Table of Contents

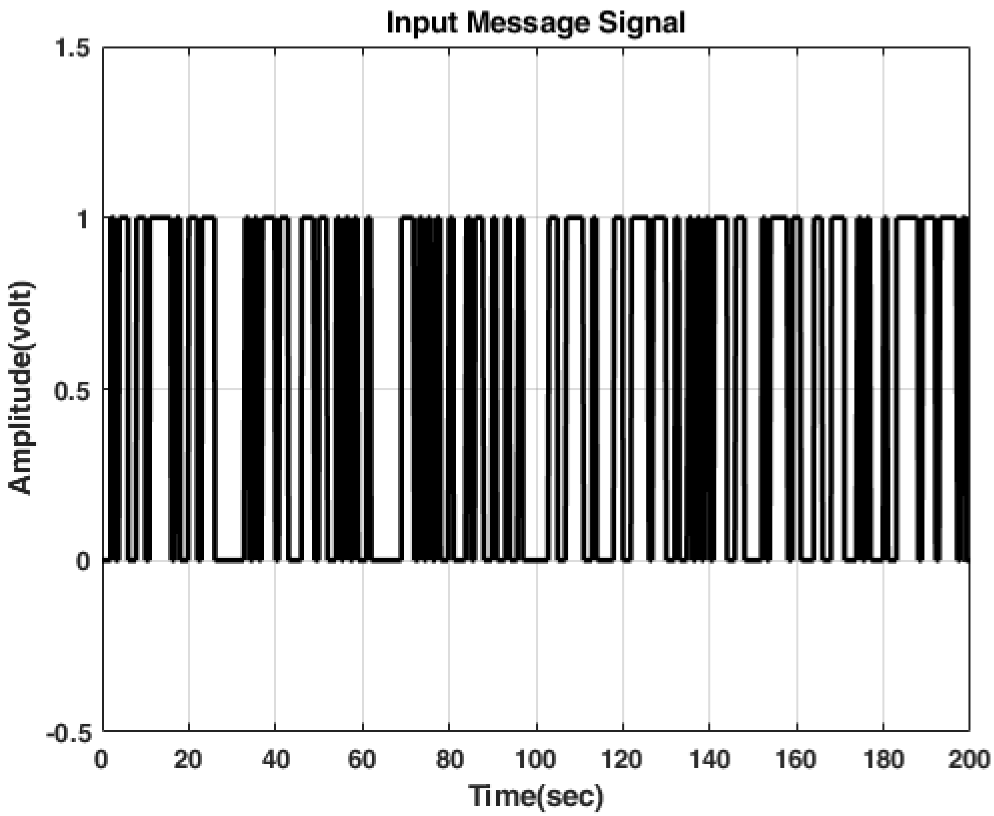

```
clc;
clear;
```

1) BFSK

Define transmitted signal (BFSK)

```
fs = 64; %Sampling Frequency
df = 32; %Frequency Separation
Tb=1; %Bit duration = 1 msec
N = 200; %Number of bits
X_input= randi([0, 1],1,N);  %Binary signal
M = 2;
m = log2(M);
ns=2;  %Number of samples per symbol
X_digit=[];
nb=10000;  %Number of points between two symbols (it's used to convert
 the symbols into continuous digital signal)
for i=1:N
```

```
    if X_input(i)==1
       x_temp=ones(1,nb);
    else
        x_temp=zeros(1,nb);
    end
    X_digit=[X_digit x_temp];
end
t_sig = Tb/nb : Tb/nb : N*Tb; %Time vector of continuous digital
 signal
%Plotting the input message signal
figure();
plot(t_sig,X_digit, 'LineWidth',2,'Color','black'); grid on; xlim([0
 Tb*N]); ylim([-0.5 1.5]);
xlabel('Time(sec)'); ylabel('Amplitude(volt)'); title('Input Message
 Signal');
```

