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)

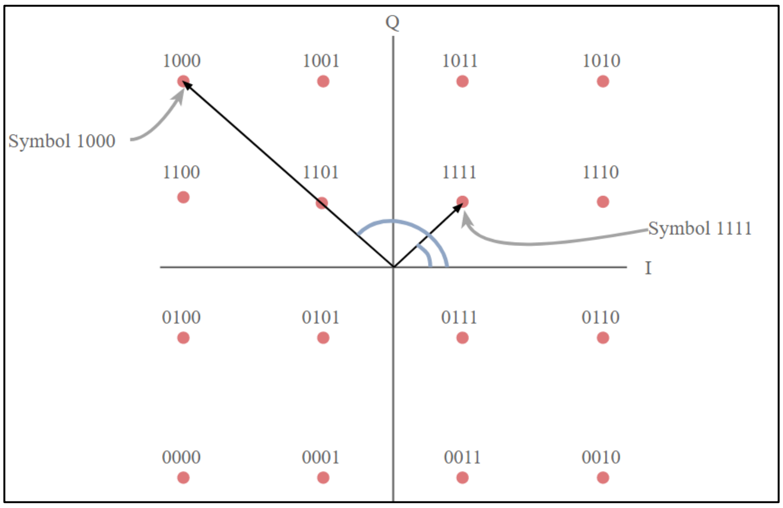
ASK, FSK and BPSK transmit one bit per symbol and hence carrier is assumed to have one of the two possible states to transmit 1 or 0.

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

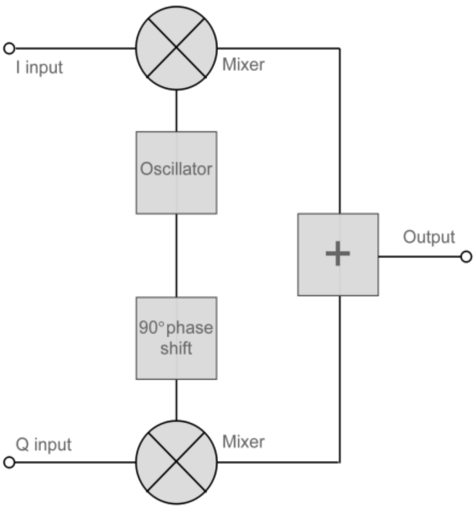
- Represents bits by changing the angle of a wave.
- As seen above, QPSK can have sixteen different phase changes as sixteen different angles.
- It is the angle of the constellation point.

QAM Modulator

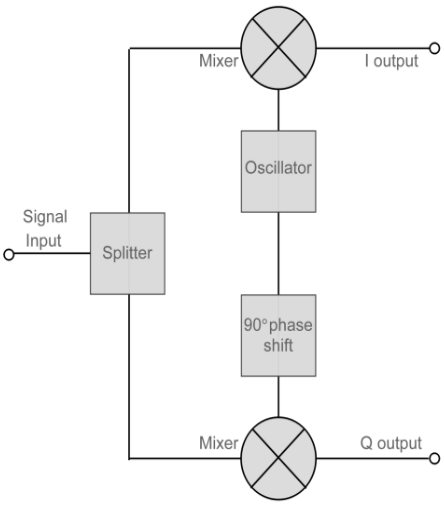
- The QAM modulator essentially follows the idea that can be seen from the basic QAM theory where there are two carrier signals with a phase shift of 90° between them.
- These are then amplitude modulated with the two data streams known as I or In-phase and the Q or quadrature data streams. These are 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
image = imread('cameraman.tif');
figure('name', 'transmitted');
imshow(image);

%% Input Matrix to 1-D Vector
imageR = image(:);
%% Decimal matrix to Binary Matrix
binary = de2bi(imageR);
binaryd = double(binary);
%% Reshaping column matrix
binaryC = binaryd(:);

%% Order of Modulation
M = 64;

%% Performing QAM Modulation
mod = qammod(binaryC, M);
%% Performing QAM Demodulation
```

```
demod = qamdemod(mod, M);

%% Reshaping
f = reshape(demod, [65536, 8]);

%% Double to uint8
g = uint8(f);

%% Binary to Decimal Conversion
h = bi2de(g);

%% Reshaping into pixel matrix
received = reshape(h, 256, 256);

%% Output
figure('name', 'received');
imshow(received);
scatterplot(mod);
scatterplot(demod);
```

2. M-PSK

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

clc;
clear all;
close all;

%% Input
image = imread('cameraman.tif');
figure('name', 'transmitted');
imshow(image);

%% Input Matrix to 1-D Vector
imageR = image(:);
%% Decimal matrix to Binary Matrix
binary = de2bi(imageR);
binaryd = double(binary);
%% Reshaping column matrix
binaryC = binaryd(:);

%% Order of Modulation
M = 64;

%% Performing QAM Modulation
mod = pskmod(binaryC, M);
%% Performing QAM Demodulation
demod = pskdemod(mod, M);
```

```
%% Reshaping
f = reshape(demod, [65536, 8]);

%% Double to uint8
g = uint8(f);

%% Binary to Decimal Conversion
h = bi2de(g);

%% Reshaping into pixel matrix
received = reshape(h, 256, 256);

%% Output
figure('name', 'received');
imshow(received);
scatterplot(mod);
scatterplot(demod);
```

OUTPUT:

1. M-QAM Output:

For M = 4 (QAM)

