Comparison of OFDM using QAM & QPSK

Introduction

- OFDM is a multicarrier modulation technique used in wireless communication
- It is improved with the use of different modulation techniques and channel coding techniques
- An in-depth analysis and comparison of the performance of OFDM system using different modulation techniques such as QAM and QPSK
- This is achieved through a performance parameter called as Bit Error Rate and simulated through MATLAB

Software Requirements

MATLAB

It is a programming and numeric computing platform used to analyze data, develop algorithms, and create models. We use MATLAB to simulate different types of modulations and compare QPSK and QAM when applied to OFDM

Fig 1: MATLAB Logo

Transmitter

- A transmitter is an electrical device used in telecommunications to generate radio waves so that data can be transmitted or sent using an antenna
- The transmitter can produce a radio frequency alternating current, which is then applied to the antenna, which radiates it as radio waves
- Transmitters come in a variety of shapes and sizes, depending on the standard and the type of device

Fig 2: Transmitter tower

Receiver

- A receiver is a device that selects a signal from among all the signals received from a communication channel, recovers the base band signal and delivers it to the user
- The receiver is the destination of the message
- The receiver's task is to interpret the sender's message, both verbal and nonverbal, with as little distortion as possible
- The process of interpreting the message is known as decoding

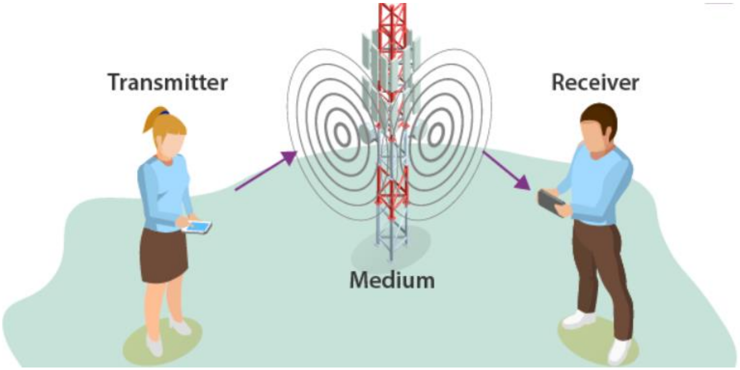


Fig 3: Transmitter and receiver

Modulation

- Modulation is the process of superimposing a message signal (also called as modulation signal) with a carrier signal
- The frequency band that is occupied by the modulation signal is called the baseband and therefore modulation signal is also known as baseband signal
- Depending on the signal there are 2 types of modulation:
  - Analog Modulation
  - Digital Modulation

Demodulation

- The process of separating the original information or signal from the modulated carrier
- In the case of amplitude or frequency modulation it involves a device, called a demodulator or detector, which produces a signal corresponding to the instantaneous changes in amplitude or frequency, respectively

Quadrature Amplitude Modulation (QAM)

- Combines Amplitude and phase shifts
- Digital Signals are created using IQ
- I is called InPhase component
- Q is called Quadrature component
- Quadrature refers to shifting by 90 degree

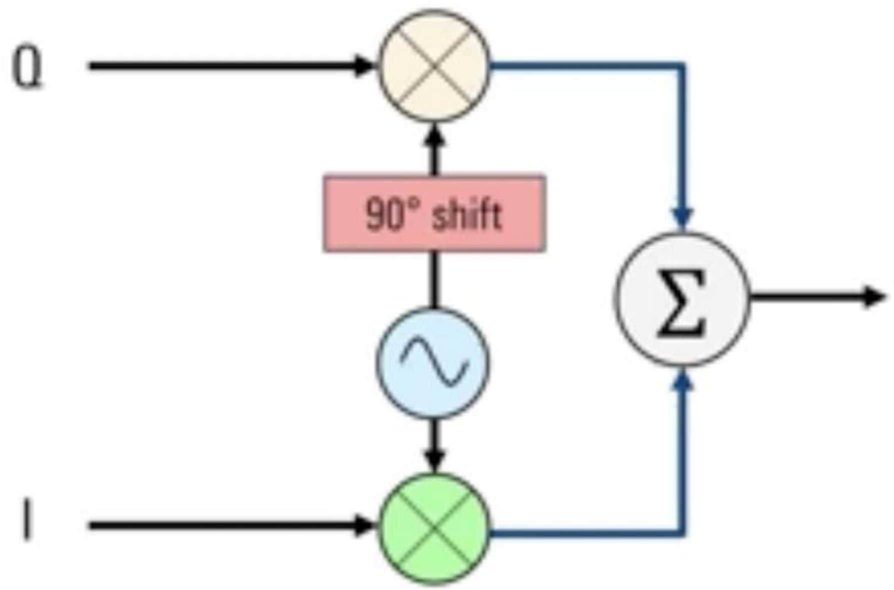


Fig 4: Transmitter Block

- The result of this IQ modulation is a constellation diagram in which points are arranged in a square shape
- Each point also referred to as symbol has a unique combination of amplitude and phase
- In 16-QAM, each symbol is represented by 4 bits
- Higher order QAMs are 64 QAM, 256 QAM, 1024 QAM, 4096 QAM
- By increasing modulation order
  - Increases bit rate
  - Reduces resistance to errors

