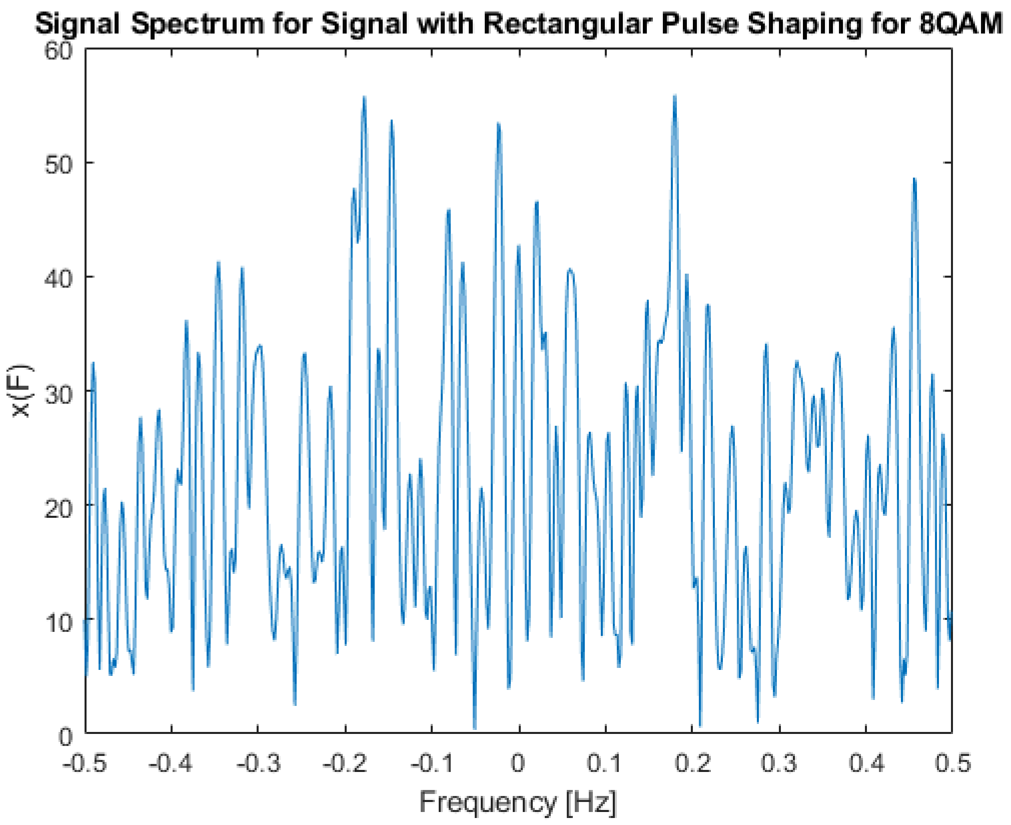
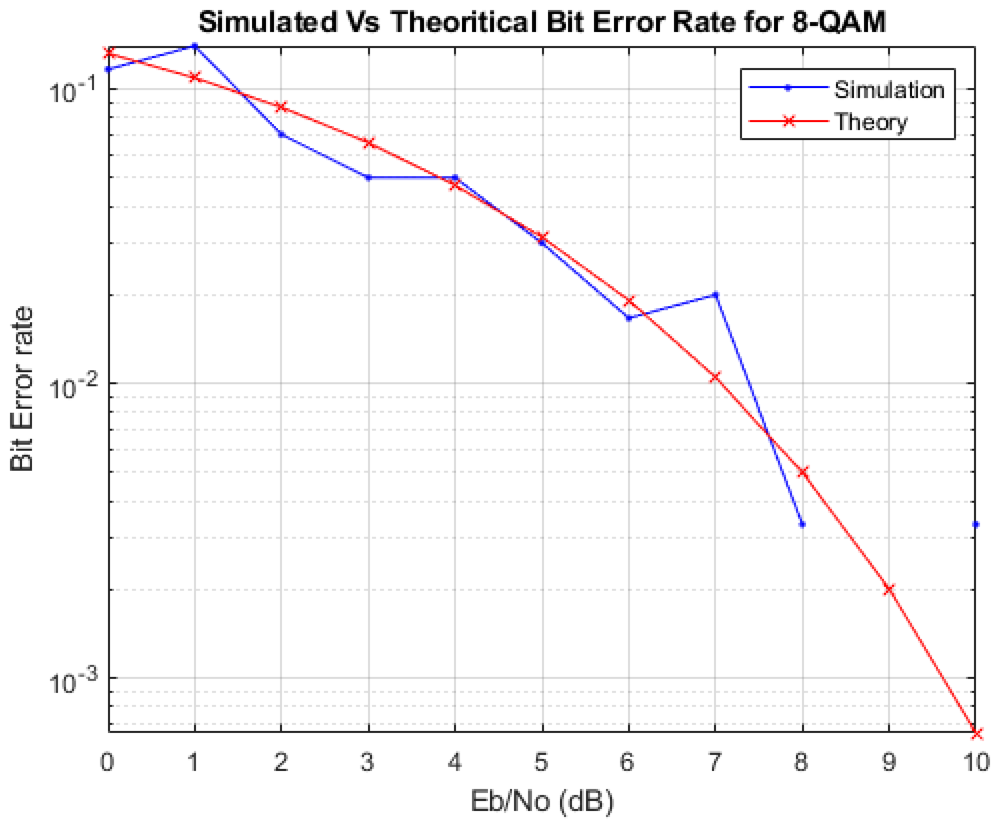
```
%BER Simulation of BPSK modulation
clc;
close all;
clear all;

bits=300;
data=double(randi(2,[1,bits])>1.5);


%---debugging---
%data=[1 1 1]
%xxxxxxxxxx
ebno=0:10;
BER=zeros(1,length(ebno));
thr=BER;
%---Transmitter---------
%Gray mapping of bits into symbols
col=length(data)/3;
I=zeros(1,col);
Q=I;
k=1;
for i=1:3:length(data)
    if(data(i:i+2)==[0 0 0])
        I(k)=1;
        Q(k)=1;
        k=k+1;
    elseif(data(i:i+2)==[0 0 1])
        I(k)=3;
        Q(k)=1;
        k=k+1;
    elseif(data(i:i+2)==[0 1 0])
        I(k)=-1;
        Q(k)=1;
        k=k+1;
    elseif(data(i:i+2)==[0 1 1])
        I(k)=-3;
        Q(k)=1;
        k=k+1;
    elseif(data(i:i+2)==[1 0 0])
        I(k)=1;
        Q(k)=-1;
        k=k+1;
    elseif(data(i:i+2)==[1 0 1])
        I(k)=3;
        Q(k)=-1;
        k=k+1;
    elseif(data(i:i+2)==[1 1 0])
        I(k)=-1;
        Q(k)=-1;
        k=k+1;
    elseif(data(i:i+2)==[1 1 1])
```

```
            I(k)=-3;
            Q(k)=-1;
            k=k+1;
        end
    end
    symb=I+1i*Q;
    %real(symb)
    %imag(symb)
    %----Filter
    psf=ones(1,1);
    Es=sum(psf.^2);
    eb=Es/3;
    eb=2;
    %----
    M=length(psf);
    for i=1:length(ebno)
        % inserting zeros between the bits
        % w.r.t number of coefficients of
        % PSF to pass the bit stream from the PSF
        z=zeros(M-1,bits/3);
        upsamp=[symb;z];
        upsamp2=reshape(upsamp,1,(M)*bits/3);
        %Passing the symbols from PSF
        %tx_symb=conv(real(upsamp2),psf)+j*conv(imag(upsamp2),psf);
        tx_symb=conv(upsamp2,psf);
        %--------CHANNEL-----------
        %Random noise generation and addition to the signal
        ebno2=10.^(ebno(i)/10);
        %no=eb/ebno2;
        %n_var=sqrt(no/2);
        n_var=sqrt(eb/(2*ebno2));
        rx_symb=tx_symb
    +(n_var*randn(1,length(tx_symb)))+1i*n_var*randn(1,length(tx_symb));
        %XXXXXXXXXXXXXXXXXXXXXXXXXX
        %-------RECEIVER-----------
        rx_match=conv(rx_symb,psf);
        rx=rx_match(M:M:length(rx_match));
        rx=rx(1:1:bits/3);
        recv_bits=zeros(1,bits);
