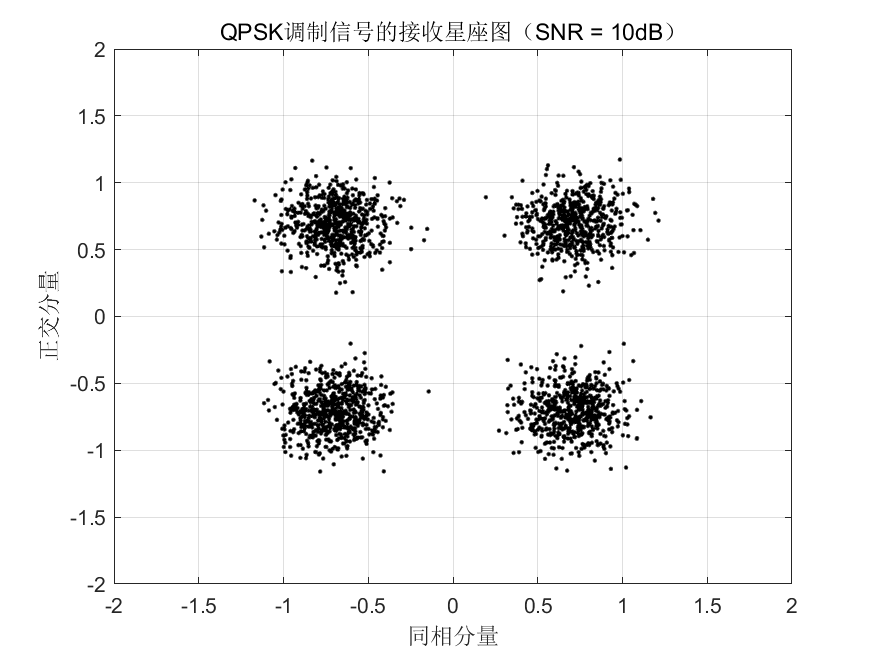
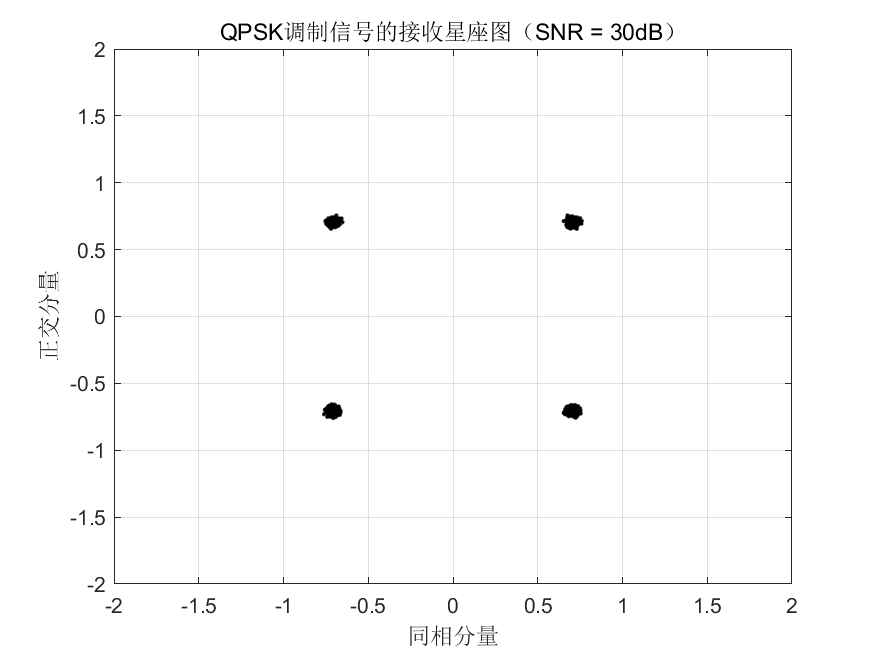


