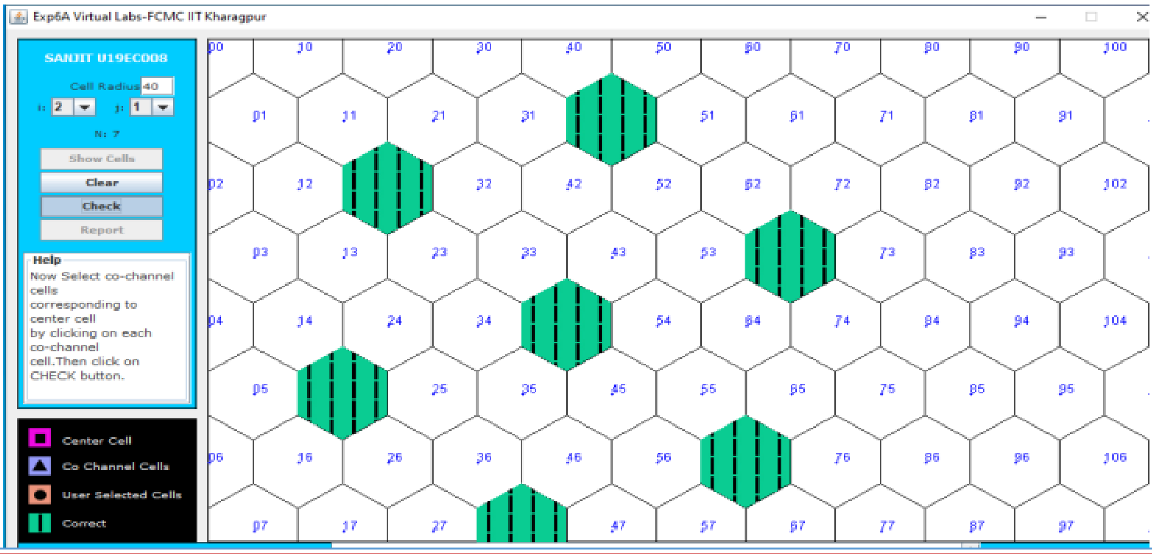
For i=1, j =1



For i=2, j =1

Exp 6A: Frequency Reuse

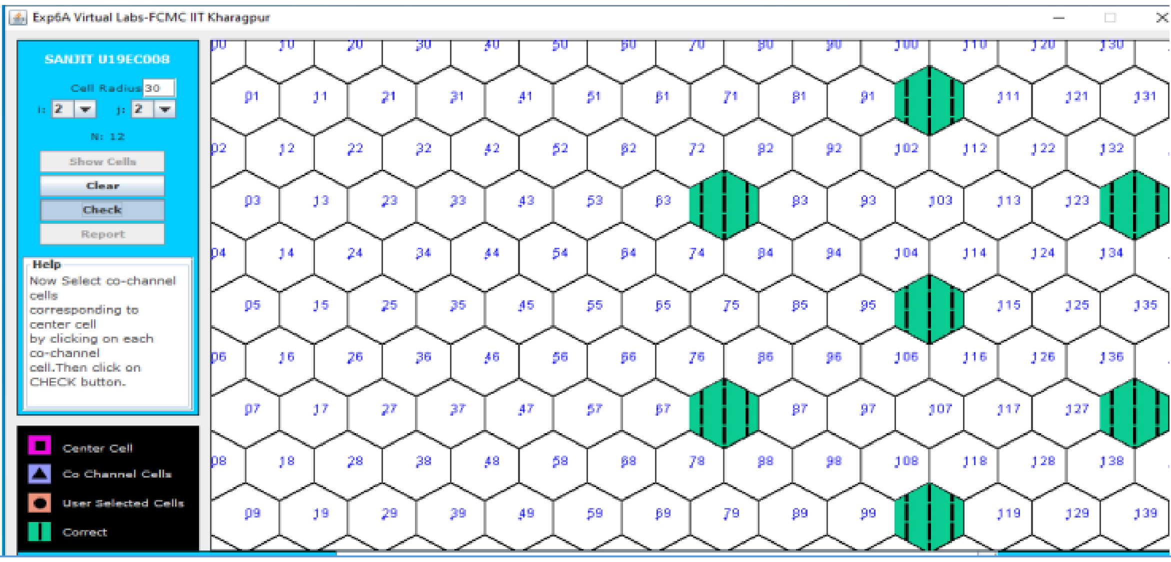
Name: SANJIT U19EC008



For i=2, j =2

Exp 6A: Frequency Reuse

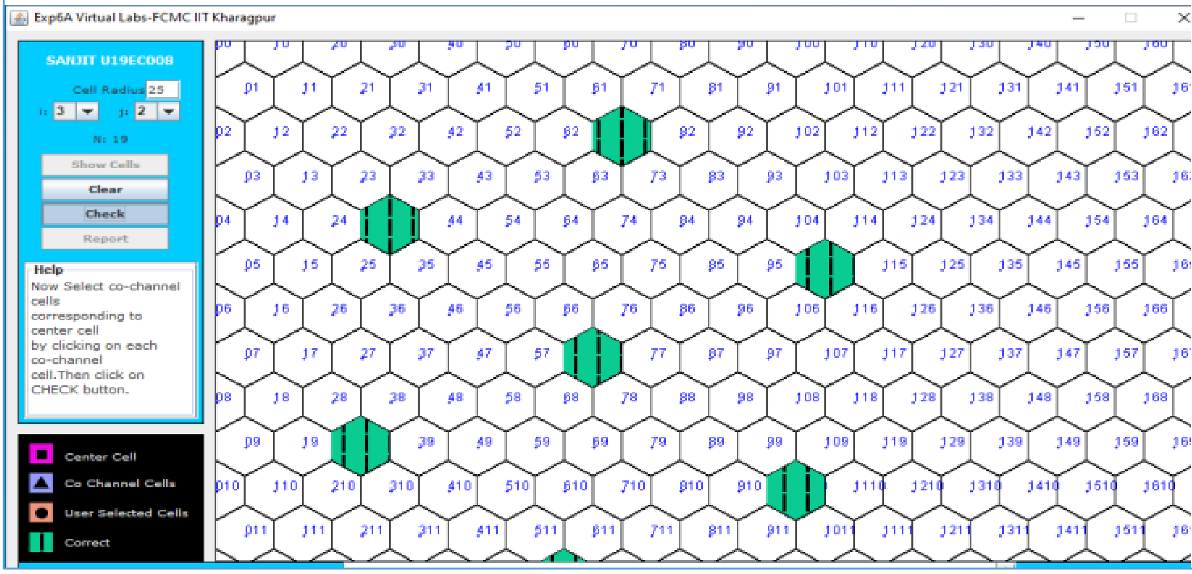
Name: SANJIT U19EC008



For i=3, j =2

Exp 6A: Frequency Reuse

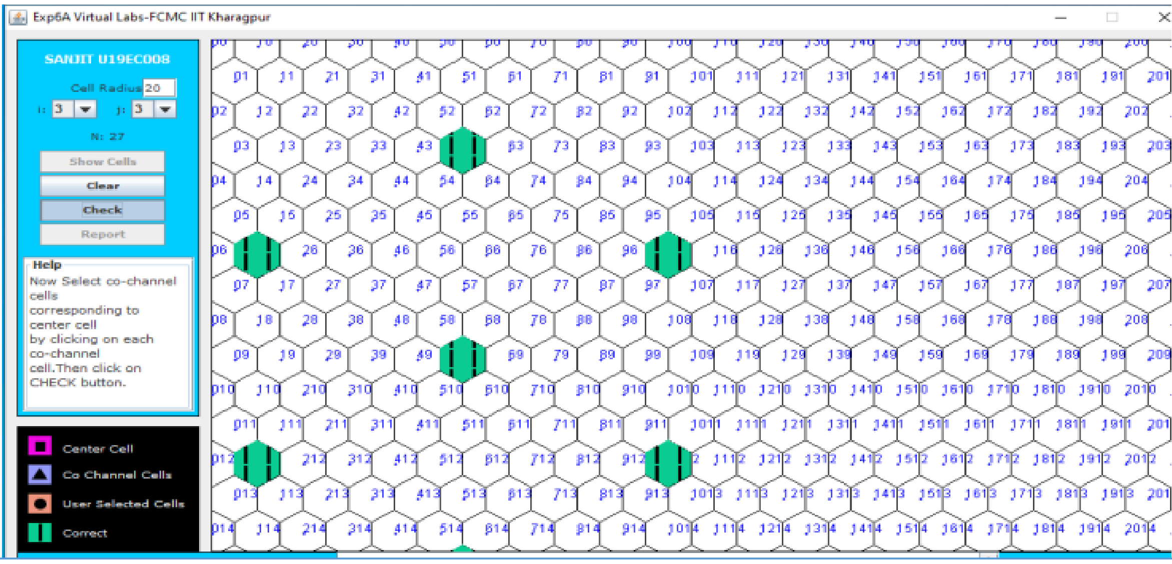
Name: SANJIT U19EC008



For i=3, j =3

Exp 6A: Frequency Reuse

Name: SANJIT U19EC008

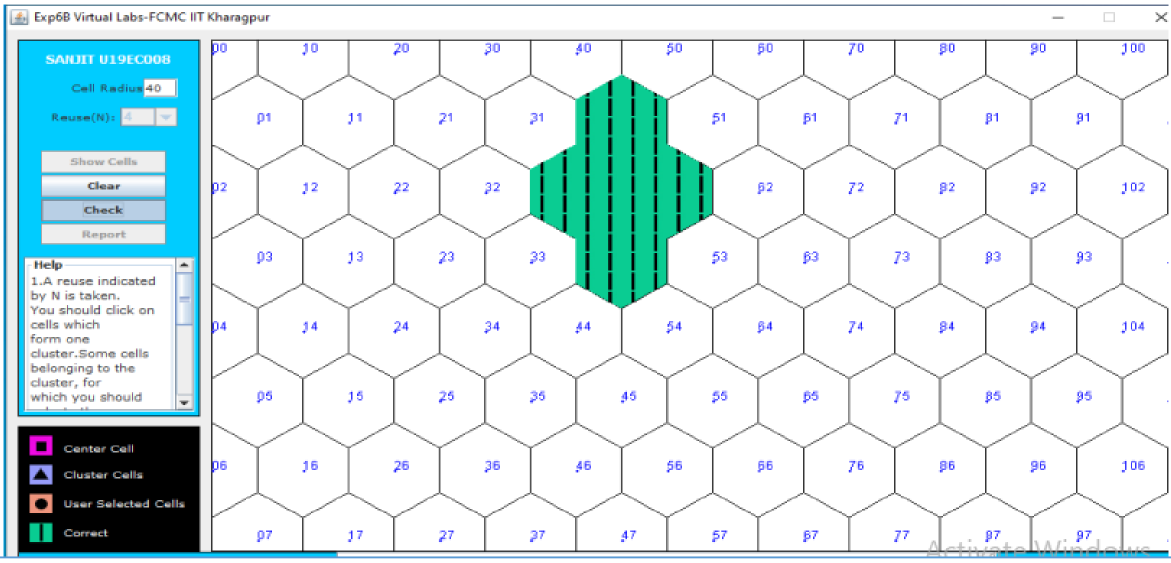


Part B: Finding the cell clusters within certain geographic area.