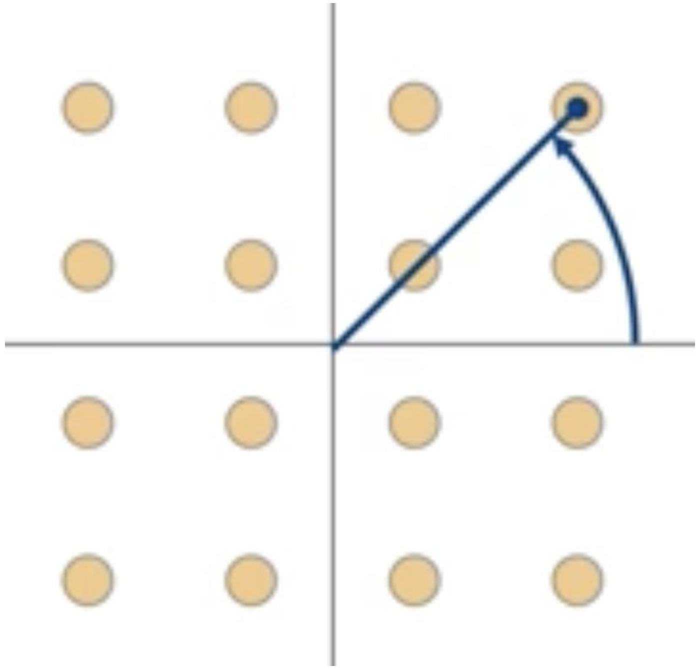


Fig 5: IQ Modulation constellation diagram for 16 QAM

Transmitter

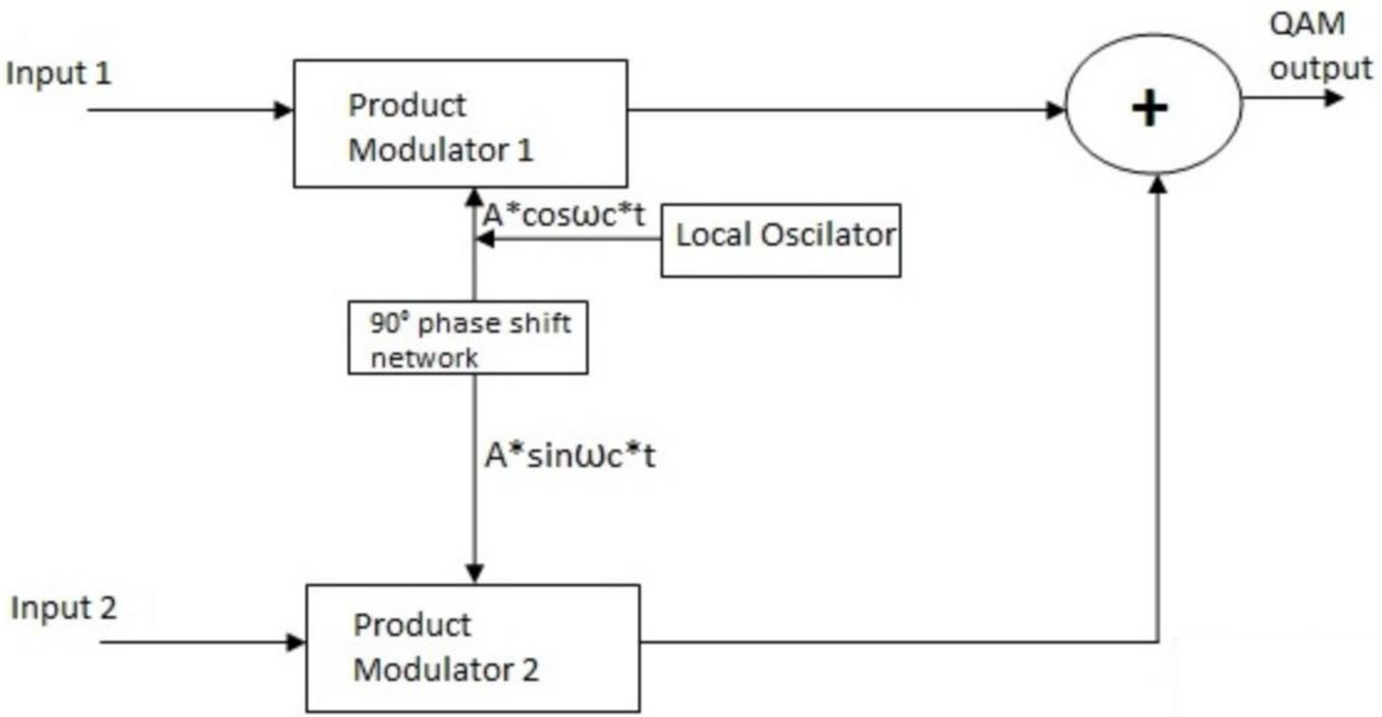


Fig 6: Transmitter block diagram for QAM

Receiver

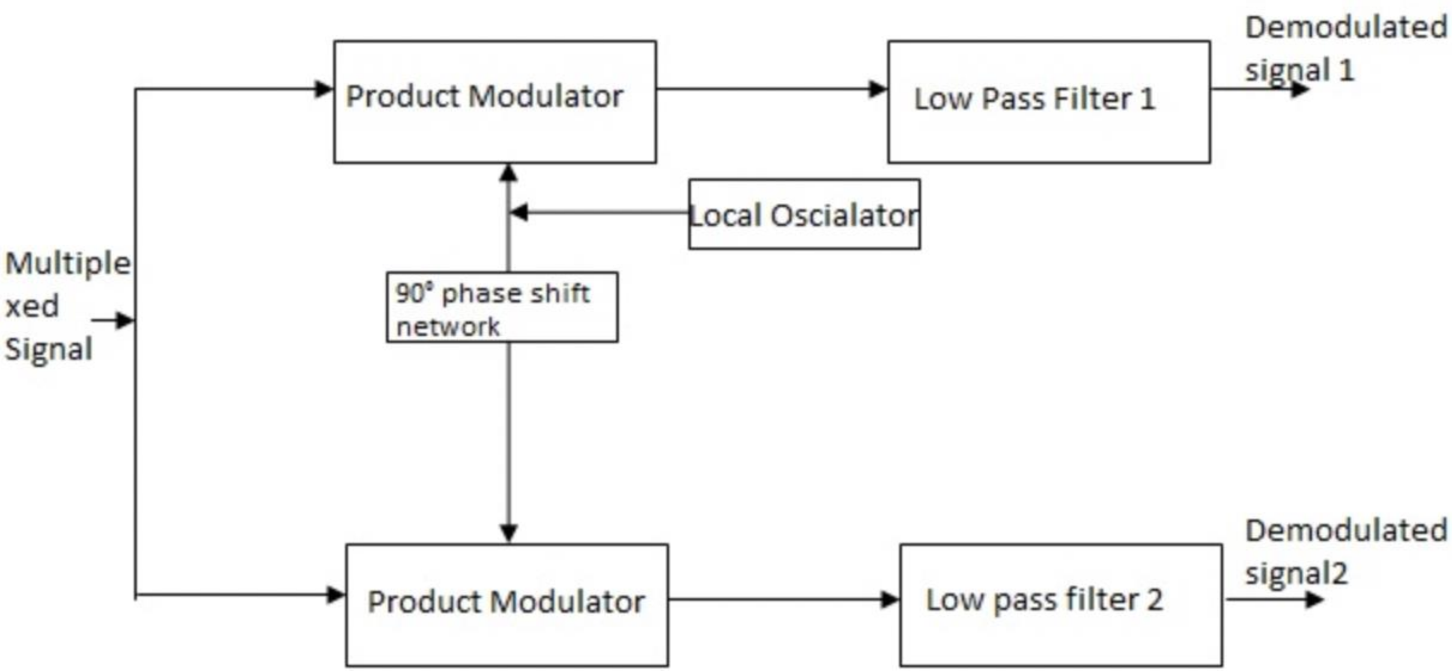


Fig 7: Receiver block diagram for QAM

MATLAB Code

```
clear;
clc;
clf;

N = 2*10^5;
M = 16;
data = [0:M-1];
W=qammod(data,M);
alpha16qam = [-3 -1 1 3]; % 16-QAM
EsNodB = [0:20];
out = zeros(1,N);
t = linspace (0, 1, N);
ip = randsrc(1,N, alpha16qam) + j*randsrc(1,N,alpha16qam);

for i=1:length(EsNodB)
    s = (1/sqrt(10))*ip; % Normalization %transmitted signal
    w0 = 1/sqrt(2)*[randn(1,N) + j*randn(1,N)]; % white guassian noise of 0 dB
    w = 10^(-EsNodB(i)/20)*w0; % extra white gaussian noise
    r = s + w; % transmitted signal with noise = received signal

```

```
    r_re = real(r);
    r_im = imag(r);
    %dem = r_re + r_im
    out_re(find(r_re < -2/sqrt(10)))              = -3;
    out_re(find(r_re >  2/sqrt(10)))              =  3;
    out_re(find(r_re > -2/sqrt(10) & r_re <= 0)) = -1;
    out_re(find(r_re > 0 & r_re <=  2/sqrt(10))) =  1;
    out_im(find(r_im < -2/sqrt(10)))              = -3;
    out_im(find(r_im >  2/sqrt(10)))              =  3;
    out_im(find(r_im > -2/sqrt(10) & r_im <= 0)) = -1;
    out_im(find(r_im > 0 & r_im <=  2/sqrt(10))) =  1;
    out = out_re + j*out_im;
    ber(i) = size(find([ip - out]),2); %counting the number of errors
end
ber
simBer = ber/N;
theoryBer = 3/2*erfc(sqrt(0.1*(10.^(EsNodB/10))));
figure
plot(t, s,'b-.'); axis([-0.1 1.2 -2 2]);
title('Transmitted Signal')

figure
plot(t,out,'r--'); axis([-0.1 1.2 -4 4]);
title('Recieved Signal')

```

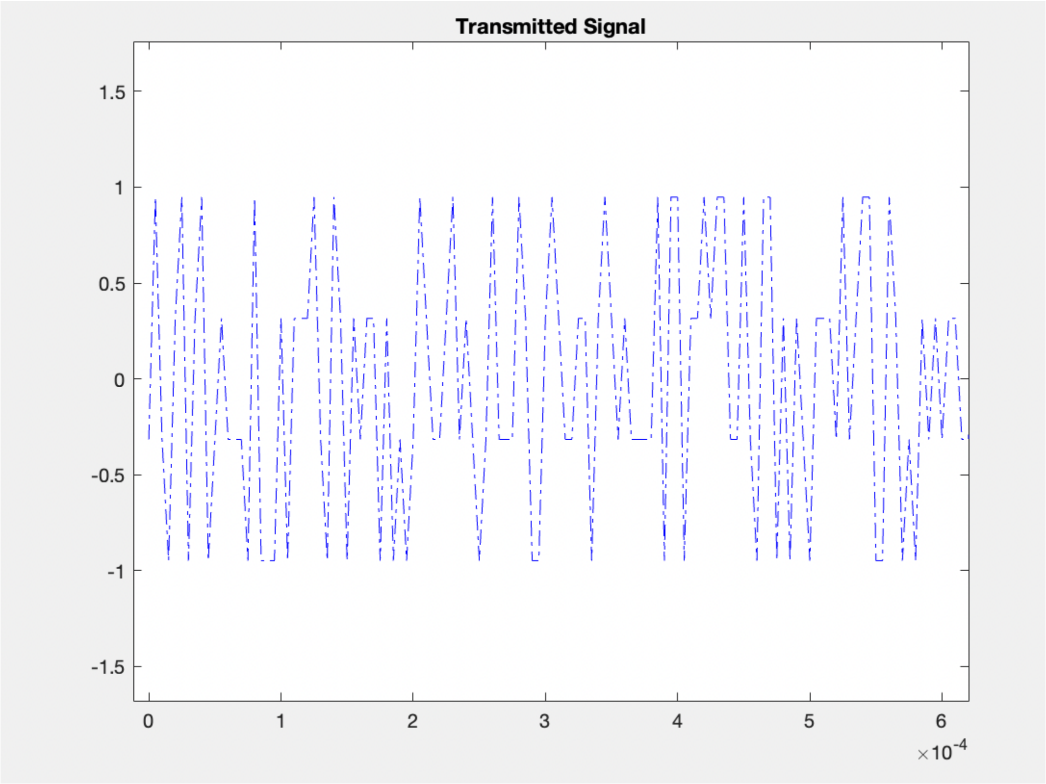

```
figure
plot(t,out,'r--'); axis([-0.1 1.2 -4 4]);
title('Recieved Signal')
```


```

figure
semilogy(EsNodB, theoryBer, 'b.-','LineWidth',2) %Plot the BER
hold on;
semilogy(EsNodB, simBer, 'mx-','LineWidth',2) %Plot the BER
axis([0 20 10^-5 1])
grid on
legend('theory', 'simulation');
xlabel('Es/No, dB');
ylabel('Symbol Error Rate')
title('Symbol error probability curve for 16-QAM modulation')

scatterplot(W,1,0,'b*');
```

