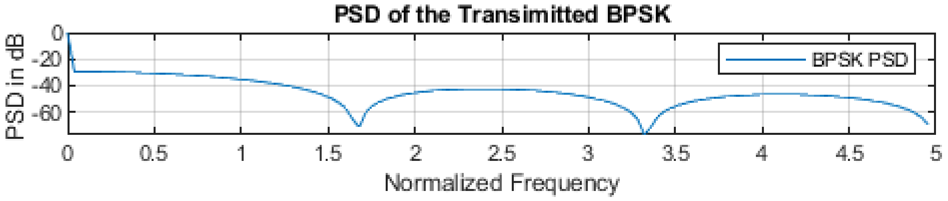
Table of Contents

PSD of BPSK

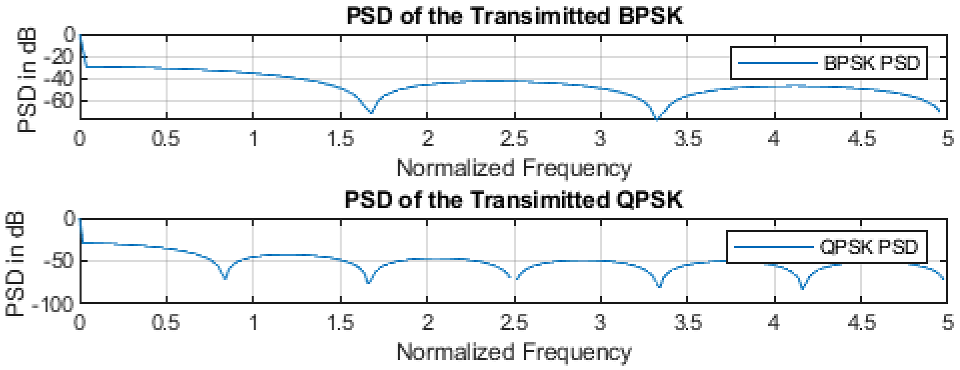
```
Tb = 5; %% bit duration in secs
Eb = 5; %% Energy per bit
N_bit = 50; %%number of samples per bit
t_bit = linspace(0,Tb,N_bit); %% time base for each bit
msg_l = 10; %% number of bits sent
t_signal = linspace(0,msg_l*Tb,msg_l*N_bit); %% total duration of the
 messag
fc=2/Tb; %% frequency of the carrier
N0=5;
message= randi([0 1],1,msg_l);
encodedMessage=[];

for i=1:1:msg_l
    if message(i)==1
        signal_seg=sqrt(Eb)*ones(1,N_bit);
    elseif message(i) == 0
        signal_seg=-sqrt(Eb)*ones(1,N_bit);
    end
    encodedMessage=[encodedMessage signal_seg];
end
base_band = abs(sqrt(2/Tb)*encodedMessage);
ts=Tb/N_bit;
[psd,f]= periodogram(base_band,[],[],1/ts*10);

psd=10*log10(psd/(2*Eb));
f=f/Tb;

subplot(4,1,1)
plot(f,psd)
xlim([0 5])
title('PSD of the Transimitted BPSK')
ylabel('PSD in dB ')
xlabel('Normalized Frequency')
grid on
legend('BPSK PSD')
```



PSD OF QPSK

```
Tb = 5; %% bit duration in secs
T = 2 * Tb;  %%dibit duration
Eb = 1; %% enerdy per bit
E = 2 * Eb; %%energy per symbol
N_bit = 50; %%number of samples per bit
N_dibit = 2 * N_bit;
t_bit=linspace(0,Tb,N_bit); %% time base for each bit
t_dibit = linspace(0,T,N_dibit);
msg_l = 10; %% number of bits sent which has to be even
t_signal = linspace(0,msg_l*Tb,msg_l*N_bit);%% total duration of the
 messag
fc=2/Tb; %% frequency of the carrier
N0=30;
message = randi([0 1],1,msg_l);
odd_bits = [];
even_bits = [];

for i=1:1:msg_l
    if mod(i,2)== 1
        odd_bits=[odd_bits message(i)];
    else
        even_bits=[even_bits message(i)];
    end
```

