PES University, Bangalore

UE21EC242B- DIGITAL COMMUNICATION

QPBSK and QAM modulation and demodulation techniques using MATLAB:

Session: Jan-May 2023

Branch: ELECTRONICS AND COMMUNICATION ENGINEERING

Semester &Section:4TH SEM , A SECTION

Sl No.	Name of the Student	SRN	Marks Allotted (Out of 5)
1.	AMOGH S RAO	PES1UG21EC033	
2.	ADITYA SHARMA	PES1UG21EC018	
3.	AKSHATH V.A.	PES1UG21EC028	
4.	AKASH RAVI BHAT	PES1UG21EC025	

Name of the Course Instructor : **Prof. BHARATHI V. KALGHATGI**

Signature of the Course Instructor(with Date) :

QPBSK and QAM modulation and demodulation techniques using MATLAB

Introduction

Digital Modulation (DM) is a process of encoding a digital information signal into the amplitude, phase, or frequency of the transmitted signal. It removes communication noise as well as provides enhanced strength for the signal intrusion and can be considered as digital-to-analog conversion and the corresponding demodulation or detection as analog-to-digital conversion.
In the following project, we aim to simulate and analyse two of the most widely used digital modulation techniques : QPSK and QAM using MATLAB. We wish to simulate generation of signals using these modulation techniques and then simulate their transmission through a noisy channel. Finally, demodulate the received signal and evaluate the bit error rate (BER) performance of each modulation technique.

Methodology

Generation of digital data: Generating a stream of digital data that will be modulated and transmitted. This can be done using the MATLAB randi() function to generate a random sequence of bits.

Implementation of QPSK and QAM modulation using MATLAB: The code includes generating the carrier signal, modulating the signal with the digital data, and transmitting the signal over a communication channel. MATLAB functions like qammod() are used.

Add noise to the channel: Introducing noise to the communication channel to simulate real-world conditions. MATLAB functions like awgn() can be used to add Gaussian noise to the modulated signal.

Implement QPSK and QAM demodulation: Programming a MATLAB code to implement QPSK and QAM demodulation. It includes receiving the modulated signal, extracting the carrier signal, demodulating the signal to extract the digital data, and outputting the decoded data. MATLAB functions like qamdemod() are used for this.

Testing the implementation: Testing the implementation by transmitting and receiving signals with known digital data. Compare the transmitted and received data to verify that the implementation is correct. MATLAB functions like biterr() are used to calculate the bit error rate.

QBPSK Modulation:

- In Quadrature Binary Phase-Shift Keying (QBPSK), two bits of digital data are mapped to a single symbol with four possible phase states (0, 90, 180, and 270 degrees).
- The carrier signal, usually a sine wave, is modulated by changing the phase of the signal to represent the digital data.
- The modulated signal is transmitted over the communication channel.

QBPSK Demodulation:

- The received signal is mixed with a local oscillator signal to extract the modulated signal.
- The extracted signal is passed through a low-pass filter to remove any high-frequency noise.
- The resulting signal is then sampled and compared to the four possible phase states to extract the digital data.

QAM Modulation:

- In Quadrature Amplitude Modulation (QAM), the digital data is mapped to a constellation of points in the amplitude and phase domain of the carrier signal.
- The carrier signal, usually a square wave, is modulated by changing both the amplitude and phase of the signal to represent the digital data.
- The modulated signal is transmitted over the communication channel.

QAM Demodulation:

- The received signal is mixed with a local oscillator signal to extract the modulated signal.
- The extracted signal is passed through a low-pass filter to remove any high-frequency noise.
- The resulting signal is then sampled and compared to the constellation points to extract the digital data.

Both QBPSK and QAM can be demodulated using a coherent receiver, which means that the receiver uses the same carrier signal as the transmitter. The receiver then extracts the digital data from the modulated signal using a demodulator circuit.

