

Error Performance of QPSK-Niya Paul U2201156

Generation of Digital Data

```
clc
clear all
close all
N=input("Enter the number of digital data : ");
digitaldata=randi([0,1],[1,N]);
% Splitting into odd and even components
d_odd=[];
d_even=[];
for i=1:N
    if mod(i,2)==0
        d_even=[d_even digitaldata(i)];
    else
        d_odd=[d_odd digitaldata(i)];
    end
end
```

Generation of Carrier wave

```
n=input("Enter the number of samples in a cycle : ");
t=linspace(0,4*pi,n);
C1=sin(t);
C2=cos(t);
```

Normalization of the Carrier

```
Energy1=C1*transpose(C1);
C1=C1./sqrt(Energy1);
Energy2=C2*transpose(C2);
C2=C2./sqrt(Energy2);
```

QPSK Generation

```
% BPSK Generation - Odd Component
BPSK_odd=[];
for i=1:length(d_odd)
    if d_odd(i)==1
        a=C1;
    else
        a=-C1;
    end
    BPSK_odd=[BPSK_odd a];
end
```

% BPSK Generation - Even Component

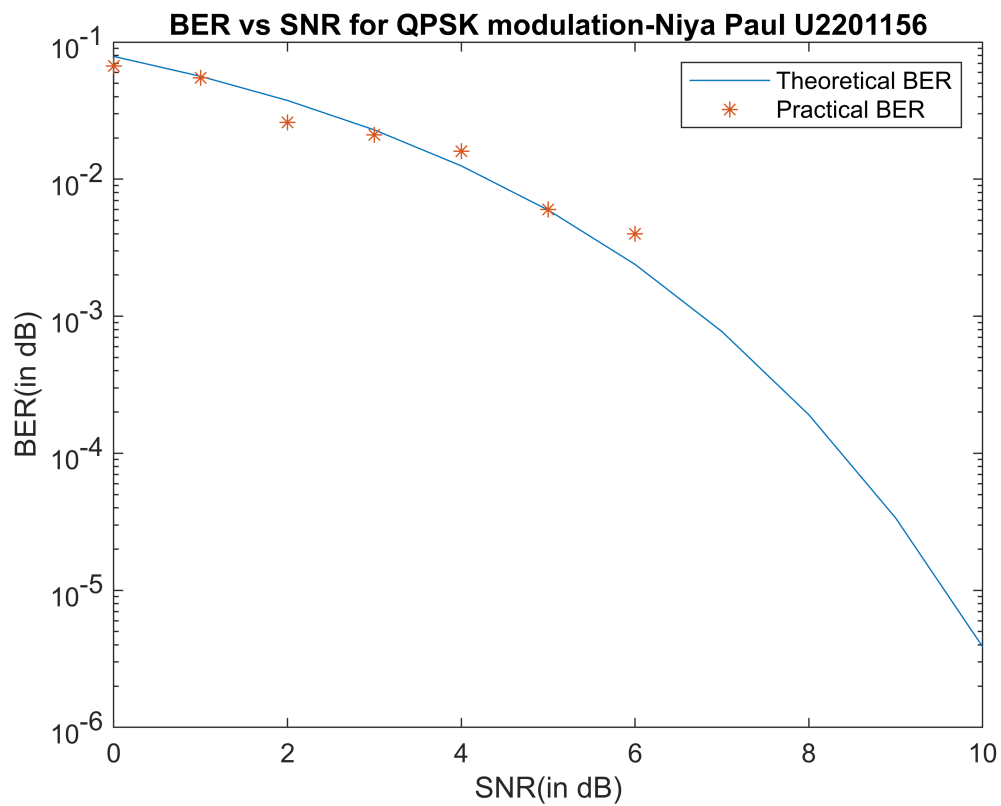
```
BPSK_even=[];  
for i=1:length(d_even)  
    if d_even(i)==1  
        a=C2;  
    else  
        a=-C2;  
    end  
    BPSK_even=[BPSK_even a];  
end  
QPSK=BPSK_even+BPSK_odd;
```

Noise Generation

```
th_ber_final=[];  
pr_ber_final=[];  
snr1=0:10;  
for snr=0:1:10  
    N0=1./(10^(snr/10));  
    th_ber=qfunc(sqrt(2/N0));  
    th_ber_final=[th_ber_final th_ber];  
    std_dev=sqrt(N0/2);  
    noise=std_dev.*(randn(1,length(QPSK))+1j*randn(1,length(QPSK)));  
    final=QPSK+noise;  
    %Demodulation  
    output_resaped=reshape(final,n,length(digitaldata)/2);  
    outputodd=C1*output_resaped;  
    outputeven=C2*output_resaped;  
    data_odd=zeros(length(outputodd));  
    data_even=zeros(length(outputeven));  
    for i=1:length(outputodd)  
        data_odd(i)=real(outputodd(i))>=0;  
    end  
    for i=1:length(outputeven)  
        data_even(i)=real(outputeven(i))>=0;  
    end  
    data=[];  
    for i=1:N/2  
        data=[data data_odd(i) data_even(i)];  
    end  
    c=0;  
    for i=1:N  
        if digitaldata(i)~=data(i)  
            c=c+1;  
        end  
    end  
    pr_ber=c/N;  
    pr_ber_final=[pr_ber_final pr_ber];  
end
```

Plotting - BER vs SNR

```
figure;  
grid on;  
semilogy(snr1,th_ber_final,snr1,pr_ber_final,'*');  
xlabel('SNR(in dB)');  
ylabel('BER(in dB)');  
title('BER vs SNR for QPSK modulation-Niya Paul U2201156')  
legend('Theoretical BER','Practical BER');
```



Plotting - Constellation diagram

```
figure
recievedSymbol=outputeven+1j*outputodd;
scatter(real(recievedSymbol),imag(recievedSymbol))
title('Constellation diagram for QPSK at SNR=10dB')
xlabel('In phase component')
ylabel('Quadrature phase component')
```