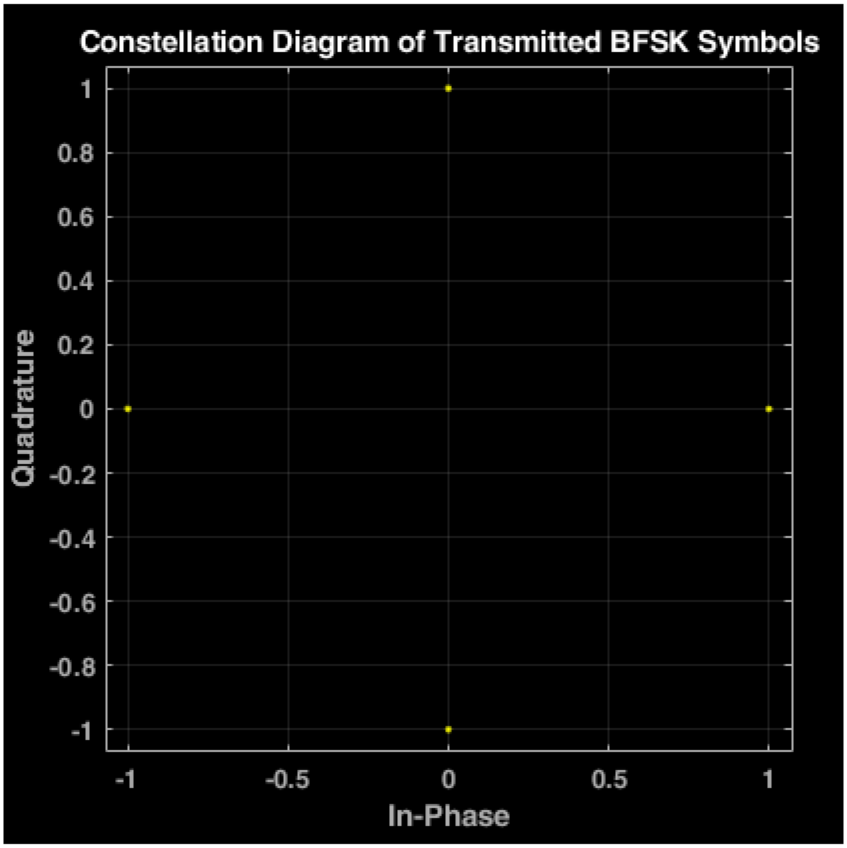


BFSK Modulation (BFSK)

```
BFSKMOD = fskmod(X_input, M, df, ns, fs); %BFSK Modulation
```

Constellation Diagram (BFSK)

```
scatterplot(BFSKMOD); grid on;
title('Constellation Diagram of Transmitted BFSK Symbols');
```

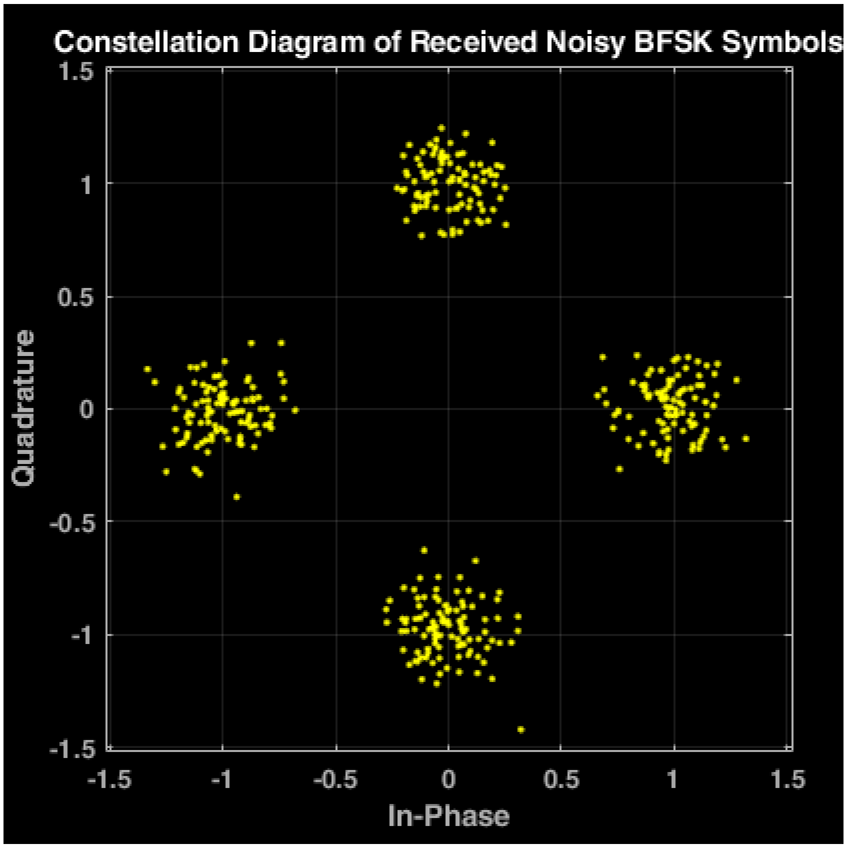


Noise in the Communication Channel (BFSK)

```
SNR = 15+10*log10(m);
Y = awgn(BFSKMOD, SNR, 'measured');  %Adds white Gaussian noise to the
 Modulated signal
```

Plotting the Noisy Signal

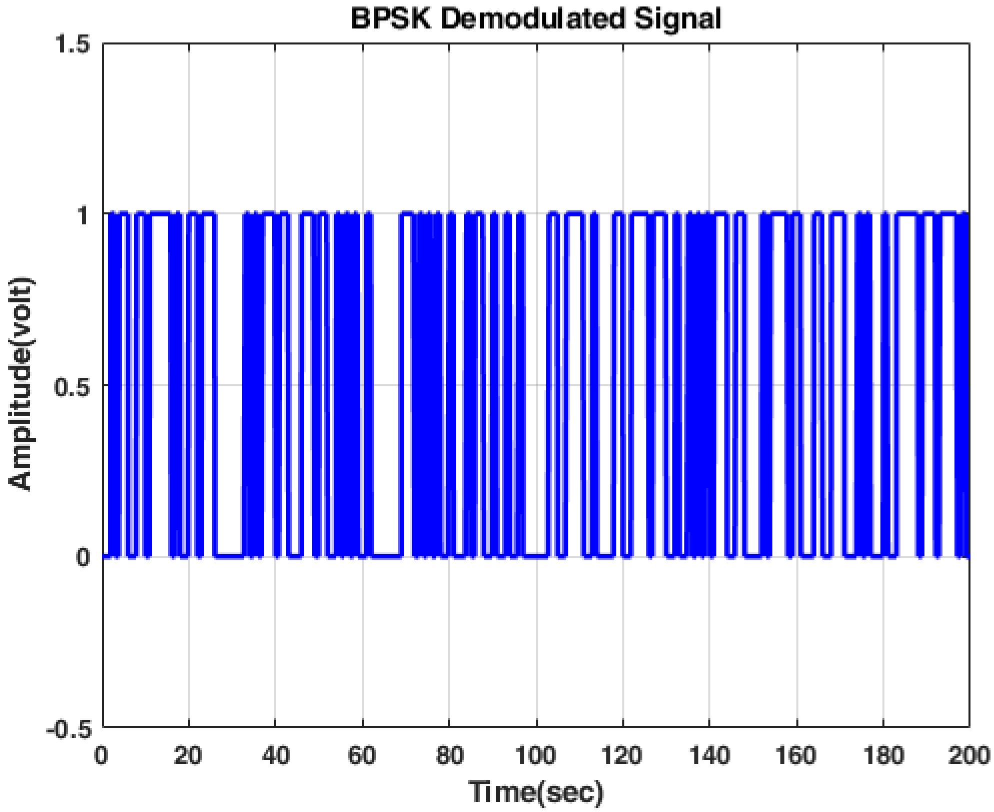
```
scatterplot(Y); grid on;
title('Constellation Diagram of Received Noisy BFSK Symbols');
```