Output



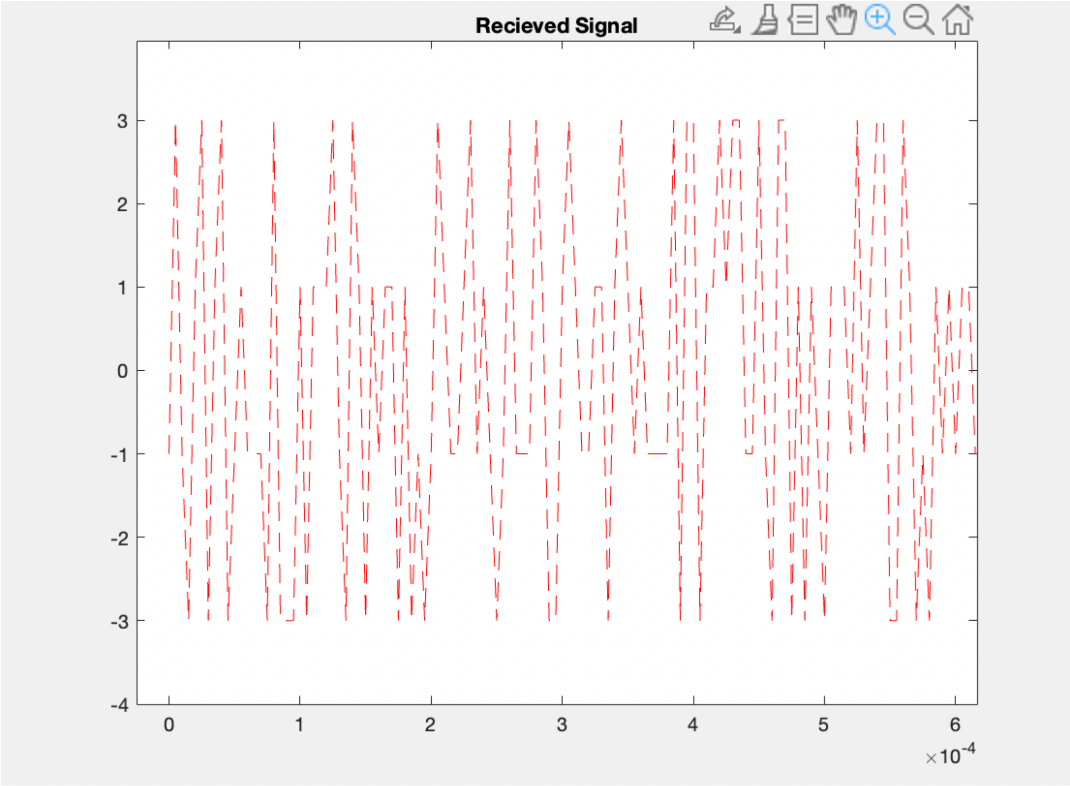
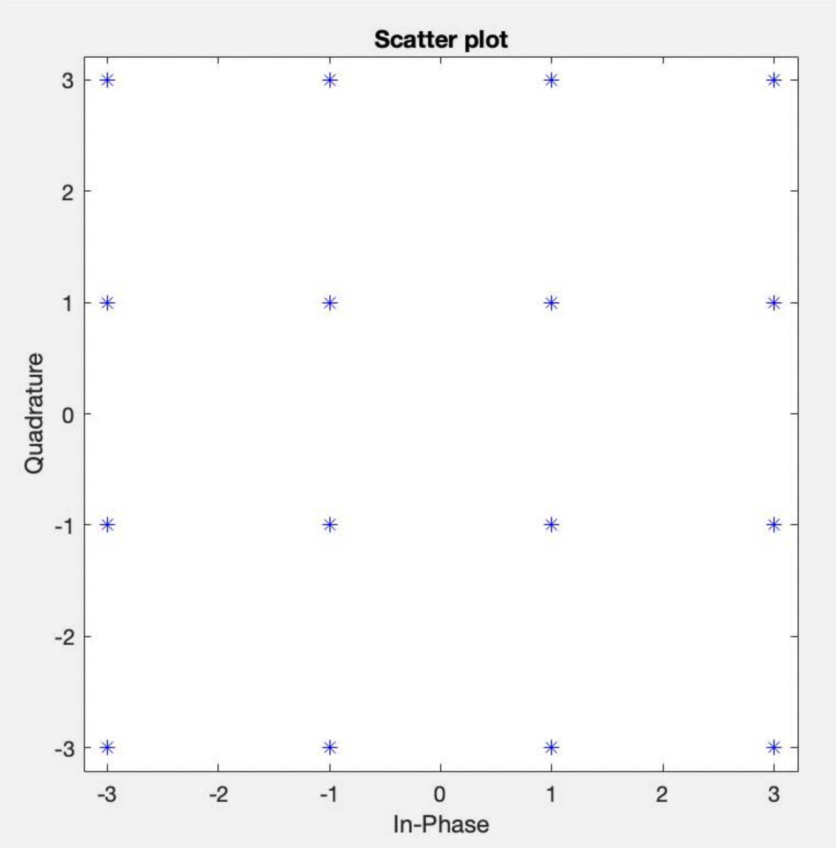


Fig 8: Transmitter and Receiver output for QAM



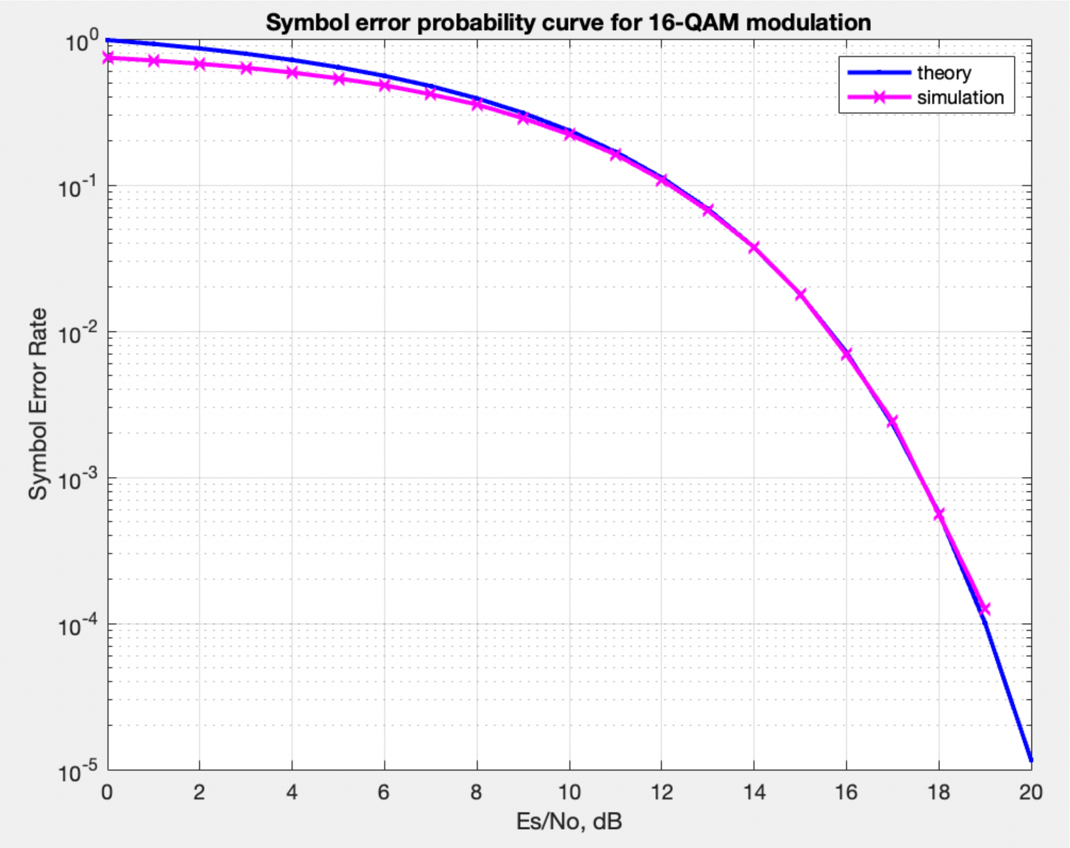
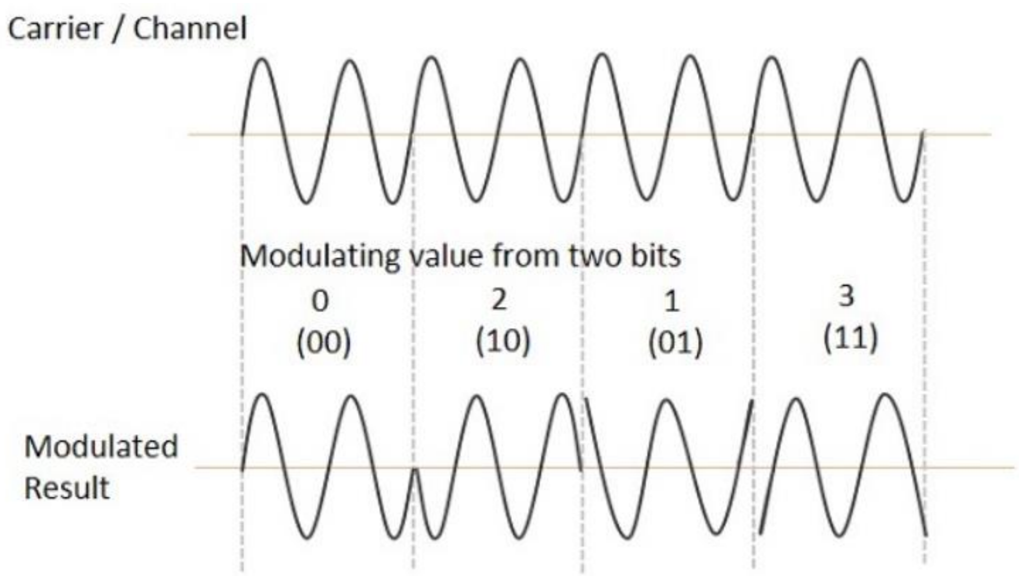