For N = 4

Exp 6B: Frequency Reuse

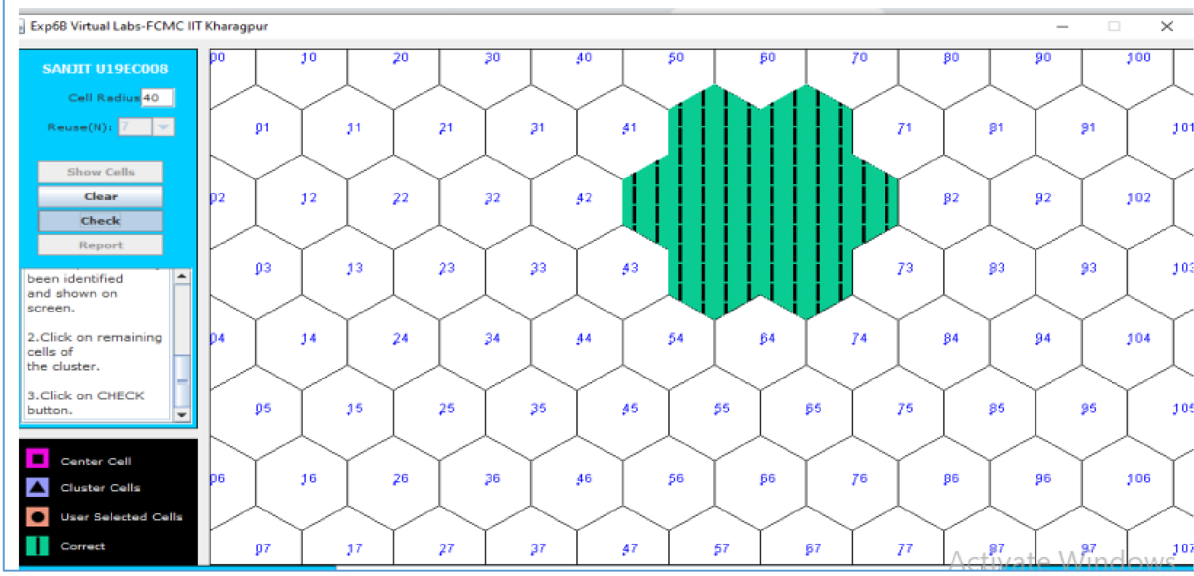
Name: SANJIT U19EC008



For N = 7

Exp 6B: Frequency Reuse

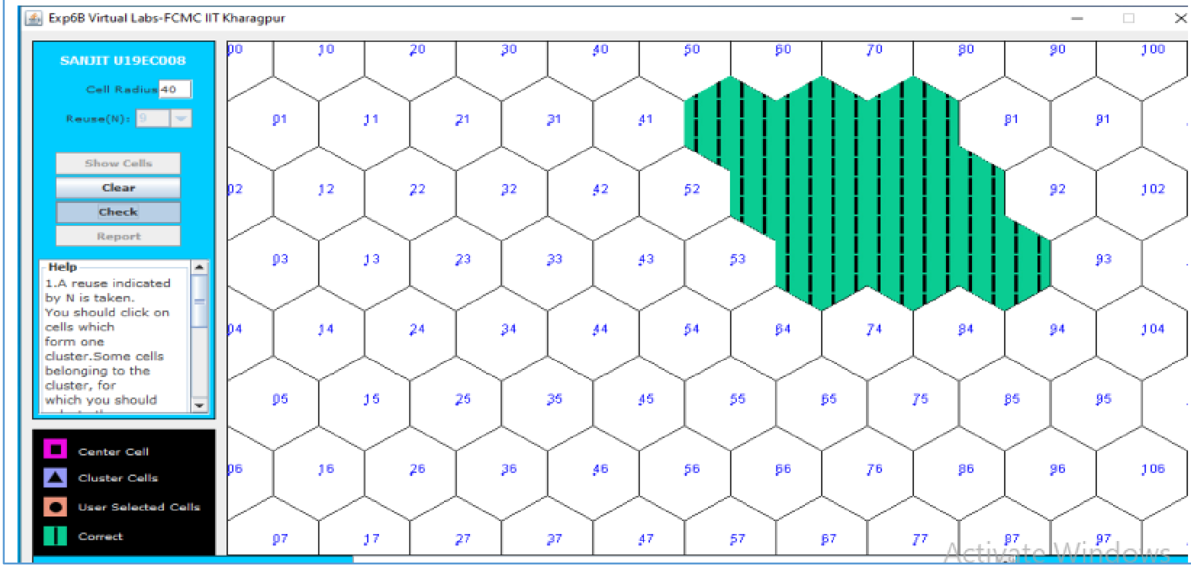
Name: SANJIT U19EC008



For N = 9

Exp 6B: Frequency Reuse

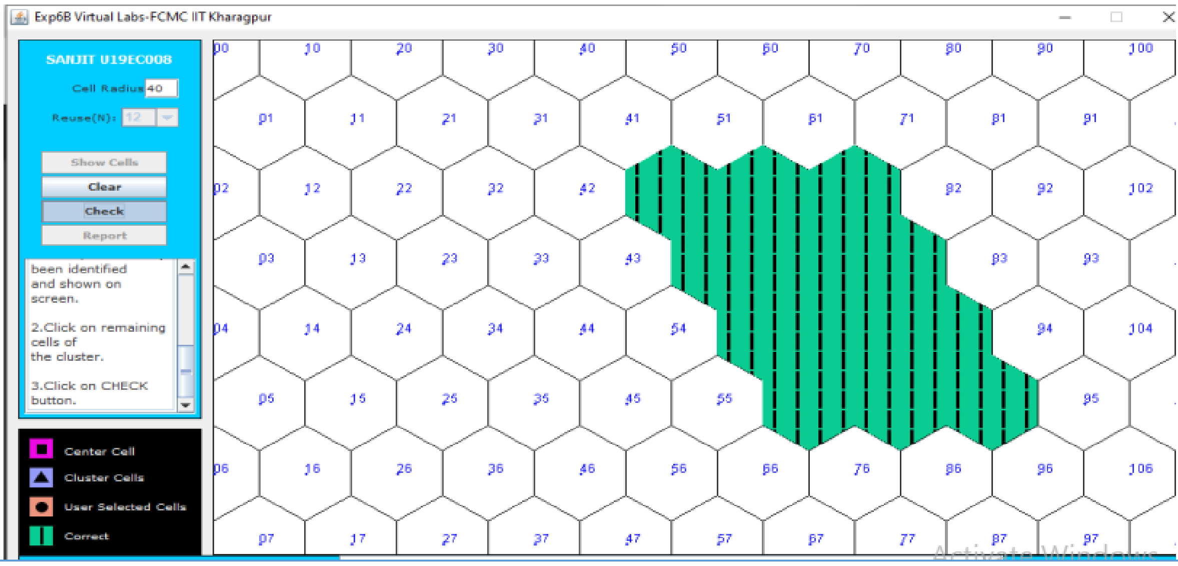
Name: SANJIT U19EC008



For N = 12

Exp 6B: Frequency Reuse

Name: SANJIT U19EC008



CONCLUSION:

In this experiment, we have implemented and simulated the co-channel and cell cluster in Visual Lab and obtained the outputs based on cells we have selected and validated with the correct output thus learning the concept of frequency reuse, co-channel cells and cell clusters.

Experiment 9

Aim: To Perform QPSK and QAM modulation Techniques using ScienTech Trainer Kit.

Experiment:-9.1

Objective: To Perform QPSK modulation Technique using ScienTech Trainer Kit.

Apparatus Required:

ST 2112 QAM Trainer
KitCRO
CRO Probes

Procedure:

1. Ensure the following initial conditions on ST2112 trainer: Power supply and SW3, SW5, SW6, SW7, SW9 should be in the OFF mode.
2. Switch on the power supply.
3. Connect Test point TP6 on Channel 1 & TP7 on Channel 2 of Oscilloscope; you will observe 1 KHz sine & cosine wave.
4. Set I & Q Channel data with the help of DIP switch SW5, SW6, SW7. As there are 24 bits data available on the trainer so, first bit is I bit then second bit is Q bit then third bit is C bit. But in this experiment you have to use I bit & Q bit so you can select combination according to your requirement.

For example:

SW5=11000010
SW6=01001010
SW7=00100010

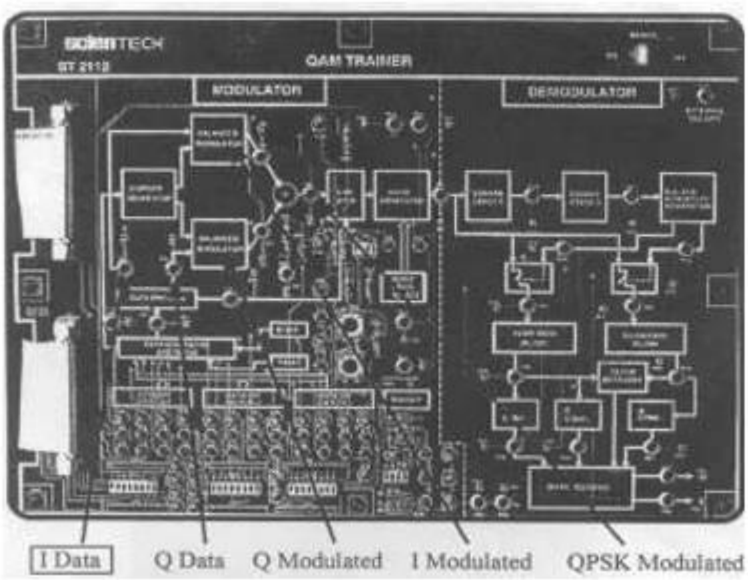


Fig 1. QPSK (Quadrature phase shift keying)

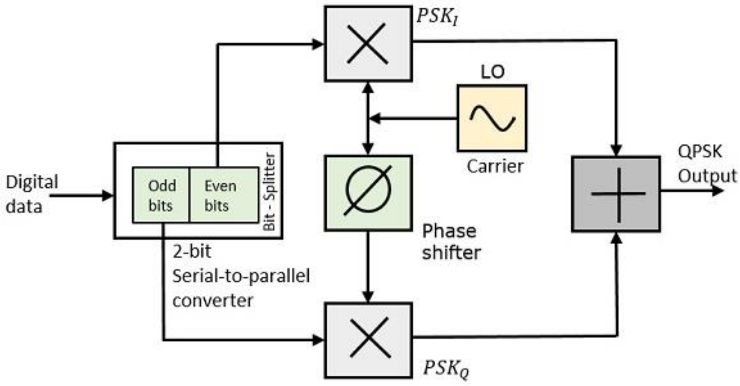
5. Switch ON all the DIP switches on SW3.
6. Now press SW8 which is reset switch then press SW4 which is start.
7. Now connect Channel 1 of Oscilloscope to TP2 & Channel 2 to TP1, you can observe Clock & Data which you have set.
8. Now to observe QPSK modulated signal with respect to data connect Channel 1 to TP1 & Channel 2 to TP8. You can observe QPSK modulated signal with respect to

data.

9. Turn OFF the power.

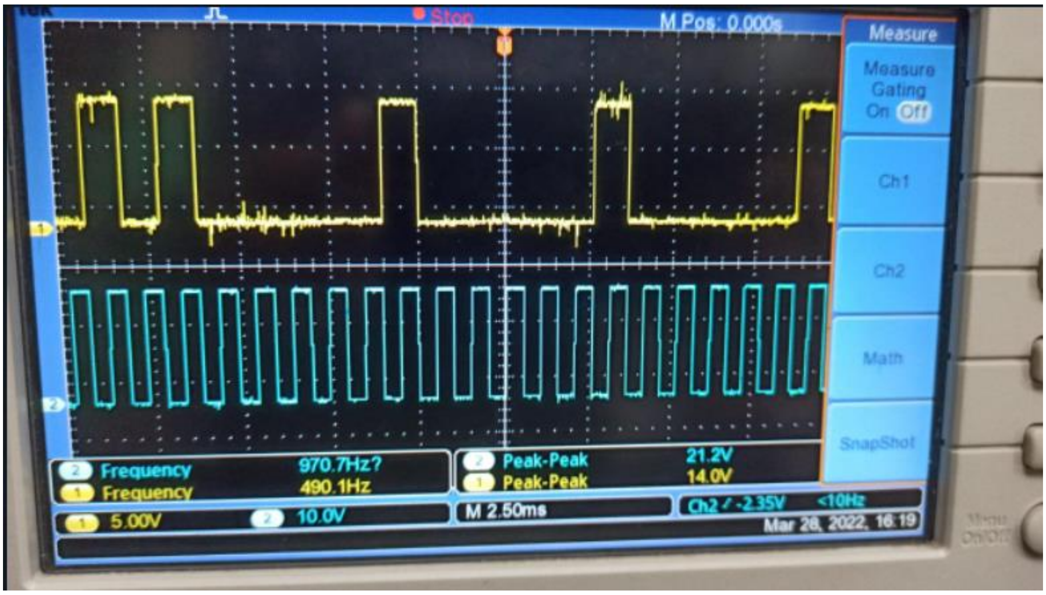
   Two 1 KHz sine carriers, shifted between them of 90°, are separately applied to two balanced modulators. The data (signals I and Q) reach the two modulators from the Dibit generator. Each modulator provides the direct sine-wave when the data signal is to low level (bit "0"), the inverted sine-wave (shifted of 180°) when the bit is "1". By adding the two outputs you get a 1 KHz sine signal, which phase can take 4 different values separated of 90° between them.

Block Diagram:

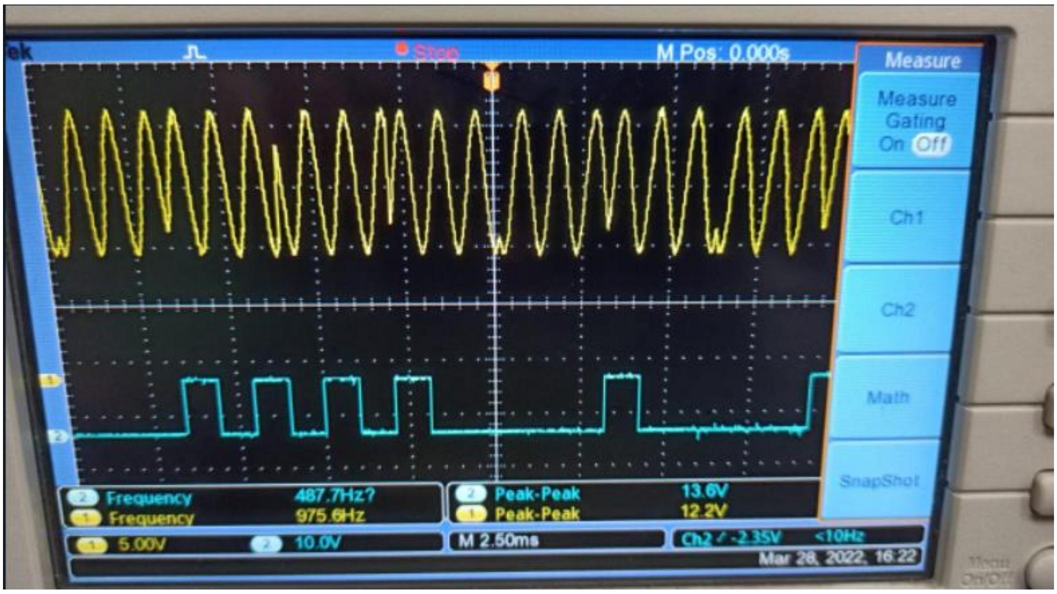


Waveforms & Constellation Diagram:

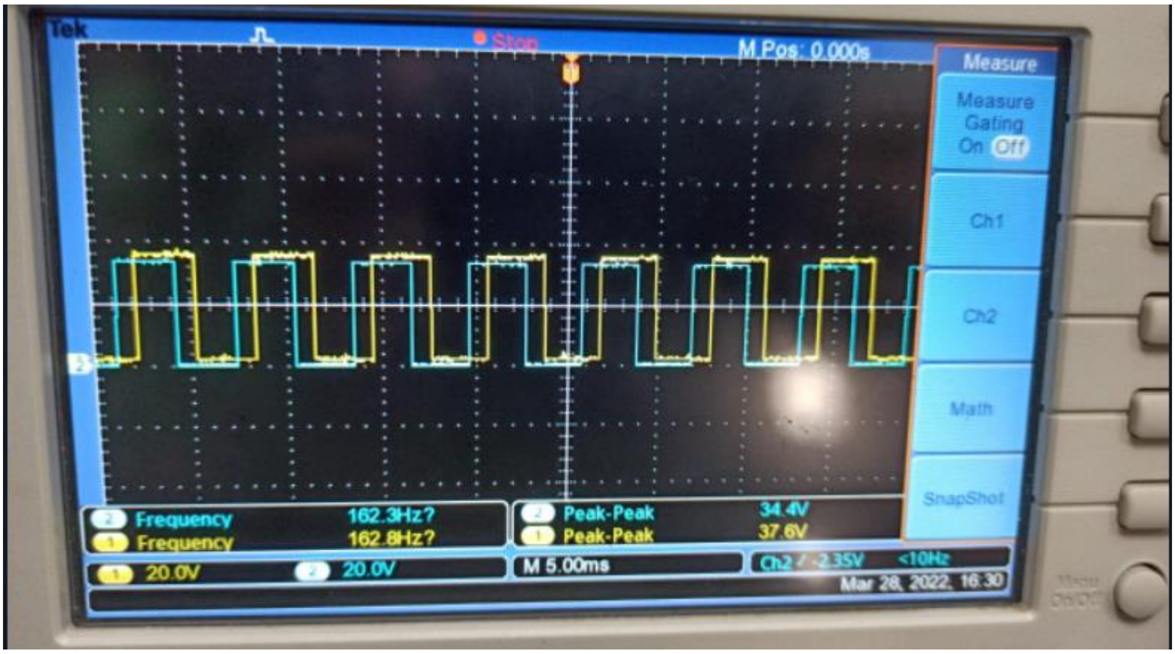
Cloak and Data:



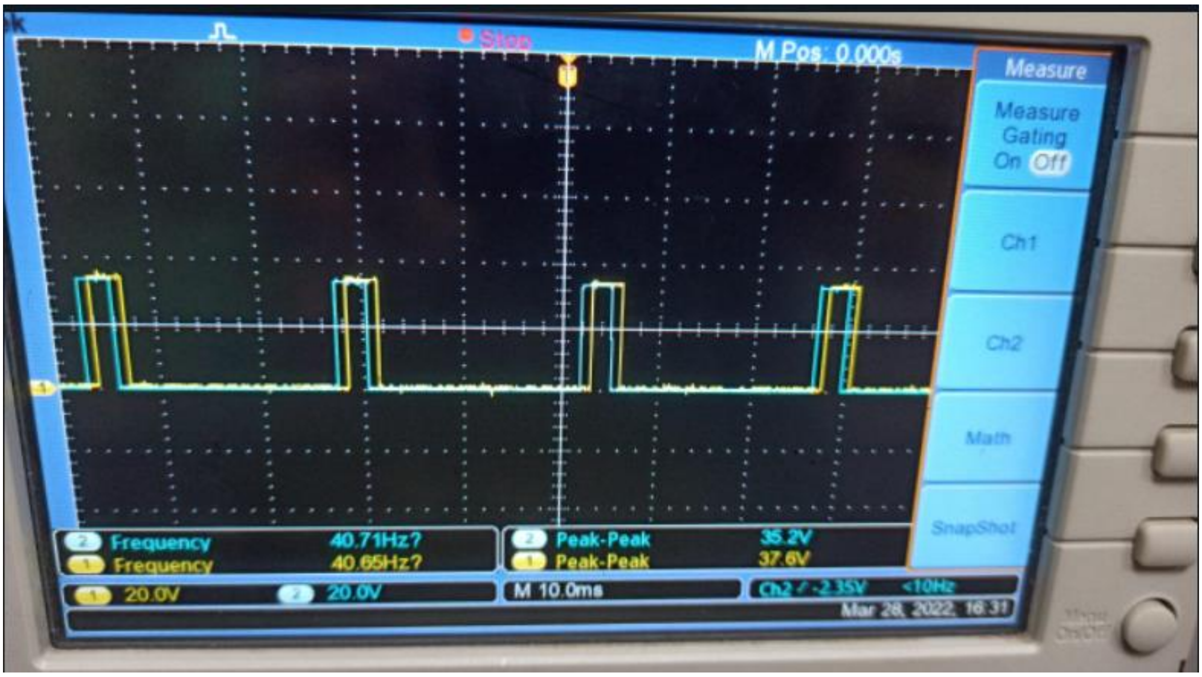
Data and QPSK:



Input I and output I(y):



Input q and Output q(y):



Constellation Diagram:

Conclusion:

In this experiment, we studies and implemented QPSK on ST 2112 QAM Trainer Kit and got the Constellation Diagram on DSO.

Experiment 9.2

Objective: To Perform QAM modulation Technique using ScienTech Trainer Kit. Apparatus

Apparatus Required:

ST 2112 QAM Trainer
KitCRO
CRO Probes

Theory:

The QAM is a digital modulation where the information is contained into the phase as well as the amplitude of the transmitted carrier.

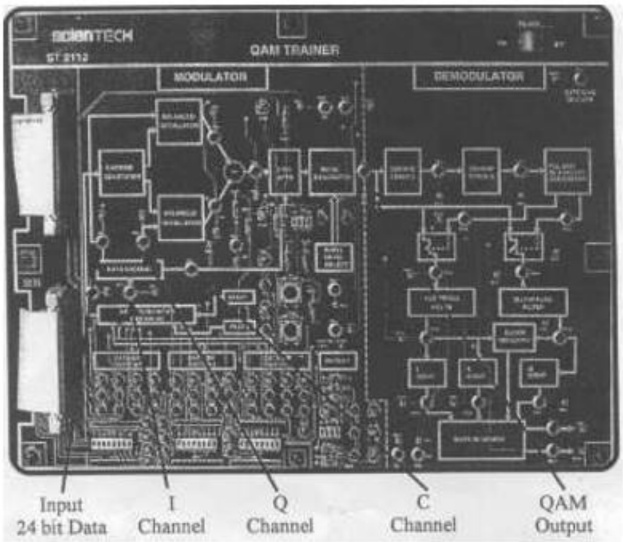


Fig.1 QAM Modulation Kit

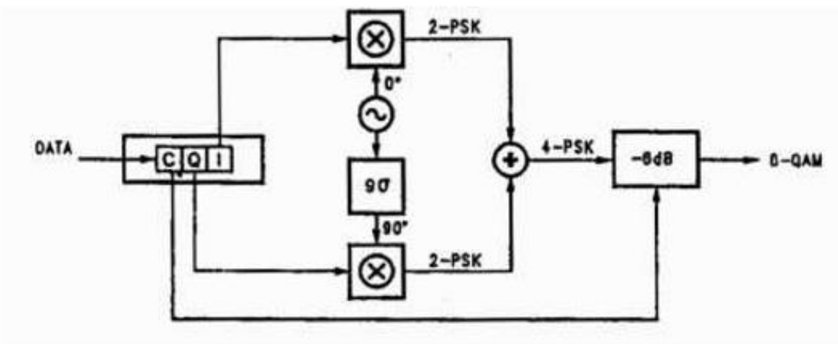


Fig.2 (Block Diagram)

8-QAM :

In the 8-QAM the data are divided into groups of 3 bits (Tri bit), one of which varies the amplitude of the carrier, the last two the phase. The modulated signal can take 4 different phases and 2 different amplitudes, for a total of 8 different states.

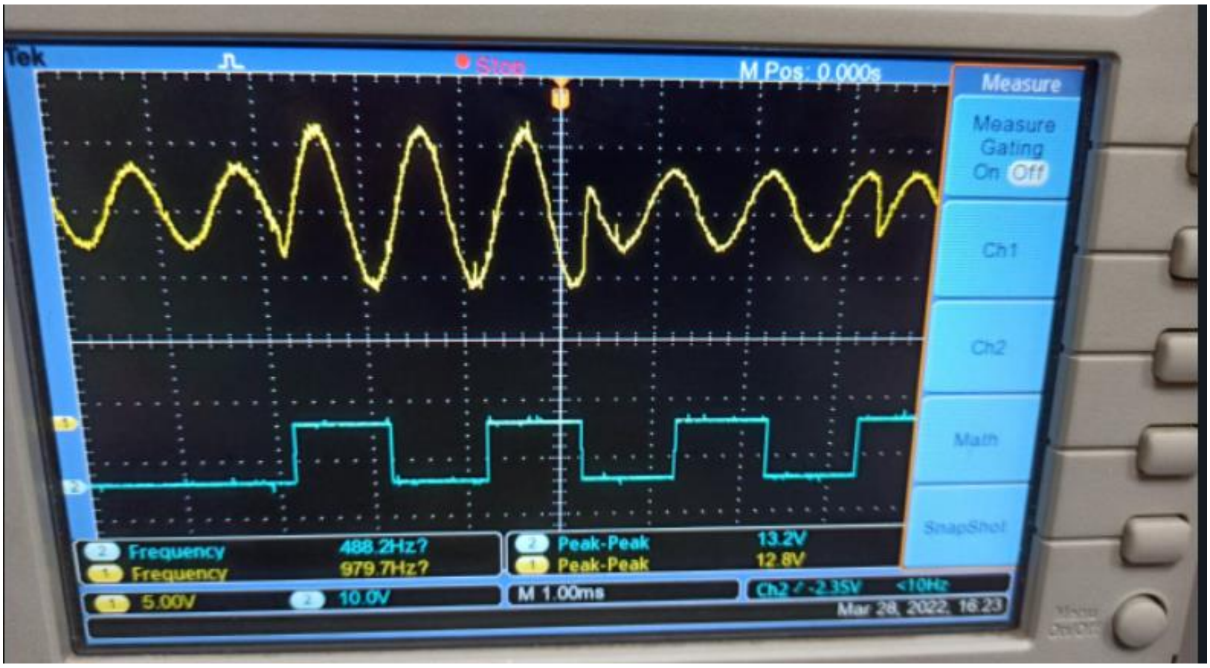
16-QAM:
In the 16-QAM the data are divided into groups of 4 bits (Quad bit). The 16 possible combinations change amplitude and phase of the carrier, which can take 16 different states.

Procedure:

1. Ensure the following initial conditions on ST2112 trainer: Power supply and SW3, SWS, SW6, SW7 and SW9 should be in the OFF mode.
2. Switch on the power supply.
3. Connect Test point TP6 on Channel 1 & TP7 on Channel 2 of Oscilloscope; you will observe 1 KHz sine & cosine wave.
4. Set I, Q & C Channel data with the help of DIP switch SW5, SW6, SW7. As there are 24 bits data available on the trainer so, first bit is I bit then second bit is Q bit then third bit is C bit. For example:
   SW5=l
   1000110
   SW6=0101100
   0
   SW7=0110001
   0
5. Switch ON all the DIP switches on SW3.
6. Now press SW8 which is reset switch then press SW4 which is start.
7. Now connect Channel 1 of Oscilloscope to TP2 & Channel 2 to TP1, you can observe Clock & Data which you have set. (If you are using logic analyzer then you are able to see all 24 bits)
8. Now to observe QAM modulated signal with respect to data, connect Channel 1 toTP1 & Channel 2 to TP9.
9. You can add noise by using DIP switch SW9 (001/010/111).
10. To observe the demodulator section, connect channel 1 of oscilloscope to the test point TP 12 you will observe squarer frequency.
11. To observe I switch & Q switch in the demodulator section, connect channel 1 of oscilloscope to TP 16 & channel 2 of the oscilloscope to TP 17.
12. To observe I, Q & C demodulated signal connect oscilloscope to TP 2O, TP 21, TP 22(if you have logic analyzer you can observe I, Q & C simultaneously).
13. To observe decoded data you have to connect oscilloscope channel 1 to TP 23 & channel 2 to TP 24.
14. Turn OFF the power.

Waveforms & Constellation Diagram:

Data and QAM:



Constellation Diagram:

Conclusion:

In this experiment, we have studied and implemented QAM on ST 2112 QAM Trainer Kit and got the Constellation Diagram on DSO. We have performed QAK modulation and demodulation technique and observed various signals such as in phase component, quadrature component, clock, input signal, modulated and demodulated signal.

Practical: 10

Aim: To study about Software Defined Platform

Theory:

Introduction of GNU based Software Defined Radio (SDR):

Software Defined Radio (SDR), sometimes shortened to software radio (SR), was introducedfor the first time in 1991 by Joseph Mitola. The word of SDR was used to show a radio class that could be re-configured or re-programmed , thus resulted a kind application of wireless communication with mode and frequency band determined by software function.

Ideally, SDR offers flexibility, re-configurability, scalability and as multi-mode as possible. SDR architecture is developed based on conventional radio functions. The difference is all functionsof signal processing on conventional radio are carried out fully by hardware while the functions of signal processing on SDR are carried out as much as possible by software. The major key in building SDR is the placement of ADC and DAC components as a divider between analog and digital domain, thus the signal processing can be carried out using software.

The more realistic SDR architecture places ADC/DAC wideband after Down Converter/Up Converter, thus the conversion from analog to digital or its reverse is carried out on Intermediate Frequency (IF) signal which possesses lower frequency than RF signal. Today, that type of architecture are being developed widely and researched for the implementation.

The Fig.1 shows the SDR architecture for both transmitter and receiver which can be represented using block diagram as shown in Fig.2. Based on Fig.2, the SDR platform performs transmitting and receiving functions.

The receiver (Rx) will perform some process such as RF signal processing, channelization, digital IF signal processing, demodulation, and base band signal processing. As shown in Fig.2, the computation process in the receiver will be more complex that in the transmitter.

GNU Radio is a free & open-source software development toolkit that provides signal processing blocks to implement software radios. It can be used with readily-available low-cost external RF hardware to create software-defined radios, or without hardware in a simulation-like environment. It is widely used in research, industry, academia, government, and hobbyist environments to support both wireless communications research and real-world radio systems.