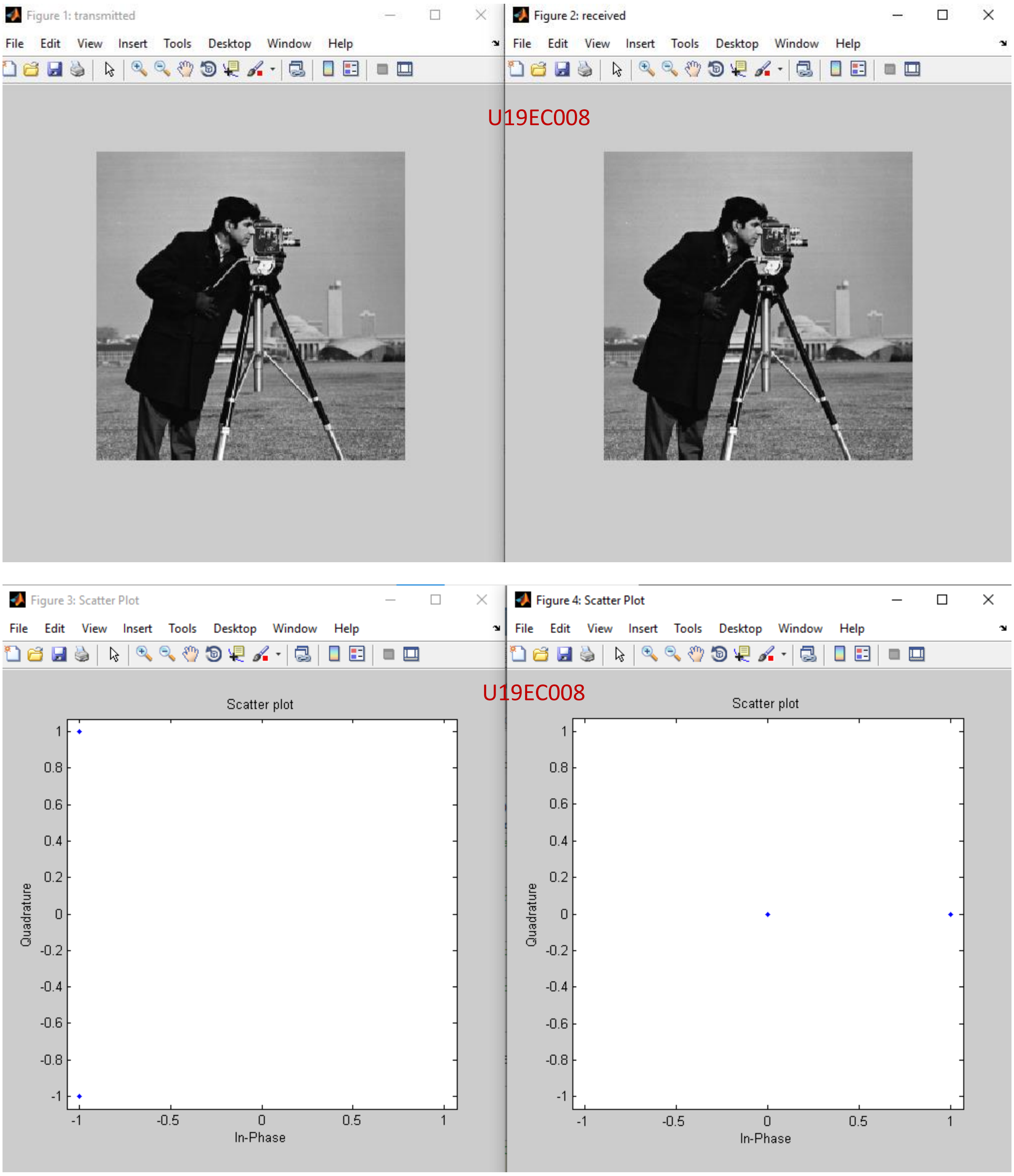


Figure 1: transmitted
File Edit View Insert Tools Desktop Window Help
Figure 2: received
File Edit View Insert Tools Desktop Window Help
U19EC008
Figure 3: Scatter Plot
File Edit View Insert Tools Desktop Window Help
Scatter plot
Quadrature
In-Phase
1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8 -1
-1 -0.5 0 0.5 1
Figure 4: Scatter Plot
File Edit View Insert Tools Desktop Window Help
U19EC008
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

For M = 8 (QAM)