BFSK Demodulation (BFSK)

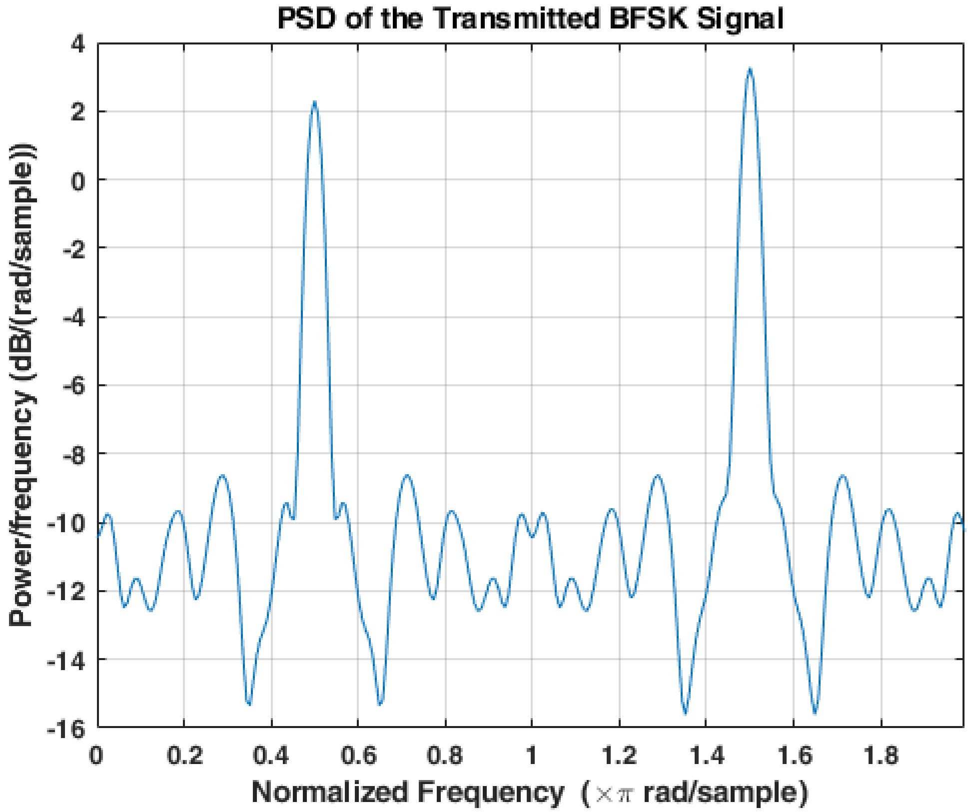
```
X_demod=fskdemod(Y, M, df, ns, fs); %Demodulation

%Convert the symbols into continuous digital signal
X_dem_sig = [];
for i=1:length(X_demod)
    if X_demod(i)==1
       x_temp_dem=ones(1,nb);
    else
        x_temp_dem=zeros(1,nb);
    end
    X_dem_sig=[X_dem_sig x_temp_dem];
end
t_sig_dem = Tb/nb : Tb/nb : length(X_demod)*Tb; %Time vector of
 continuous digital signal
figure();
plot(t_sig_dem,X_dem_sig, 'LineWidth',2,'Color','blue'); grid on;
 xlim([0 Tb*length(X_demod)]); ylim([-0.5 1.5]);
xlabel('Time(sec)'); ylabel('Amplitude(volt)'); title('BPSK
 Demodulated Signal');
```