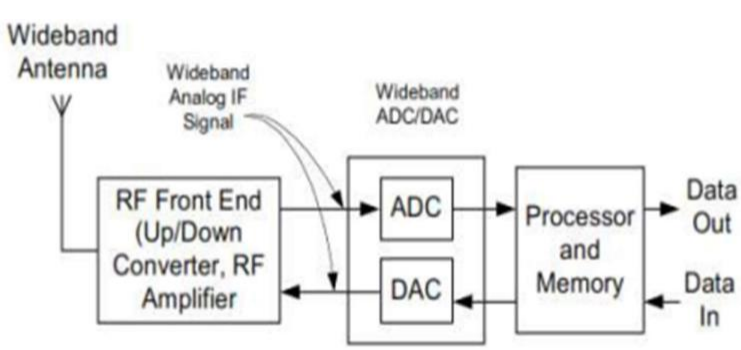


Fig. 1 Realistic SDR Architecture

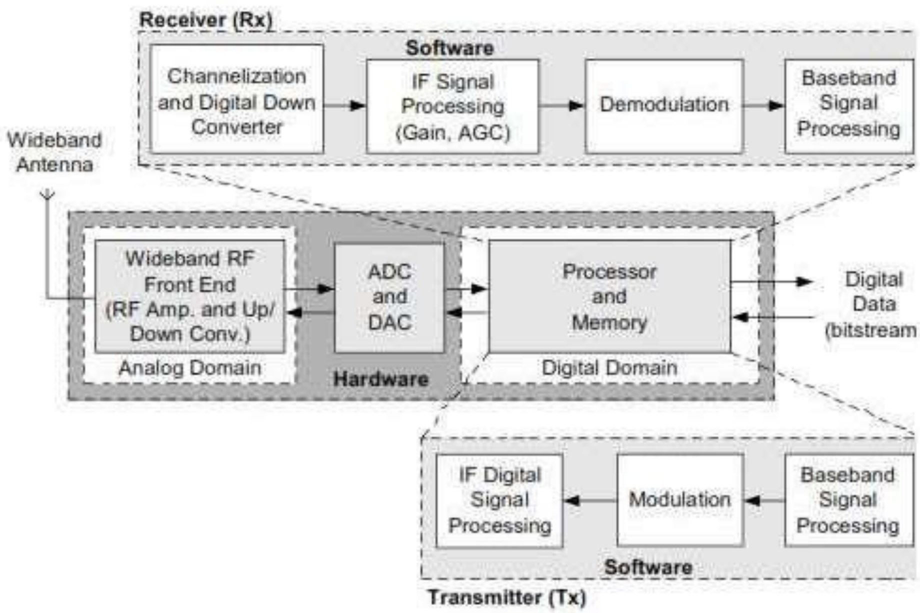
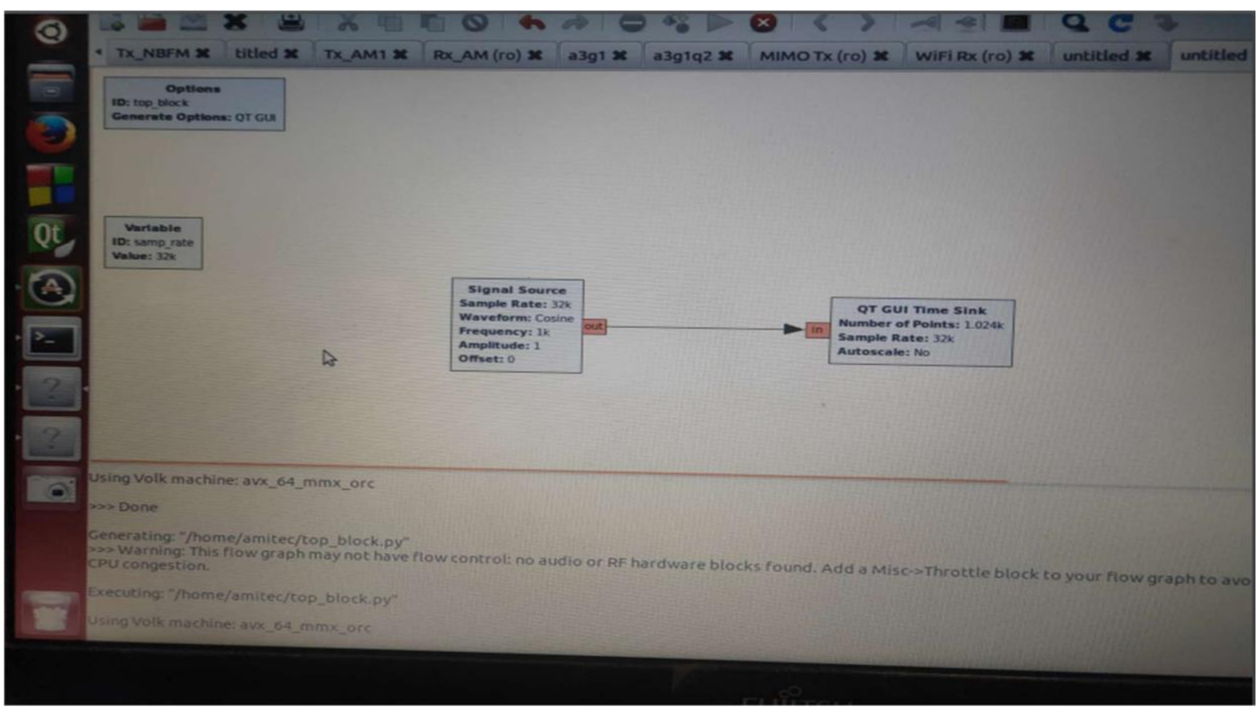
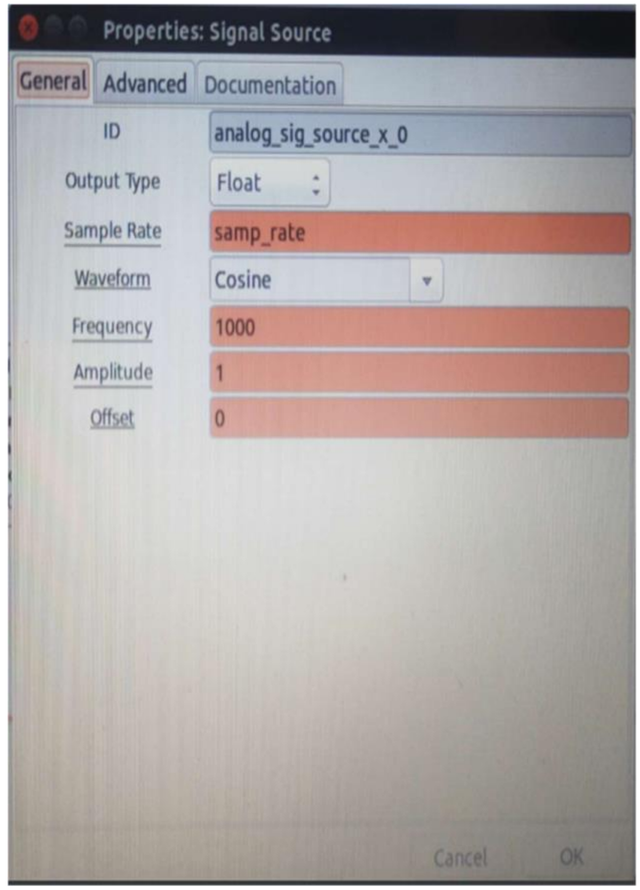