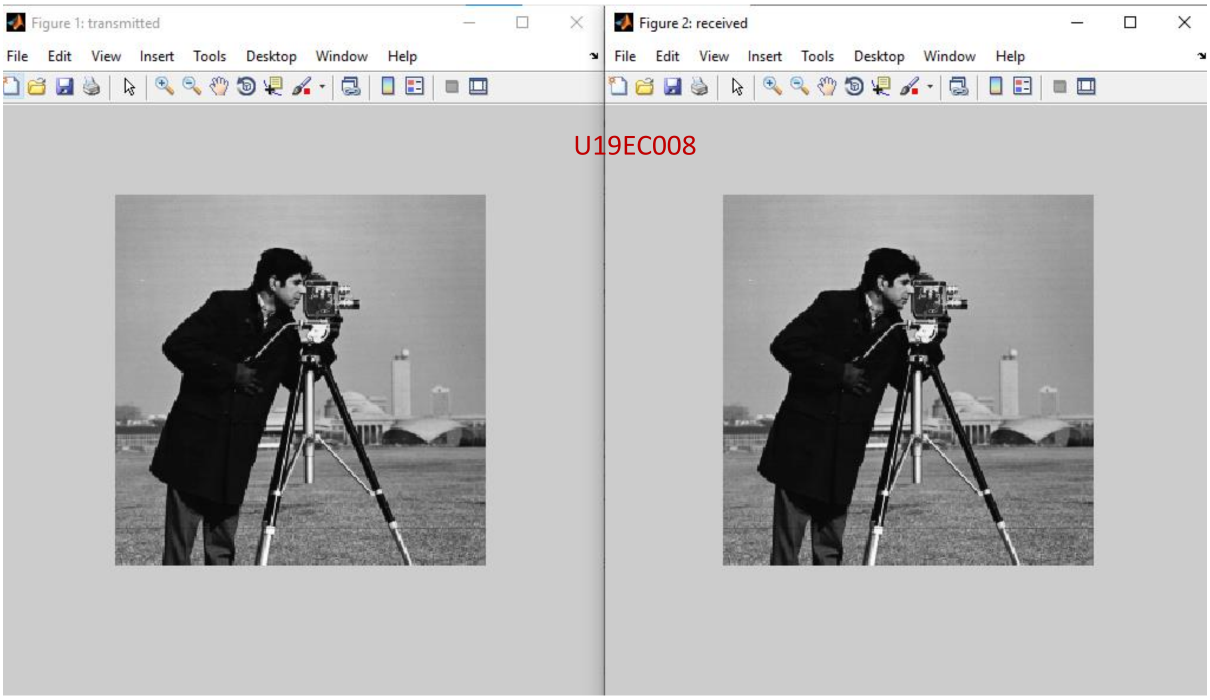


Figure 1: transmitted
Figure 2: received
U19EC008

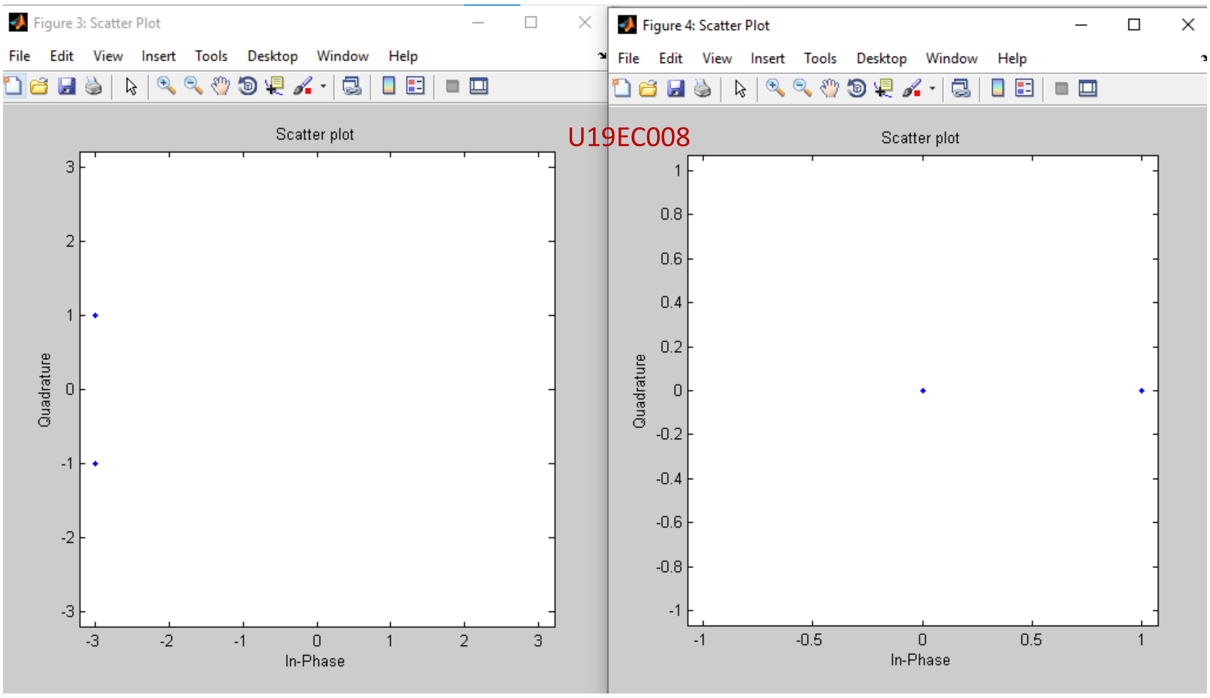


Figure 3: Scatter Plot
Figure 4: Scatter Plot
U19EC008
Scatter plot
Quadrature
In-Phase
Scatter plot
Quadrature
In-Phase

For M = 16 (QAM)

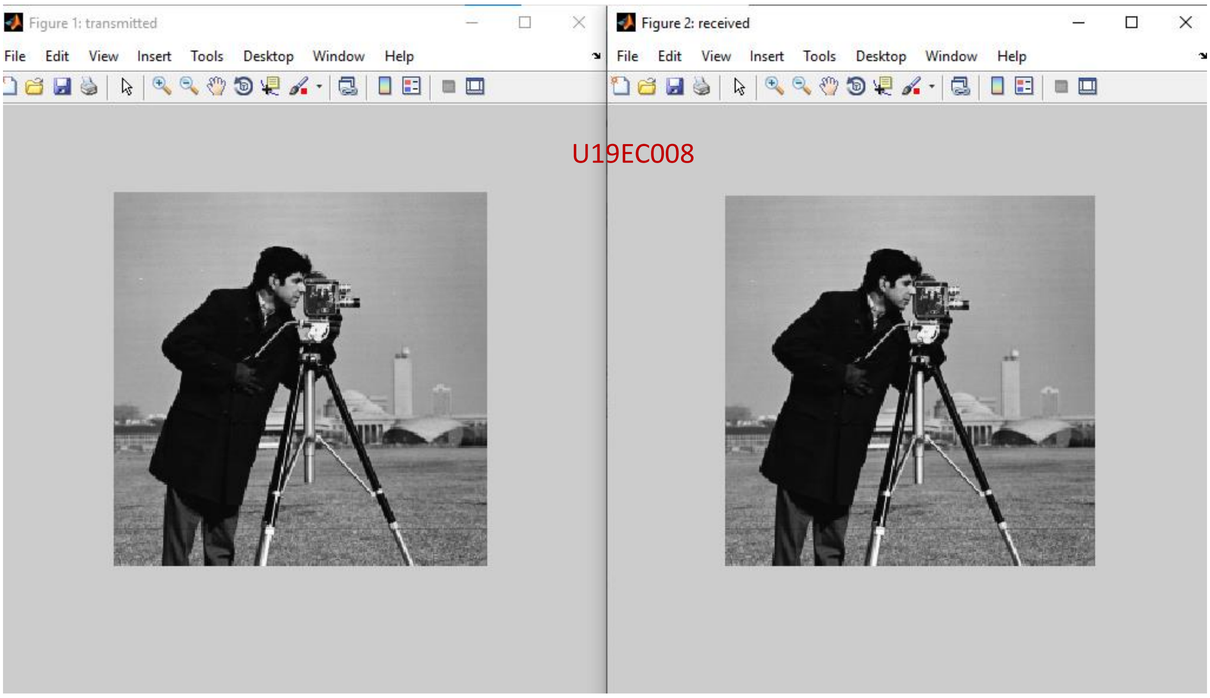


Figure 1: transmitted
File Edit View Insert Tools Desktop Window Help
Figure 2: received
File Edit View Insert Tools Desktop Window Help
U19EC008