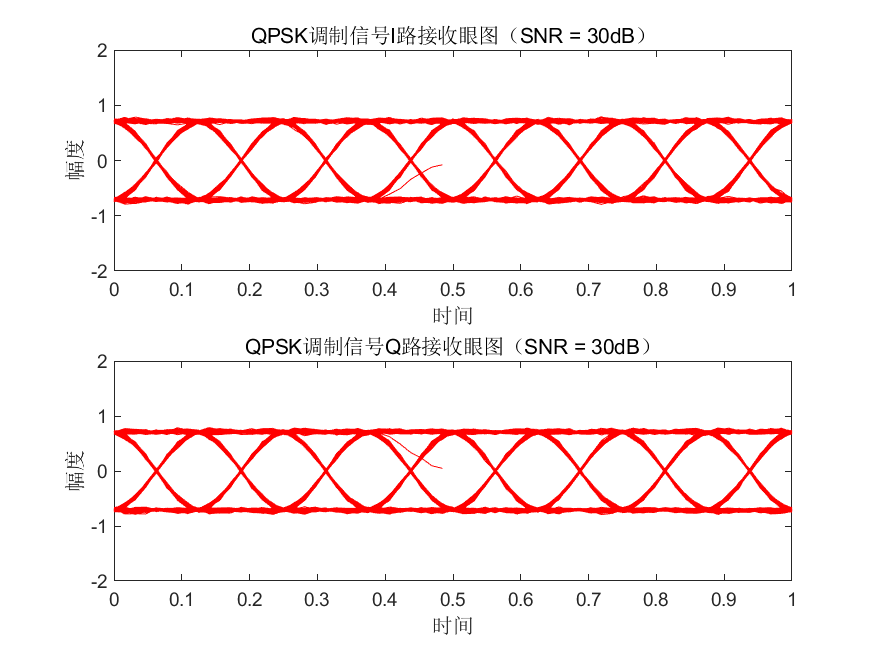
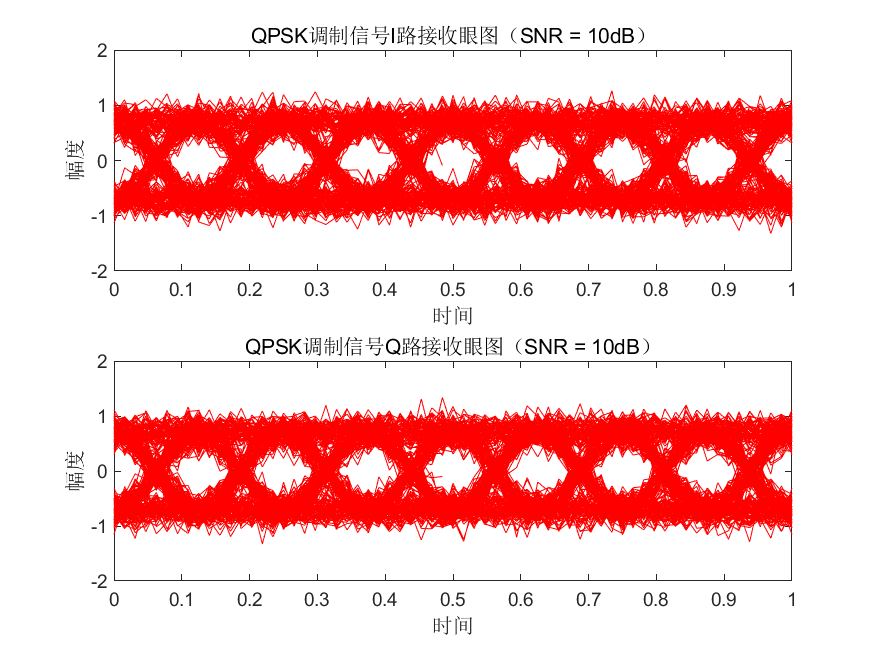
3、给定进制的PSK调制，以QPSK为例，仿真不同信噪比下的波形