Fig. 2 SDR Architecture for Transmitter and Receiver

Part 1 Block Diagram

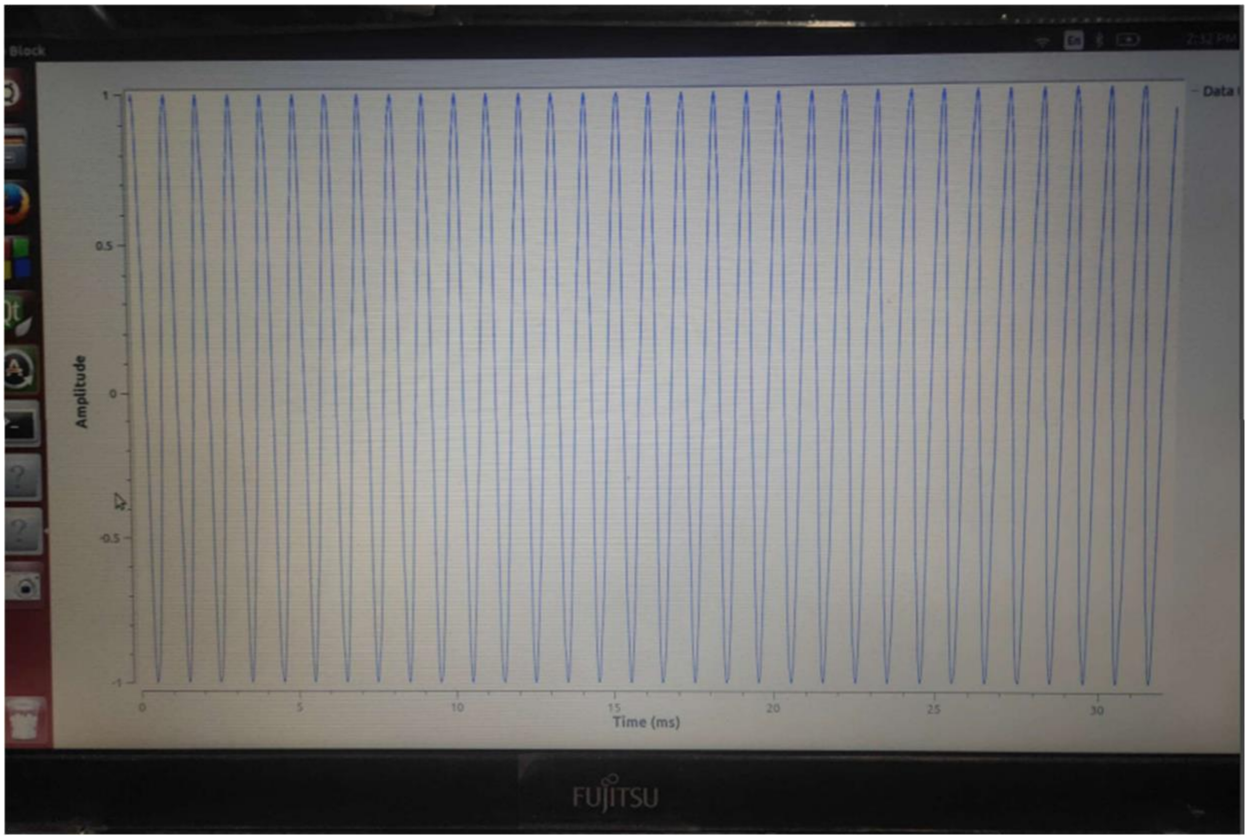


Properties QT GUI Time Sink

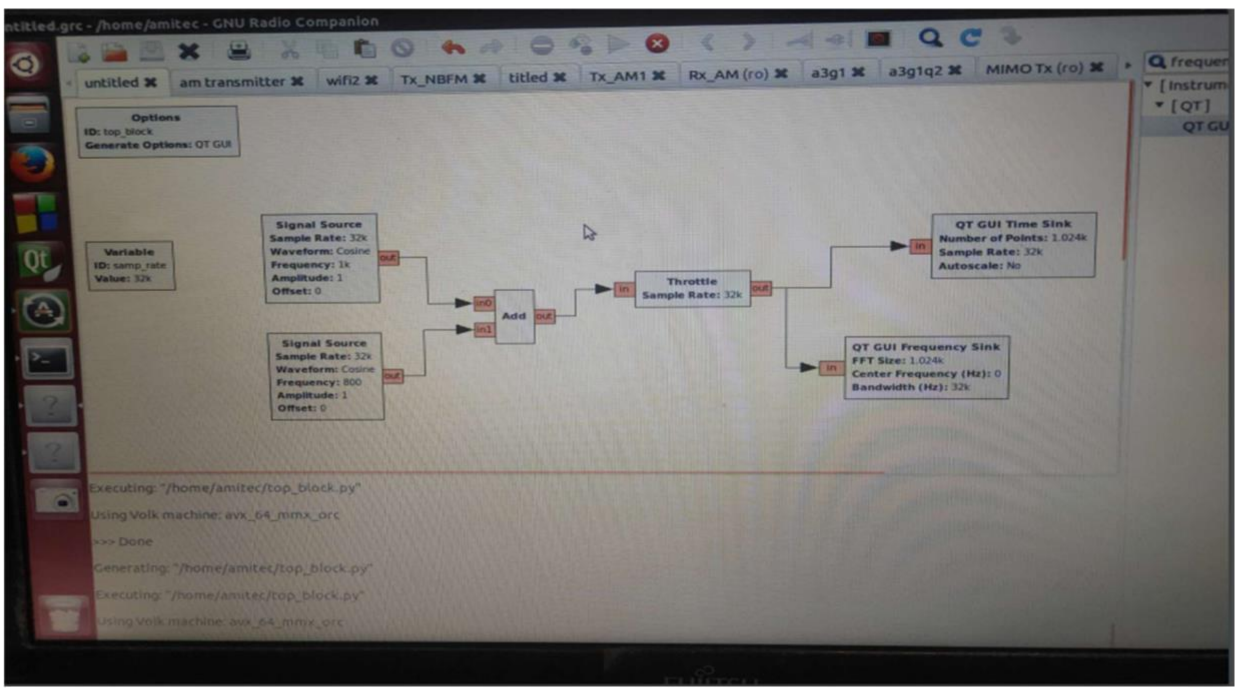


Properties Signal Source

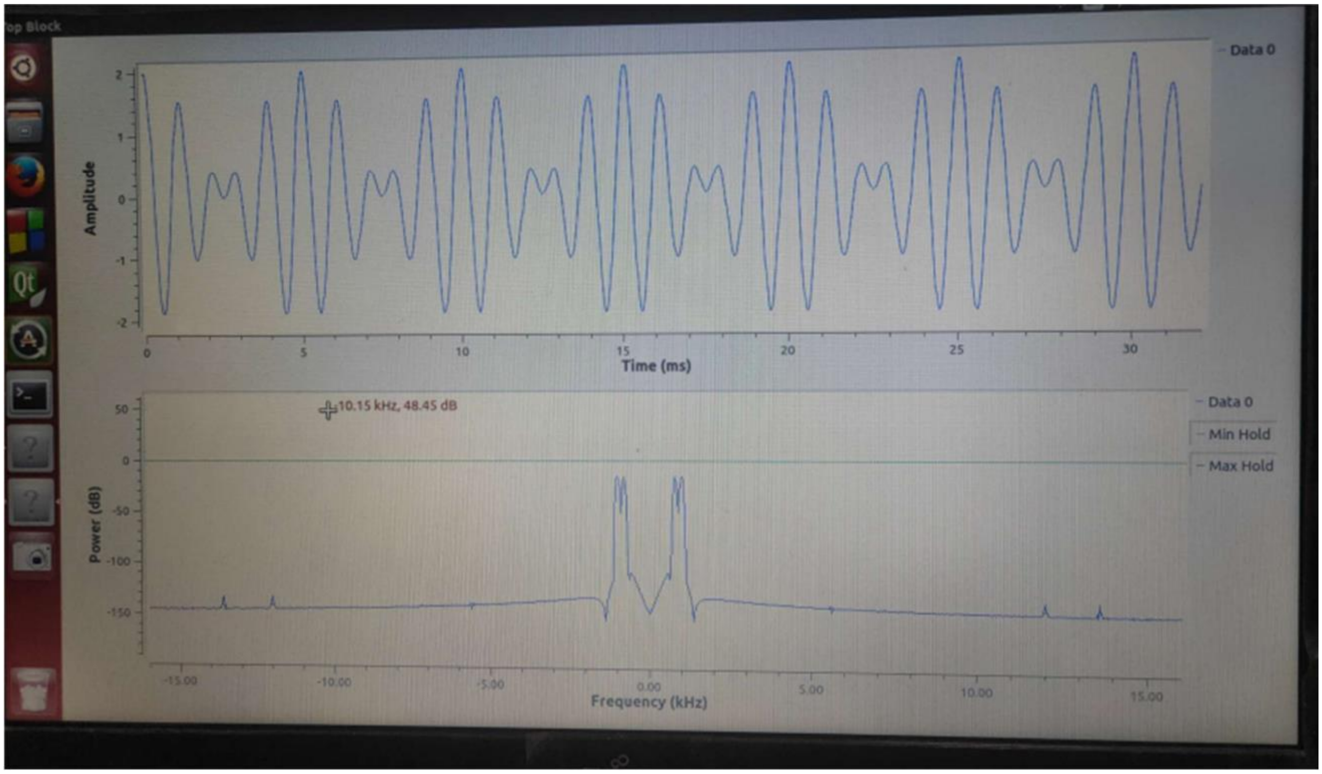
Result: Waveforms/Graphs



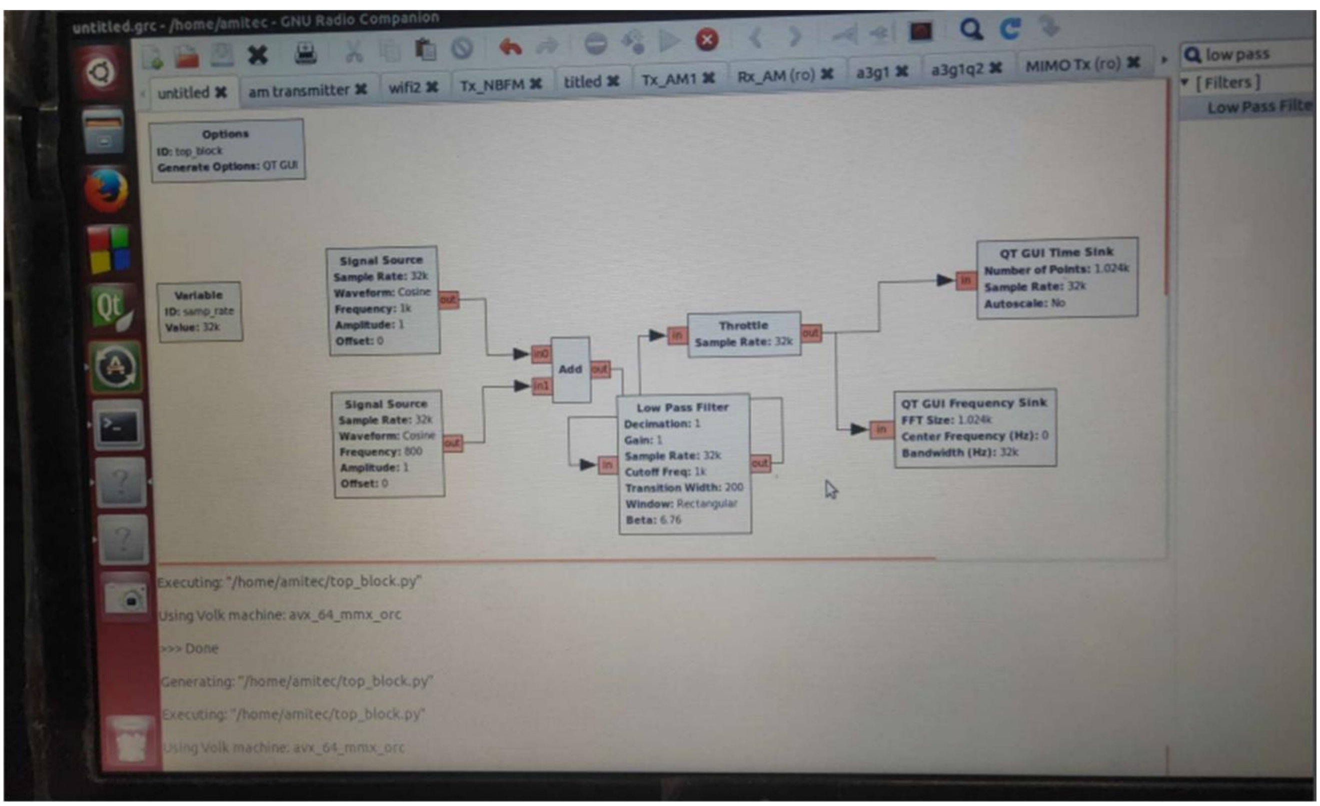
PART 2 Block Diagram

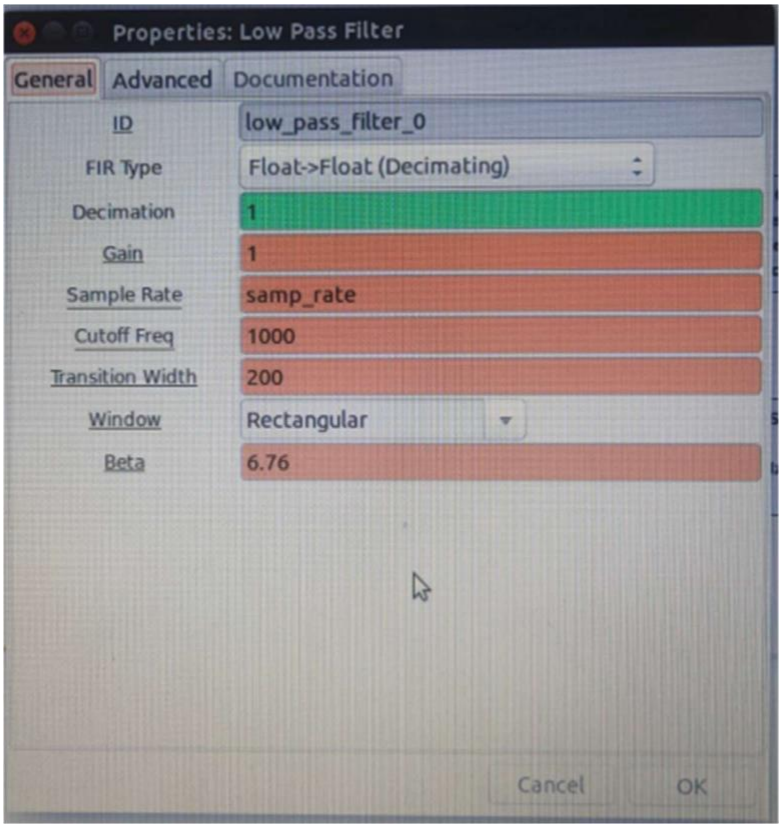


Result: Waveform/Graphs



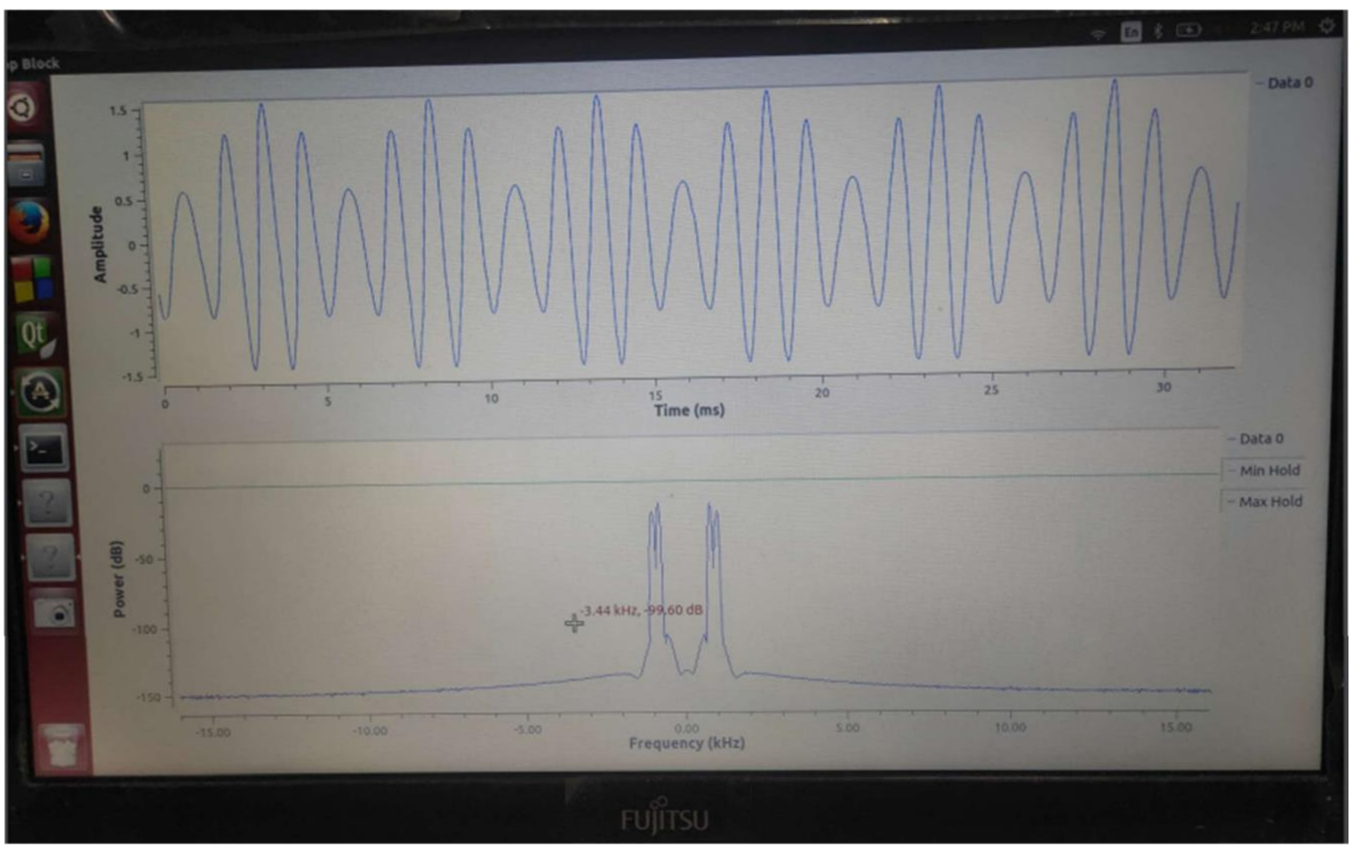
PART 3 Block Diagram





Properties of Low Pass FIlter

Result: Waveform/ Graphs



Conclusion:

The basic understanding of SDR and GRC graphic tool was developed and the use of different block in different categories was understood. In this way, different functions can be generated using blocks from different categories.

Experiment: 11

Study of Direct Sequence Spread Spectrum(DSSS) Modulation and Demodulation process

Experiment-10.1:

Objective: Study of Spreading and Despreading based on Spread Spectrum Technique.

Apparatus Required:

1. ST 2115 CDMA Trainer Kit
2. CRO
3. CRO Probes, Patch Codes

Procedure: Refer to the figure 1. while configuring setup for the experiment.

1. Switch data switches to 1 or 0 as per your choice of binary data pattern.
2. Connect any two of the four taps viz. A, B, C or D to the inputs of EX-OR gate of PN sequence generator. Connect 240 KHz clock signal on board to the clock input of the PN sequence generator.
3. Now switch ON the power supply and observe the output of Binary Data Generator and PN sequence generator. Since the data generator frequency used here is 30 KHz and that of PN Sequence Generator is 240 KHz, and hence there are 8 PN sequence bits per data bits for spreading the binary signal.

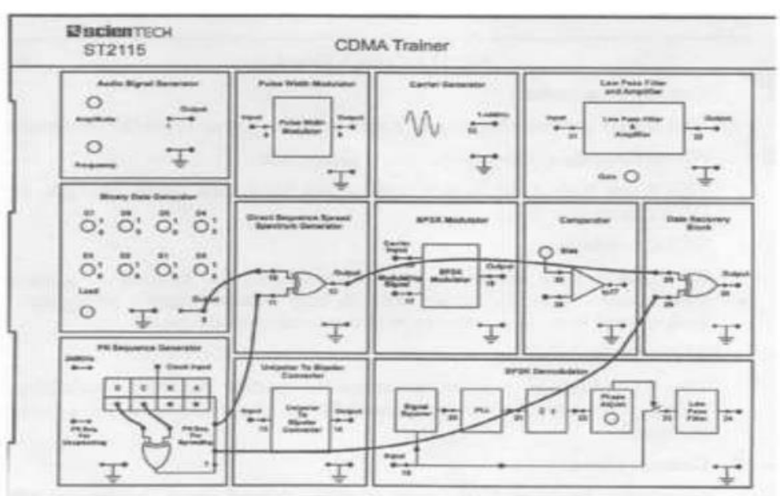
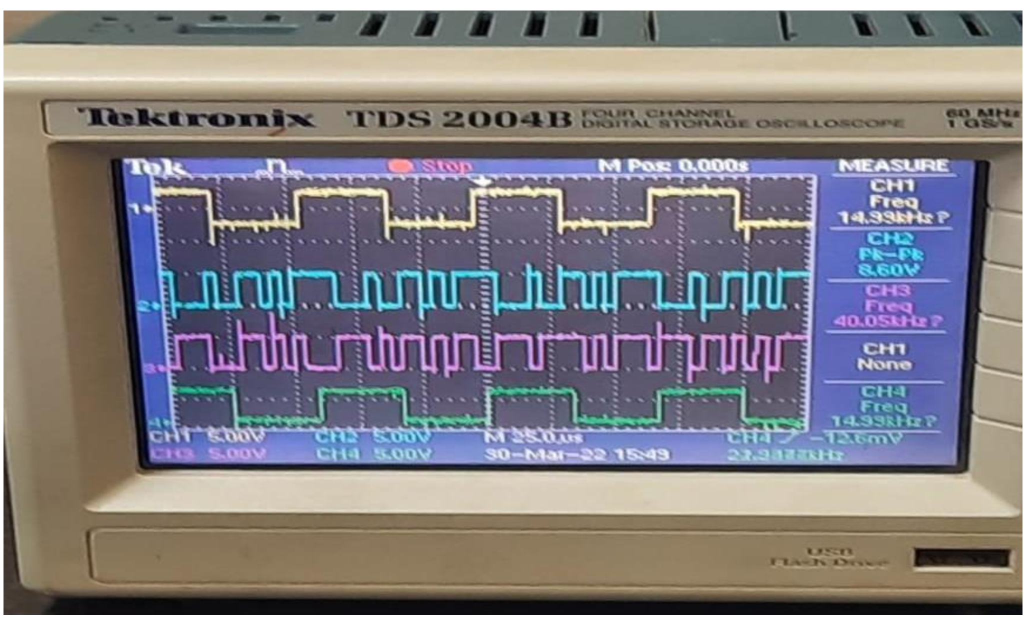


Fig 1: Spreading and Despreading Set Up

4. Change the positions of taps for feedback in the PN Sequence Generator Block to obtain deferent patterns of the PN sequences. Now, switch OFF and then ON the power supply to reload the changes, if changes do not appear in the output on changing the tap positions.