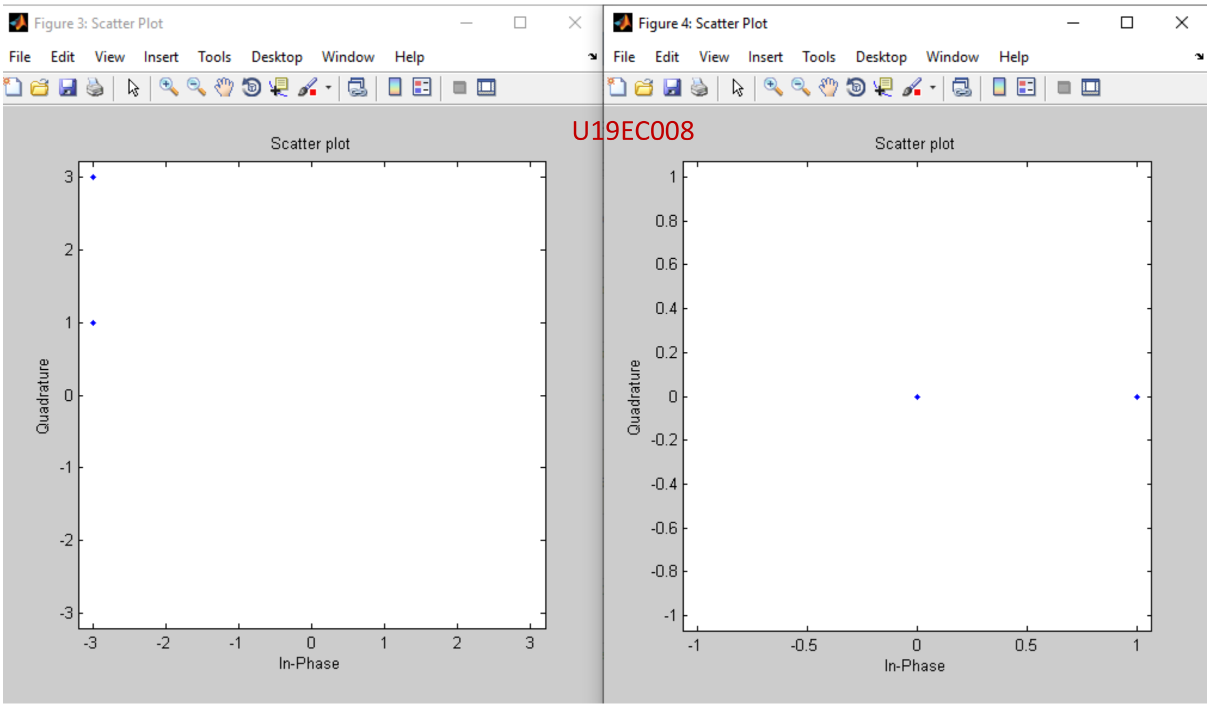


Figure 3: Scatter Plot
File Edit View Insert Tools Desktop Window Help
Scatter plot
Quadrature
In-Phase
3
2
1
0
-1
-2
-3
Figure 4: Scatter Plot
File Edit View Insert Tools Desktop Window Help
U19EC008
Scatter plot
Quadrature
In-Phase
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0.5

For M = 32 (QAM)

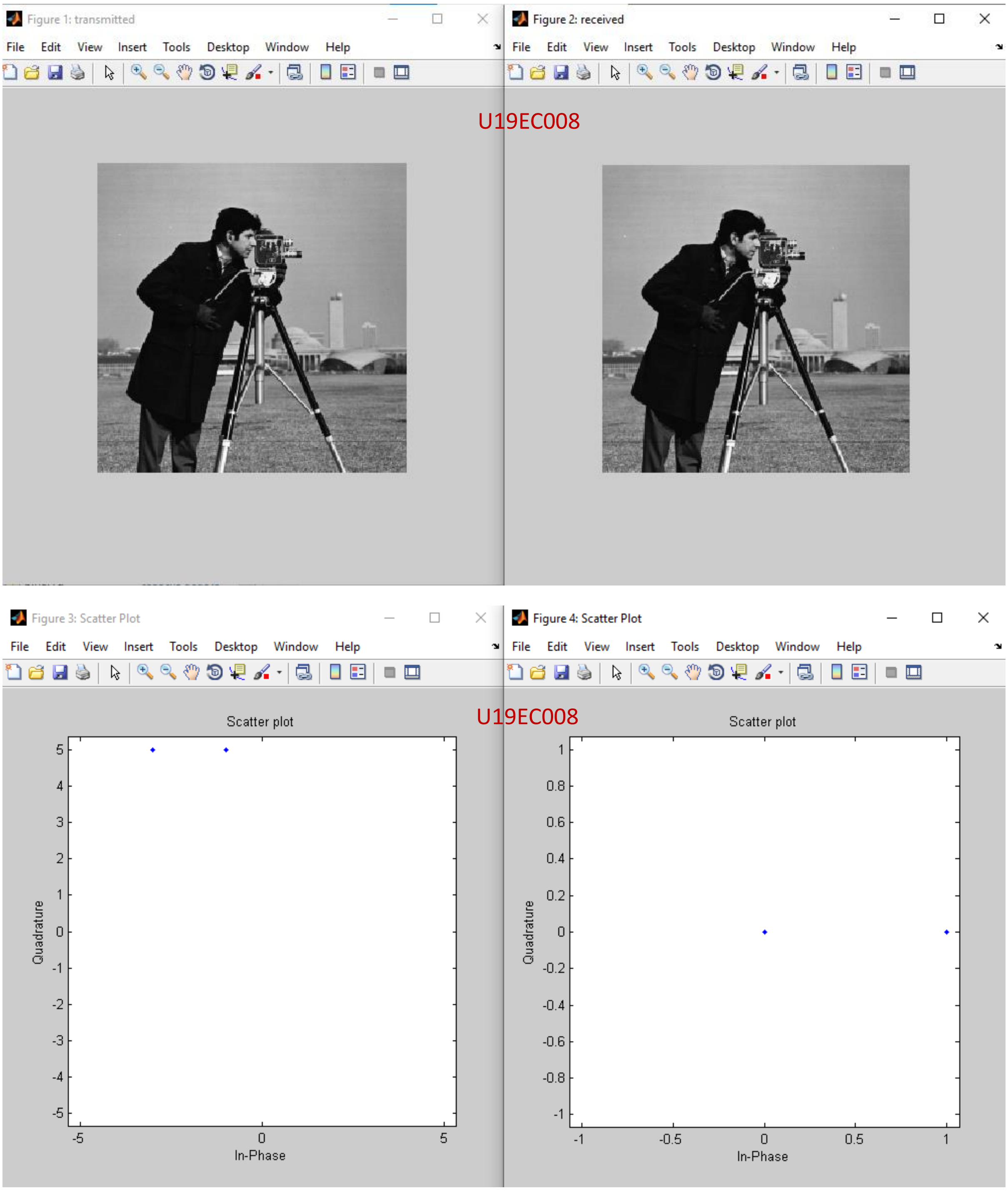


Figure 1: transmitted
Figure 2: received
U19EC008
Figure 3: Scatter Plot
Figure 4: Scatter Plot
U19EC008
Scatter plot
Quadrature
In-Phase
Scatter plot
Quadrature
In-Phase