MATLAB CODE

QBPSK

```
clc;
clear all;
close all;
data=[0  1 0 1 1 1 0 0 1 1]; % information

%Number_of_bit=1024;
%data=randint(Number_of_bit,1);

figure(1)
```

```
stem(data, 'linewidth',3), grid on;
title('  Information before Transmiting ');
axis([ 0 11 0 1.5]);

data_NZR=2*data-1; % Data Represented at Polar form for QPSK modulation
s_p_data=reshape(data_NZR,2,length(data)/2);  % S/P convertion of data


br=10.^6; %Let us transmission bit rate  1000000
f=br; % minimum carrier frequency
T=1/br; % bit duration
t=T/99:T/99:T; % Time vector for one bit information



% XXXXXXXXXXXXXXXXXXXXXXX QPSK modulatio
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
y=[];
y_in=[];
y_qd=[];
for(i=1:length(data)/2)
   y1=s_p_data(1,i)*cos(2*pi*f*t); % inphase component
   y2=s_p_data(2,i)*sin(2*pi*f*t) ;% Quadrature component
   y_in=[y_in y1]; % inphase signal vector
   y_qd=[y_qd y2]; %quadrature signal vector
   y=[y y1+y2]; % modulated signal vector
end
Tx_sig=y; % transmitting signal after modulation
tt=T/99:T/99:(T*length(data))/2;

figure(2)

subplot(3,1,1);
plot(tt,y_in,'linewidth',3), grid on;
title(' wave form for inphase component in QPSK modulation ');
xlabel('time(sec)');
ylabel(' amplitude(volt0');

subplot(3,1,2);
plot(tt,y_qd,'linewidth',3), grid on;
title(' wave form for Quadrature component in QPSK modulation ');
xlabel('time(sec)');
ylabel(' amplitude(volt0');


subplot(3,1,3);
plot(tt,Tx_sig,'r','linewidth',3), grid on;
title('QPSK modulated signal (sum of inphase and Quadrature phase signal)');
```

```
xlabel('time(sec)');
ylabel(' amplitude(volt0');

%QPSK demodulation
Rx_data=[];
Rx_sig=Tx_sig; % Received signal
for(i=1:1:length(data)/2)

   %%XXXXXX inphase coherent dector XXXXXXX
   Z_in=Rx_sig((i-1)*length(t)+1:i*length(t)).*cos(2*pi*f*t);
   % above line indicat multiplication of received & inphase carred signal

   Z_in_intg=(trapz(t,Z_in))*(2/T);% integration using trapizodial rull
   if(Z_in_intg>0) % Decession Maker
      Rx_in_data=1;
   else
     Rx_in_data=0;
   end

   %%XXXXXX Quadrature coherent dector XXXXXX
   Z_qd=Rx_sig((i-1)*length(t)+1:i*length(t)).*sin(2*pi*f*t);
   %above line indicat multiplication ofreceived & Quadphase carred signal

   Z_qd_intg=(trapz(t,Z_qd))*(2/T);%integration using trapizodial rull
      if (Z_qd_intg>0)% Decession Maker
      Rx_qd_data=1;
      else
     Rx_qd_data=0;
      end


      Rx_data=[Rx_data  Rx_in_data  Rx_qd_data]; % Received Data vector
end


figure(3)
stem(Rx_data,'linewidth',3)
title('Information after Receiveing ');
axis([ 0 11 0 1.5]), grid on;
```