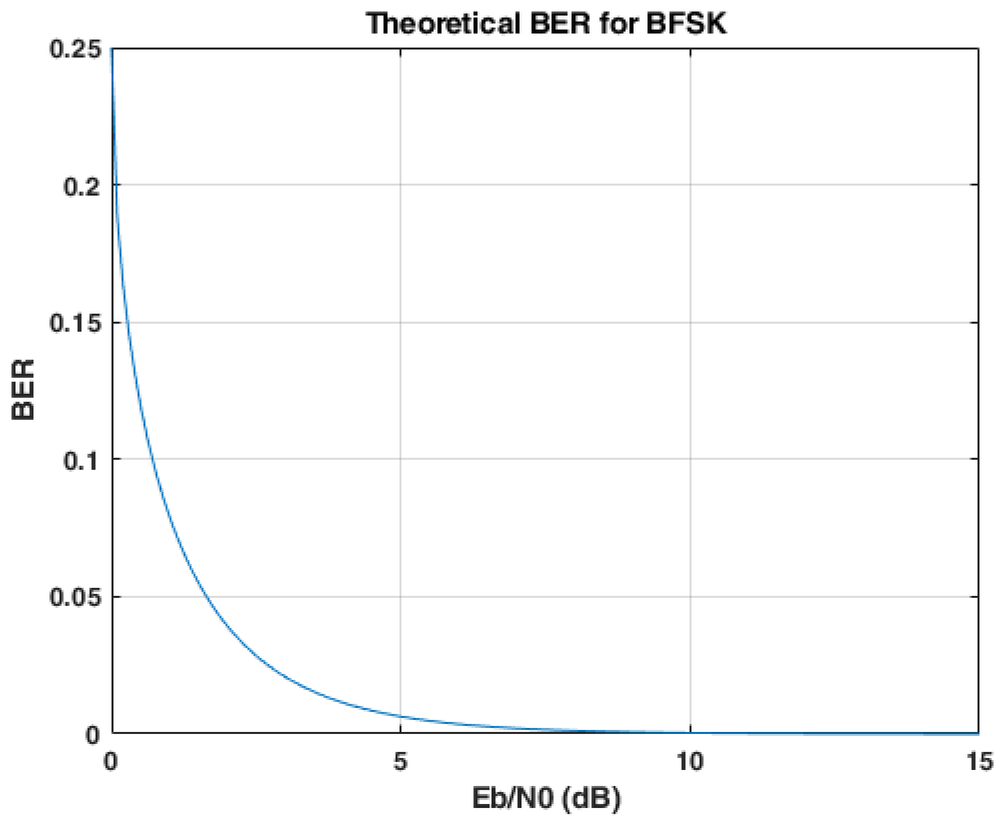
PSD

```
power_Spec = spectrum.welch;
PSD_BFSK=psd(power_Spec,BFSKMOD);
figure(); plot(PSD_BFSK); title('PSD of the Transmitted BFSK Signal');
 grid on;
```



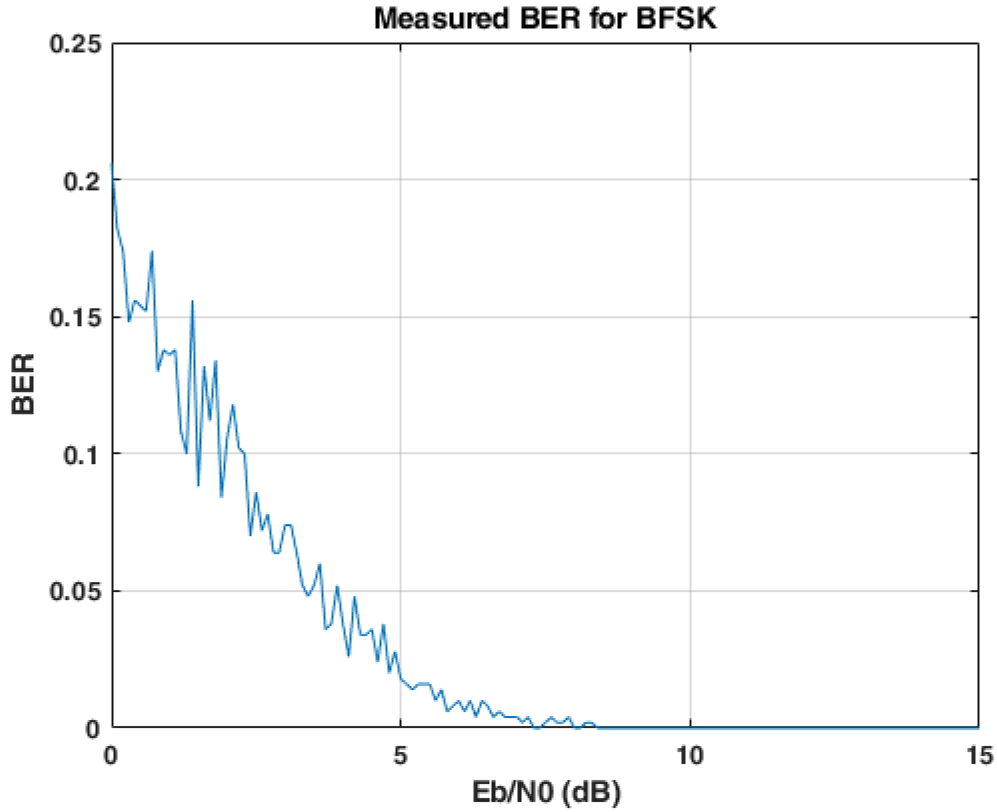
Theoretical BER (BFSK)

```
EbNo=15;
BER_Theo=zeros(1,EbNo/0.1);
j=1;
for i=0:0.1:EbNo
    BER_Theo(j)=0.25*erfc(sqrt(i/2));
    j=j+1;
end
Eb_No=0:0.1:15;
figure(); plot(Eb_No,BER_Theo); grid on;
ylabel('BER'); xlabel('Eb/N0 (dB)'); title('Theoretical BER for
 BFSK');
```



