****

**MARKS:**

**AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH (AIUB)**

**Data Communication Laboratory**

**LAB PROJECT**

.

**Section: [G]**

**Semester: Spring 20-21**

**Course Teacher: MD MEHEDI HASAN**

**Student Name:** DEBORAJ ROY

**Student ID:** 19-40158-1

**Data Communication**

Lab Project

Develop a communication system using MATLAB/Octave that will send and receive images in a form of analog signal. Consider 10X10 pixel color images. You will require 24 bits to represent 1 pixel. The system uses synchronous transmission considering QPSK modulation and demodulation.

1. Show the transmuted bits at the sender.

2. Show the analog signals after modulation.

3. Show received signal considering AWGN channel.

4. Demonstrate your system with an input image (show the output image).

Hint: You should create a function to encode the 10X10 pixel color image into binary bit sequence. Then, convert the bit sequence to analog signal and add noise. Finally, decode at the receiver.

**Function:**

function DEB = imageTobinConverter(image)

figure

subplot(4,1,1);

imshow(image);

title('Original Image');

subplot(4,1,2);

grimage = rgb2gray(image);

grimageAdj = imadjust(grimage);

title('GrayScale Image');

binimage = imbinarize(grimageAdj);

subplot(4,1,3);

imshow(binimage)

title('Binary Image');

s = sum(binimage,2);

subplot(4,1,4);

plot(s)

title('Sum of columns');

disp("Original binary matrix for Image:");

disp(binimage);

DEB = reshape(binimage, 1, numel(binimage));

end

**Code:**

% DEBORAJ ROY 19-40158-1.

clc;

close all;

workspace;

fontSize = 14;

image = imread("40158.png"); % can use any size image.

binaryData = imageTobinConverter(image);

disp("Message transmitter: ");

figure

stem(binaryData, 'Linewidth',1), grid on;

title('Information before Transmiting ');

axis([ 0 99 0 1.5]);

disp(binaryData);

size(binaryData);

data\_NZR=2\*binaryData-1;

SPData=reshape(data\_NZR,2,length(binaryData)/2);

br=10^6;

f=br;

T=1/br;

t=T/99:T/99:T;

y=[];

yInPhase=[];

yQuadrature=[];

rcvddata=image;

for i=1:length(binaryData)/2

y1=SPData(1,i)\*cos(2\*pi\*f\*t);

y2=SPData(2,i)\*sin(2\*pi\*f\*t) ;

yInPhase=[yInPhase y1];

yQuadrature=[yQuadrature y2];

y=[y y1+y2];

end

transmittedSignal=awgn(y,10);

tt=T/99:T/99:(T\*length(binaryData))/2;

figure

subplot(3,1,1);

plot(tt,y\_inPhase,'Linewidth',3), grid on;

title('QPSK modulation');

xlabel('time(sec)');

ylabel('Amplitude(volt0');

subplot(3,1,2);

plot(tt,y\_quadrature,'Linewidth',3), grid on;

title('QPSK modulation ');

xlabel('time(sec)');

ylabel('Amplitude(volt0');

subplot(3,1,3);

plot(tt,transmittedSignal,'r','Linewidth',3), grid on;

title('QPSK modulated signal (sum of inphase and Quadrature phase signal)');

xlabel('time(sec)');

ylabel('Amplitude(volt0');

receivedData=[];

receivedSignal=transmittedSignal;

for i=1:1:length(binaryData)/2

ZInPhase=receivedSignal((i-1)\*length(t)+1:i\*length(t)).\*cos(2\*pi\*f\*t);

ZInPhase\_intg=(trapz(t,ZInPhase))\*(2/T);

if(ZInPhase\_intg>0)

receivedInphaseData=1;

else

receivedInphaseData=0;

end

Quadrature=receivedSignal((i-1)\*length(t)+1:i\*length(t)).\*sin(2\*pi\*f\*t);

Quadrature\_intg=(trapz(t,Quadrature))\*(2/T);

if (Quadrature\_intg>0)

receivedQuadratureData=1;

else

receivedQuadratureData=0;

end

receivedData=[receivedData receivedInphaseData receivedQuadratureData]; % Received Data vector

end

figure

stem(receivedData,'Linewidth',1);

title('Information after Receiveing ');

axis([ 0 99 0 1.5]), grid on;

figure

subplot(2,1,1);

imshow(rcvddata);

title('Output Image');

De=reshape(receivedData,10,[]);

disp("binary matrix after receiving & demodulation");

disp(De);

GrayImage = uint8(255 \*receivedData);