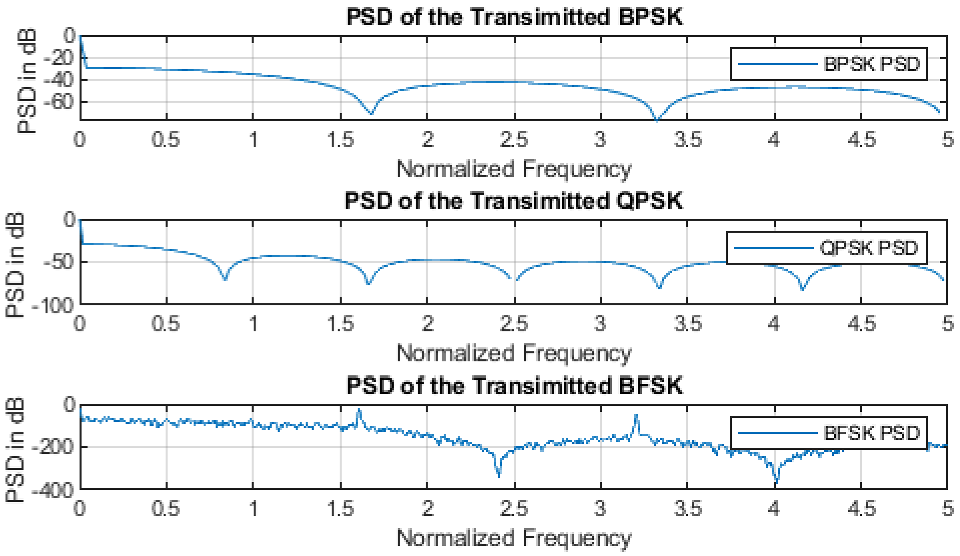
```
end

encodedodd =[];
encodedeven =[];

for i=1:1:length(odd_bits)
    if odd_bits(i)==1
        signal_seg=sqrt(E)*ones(1,2*N_bit);
    elseif odd_bits(i) == 0
        signal_seg=-sqrt(E)*ones(1,2*N_bit);
    end
    encodedodd=[encodedodd signal_seg];
end

for i=1:1:length(even_bits)
    if even_bits(i)==1
        signal_seg=sqrt(E)*ones(1,2*N_bit);
    elseif even_bits(i) == 0
        signal_seg=-sqrt(E)*ones(1,2*N_bit);
    end
    encodedeven=[encodedeven signal_seg];
end
base_band_I = abs(sqrt(2/T)*encodedeven);
ts=T/N_dibit;
[psd_I,f]= periodogram(base_band_I,[],[],1/ts*10);

base_band_Q = abs(sqrt(2/T)*encodedodd);
ts=T/N_dibit;
[psd_Q,f]= periodogram(base_band_Q,[],[],1/ts*10);

psd = psd_I + psd_Q;
psd=10*log10(psd/(2*E));
f=f/T;


subplot(4,1,2)
plot(f,psd)
xlim([0 5])
title('PSD of the Transimitted QPSK')
ylabel('PSD in dB ')
xlabel('Normalized Frequency')
grid on
legend('QPSK PSD')
```



PSD OF BFSK

```
Tb = 5; %% bit duration in secs
Eb = 5; %% Energy per bit
N_bit = 500; %%number of samples per bit
t_bit = linspace(0,Tb,N_bit); %% time base for each bit
msg_l = 500 ; %% number of bits sent
t_signal = linspace(0,msg_l*Tb,msg_l*N_bit); %% total duration of the
 messag
N0 = 100;
f1 =2+1/Tb;
f2 =2*2/Tb;
fc =2/Tb;
message = randi([0 1],1,msg_l);

base_I = sqrt(2*Eb/Tb)* cos(pi*t_bit/Tb);
base_Q = sqrt(2*Eb/Tb)* sin(pi*t_bit/Tb);
base_band = [];

for i = 1:1:length(message)
    seg = base_I + (-1)^message(i)*base_Q;
   base_band = [base_band seg];
end

ts = Tb/N_bit;
```