Measured BER (BFSK)

```
k=1;
l=500;
BER_Measured = zeros(1,EbNo/0.1);
for i=0:0.1:EbNo
    rand_bits = randi([0 1],l,1);
    BFSKmodul=fskmod(rand_bits, M, df, ns, fs);
    yy=awgn(BFSKmodul,i+10*log10(m),'measured');
    demod=fskdemod(yy,M,df,ns,fs);
    [ou ,e_ratio]=biterr(demod,rand_bits);
    BER_Measured(k)=e_ratio;
    k=k+1;
end
figure(); plot(Eb_No,BER_Measured); grid on;
ylabel('BER'); xlabel('Eb/N0 (dB)'); title('Measured BER for BFSK');
```



%%%

1) 16-QAM

Define transmitted signal (16-QAM)

```
Tb=1; %Bit duration = 1 msec
N = 2000; %Number of bits
X_input= randi([0, 1],1,N);  %Binary signal
M = 16;
m = log2(M);
X_digit=[];
nb=10000;  %Number of points between two symbols (it's used to convert
 the symbols into continuous digital signal)
```

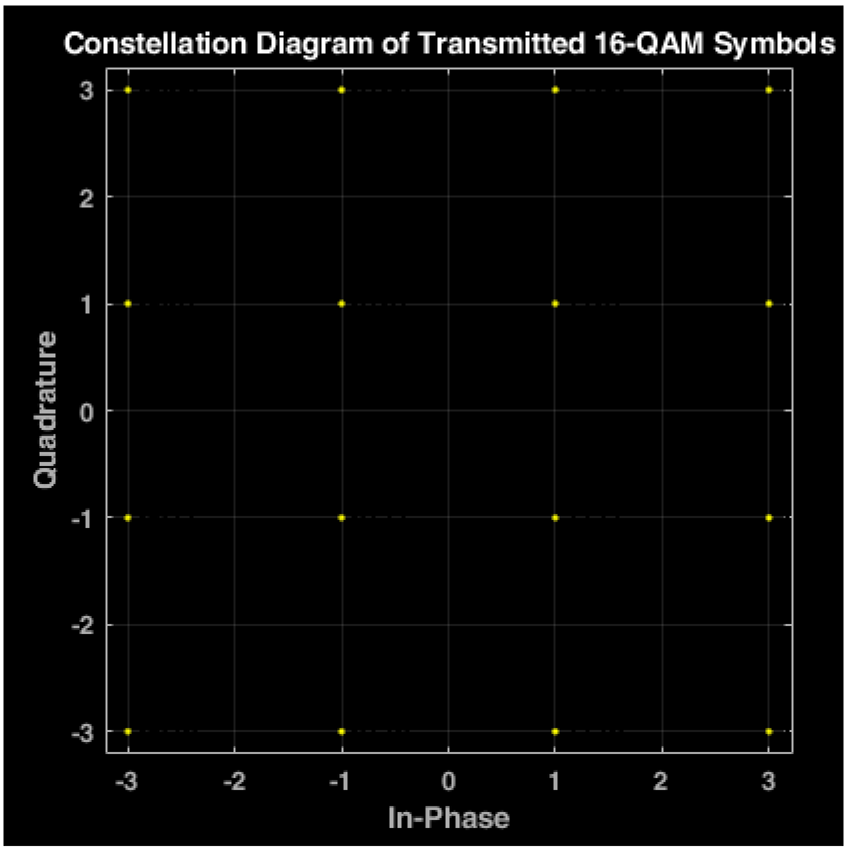
16-QAM Modulation (QAM)

```
k = N/m;
```

```
QAM_Sig=reshape(X_input, k, m); %Dividing the bits into signals each
 of which with 4 bits.
QAM_Sig_dec = bi2de(QAM_Sig); %converting the binary values into
 decimal numbers.
QAM_ModSig = qammod(QAM_Sig_dec,M); %16-QAM Modulation and Gray Coding
```

Constellation Diagram (QAM)

```
scatterplot(QAM_ModSig); grid on;
title('Constellation Diagram of Transmitted 16-QAM Symbols')
text(real(QAM_ModSig)+0.1, imag(QAM_ModSig), dec2bin(QAM_Sig_dec))
```