5. Connect output of binary data generator to one of the inputs of Direct SequenceSpread Spectrum generator input.
6. Connect output of PN sequence generator to the other input of DSSS EX-OR gate.
7. Now turn ON power supply and observe the output of DSSS generator block. This is our DSSS signal.
8. Now connect output of this DSSS block to the one of the inputs of EX-OR gate of Data Recovery Block. Connect the same output of PN sequence generator, which we have taken for spreading to the other input of this recovery gate for dispreading. Note that the PN sequence used for dispreading is taken from the same output pin from where the PN sequence is taken for spreading the signal. This is because of the fact that there is complete synchronization between the spread signal and PN sequence. In other words there is not any significant delay involved in spreading process.
9. Observe the output of this data recovery block. This is recovered output without almost any error.
10. Now change the tape positions of shift registers (A, B, C or D) to get a new PN sequence and repeat the above process again. Thus you will observe that with each different sequence we are quite able to recover the original data also with different PN sequences, the modulated (Spread) data looks different i.e. we can recover the data if and only if we are using the same PN sequence for both modulation and demodulation. This is the reason that this DSSS technique has a large potential for being a multiple access technique. This multiple access technique is known as “Code Division Multiple Access” (CDMA) technique.

Waveform:



Conclusion: We were able to understand and perform spreading and dispreading based on spread spectrum technique using CDMA Trainer kit.

Experiment-10.2:

Objective: Study of Direct Sequence Spread Spectrum(DSSS) Modulation/Demodulation using an Analog signal and Digital signal as inputs individually.

Apparatus Required:

1. ST 2115 CDMA Trainer Kit
2. CRO
3. CRO Probes, Patch Codes

Procedure for Analog signal:

Refer to the figure 1. while configuring setup for the experiment.

1. Make the connections as shown in the figure 1.

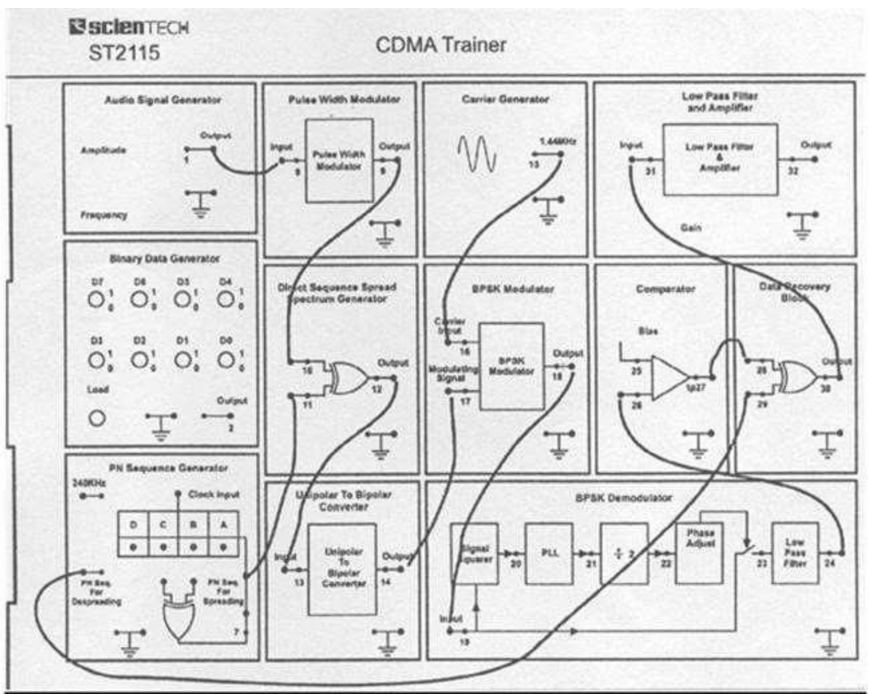


Fig 1. DSSS Set Up

2. Observe the output of audio signal generator block.
3. Observe the output of DSSS block.
4. Observe the output of BPSK modulator.
5. Observethe output of data recovery block. Adjust phase of recovered carrier and bias of comparator until you see an exact replica of pulse width modulated data at the output of this recovery block.

6. Observe the output of low pass filter section and compare it with the input Audio signal. Change gain of the amplifier to remove any nonlinearity errors. If still outputis not proper then change amplitude of the input audio signal and adjust the gain of the output amplifier to remove distortions.

Procedure for Digital signal:

Refer to the figure 2 while configuring setup for the experiment.

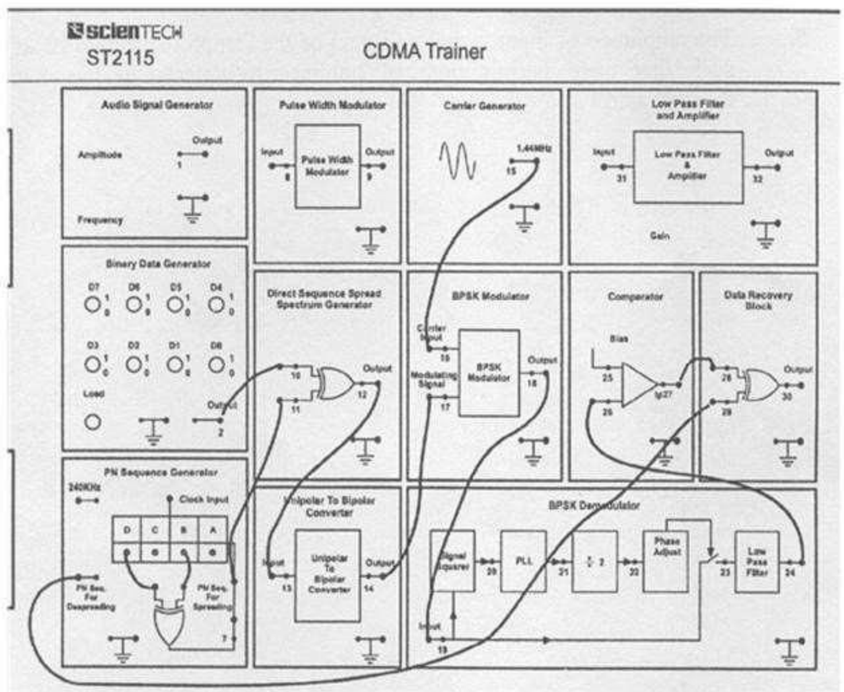


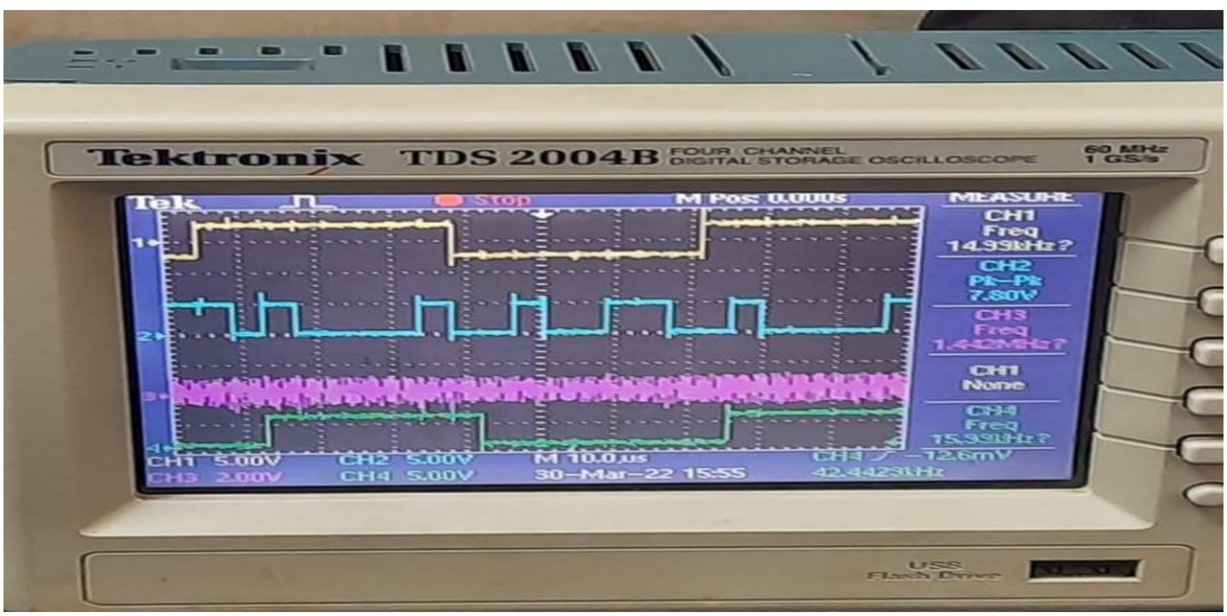
Fig 2. CDMA trainer set up

1. Switch data switches to 1 or 0 as per your choice of binary data pattern.
2. Connect any two of the four taps viz. A, B, C, or D to the inputs of EX-OR gate of PN Sequence generator. Connect 240 KHz clock signal on board to the clock input of the PN sequence generator.
3. Now switch ON the power supply and observe the output of Binary Data generator and PN sequence generator. Since the data generator frequency used here is 30 KHz and that of PN Sequence Generator is 240 KHz, and hence there are 8 PN sequence bits per data bits for spreading the binary signal.
4. There are two outputs of PN Sequence Generator shown on the board. One of the outputs is for spreading the binary data signal and the other one is for dispreading the coded signal to recover back the original data(when BPSK modulation is used for RF modulation of Spread signal)
5. Connect binary data and PN sequence outputs to the EX-OR gate of DSSS block. Connect the output of DSSS block to the input of unipolar to

bipolar converter. Take the output of this converter to the input of BPSK modulator. Connect sinusoidal carrier from Carrier Generator to the carrier input of BPSK modulator. This completes the modulator connections.

6. Now connect output of BPSK modulator to the input of BPSK demodulator block. Connect output of this block to the comparator input. Here we would receive original chipped data.
7. Connect the recovered chipped data (output of comparator) to one of the inputs of data recovery block. Connect 'PN Sequence for Dispreading' output of PN sequence generator block to the other input of data recovery gate.
8. Now turn power supply ON. Observe data and PN Sequence at their respectiveoutput pins. Press load button if data is not appearing.
9. Observe the output of DSSS block. This is called 'Chipped Data'.
10. Observe the output of BPSK modulator. This is RF modulated chipped data.
11. Observe the output of comparator and then Data Recovery Block. Adjust phase of recovered carrier in BPSK modulator section and bias of comparator until you see a complete replica of original binary data.
12. Change data pattern and repeat the whole procedure with this new data. Again adjust phase and bias of comparator so as to recover the data completely.
13. Change chip (PN Sequence) pattern and observe the results

Waveform:



Conclusion: We were able to understand and perform the Direct Sequence Spread Sprectrum modulation and demodulation using CDMA by taking analog and digital signals as inputs.

Experiment - 12

Objective: To Study about Mobile Phone Trainer Kit.

Apparatus Required: **TechBook Scientech 2139**

Technical Specification :

Cellular System	:	EGSM/GSM 900.
Rx Frequency Band	:	EGSM 925....960 MHz
	:	GSM 900 935 ...960 MHz
Tx Frequency Band	:	EGSM 880....890 MHz
	:	GSM 900 890 ......915 MHZ
Output power	:	+5... +33dBm/3.2 mW...2 W
Channel spacing	:	200 KHz
-Antenna --	:	Loop type, 50 ohms
Display	:	84 x 48 pixels
On board sections	:	Antenna, Keypad, SIM, Charging Circuit, Clock, User interface: Buzzer, Vibrator, LEDs.
No. of Test points	:	41
No. of Switched fault	:	25
Features that can be set	:	Screen saver, Ring tones, Logos, SMS etc.
Power supply	:	220V f 10%, 50 Hz
Power consumption	:	3.6 Watts (Approx.)
Fuse	:	1.5 amps
Dimension (mm)	:	W 450 x H 113 x D 280.
Weight	:	2.6 Kg (Approx.)

An overview of GSM network:

1. ***GSM network:***
   The GSM Network comprises three parts, Mobile Station (MS) which is similar to a cordless phone with extra features, the Base Transceiver Station (BTS) that controls the connection with the Mobile Station, the Base Station Controller (BSC) that controls multiply Base Transceiver Station's and then the rest of the network covered further below.

2. ***Mobile Station (MS):***
   A Digital Mobile Phone and a SIM card make up the Mobile Station. The SIM (Subscriber Identity Module) is a card that first into your handset. The SIM microprocessor is based on a silicon chip which is designed to tolerate temperatures between -25°C and +70°C, and will also withstand up to 85% humidity. However silicon is fragile and therefore, if the card is tampered with, physically or electronically, the card will be rendered useless. The SIM contains all of your identification details, such as your IMSI (International Mobile Subscriber Identity).This is a numeric string, where the first 3 digits represent the country where the SIM is from, the next represent the operator in that specific country. The other digits represent the subscriber's identity in his home-network), phone memories, billing Information, SMS text messages, pin numbers and international roaming information.