QAM

```
clc;
clear all;
close all;
```

```
M=4;
%M=input(' enter the value of M array for QAM modulation : ');
fprintf('\n\n\n');
%input  chaking loop
Ld=log2(M);
ds=ceil(Ld);
dif=ds-Ld;
if(dif~=0)
  error('the value of M is only acceptable if log2(M)is an integer');
end
%binary Information Generation
nbit=16;                                %number of information bits
msg=round(rand(nbit,1));           % information generation as binary form
disp(' binary information at transmitter ');
disp(msg);
fprintf('\n\n');
%representation of transmitting binary information as digital signal
x=msg;
bp=.000001;                                          % bit period
bit=[];
for n=1:1:length(x)
   if x(n)==1;
     se=ones(1,100);
   else x(n)==0;
      se=zeros(1,100);
   end
    bit=[bit se];
end
t1=bp/100:bp/100:100*length(x)*(bp/100);
figure(1)
subplot(3,1,1);
plot(t1,bit,'lineWidth',2.5);grid on;
axis([ 0 bp*length(x) -.5 1.5]);
ylabel('amplitude(volt)');
xlabel(' time(sec)');
title('transmitting information as digital signal');
% binary information convert into symbolic form for M-array QAM modulation
M=M;                                    % order of QAM modulation
msg_reshape=reshape(msg,log2(M),nbit/log2(M))';
disp(' information are reshaped for convert symbolic form');
disp(msg_reshape);
fprintf('\n\n');
size(msg_reshape);
for(j=1:1:nbit/log2(M))
  for(i=1:1:log2(M))
     a(j,i)=num2str(msg_reshape(j,i));
  end
end
```