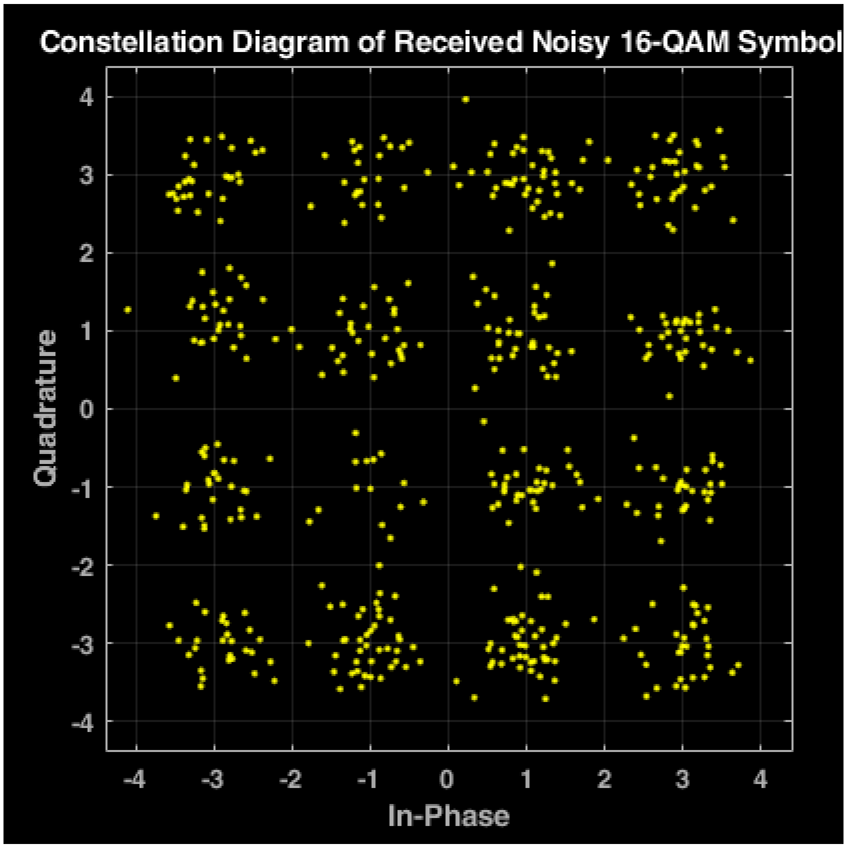


Noise in the Communication Channel (BFSK)

```
SNR = 10+10*log10(m);
Y = awgn(QAM_ModSig, SNR, 'measured');
```

Plotting the Noisy Signal

```
scatterplot(Y); grid on;
title('Constellation Diagram of Received Noisy 16-QAM Symbols');
```



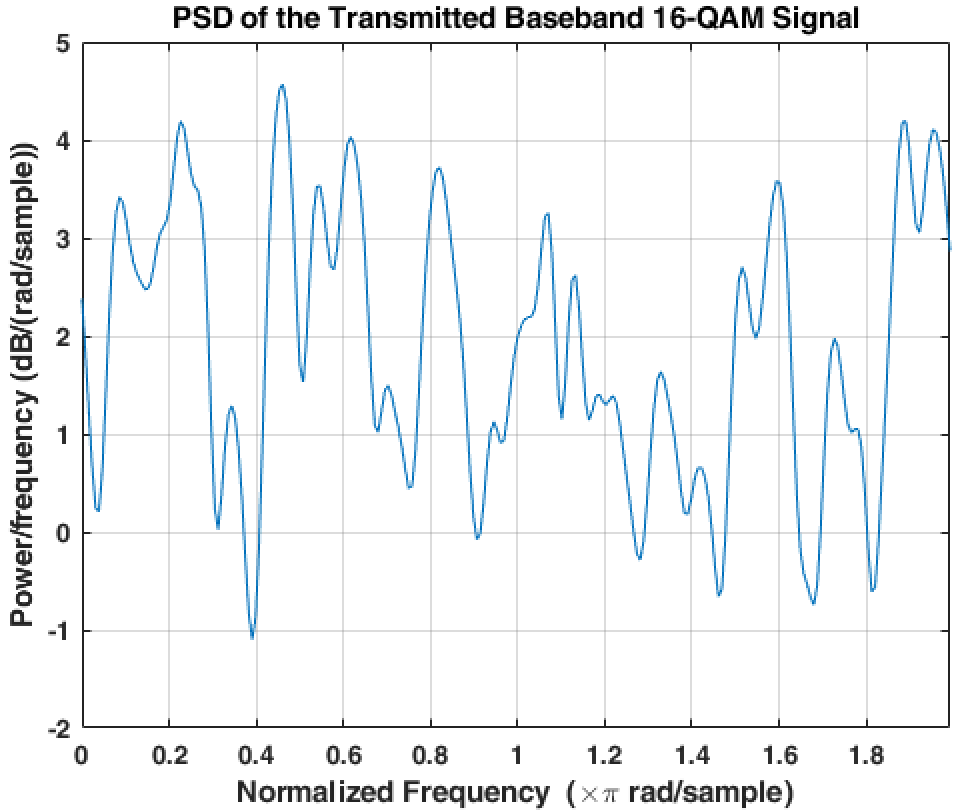
BFSK Demodulation (QAM)

```
X_demod_De=qamdemod(Y, M); %QAM Demodualtion
X_demod_Bi = de2bi(X_demod_De, m); %Converting the recieved decimal
 values to binary symbols
X_demod = reshape(X_demod_Bi, N, 1); %Reshaping the symbols in a
 single row vector

%Convert the symbols into continuous digital signal
```