Fig 9: Scatter plot and Symbol error probability curve for 16 - QAM

Quadrature Phase Shift Keying (QPSK)

- A form of Phase Shift Keying in which two bits are modulated at once
- Variation of Binary Phase Shift Keying (BPSK)
- It is a Double side band suppressed carrier (DSBSC) modulation scheme, which sends two bits of digital information at a time
- Selects one of the four carrier phase shifts [0, 90, 180, 270]



MATLAB Code

```
clc;
clear all;
close all;

%generating quadrature carrier signal
Tb = 1;
t = 0:(Tb/100):Tb;
fc = 1;
c1 = sqrt(2/Tb) * cos(2 * pi * fc * t);
c2 = sqrt(2/Tb) * sin(2 * pi * fc * t);

%plotting carriers c1 and c2
subplot(3,2,1);
plot(t, c1);
title ('carrier signal 1');
xlabel('t');
ylabel('c1(t)');
grid on;
subplot(3,2,2);
plot(t, c2);
title('carrier signal 2');
xlabel('t');
ylabel('c2(t)');
grid on;
```

Output

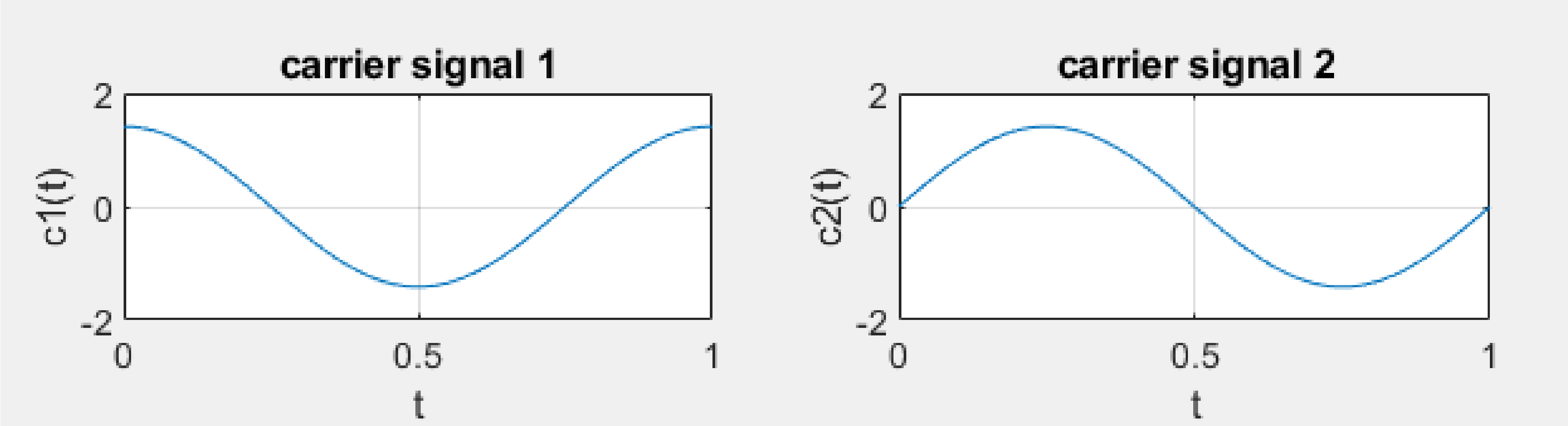


Fig 10: Carrier signals

Transmitter

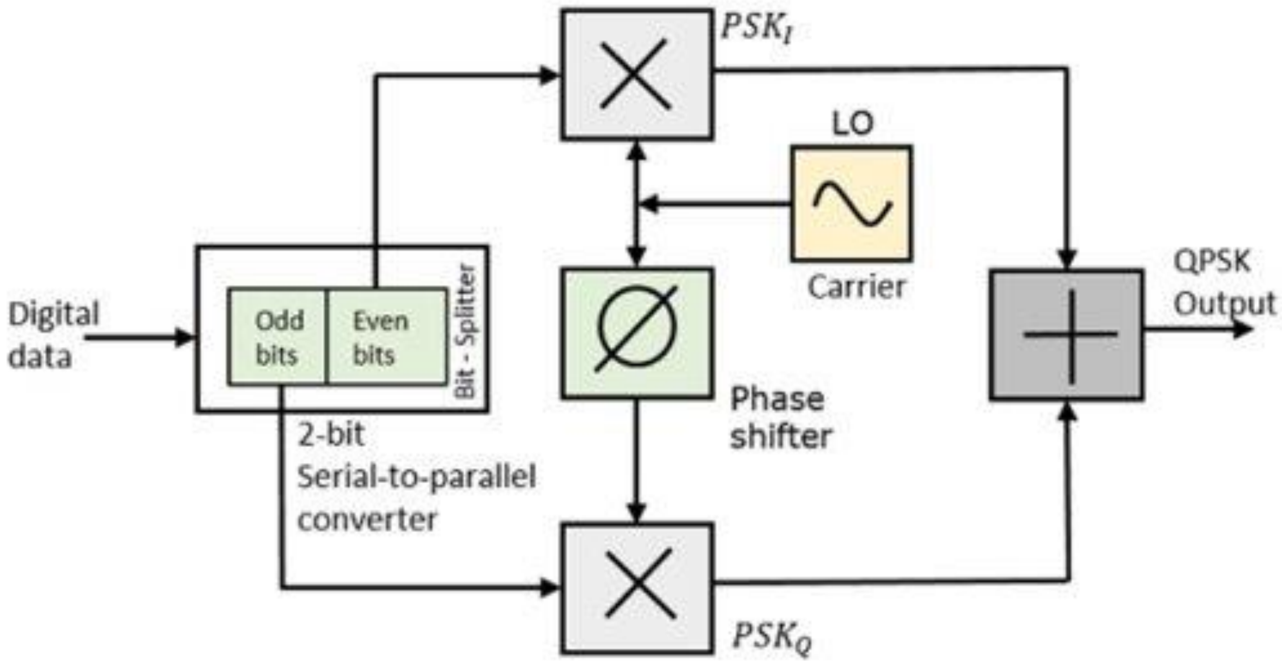


Fig 11: Transmitter block diagram for QPSK

MATLAB Code

```matlab
%Let us generate the message signal
N = 16;
m = rand(1,N);
t1 = 0;
t2 = Tb;
for i = 1:2:(N-1)
    t = t1:(Tb/100):t2;
    if m(i) > 0.5
        m(i) = 1;
        m_s = ones(1, length(t));
    else
        m(i) = 0;
        m_s = -1 * ones(1, length(t));
    end
    odd_sig(i, :) = c1.*m_s;

    if m(i+1) > 0.5
        m(i+1) = 1;
        m_s = ones(1, length(t));
    else
        m(i+1) = 0;
        m_s = -1 * ones(1, length(t));
    end
    even_sig(i, :) = c2.*m_s;

    qpsk = odd_sig + even_sig;

    %Let us plot the QPSK signal
    subplot(3, 2, 4);
    plot(t, qpsk(i,:));
    title('Qpsk signal');
    xlabel('t');
    ylabel('s(t)');
    grid on;
    hold on;
    t1 = t1 + (Tb + 0.1);
    t2 = t2 + (Tb + 0.1);
end

hold off;

%Plotting message signal
subplot(3,2,3);
stem(m);
title ('Binary data in');
xlabel('n');
ylabel('b(n)');
grid on;
```