For M = 64 (QAM)


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

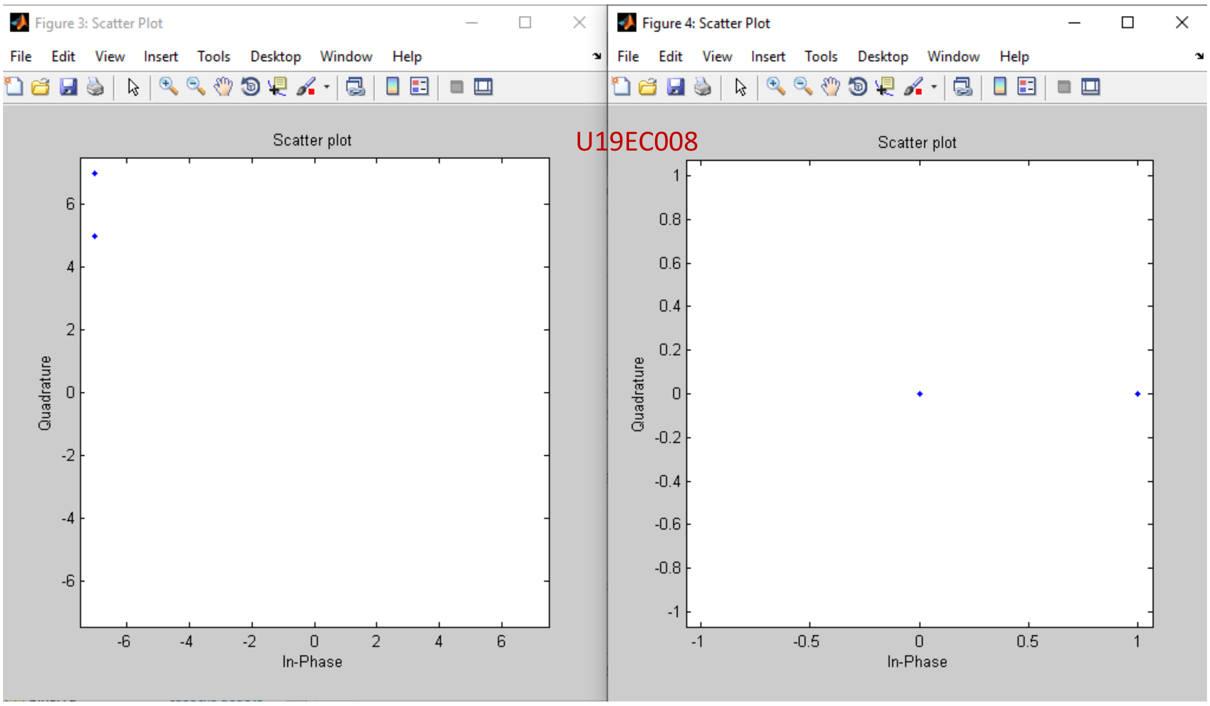
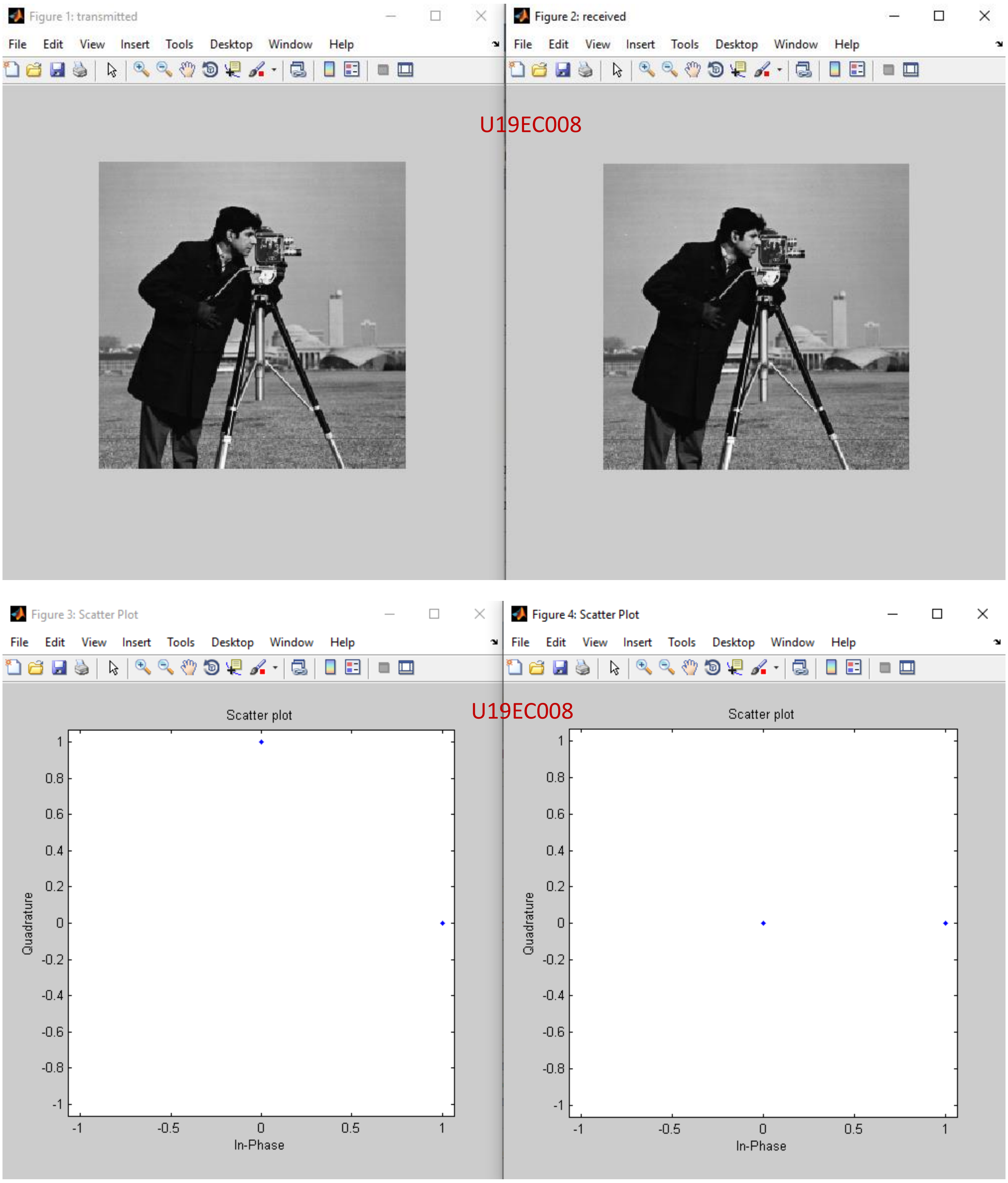