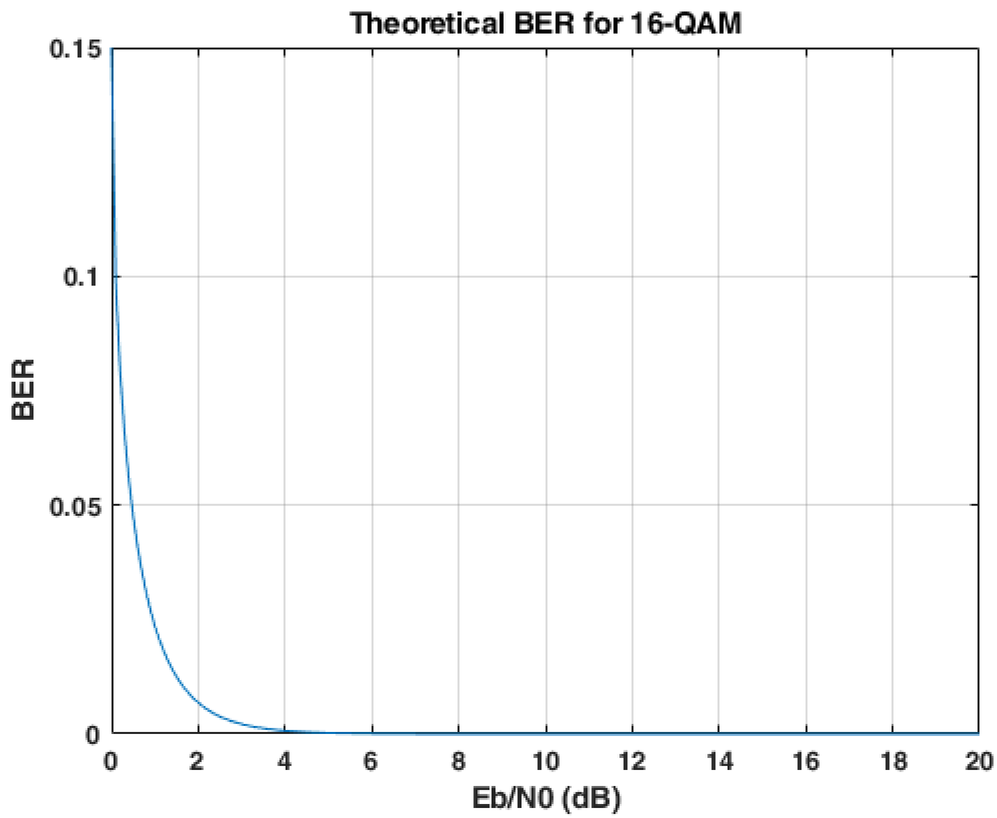
PSD

```
power_Spec = spectrum.welch;
PSD_QAM = psd(power_Spec, QAM_ModSig);
figure(); plot(PSD_QAM); title('PSD of the Transmitted Baseband 16-QAM
 Signal'); grid on;
```



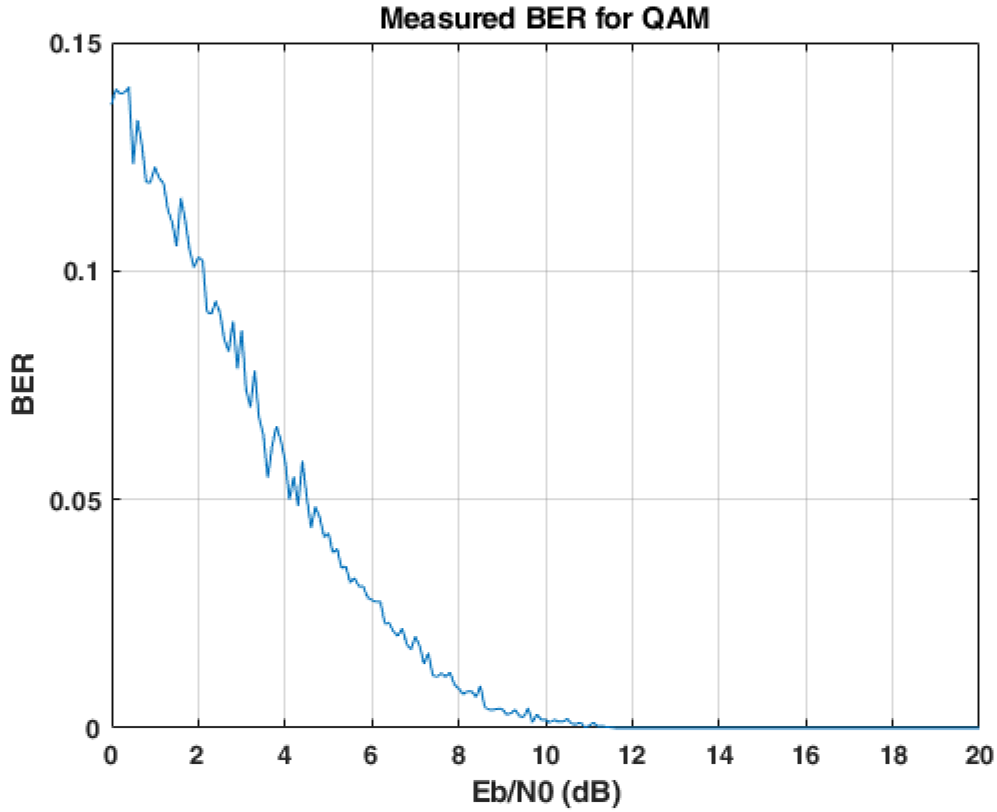
Theoretical BER (QAM)

```
EbNo=20;
BER_Theo=zeros(1,EbNo/0.1);
j=1;
k = 1/sqrt(M);
for i=0:0.1:EbNo
    BER_Theo(j)=0.2*(1-k)*erfc(sqrt(i));
    j=j+1;
end
Eb_No=0:0.1:20;
figure(); plot(Eb_No,BER_Theo); grid on;
ylabel('BER'); xlabel('Eb/N0 (dB)'); title('Theoretical BER for 16-
QAM');
```



Measured BER (QAM)

```
k=1;
l = 5000;
o = l/m;
BER_Measured = zeros(1,EbNo/0.1);
for i=0:0.1:EbNo
    rand_bits = randi([0 1],l,1);
    randSeq=reshape(rand_bits, o, m);
    randDec = bi2de(randSeq);
    QAMMMod=qammod(randDec, M);
    YY=awgn(QAMMMod,i+10*log10(m),'measured');
    yy_o = qamdemod(YY,M);
    OUT=de2bi(yy_o, m);
    OUT=reshape(OUT, l, 1);
    [er,ratio]=biterr(OUT, rand_bits);
    BER_Measured(k)=ratio;
    k=k+1;
end
figure(); plot(Eb_No,BER_Measured); grid on;
ylabel('BER'); xlabel('Eb/N0 (dB)'); title('Measured BER for QAM');
```



Comment:

```
disp("The Graphs show that the BER for QAM is less than the BER for
 BFSK while the BER for both of them decreases with the increase of
 SNR. ");
disp("This is applied for both the theoritical and measured BERs. This
 also agrees with what we obtained theoritically in the lecutres. ");
```

The Graphs show that the BER for QAM is less than the BER for BFSK while the BER for both of them decreases with the increase of SNR. This is applied for both the theoritical and measured BERs. This also agrees with what we obtained theoritically in the lecutres.