Output

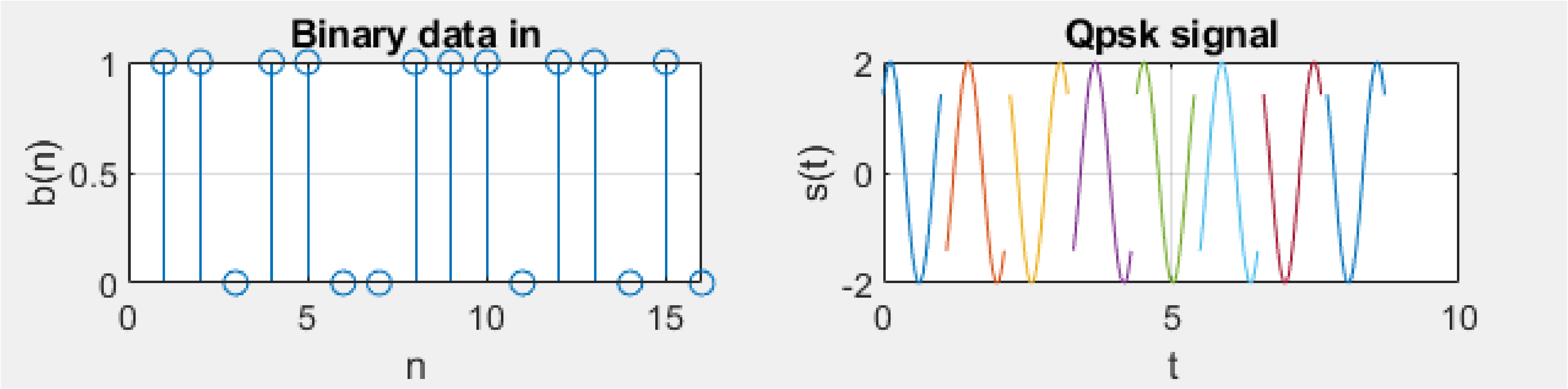


Fig 12: Modulated QPSK Signal

Receiver

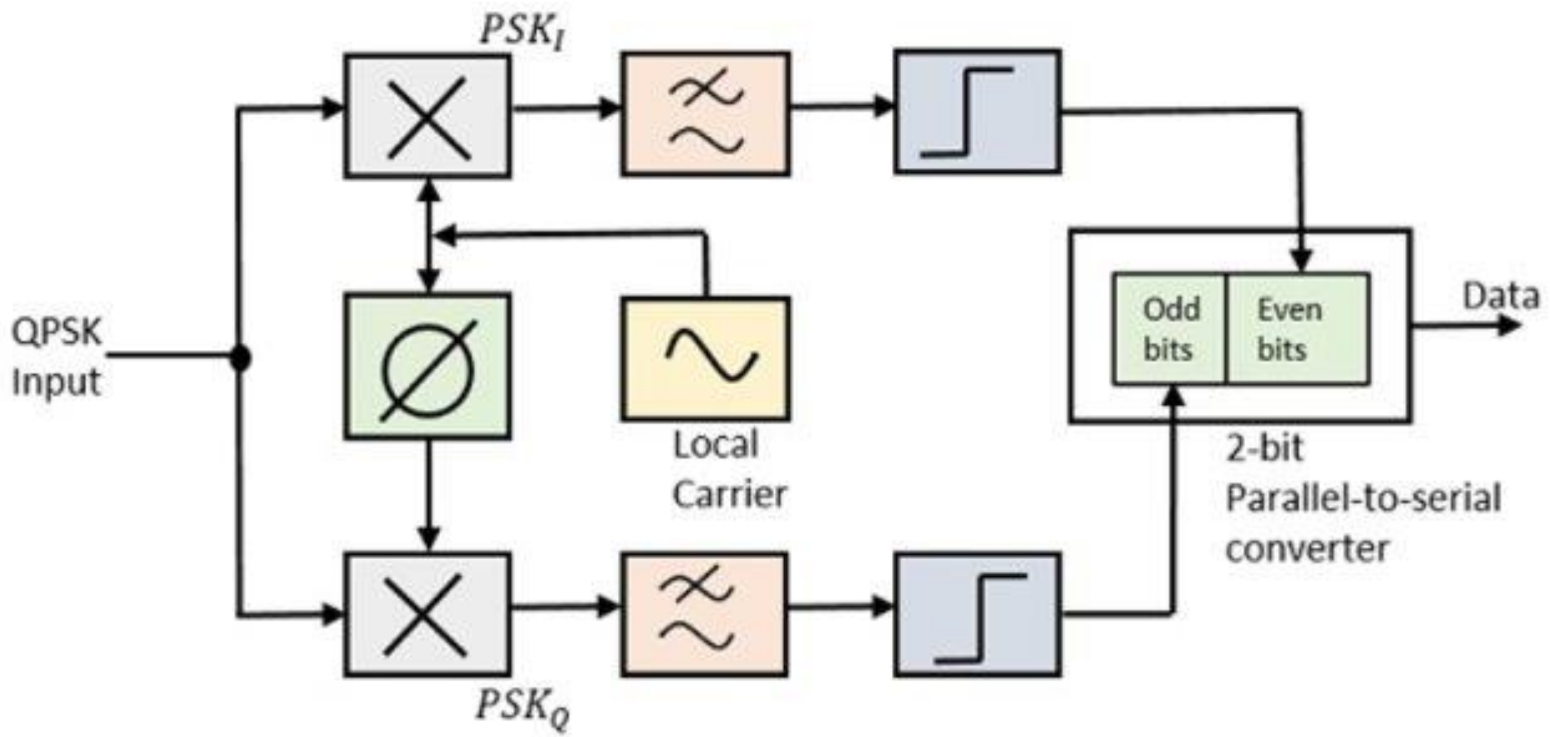


Fig 13: Receiver block diagram for QPSK

MATLAB Code

```
%demodulation
t1 = 0;
t2 = Tb;
for i = 1:N-1
    t = t1:(Tb/100):t2;
    x1 =  sum(c1.*qpsk(i,:));
    x2 =  sum(c2.*qpsk(i,:));
    if (x1 > 0 && x2 > 0)
        demod(i) = 1;
        demod(i+1) = 1;

    elseif (x1 > 0 && x2 < 0)
        demod(i) = 1;
        demod(i+1) = 0;

    elseif (x1 < 0 && x2 < 0)
        demod(i) = 0;
        demod(i+1) = 0;

     elseif (x1 < 0 && x2 > 0)
        demod(i) = 0;
        demod(i+1) = 1;
    end

    t1 = t1 + (Tb + 0.01);
    t2 = t2 + (Tb + 0.01);
end
```

```
subplot(3,2,5);
stem(demod);
title('qpsk demodulated bits');
xlabel('n');
ylabel('b(n)');
grid on;
```