   An IMEI (International Mobile Equipment Identity) card is the serial number of the GSM phone. The SIM card contains a IMSI (International Mobile Subscriber Identity) number that identifies the user to the network along with other user and security information.

3. ***Base Transceiver Station (BTS):***
   The Base Transceiver Station consists of a radio transceiver with antenna that coversa single cell. It handles the communications with the MS via radio interface. BTS areall connected together to allow you to move from one cell to another. The antenna can take on various forms.

4. ***Base Station Controller (BSC) :***
   The Base Station Controller manages multiple BTS's. It controls the allocation and release of radio channels and handovers between cells. A series of BTS's are connected. To each Base Station Controller, the BSC keeps an eye on each call and decides when to pass the call off to another BTS and to which one.

The Rest of the Network:

Several BTC's are controlled by the Mobile service Switching Center (MSC), the MSC works with four databases (HLR, VLR, EIR and the AUC) and together they manage the communications between Mobile Station user and the other network types. Each of the databases has a separate job.

5. ***Mobile Switching Center (MSC):***
   The Mobile Switching Center is the interface between the base station system and the switching subsystem of the mobile phone network. Furthermore, the MSC is also the interface between the cellular network and the PSTN. The MSC generates all billing records and ensures that all usage is directed to the appropriate account. The MSC has a relatively complex task, as unlike a conventional telephone exchange, when GSM subscribers make calls they could be anywhere within the network. The MSC must ensure that calls are routed through to those subscribers, wherever they are and wherever they move to throughout the duration of each cell. This situation becomes even more complex when two mobile subscribers wish to contact each other from two distant locations.

   In order to simplify the subscriber management function, a specific service area is allocated to each MSC. The MSC has to control the switching of tariff to and from the subscribers within its service area which involves the co-ordination of all radio resources and the inter cell hand-off activities.

Home Location Register (HLR) :

The HLR is the central data base for all the subscribers which contain detaiIs on the identity of each subscriber, the services to which they have access and the locations where the subscriber was last registered. Once the Mobile Station's MSISDN has been used to identify the IMSI, the HLR verifies the subscription records to ensure that the call can be delivered to the last known location of the Mobile Station.

Visitor's Location Register (VLR):

The VLR is a database that is linked to an MSC and temporarily stares information about each Mobile Station within the area served by that MSC.

Equipment Identity Register (EIR):

The EIR ensures that all Mobile Equipment's, are valid and authorized to function on the PLMN.

Authentication Center (AUC):

The authentication center is used to validate the SIM Card being used by the Mobile Station. Secret information that is held in the AUC and which is also contained within the SIM Card is used to perform a complex mathematical calculation. Authentication occurs if the results of these two calculations agree.

SMSC (SMS Center or Service Center):

The SMSC handled all the SMS messages that are sent. The messages are sent on a data channel so you can receive them while on a call.

GMSC (Gateway MSC):

It is a gateway switch where the call is directed when setting up a call to a GSM user. The GMSC looks for the subscriber by interrogating the right HLR which then interrogates the VLR and routes the incoming call towards the MSC where the subscriber can be reached.

Outstanding Features:

1. ***Quality:***
   With digital, sound quality is sharp and clear. Background sounds and static are vastly reduced and crossed-line conversations are also eliminated. In comparison with analog there are also far fewer dropouts, and overall the quality is more like that of a fixed telephone.

2. ***Security:***
   Unlike analog, everything you say and send within the digital network is safe and secure. Some features are user authentication that prohibits unauthorized access, encryption key distribution that guarantees the privacy of the call and caller Identification restrictions that can prevent the delivery & identify calling-user's number to the receiver.

3. ***Convenience:***
   With digital, better technology means better battery life. You get up to twice as much talk time from each battery charge, compared with analogue. In addition the digital service allows more calls to be handled at any one time, therefore reducing congestion in areas of dense population and high usage.

4. ***Roaming:***
   Roaming is one more feature of GSM technology. With digital, you are able to use your mobile phone, and number in other countries around the world that operate a GSM network.

Features:

1. Call Forwarding
2. All Calls
3. No Answer
4. Engaged
5. Unreachable
6. Call Barring
7. Outgoing - Bar certain outgoing calls (e.g. ISD)
8. Incoming - Bar certain incoming calls (Useful if in another country)

9. Global roaming - Visit any other country with GSM and a roaming agreement and use your phone.
10. SMS - Short Message Service - Allows you to send text messages to and from phones
11. Multi Party Calling - Talk to five other parties, as well as yourself at the same time
12. Call Holding - Place a call on Hold
13. Call Waiting - Notifies you of another call whilst on a call
14. Mobile Data Services - Allows handsets to communicate with computers
15. Mobile Fax Service - Allows handsets to send, retrieve and receive faxes
16. Calling Line Identity Service - This facility allows you to see the telephone number of the incoming caller on our handset before answering
17. Advice of Charge - Allows you to keep track of call costs
18. Cell Broadcast - Allows you to subscribe to local news channels
19. Mobile Terminating Fax - Another number you are. Issued with that receives faxes that you can then download to the nearest fax machine.
20. Upgrade and improvements to existing services
21. Majority of the upgrade concerns data transmission, including bearer services and packet switched data at 64 Kbits and above
22. SIM enhancements

GMSK Modulation:

GSM uses a digital modulation format called 0.3GMSK (Gaussian minimum shift keying). The 0.3 describes the bandwidth of the Gaussian filter with relation to the bit rate. The bandwidth of 0.3 was chosen as a compromise between spectral efficiency and inters symbol interference.

GMSK is a special type of digital FM modulation. 1's and. 0's are represented by shifting the RF carrier by plus or minus 67.-708. KHz. Modulation techniques which use two frequencies to represent one and zero` are denoted FSK (frequency shift keying). In the case of GSM the data rate of 270.833kbit/sec is chosen to be exactly four times the RF frequency shift. This has the effect of minimizing the modulation spectrum and improving channel efficiency. FSK modulation where the bit rates exactly four times the frequency shift is called MSK (minimum-shift keying). In GSM, the modulation spectrum is further reduced by applying a Gaussian pre-modulation filter. This slows down the rapid frequency transitions, which would otherwise spread energy into adjacent channels.

0.3GMSK is not phase modulation (i.e. information is not conveyed by absolute phase states, as in QPSK for example). It's the frequency shift or change of phase statewhich conveys information. GMSK can be visualized from an I/Q diagram. Without the Gaussian filter, if a constant stream of l's is being - transmitted, MSK will effectively stay 67.708 KHz above the carrier center frequency. If the carrier center frequency is taken as a stationary phase reference, the 67.708 KHz signal will cause ; steady increase in phase. The phases will role 360 degrees at a rate of 67,701 revolutions per second. In one bit period (1/270.833 KHz), the phase will get a quarter of the way round the I/Q diagram or 90 degrees. I's are seen as a phase increase of 91 degrees. Two l's causes a phase increase of

180 degrees, three l's 270 degrees and s+ on. 0's cause the same phase change in the opposite direction.

The exact phase trajectory is very tightly controlled. GSM use's digital filters and I/(or digital FM modulators to accurately generate the correct trajectory. The GSM specification allows no more than 5 degrees rms and 20 degrees peak deviation four theideal trajectory.

GMSK, is a form of modulation used in a variety of digital radio communications systems. It has advantages of being able to carry digital modulation while still using the spectrum efficiently. One of the problems with other forms of phase shift keying is that the sidebands extend outwards from the main carrier and these can cause interference to other radio communications systems using nearby channels.

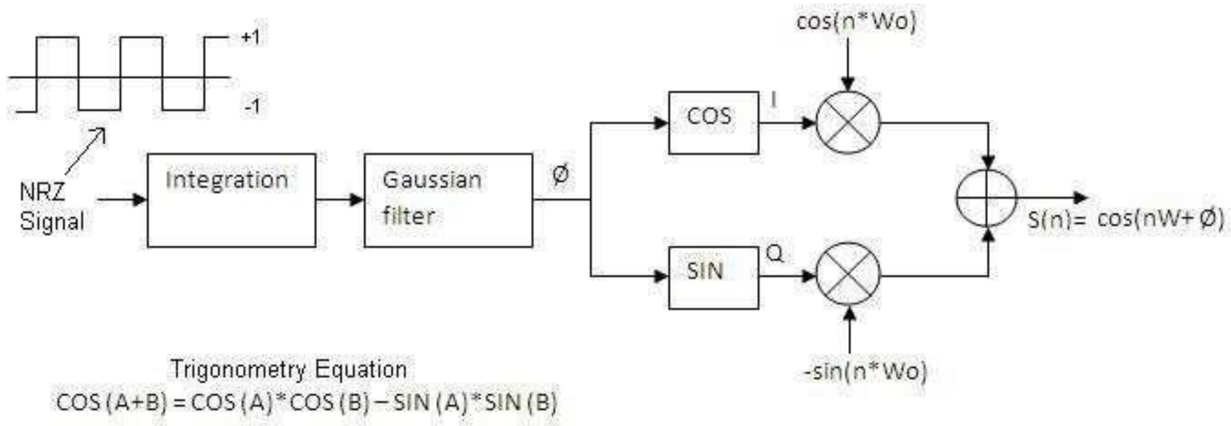


Fig. 1 GMSK Transmitter Block Diagram

GETTING STARTED:

1. Check the battery is received in proper condition.
2. Insert SIM card of any 900MHz service provides in the SIM assembly.
3. Now, insert the battery so that the battery contacts and assembly Terminal match.
4. Switch ON the trainer by pressing the power ON/OFF switch, located at the top right most corner.

Note: Whenever the switch is operated ON/OFF LED operates.

5. When/If the battery is low. Connect the mains cord and operate the Charging switch located at the top right most corner below the power switch.
6. Mode: This is an improved feature of ST2132. The mode has to be selected prior to switching ON the trainer.

   DC Mode: In normal switch position, trainer operates with battery supplied & charging facility functions normally.

   AC Mode :When switch pressed trainer operates on AC & mains cord is a must for the supply. The trainer automatically disconnects the battery contacts when the mode is changed from DC to AC. So, physical presence of the battery in the battery assembly doesn't have any effect. The charging On/Off switch stops the functioning in this mode.

Note: Switch OFF the trainer before switching between the operating modes.

Task 1: To study the T_x IQ and R_x IQ signals.

Procedure:

1. Insert the SIM and power ON the trainer.
2. Make a Call to the trainer or from the trainer.
3. Keep the Call ON

Connect the probe of spectrum analyzer at Tp. 1 and observe the signal in the T_x band.

Now connect the probe to Tp. 2 observe the signal in the R_x band.

Connect two probe of CRO one at Tp. 3 and the other on Tp. 4 observe the R_x burst. A similar R_x burst can be observed by connecting probe Tp. 5 and Tp.6 respectively. Signal fig.2 shows a IQ T_x and R_x burst signal.

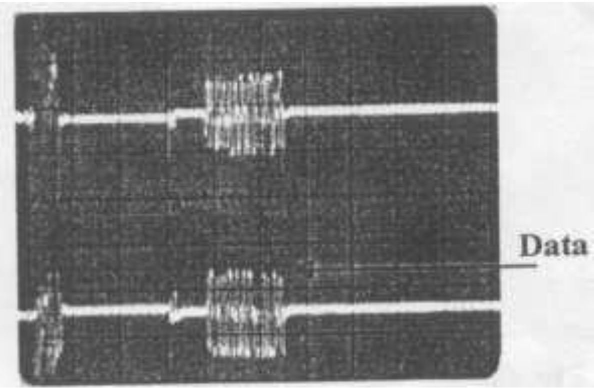


Fig. 2 Burst Signal

Task 2: To observe signal constellation of GMSK signal.

Procedure:

1. Make a Call to the trainer.
2. Receive the Call and Keep the Call ON
3. Connect two probe of CRO one at Tp. 3 and the other on Tp. 4 observe the R_x burst signal fig.3 shows a IQ R_x burst signal.

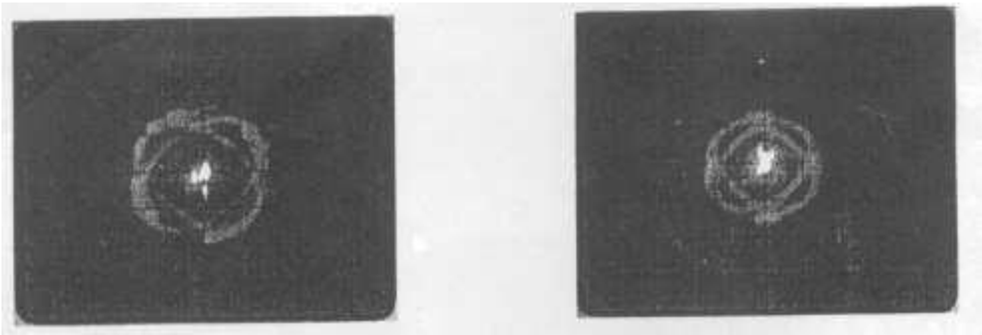


Fig 3. Constellation at Receiver end

Task 3: GSM Data Rate and GMSK encoded signal

Procedure

1. Make a call to the trainer.
2. Connect the probe of CRO at TP. 5 or TP 6.
3. Observe the signal As shown in fig.4 (a).
4. Expand the signal to see the eye pattern.
5. Expand the signal.
6. Observe the GMSK encoded signal

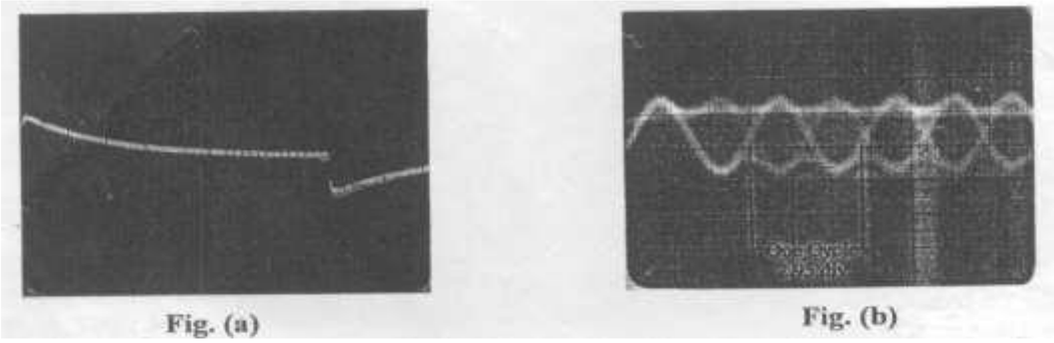


Fig. 4 Waveform observed during a call

Conclusion:

Using Mobile Phone Trainer Kit TechBook Scientech 2139, we studied and analyzed various signals and operations on DSO and also analyzed frequency spectrum of ongoing call on Frequency Analyzer.