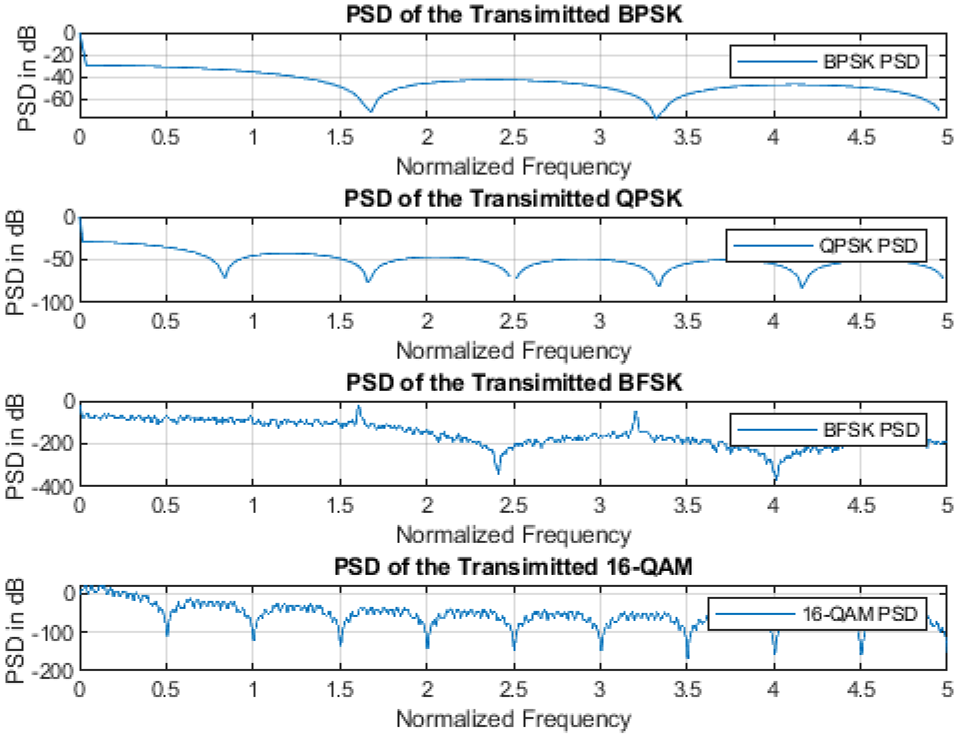
```
[psd,f]= periodogram(base_band,[],[],1/ts*40);

f = f/Tb;
psd = 10*log10(psd/(2*Eb));
psd = filter([ 0.8 0.8 0.8 0.8 0.8],1,psd);

subplot(4,1,3)
plot(f,psd)
xlim([0 5])

title('PSD of the Transimitted BFSK')
ylabel('PSD in dB ')
xlabel('Normalized Frequency')
grid on
legend('BFSK PSD')
```



PSD of 16-QAM

```
Tb = 5; %% bit duration in secs
T = 4 * Tb;  %%quadbit duration
E0 = 1; %% enerdy of the signal with the smallest amplitude
M = 16;
Eav = 2/3 * (M-1) * E0;
N_bit = 5000; %%number of samples per bit
N_quadbit = 4 * N_bit;
t_bit = linspace(0, Tb ,N_bit); %% time base for each bit
```

```
t_quadbit = linspace(0, T , N_quadbit);
msg_l = 500 ; %% number of bits sent which has to be divisible by 4---
> 300 point
t_signal = linspace(0, msg_l*Tb, msg_l*N_bit);%% total duration of the
 messag
fc = 2/Tb; %% frequency of the carrier
N0 = 30;
message = randi([0 1],1,msg_l);
encodedMessage = [];
for i = 1:4:length(message)
    d = bi2de(flip(message(i:i+3)));
    encodedMessage = [encodedMessage d];
end
a = [-3,-3,-3,-3,-1,-1,-1,-1,3,3,3,3,1,1,1,1]*sqrt(E0);
b = [-3,-1,3,1,-3,-1,3,1,-3,-1,3,1,-3,-1,3,1]*sqrt(E0);
base_band = [];

for i = 1:1:length(encodedMessage)
    index = encodedMessage(i) + 1;
    seg = a(index)*ones(1,N_quadbit) + b(index)* ones(1,N_quadbit);
    base_band = [base_band seg];
end

ts = Tb/N_bit;
[psd,f]= periodogram(base_band,[],[],1/ts*50);

f = f/Tb;

psd = 10*log10(psd/(2*E0));
psd = filter([ 0.5 0.5 0.5 0.5 0.5],1,psd);

subplot(4,1,4)
plot(f,psd)
xlim([0 5])
title('PSD of the Transimitted 16-QAM')
ylabel('PSD in dB ')
xlabel('Normalized Frequency')
grid on
legend('16-QAM PSD')
```



Published with MATLAB® R2019b