        k=1;
        %demappingk=1;
        for n=1:bits/3
            I=real(rx(n));
            Q=imag(rx(n));
            if (I > 0) && (I < 2) && (Q > 0)
                recv_bits(k:k+2)=[0 0 0];
            elseif (I > 0) && (I < 2) && (Q < 0)
                recv_bits(k:k+2)=[1 0 0];
            elseif (I > 2) && (Q >0)
                recv_bits(k:k+2)=[0 0 1];
            elseif (I > 2) && (Q < 0)
                recv_bits(k:k+2)=[1 0 1];
            elseif (I < 0) && (I > -2) && (Q > 0)
                recv_bits(k:k+2)=[0 1 0];
            elseif (I < 0) && (I > -2) && (Q < 0)
                recv_bits(k:k+2)=[1 1 0];
            elseif (I < -2) && (Q > 0)
                recv_bits(k:k+2)=[0 1 1];
            elseif (I < -2) && (Q < 0)
                recv_bits(k:k+2)=[1 1 1];
            end
            k=k+3;
        end
        tx_symb;
        rx_symb;
        data;
        recv_bits;
        %xxxxxxxxxxxxxxxxxxxxxxxxxxxx
        %---SIMULATED BIT ERROR RATE----
        errors=find(xor(recv_bits,data));
        errors=size(errors,2);
        BER(i)=errors/bits;
        ebno_lin=(10^(ebno(i)/10));
        thr(i)=(5/12)*erfc(sqrt(ebno_lin/2));
        %xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    end
    fs=1;
    n_pt=2^9;
    tx_spec=fft(tx_symb,n_pt);
    f= -fs/2:fs/n_pt:fs/2-fs/n_pt;
    figure;
    plot(f,abs(fftshift(tx_spec)));
    title('Signal Spectrum for Signal with Rectangular Pulse Shaping for
     8QAM');
    xlabel('Frequency [Hz]');
    ylabel('x(F)');
    figure;
    semilogy(ebno,BER,'b.-');
    hold on;
    %ebno2=(10.^(ebno/10));%thr=(5/12).*erfc(sqrt((10.^(ebno/10))./2));
    semilogy(ebno,thr,'rx-');
    xlabel('Eb/No (dB)');
    ylabel('Bit Error rate')
    title('Simulated Vs Theoritical Bit Error Rate for 8-QAM');
    legend('Simulation','Theory');
    grid on;
```


Signal Spectrum for Signal with Rectangular Pulse Shaping for 8QAM
x(F)
Frequency [Hz]
0
10
20
30
40
50
60
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5


Simulated Vs Theoritical Bit Error Rate for 8-QAM
Bit Error rate
Eb/No (dB)
Simulation
Theory
10^{-1}
10^{-2}
10^{-3}
0
1
2
3
4
5
6
7
8
9
10

Published with MATLAB® R2019b