Wireless and Mobile Communication

Lab Assignment

Aim: To give the factorial of a number entered by user

Software Used: Android Studio

Libraries Used: Flutter

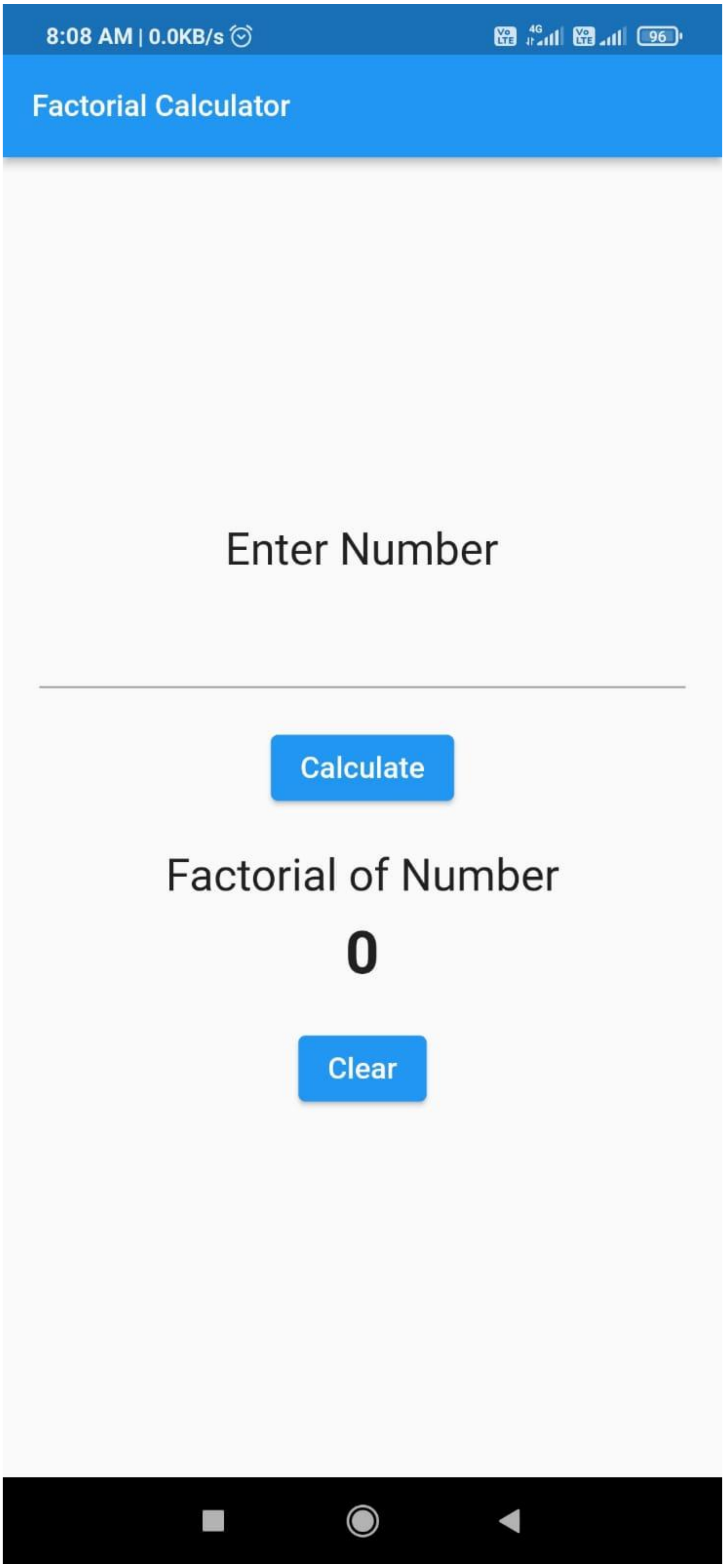
Code:

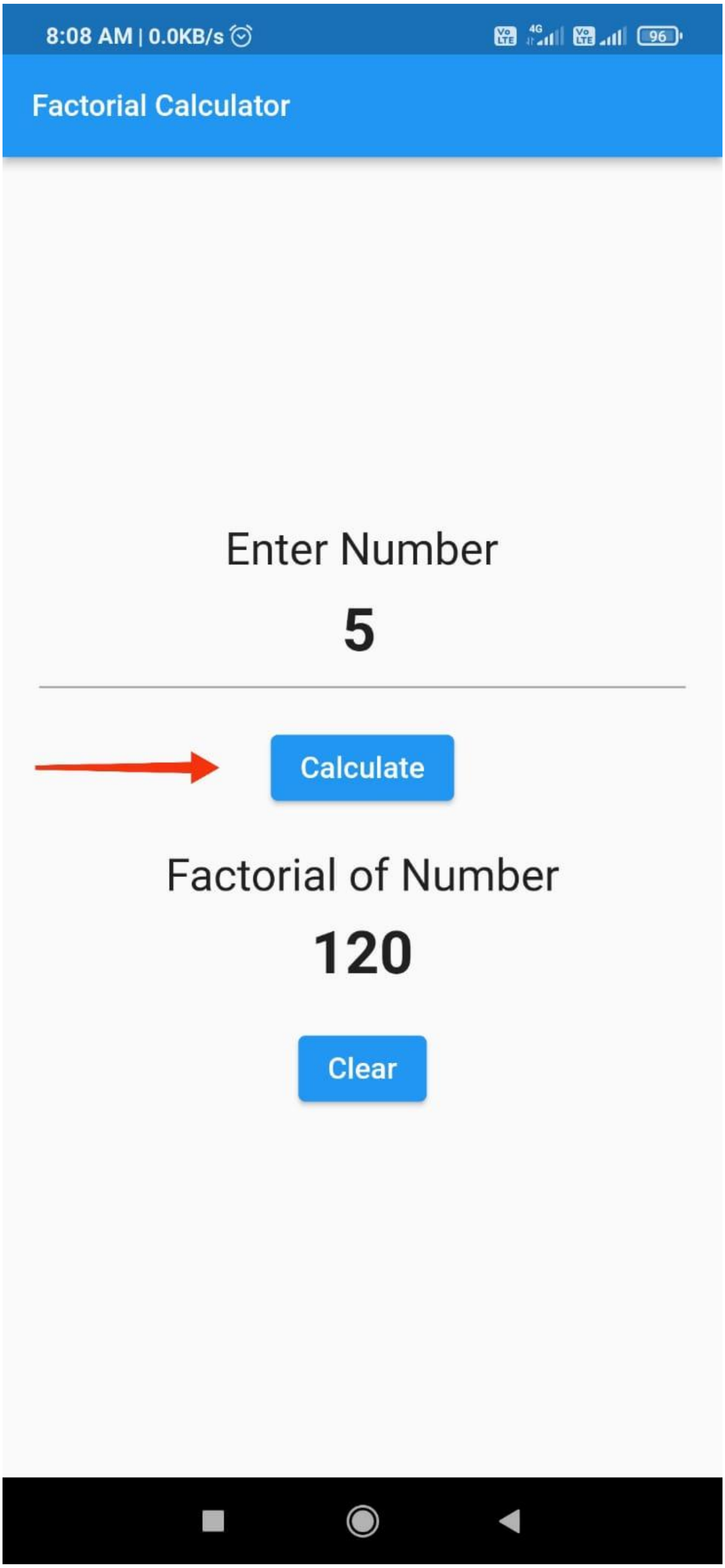
```
import 'package:flutter/material.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: const Factorial(),
    );
  }
}

class Factorial extends StatefulWidget {
  const Factorial({Key? key}) : super(key: key);
  @override
  _FactorialState createState() => _FactorialState();
}

class _FactorialState extends State<Factorial> {
  final controller = TextEditingController();
  String number="";
  int res=0;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
        resizeToAvoidBottomInset: false,
```

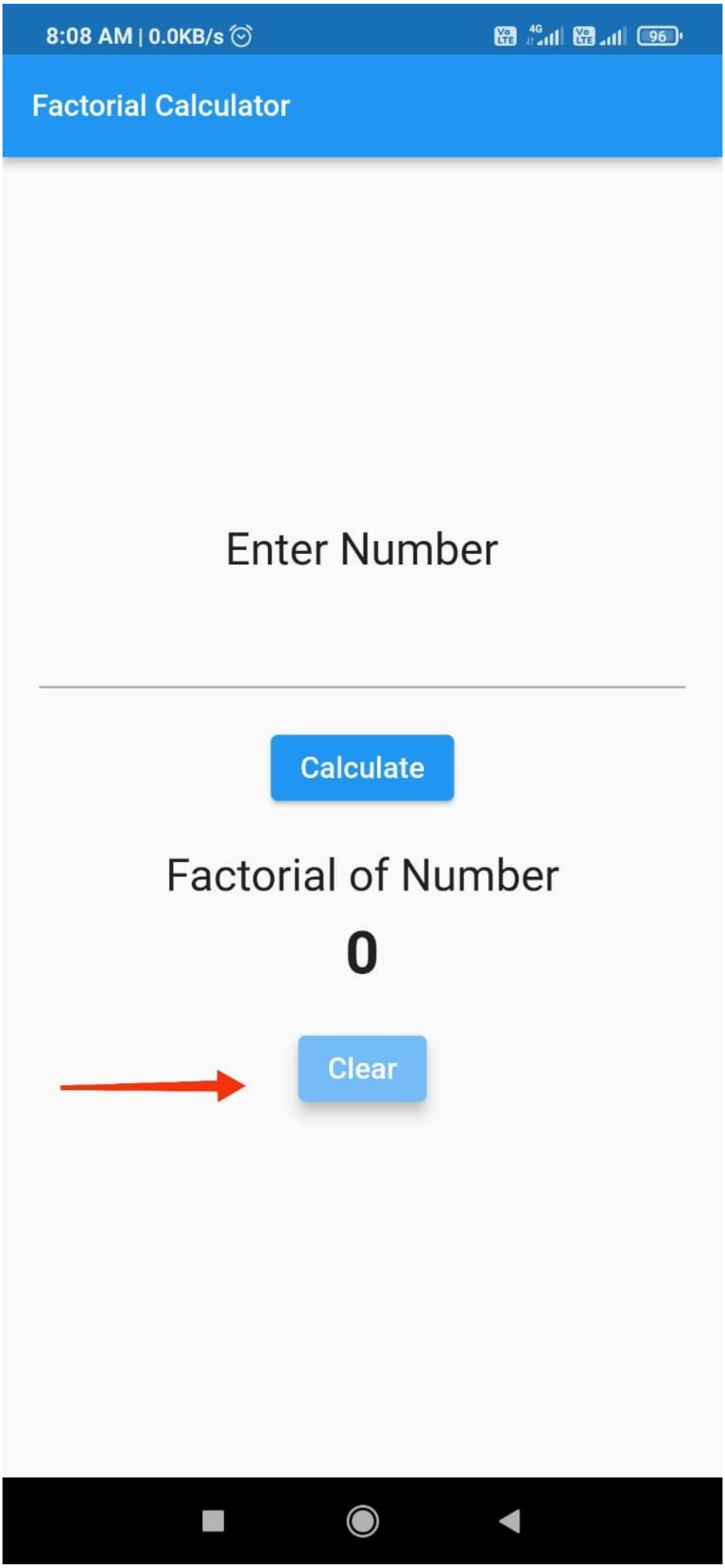
```
        appBar: AppBar(
            title: Text('Factorial Calculator')
        ),
        body: Padding(
          padding: const EdgeInsets.all(20.0),
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              const Text('Enter Number',style:TextStyle(
                  fontSize:30
              ),),
              Container(

                child: TextFormField(
                  textAlign: TextAlign.center,
                  style: TextStyle(
                    fontSize: 40,
                    fontWeight: FontWeight.bold,
                  ),
                  controller: controller,
                  onChanged: (val)=>setState(() {
                    number=val;
                  }),
                ),
              ),
              SizedBox(height: 20,),
              ElevatedButton(
                  child:Text('Calculate',style: TextStyle(
                    fontSize: 20,
                  )),
                  onPressed: (){
                    FocusScopeNode           currentFocus           =
FocusScope.of(context);
                    if (!currentFocus.hasPrimaryFocus) {
                      currentFocus.unfocus();
                    }
                    setState(() {
                      res=factorial(number);
                    });
                  }),
              SizedBox(height:20),
              Text('Factorial of Number',style:TextStyle(
                  fontSize:30
              ),),
              SizedBox(height: 10,),
              Text(
                  '$res',style:TextStyle(
                fontSize: 40,
```

```
                fontWeight: FontWeight.bold,
              )
              ),
              SizedBox(height: 20,),
              ElevatedButton(
                  child:Text('Clear',style: TextStyle(
                    fontSize: 20,
                  ),),
                  onPressed: ()
                  {
                    setState(() {
                      controller.text="";
                      res=0;
                      number="";
                    });
                  }),
            ],
          ),
        )
    );
  }
}
int factorial(String number)
{
  int a=int.parse(number);
  int res=1;
  for(int i=2;i<=a;i++)
    res=res*i;
  return res;
}
```

Result:


8:08 AM | 0.0KB/s
96
Factorial Calculator
Enter Number
Calculate
Factorial of Number
0
Clear


8:08 AM | 0.0KB/s
Factorial Calculator
Enter Number
5
Calculate
Factorial of Number
120
Clear


8:08 AM | 0.0KB/s
Factorial Calculator
Enter Number
Calculate
Factorial of Number
0
Clear