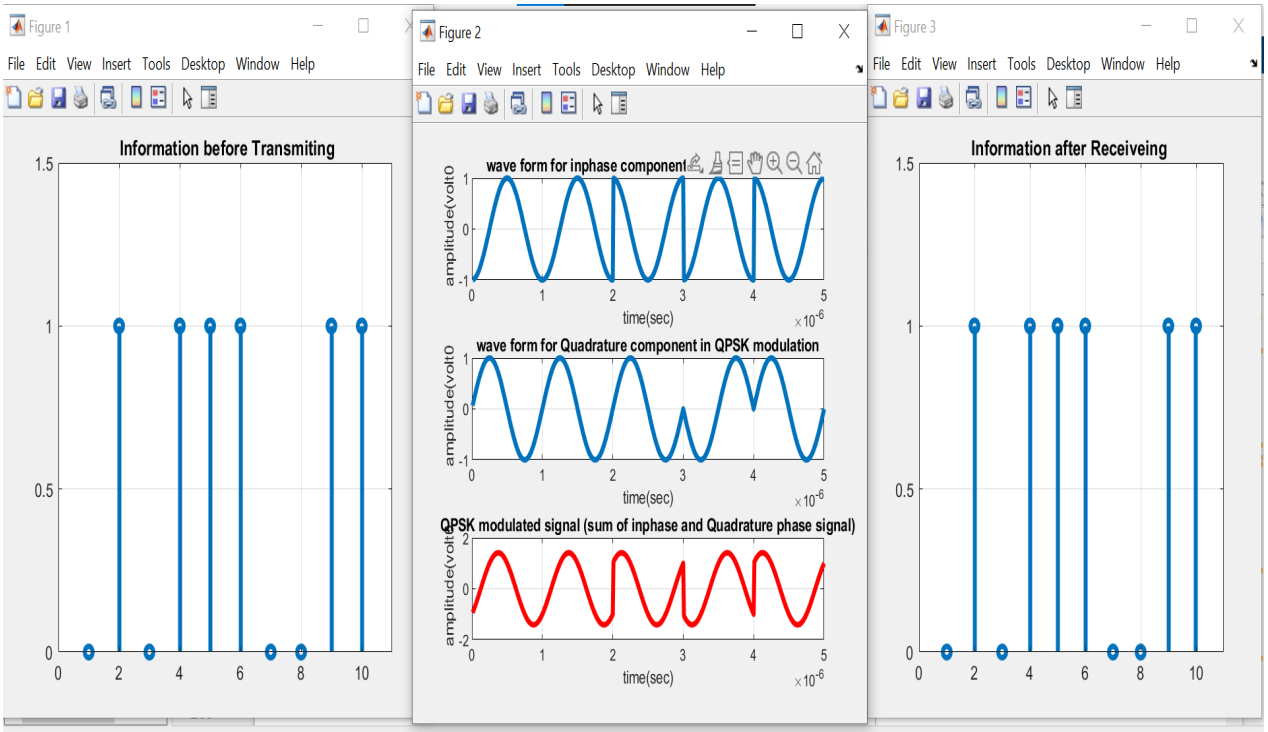
```
as=bin2dec(a);
ass=as';
figure(1)
subplot(3,1,2);
stem(ass,'Linewidth',2.0);
title('serial symbol for M-array QAM modulation at transmitter');
xlabel('n(discrete time)');
ylabel(' magnitude');
disp('symbolic form information for M-array QAM ');
disp(ass);
fprintf('\n\n');
%Mapping for M-array QAM modulation
M=M;                                    %order of QAM modulation
x1=[0:M-1];
p=qammod(ass,M)     %constalation design for M-array QAM acording to symbol
sym=0:1:M-1;     % considerable symbol of M-array QAM, just for scatterplot
pp=qammod(sym,M);                   %constalation diagram for M-array QAM
scatterplot(pp),grid on;
title('consttelation diagram for M-array QAM');
%  M-array QAM modulation
RR=real(p)
II=imag(p)
sp=bp*2;                                %symbol period for M-array QAM
sr=1/sp;                                        % symbol rate
f=sr*2;
t=sp/100:sp/100:sp;
ss=length(t);
m=[];
for(k=1:1:length(RR))
   yr=RR(k)*cos(2*pi*f*t);                  % inphase or real component
   yim=II(k)*sin(2*pi*f*t);           % Quadrature or imagenary component
   y=yr+yim;
   m=[m y];
end
tt=sp/100:sp/100:sp*length(RR);
figure(1);
subplot(3,1,3);
plot(tt,m);
title('waveform for M-array QAM modulation acording to symbolic information');
xlabel('time(sec)');
ylabel('amplitude(volt)');
%M-array QAM demodulation
m1=[];
m2=[];
for n=ss:ss:length(m)
  t=sp/100:sp/100:sp;
  y1=cos(2*pi*f*t);                         % inphase component
  y2=sin(2*pi*f*t);                        % quadrature component
```

```
  mm1=y1.*m((n-(ss-1)):n);
  mm2=y2.*m((n-(ss-1)):n);
  z1=trapz(t,mm1)                                % integration
  z2=trapz(t,mm2)                                % integration
  zz1=round(2*z1/sp)
  zz2=round(2*z2/sp)
  m1=[m1 zz1]
  m2=[m2 zz2]
end
%de-mapping for M-array QAM modulation
clear i;
clear j;
for (k=1:1:length(m1))
gt(k)=m1(k)+j*m2(k);
end
gt
ax=qamdemod(gt,M);
figure(3);
subplot(2,1,1);
stem(ax,'linewidth',2);
title(' re-obtain symbol after M-array QAM demodulation ');
xlabel('n(discrete time)');
ylabel(' magnitude');
disp('re-obtain symbol after M-array QAM demodulation ');
disp(ax);
fprintf('\n\n');
bi_in=dec2bin(ax);
[row col]=size(bi_in);
p=1;
 for(i=1:1:row)
     for(j=1:1:col)
         re_bi_in(p)=str2num(bi_in(i,j));
         p=p+1;
     end
 end
disp('re-obtain binary information after M-array QAM demodulation');
disp(re_bi_in')
fprintf('\n\n');
% representation of receiving binary information as digital signal
x=re_bi_in;
bp=.000001;                                    % bit period
bit=[];
for n=1:1:length(x)
    if x(n)==1;
       se=ones(1,100);
    else x(n)==0;
        se=zeros(1,100);
    end
```