Figure 3: Scatter Plot
File Edit View Insert Tools Desktop Window Help
Scatter plot
Quadrature
In-Phase
6
4
2
0
-2
-4
-6
Figure 4: Scatter Plot
File Edit View Insert Tools Desktop Window Help
U19EC008
Scatter plot
Quadrature
In-Phase
1
0.8
0.6
0.4
0.2
-0.2
-0.4
-0.6
-0.8
-1
-0.5
0.5

2. M-PSK Output:

For M = 4 (PSK)



For M = 8 (PSK)

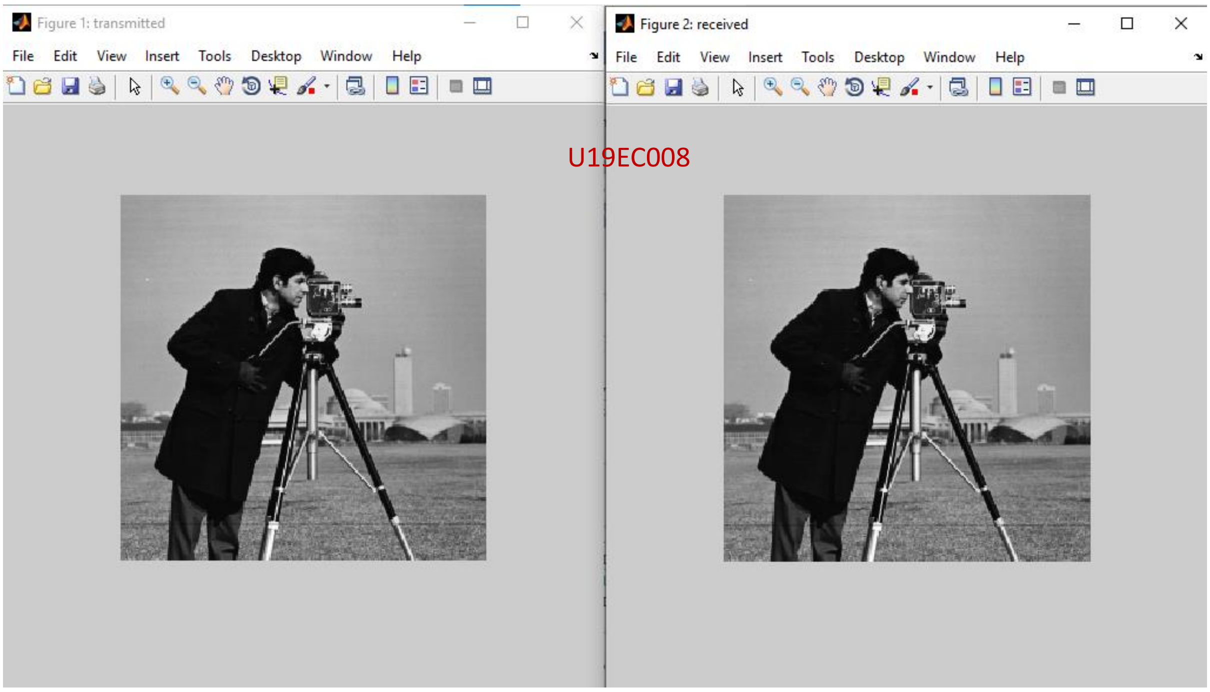


Figure 1: transmitted
File Edit View Insert Tools Desktop Window Help
Figure 2: received
File Edit View Insert Tools Desktop Window Help
U19EC008