Comparison between BPSK , QPSK, BFSK, QAM

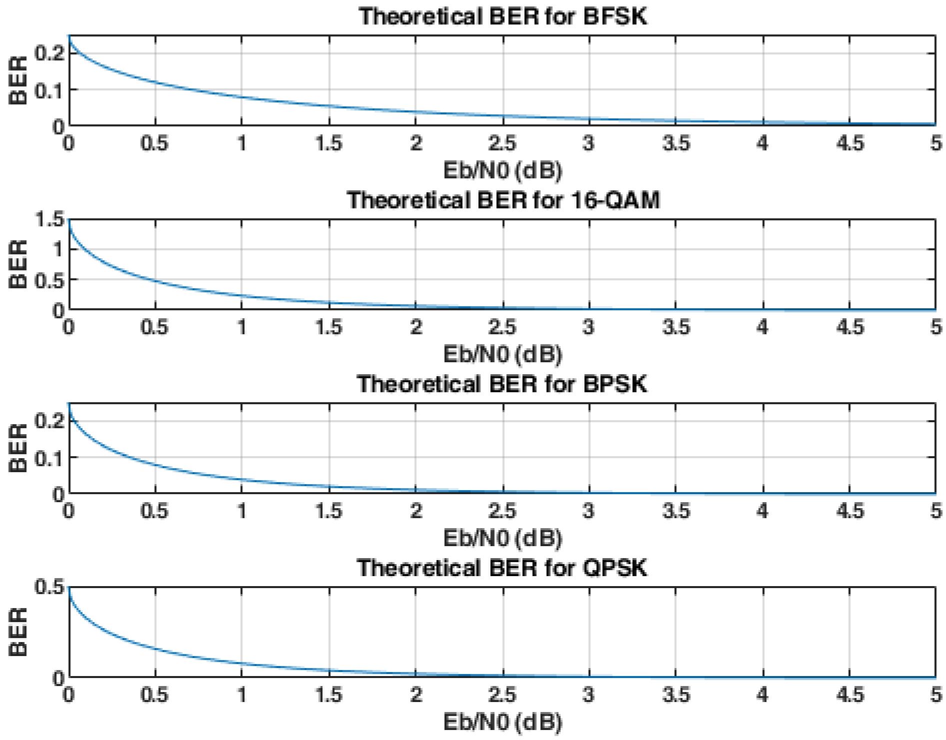
```
%Theoretical BER (BFSK)
EbNo=5;
BER_Theo=zeros(1,EbNo/0.1);
j=1;
for i=0:0.01:EbNo
    BER_Theo(j)=0.25*erfc(sqrt(i/2));
    j=j+1;
end
```

```
Eb_No=0:0.01:5;
figure(); subplot(4,1, 1); plot(Eb_No,BER_Theo); grid on;
ylabel('BER'); xlabel('Eb/N0 (dB)'); title('Theoretical BER for
 BFSK');


% Theoretical BER (QAM)
EbNo=5;
BER_Theo=zeros(1,EbNo/0.1);
j=1;
k = 1/sqrt(16);
for i=0:0.01:EbNo
    BER_Theo(j)=2*(1-k)*erfc(sqrt(i));
    j=j+1;
end
Eb_No=0:0.01:5;
subplot(4,1, 2); plot(Eb_No,BER_Theo); grid on;
ylabel('BER'); xlabel('Eb/N0 (dB)'); title('Theoretical BER for 16-
QAM');


%Theoretical BER (BPSK)
EbNo=5;
BER_Theo=zeros(1,EbNo/0.1);
j=1;
for i=0:0.01:EbNo
    BER_Theo(j)=0.25*erfc(sqrt(i));
    j=j+1;
end
Eb_No=0:0.01:5;
subplot(4,1, 3); plot(Eb_No,BER_Theo); grid on;
ylabel('BER'); xlabel('Eb/N0 (dB)'); title('Theoretical BER for
 BPSK');


%Theoretical BER (QPSK)
EbNo=5;
BER_Theo=zeros(1,EbNo/0.1);
j=1;
for i=0:0.01:EbNo
    BER_Theo(j)=0.5*erfc(sqrt(i));
    j=j+1;
end
Eb_No=0:0.01:5;
subplot(4,1, 4); plot(Eb_No,BER_Theo); grid on;
ylabel('BER'); xlabel('Eb/N0 (dB)'); title('Theoretical BER for
 QPSK');
```



Comment

```
disp("The Graphs show that the BER for QAM is the most, then BER of
 QPSK, then BFSK, then BPSK");
```

```
The Graphs show that the BER for QAM is the most, then BER of QPSK,
 then BFSK, then BPSK
```

Published with MATLAB® R2021a