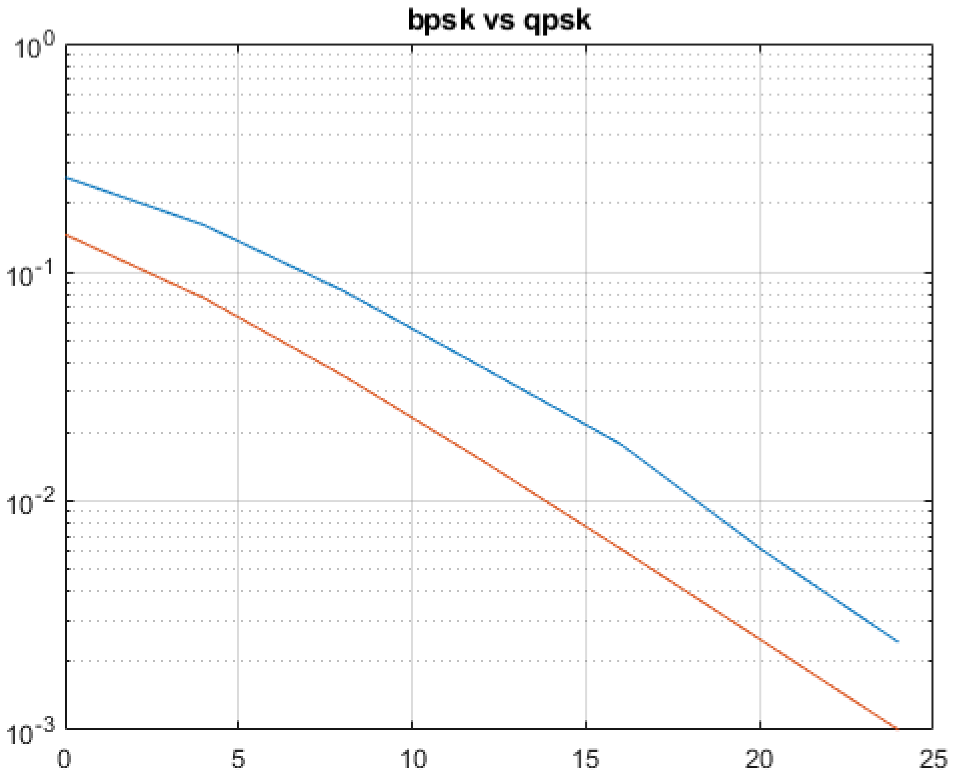
```
clc;
clear all;
close all;

N= 10000;

% r =[0 0 0 1 1 0 1 1];
% N = length(r);
r = randi([0,1],1,N);
k =1;
for i=1:2:N                         %converted to bpsk

        if((r(i)==0)&&(r(i+1)==0))
            r1(k) = 1+1j;

        elseif((r(i)==0)&&(r(i+1)==1))
            r1(k) = -1+1j;

        elseif((r(i)==1)&&(r(i+1)==0))
            r1(k) = -1-1j;

        elseif((r(i)==1)&&(r(i+1)==1))
            r1(k) = 1-1j;

        end
    k = k+1;

end
h = (randn(1,N/2) + 1i*randn(1,N/2))*sqrt(1/2);  %h complex random ;
 ray light faded channel
n = randn(1,N/2)+j*randn(1,N/2);


snr_db = 0:4:24;
kk=1;% snr in db

for k = 1: length(snr_db)

snr_linear(k) = 10.^(snr_db(k)/10);          % converted snr to
 linear
sigma(k) = 1./(snr_linear(k)).^(1/2);                % find sigma

y = h.*r1+ sigma(k).*n;                 % find y = bpsk_signal +
 sigma*noise
y1 = y./h;

% convert y sequence into bpsk take threshold value = 0
% z is your constructed signal
% y1 is output signal with noise
for j=1:N/2
    a=real(y1(j));
    b = imag(y1(j));
```

```
   if((a>=0)&&(b>=0))
            z(2*j-1)=0;
            z(2*j)=0;

   elseif((a<0)&&(b>=0))
            z(2*j-1)=0;
            z(2*j)=1;

   elseif((a<0)&&(b<0))
            z(2*j-1)=1;
            z(2*j)=0;

   elseif((a>=0)&&(b<0))
            z(2*j-1)=1;
            z(2*j)=1;
   end
end

% check bit by bit that r and z is same or not

ber_th(k) = (1/2)*(1-(snr_linear(k)/(1+snr_linear(k))).^(1/2));

% check bit by bit that r and z is same or not

count_error(k)=0;
for jj=1:N
    if(r(jj)~=z(jj))
        count_error(k)=count_error(k)+1;
%     else
%         count_error;
    end
end
end

ber_prac  = count_error./N;
semilogy(snr_db,ber_prac);
grid on;
hold on;
semilogy(snr_db,ber_th);
title("bpsk vs qpsk");
```



Published with MATLAB® R2018b