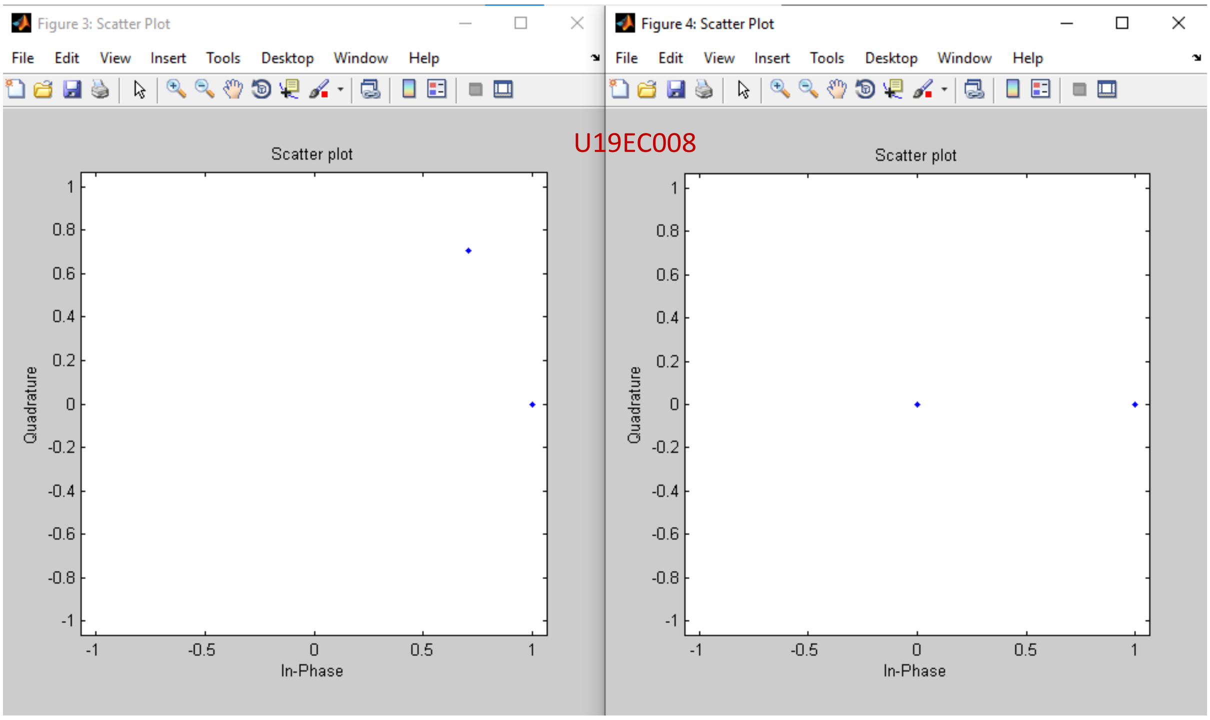


Figure 3: Scatter Plot
File Edit View Insert Tools Desktop Window Help
Figure 4: Scatter Plot
File Edit View Insert Tools Desktop Window Help
U19EC008
Scatter plot
Quadrature
1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8 -1
-1 -0.5 0 0.5 1
In-Phase
Scatter plot
Quadrature
1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8 -1
-1 -0.5 0 0.5 1
In-Phase

For M = 16 (PSK)

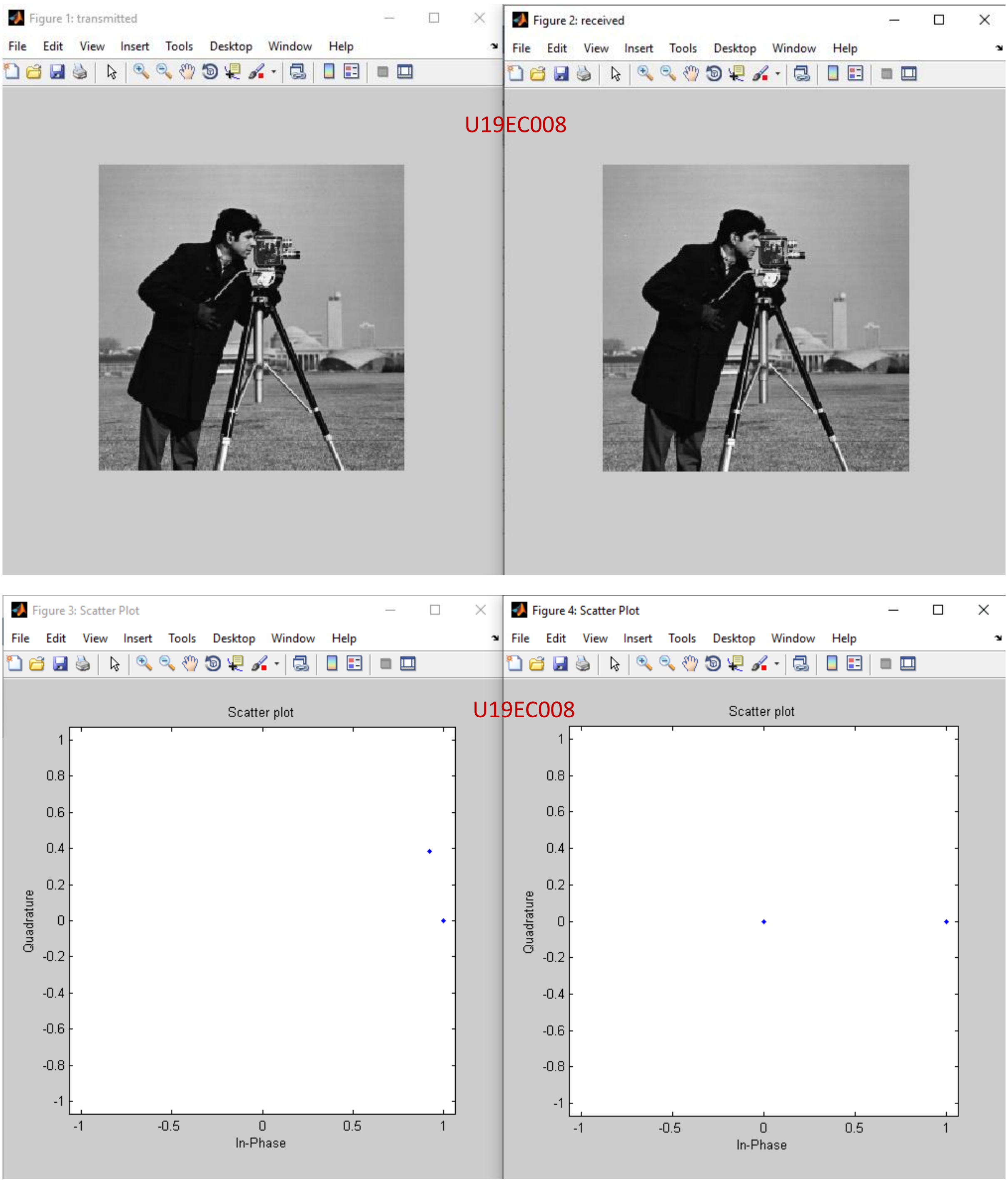


Figure 1: transmitted
Figure 2: received
U19EC008
Figure 3: Scatter Plot
Figure 4: Scatter Plot
U19EC008
Scatter plot
Scatter plot
Quadrature
In-Phase
Quadrature
In-Phase