Output

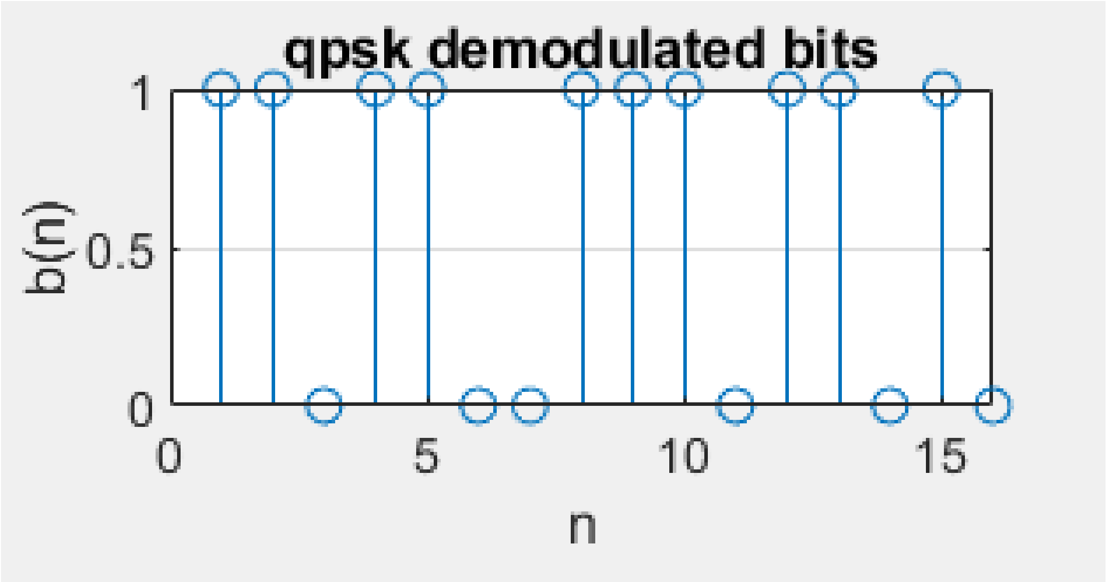


Fig 14: Demodulated QPSK Signal

QPSK Output

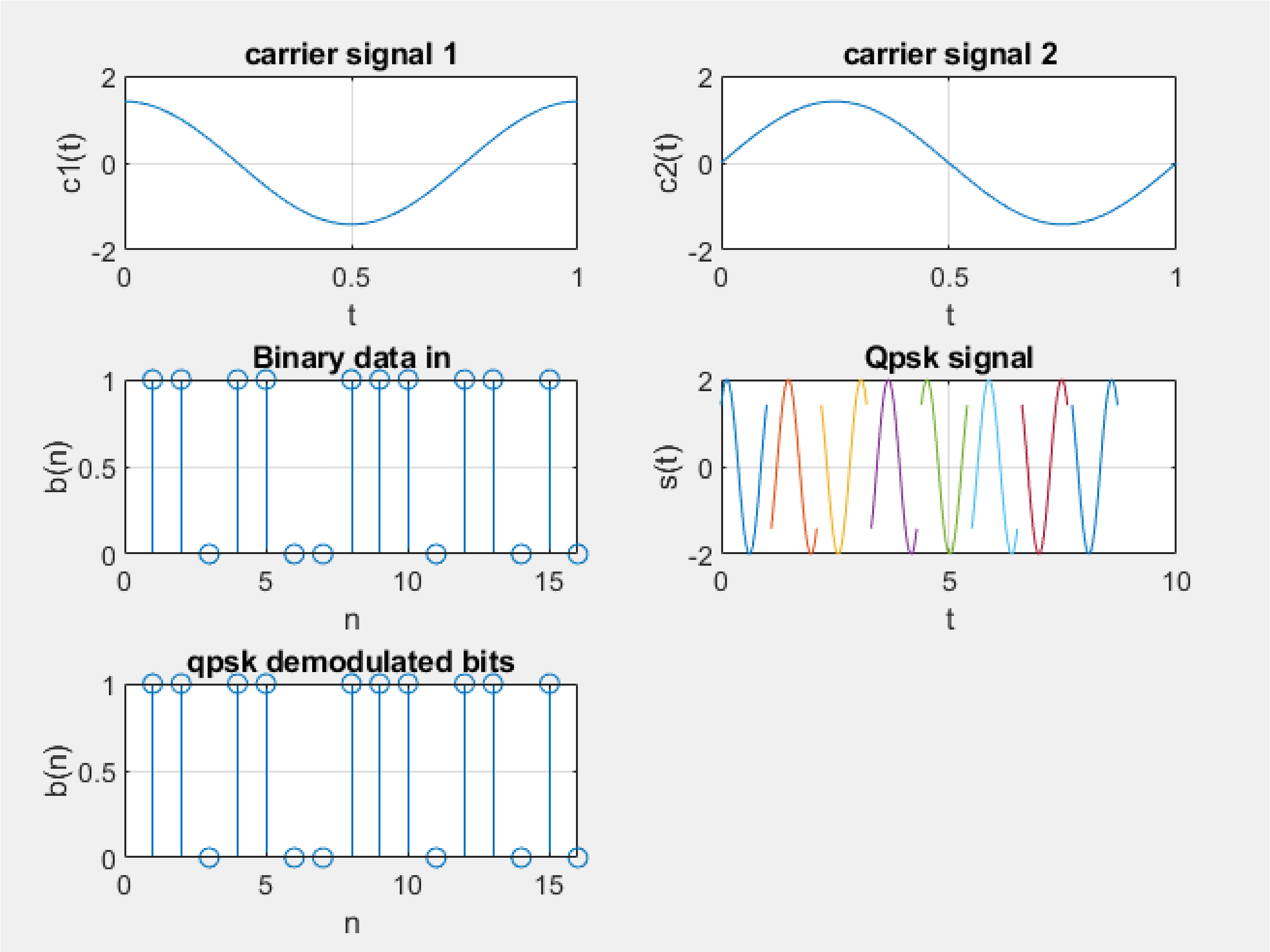


Fig 15: QPSK Signal output

Orthogonal frequency division multiplexing (OFDM)

Orthogonal frequency-division multiplexing is a method of data transmission where a single information stream is split among several closely spaced narrowband subchannel frequencies instead of a single Wideband channel frequency