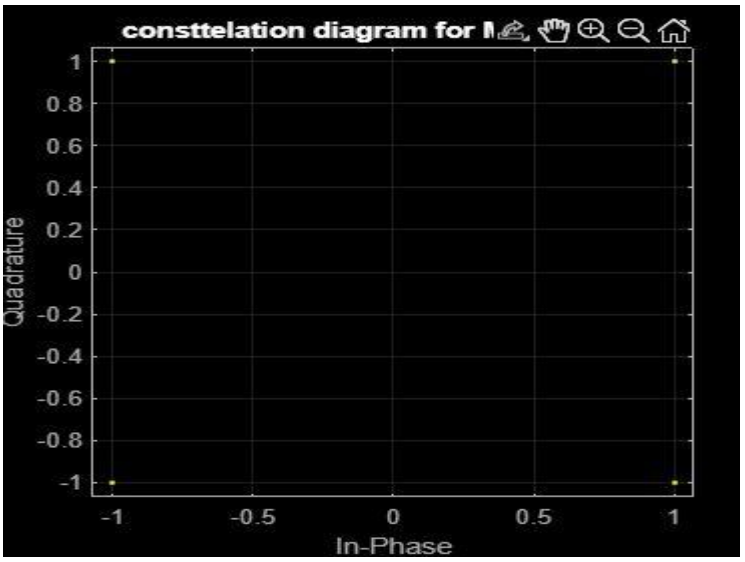
```
    bit=[bit se];
end
t1=bp/100:bp/100:100*length(x)*(bp/100);
figure(3)
subplot(2,1,2);
plot(t1,bit,'lineWidth',2.5);grid on;
axis([ 0 bp*length(x) -.5 1.5]);
ylabel('amplitude(volt)');
xlabel(' time(sec)');
title('receiving information as digital signal after M-array QAM demoduation');
% end of program
```

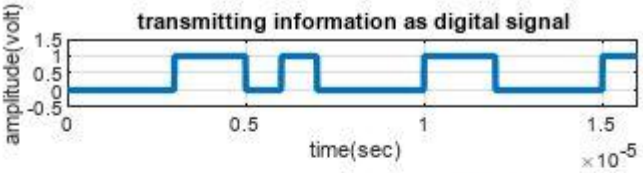
OUTPUT

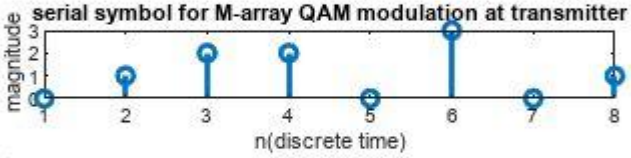
QBPSK

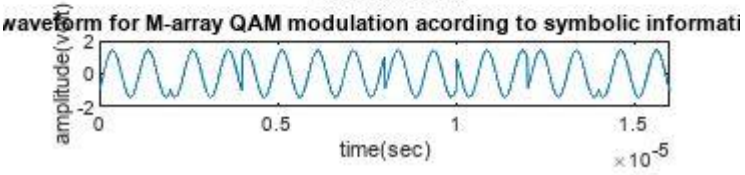


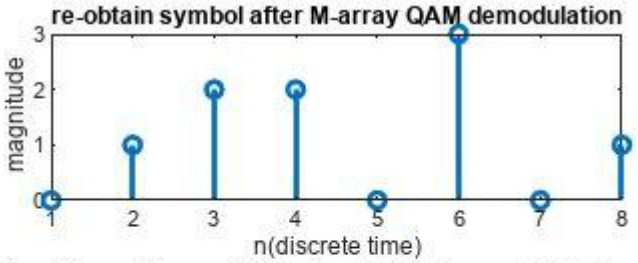
QAM

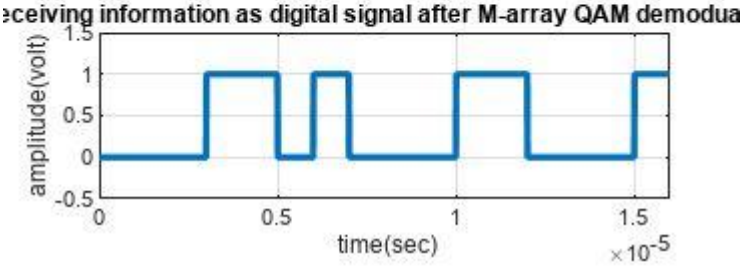

consttelation diagram for M
Quadrature
In-Phase


transmitting information as digital signal
amplitude(volt)
time(sec)
×10^-5


serial symbol for M-array QAM modulation at transmitter
magnitude
n(discrete time)


waveform for M-array QAM modulation acording to symbolic informati
amplitude(volt)
time(sec)
×10^-5


re-obtain symbol after M-array QAM demodulation
magnitude
n(discrete time)


eceiving information as digital signal after M-array QAM demodua
amplitude(volt)
time(sec)
×10^-5