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

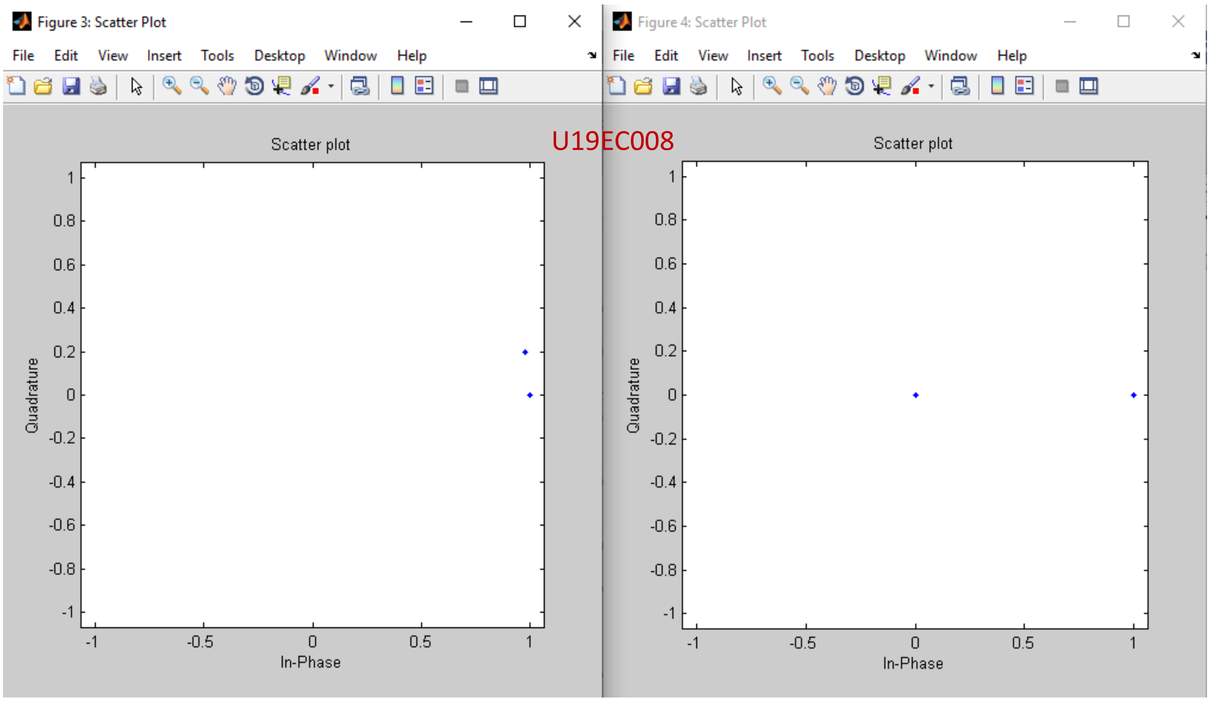
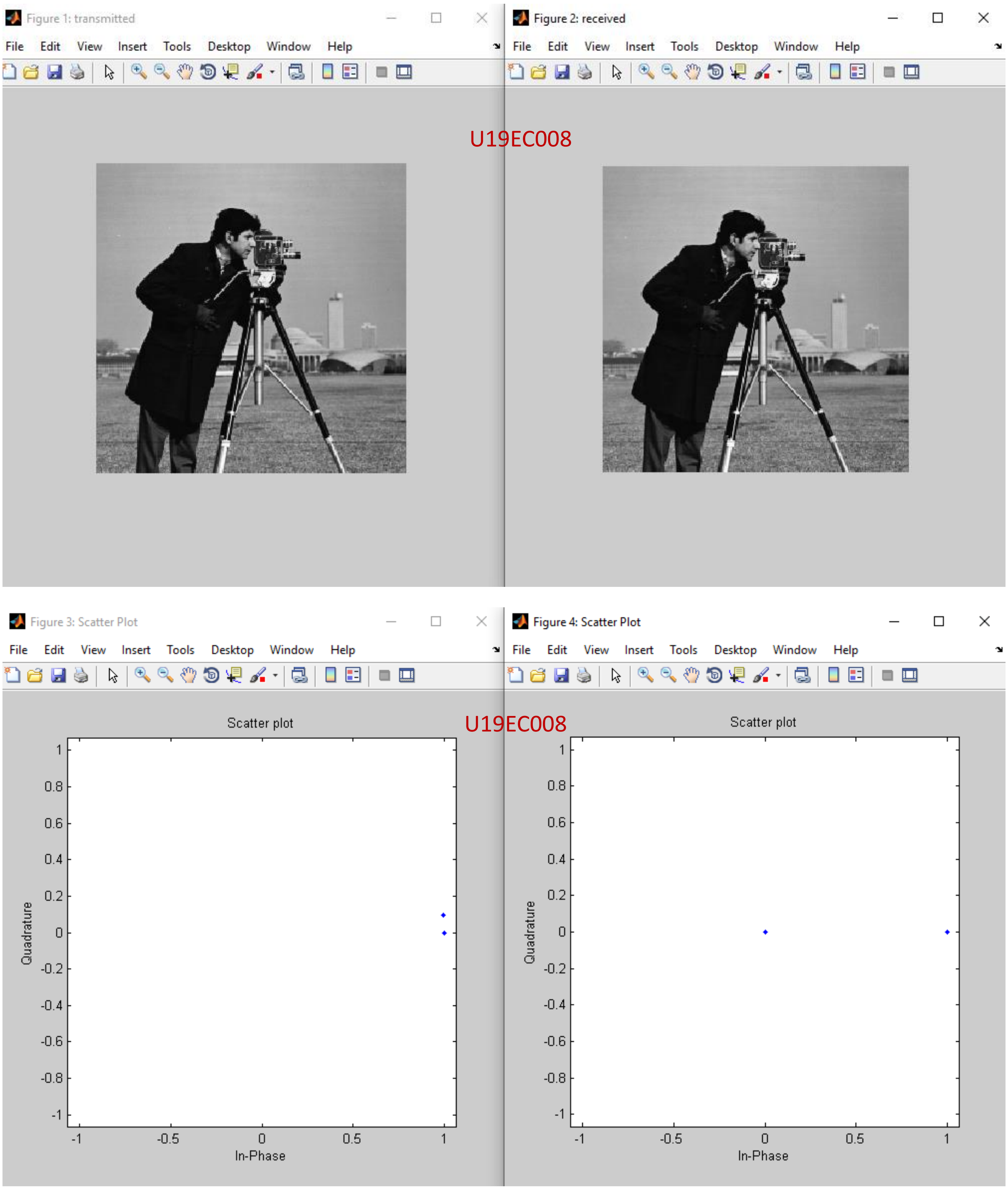


Figure 3: Scatter Plot
File Edit View Insert Tools Desktop Window Help
Scatter plot
Quadrature
In-Phase
1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8 -1
-1 -0.5 0 0.5 1
U19EC008
Figure 4: Scatter Plot
File Edit View Insert Tools Desktop Window Help
Scatter plot
Quadrature
In-Phase
1 0.8 0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8 -1
-1 -0.5 0 0.5 1

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.