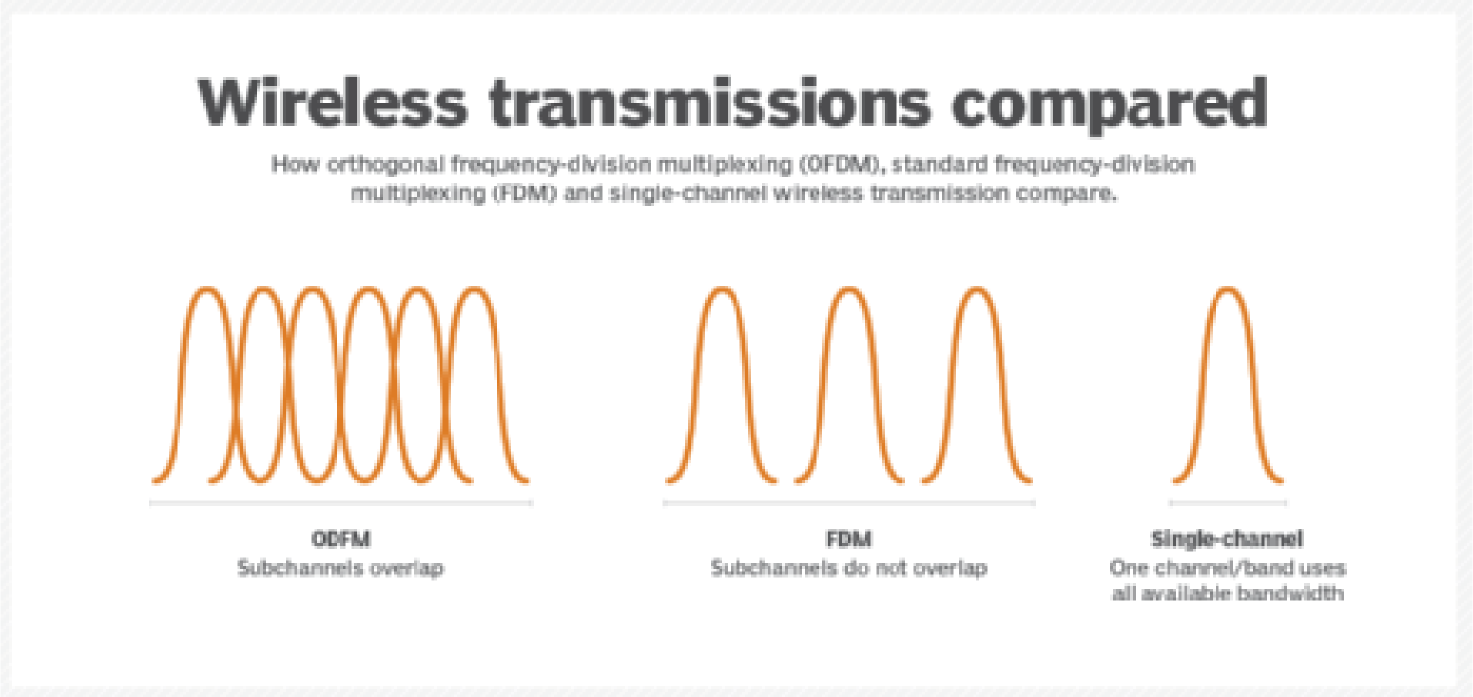


Fig 16: Introduction to OFDM

Transmitter and Receiver

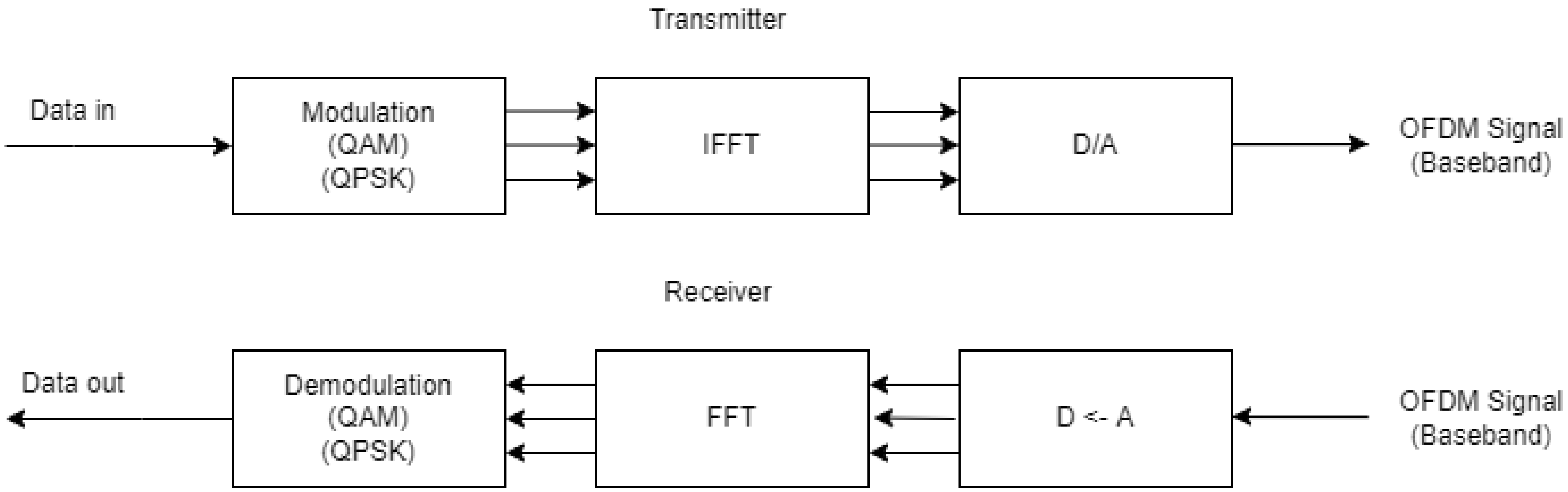


Fig 16: Transmitter & Receiver blocks for OFDM

OFDM - QAM vs OFDM - QPSK

MATLAB Code for OFDM using QPSK

```
clear; clc;
```

```
M = 4;
k = log2(M);
numSC = 128;
cpLen = 32;
maxBitErrors = 100;
maxNumBits = 1e7;
xTrial = (0:M-1)';
```

```
qpskMod = comm.QPSKModulator('BitInput',true);
qpskDemod = comm.QPSKDemodulator('BitOutput',true);

ofdmMod = comm.OFDMModulator('FFTLength',numSC,'CyclicPrefixLength',cpLen);
ofdmDemod = comm.OFDMDemodulator('FFTLength',numSC,'CyclicPrefixLength',cpLen);

channel = comm.AWGNChannel('NoiseMethod','Variance', 'VarianceSource','Input port');

errorRate = comm.ErrorRate('ResetInputPort',true);

ofdmDims = info(ofdmMod)
```

```
ofdmDims = struct with fields:
    DataInputSize: [117 1]
       OutputSize: [160 1]
```

```
numDC = ofdmDims.DataInputSize(1)
```

```
numDC = 117
```

```
frameSize = [k*numDC 1];

EbNoVec = (0:10)';
snrVec = EbNoVec + 10*log10(k) + 10*log10(numDC/numSC);

berVec = zeros(length(EbNoVec),3);
errorStats = zeros(1,3);
```

```
for m = 1:length(EbNoVec)
    snr = snrVec(m);
    while errorStats(2) <= maxBitErrors && errorStats(3) <= maxNumBits
        dataIn = randi([0,1],frameSize);
        qpskTx = qpskMod(dataIn);
        txSig = ofdmMod(qpskTx);
        powerDB = 10*log10(var(txSig));
        noiseVar = 10.^(0.1*(powerDB-snr));
        rxSig = channel(txSig,noiseVar);
        qpskRx = ofdmDemod(rxSig);
        dataOut = qpskDemod(qpskRx);
        errorStats = errorRate(dataIn,dataOut,0);
    end

    berVec(m,:) = errorStats;
    errorStats = errorRate(dataIn,dataOut,1);
end
```

```
berTheory = berawgn(EbNoVec,'psk',M,'nondiff');

figure
semilogy(EbNoVec,berVec(:,1),'*')
hold on
semilogy(EbNoVec,berTheory)
title('BER vs SNR');
legend('Simulation','Theory','Location','Best')
xlabel('SNR (dB)')
ylabel('Bit Error Rate')
grid on
hold off
```

Output

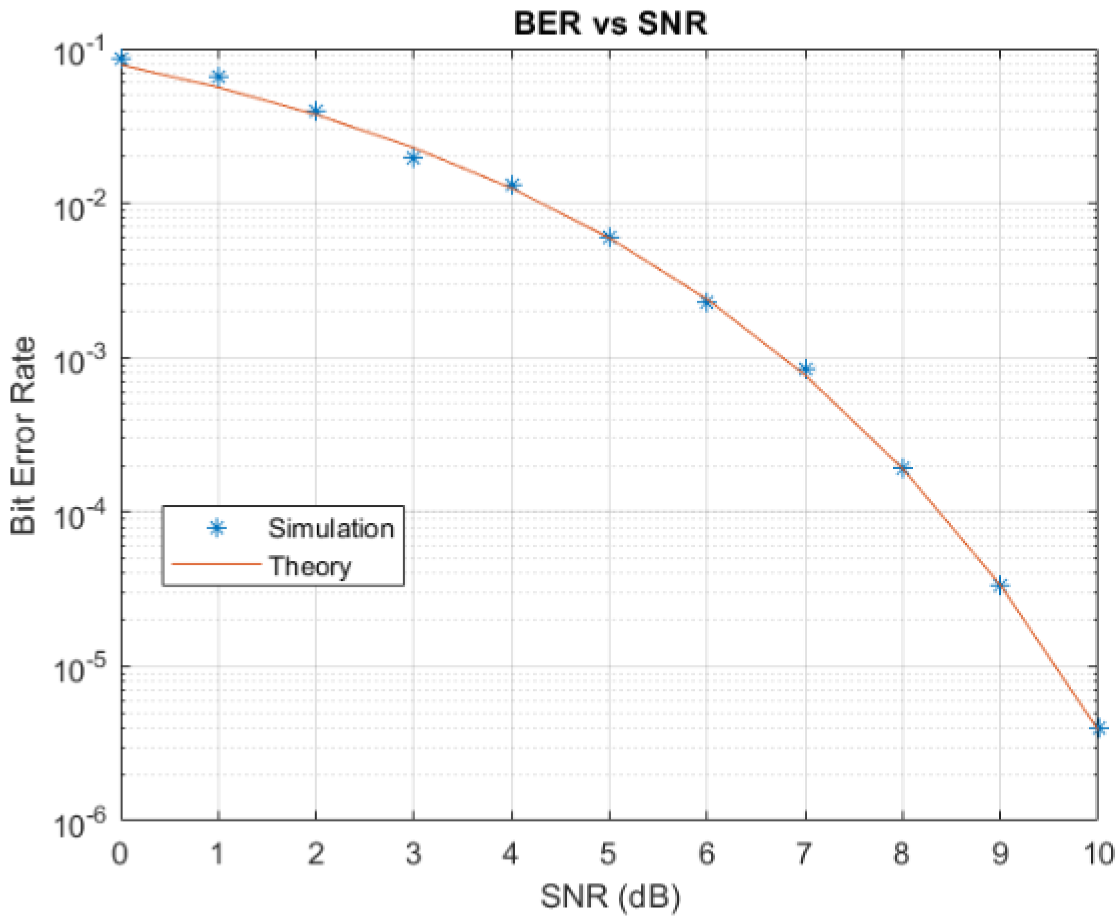


Fig 18: Bit error rate vs Signal to noise ratio for OFDM - QPSK

MATLAB Code for OFDM using QAM

```
close all
clear all
clc
```

Generating and coding data

```
t_data = randi([0,1],9600,1)';
x = 1;
si = 1; % for BER rows
```

```
for d = 1:100;
data = t_data(x:x+95);
x = x+96;
k = 3;
n = 6;
s1 = size(data,2); % Size of input matrix
j = s1/k;
```

Convolutionally encoding data

```
constlen = 7;
codegen = [171 133]; % Polynomial
trellis = poly2trellis(constlen, codegen);
codedata = convenc(data, trellis);
```

```
%Interleaving coded data

s2 = size(codedata,2);
j = s2/4;
matrix = reshape(codedata,j,4);

intlvddata = matintrlv(matrix',2,2)'; % Interleave
intlvddata = intlvddata';
```

Binary to decimal conversion

```
dec = bi2de(intlvddata','left-msb');
```

```
%16-QAM Modulation
M = 16;
y = qammod(dec,M);
% scatterplot(y);
```

Pilot insertion

```
lendata=length(y);
pilt=3+3j;
nofpits=4;

k=1;

for i=(1:13:52)

    pilt_data1(i)=pilt;

    for j=(i+1:i+12);
        pilt_data1(j)=y(k);
        k=k+1;
    end
end

pilt_data1=pilt_data1';   % size of pilt_data =52
pilt_data(1:52)=pilt_data1(1:52);    % upsizing to 64
pilt_data(13:64)=pilt_data1(1:52);   % upsizing to 64

for i=1:52

    pilt_data(i+6)=pilt_data1(i);

end
```

MATLAB Code for OFDM using QAM

IFFT

```
ifft_sig=ifft(pilt_data',64);
```

Adding Cyclic Extension

```
cext_data=zeros(80,1);
cext_data(1:16)=ifft_sig(49:64);
for i=1:64
    cext_data(i+16)=ifft_sig(i);
end
```

Channel

```
% SNR
o=1;
for snr=0:2:50
ofdm_sig=awgn(cext_data,snr,'measured'); % Adding white Gaussian Noise
```

RECEIVER

```
%Removing Cyclic Extension
for i=1:64
    rxed_sig(i)=ofdm_sig(i+16);
end
```

FFT

```
ff_sig=fft(rxed_sig,64);
```

Pilot

```
for i=1:52
    synched_sig1(i)=ff_sig(i+6);
end

k=1;

for i=(1:13:52)
    for j=(i+1:i+12);
        synched_sig(k)=synched_sig1(j);
        k=k+1;
    end
end
```

Demodulation

```
dem_data= qamdemod(synched_sig,16);
```

Decimal to binary conversion

```
bin=de2bi(dem_data','left-msb');
bin=bin';
```

De-Interleaving

```
deintlvddata = matdeintrlv(bin,2,2); % De-Interleave
deintlvddata=deintlvddata';
deintlvddata=deintlvddata(:)';
```

```
% Decoding data
n=6;
k=3;
decodedata =vitdec(deintlvddata,trellis,5,'trunc','hard');  % decoding datausing veterbi decoder
rxed_data=decodedata;
```

AMRITA VISHWA VIDYAPEETHAM
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF THE UGC ACT, 1956