QBPSK AND QAM MODULATION ADVANTAGES

Here are some advantages of QBPSK and QAM modulation and demodulation:

Higher data rates: Both QBPSK and QAM modulation schemes offer higher data rates compared to traditional modulation schemes like BPSK and QPSK. This is because they can transmit multiple bits per symbol.

Better spectral efficiency: QBPSK and QAM modulation schemes are more spectrally efficient compared to traditional modulation schemes. This means that they can transmit more information per unit of bandwidth, which can result in better overall system performance.

Robustness to noise: QBPSK and QAM modulation schemes are more robust to noise compared to traditional modulation schemes. This is because they can distinguish between multiple levels of amplitude and phase, which makes it easier to detect the transmitted signal even in the presence of noise.

Lower power consumption: QBPSK and QAM modulation schemes typically require lower power consumption compared to traditional modulation schemes. This is because they can transmit more bits per symbol, which reduces the overall number of symbols that need to be transmitted.

Versatility: Both QBPSK and QAM modulation schemes can be used in a variety of applications, including wireless communication, satellite communication, and cable television. They are also widely used in digital communication systems such as Wi-Fi, Bluetooth, and digital television.

Overall, QBPSK and QAM modulation and demodulation offer several advantages over traditional modulation schemes, including higher data rates, better spectral efficiency, robustness to noise, lower power consumption, and versatility.

Digital modulation techniques applications

Digital modulation techniques have various applications in the fields of telecommunications, wireless communications, and signal processing. Some of the widely used applications include

Wireless Local Area Network (WLAN) System: WLAN system uses Quadrature Amplitude Modulation (QAM) modulation to transmit data wirelessly. Implementation of channel coding techniques such as convolutional coding or Reed-Solomon coding improves the system's error correction capabilities. System's performance is measured in terms of throughput and coverage range.

Digital Television Broadcasting (DTB) System: Implementing a DTB system using 8-VSB (Vestigial Sideband) modulation to transmit digital television signals over a wide range of frequencies.

Radio Frequency Identification (RFID) System: Developing an RFID system that uses Amplitude Shift Keying (ASK) or Frequency Shift Keying (FSK) modulation to transmit data wirelessly. Implementing collision detection and resolution techniques improves the system's efficiency in handling multiple tags.

Software-Defined Radio (SDR) System: Building an SDR system uses digital modulation techniques such as BPSK, QPSK, and 16-QAM to transmit and receive radio signals. Implementation of digital signal processing techniques such as filtering, demodulation, and equalization to improve the system's performance.

REFERENCE:

"Digital Communications: Fundamentals and Applications" by Bernard Sklar
"Digital Modulation Techniques" by Fuqin Xiong
"Digital Modulation Techniques, Second Edition" by Fuqin Xiong
"Digital Modulation Techniques, Third Edition" by Fuqin Xiong
"Modern Digital Modulation Techniques" by Andrzej Jajszczyk and Tadeusz A. Wysocki
"Wireless Communications: Principles and Practice, Second Edition" by Theodore S. Rappaport
"Digital Modulation Techniques for Wireless Communications" by K. Vasudevan and C. Chellappan
"Digital Modulation Techniques: Second Edition" by Surendra Prasad and K. S. Shivaprakasha
"Digital Modulation Techniques for Satellite Communications, Part 1: Channel Coding and Modulation Schemes" by A. Barbieri and L. Tomasoni

"Digital Modulation Techniques for Satellite Communications, Part 2: Modulation and Demodulation Techniques" by A. Barbieri and L. Tomasoni.