MATLAB Code for OFDM using QAM

Calculating BER

```
rxed_data=rxed_data(:)';
errors=0;
c=xor(data,rxed_data);
errors=nnz(c);
BER(si,o)=errors/length(data);
o=o+1;
 end % SNR loop ends here
 si=si+1;
end % main data loop
```

Time averaging for optimum results

```
for col=1:25;
    ber(1,col)=0;
for row=1:100;
        ber(1,col)=ber(1,col)+BER(row,col);
    end
end
ber = ber./100;
```

```
figure
i=0:2:48;
semilogy(i,ber, 'r*');
title('BER vs SNR');
ylabel('BER');
xlabel('SNR (dB)');
grid on
hold on
EsNodB = [0:16];
theoryBer = 3/2*erfc(sqrt(0.1*(10.^(EsNodB/10))));
semilogy(EsNodB, theoryBer, 'b.-','LineWidth',2) %Plot the BER
legend('Simulation','Theory','Location','Best')
```

Output

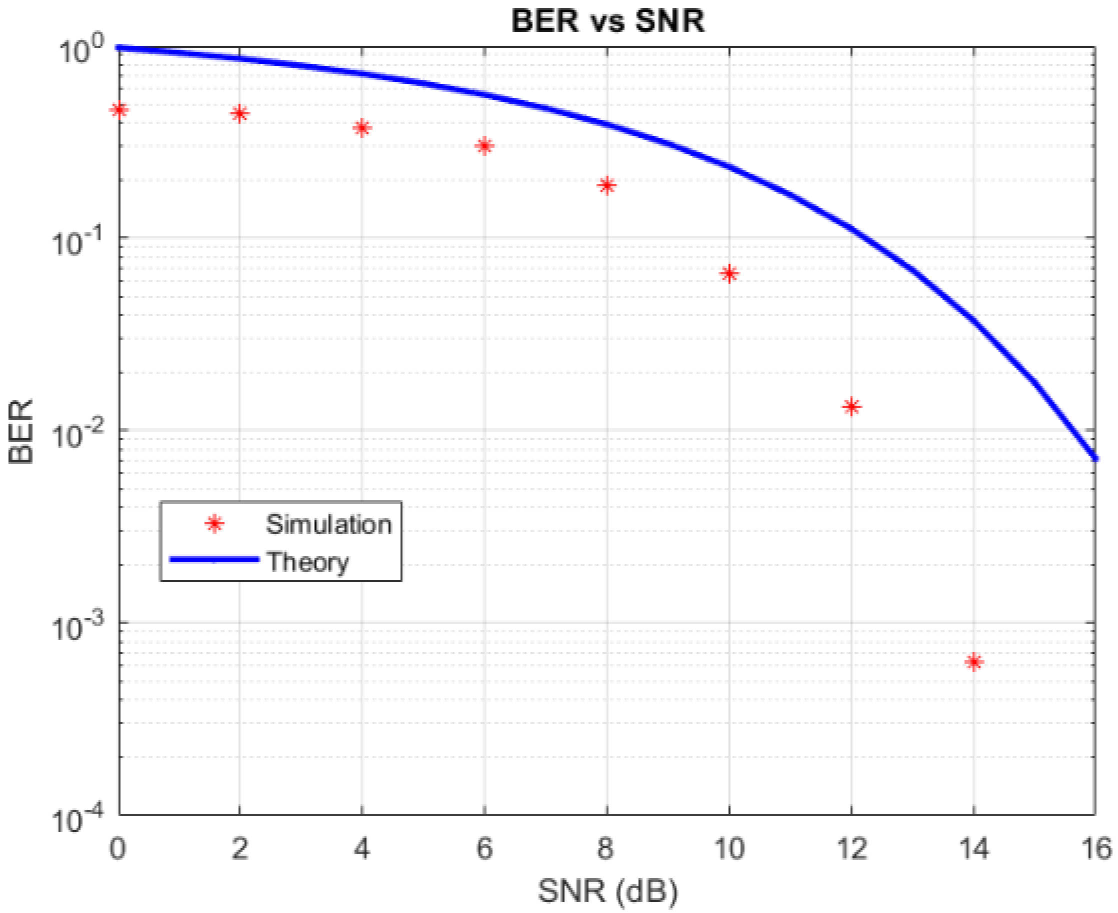
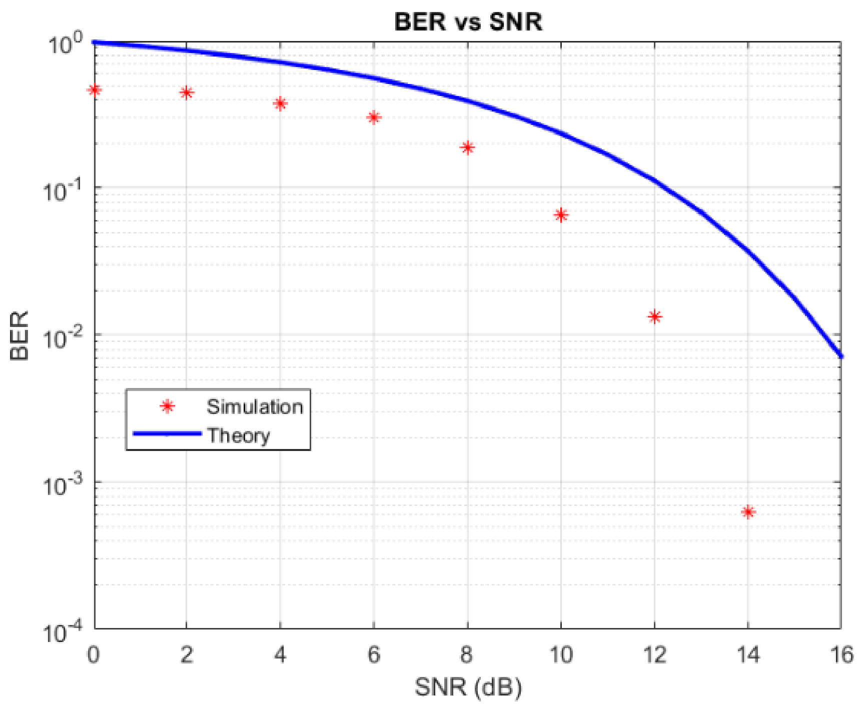
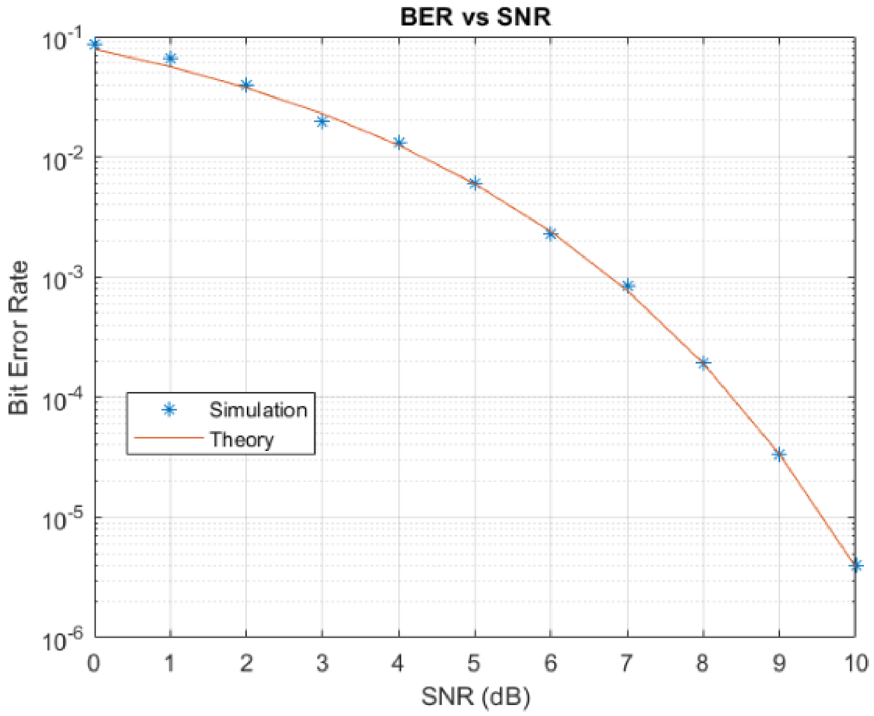


Fig 19: Bit error rate vs Signal to noise ratio for OFDM - QAM

Comparison



OFDM - QAM



OFDM - QPSK

Fig 20: Comparison of Bit error rate vs Signal to noise ratio for OFDM – QAM vs OFDM - QPSK

Conclusion

- It is observed that for Signal to Noise Ratio, the BER for QAM is higher than QPSK
- For example, when SNR = ‘4’
  - For QAM, in 100 bits, 5 bits have a probability of error
  - For QPSK, in 100 bits, 1 bit has a probability of error
- Observing the graph, we deduce that OFDM using QPSK's theoretical and simulated points are closer than that of QAM
- Hence, the performance of QPSK is better than 16 QAM because the BER values with respect to the Average received SNR (in dB) in case of QPSK are lower than the values obtained in the case of 16 QAM

Thank you!