3.1 比较不同信噪比下的QPSK星座图

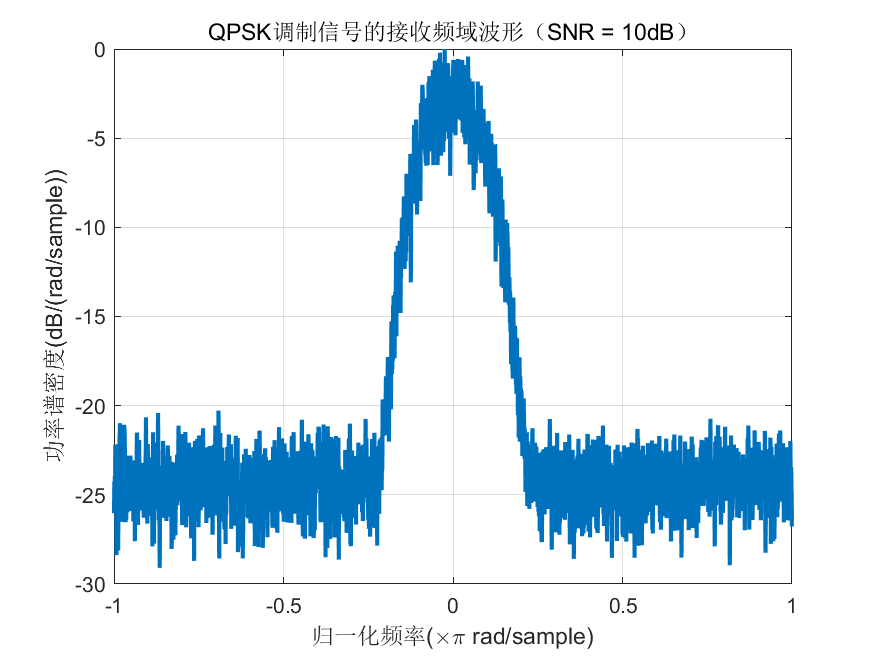


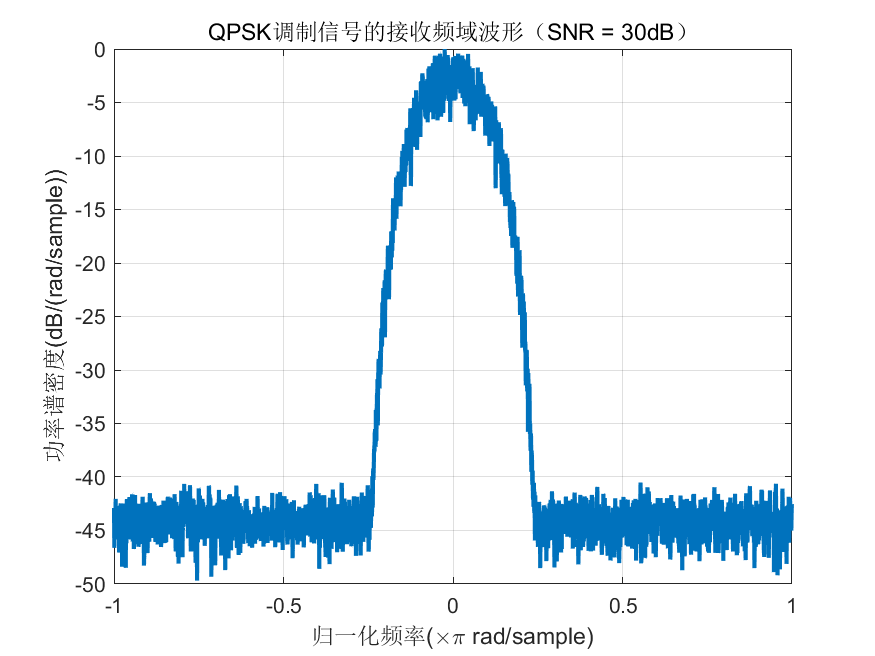


3.2 比较不同信噪比下的QPSK眼图

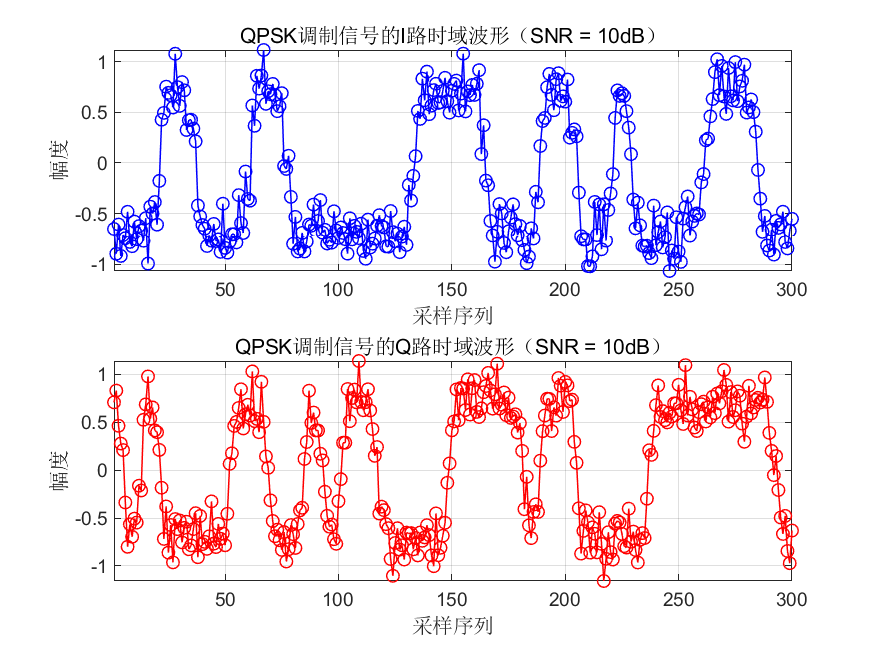


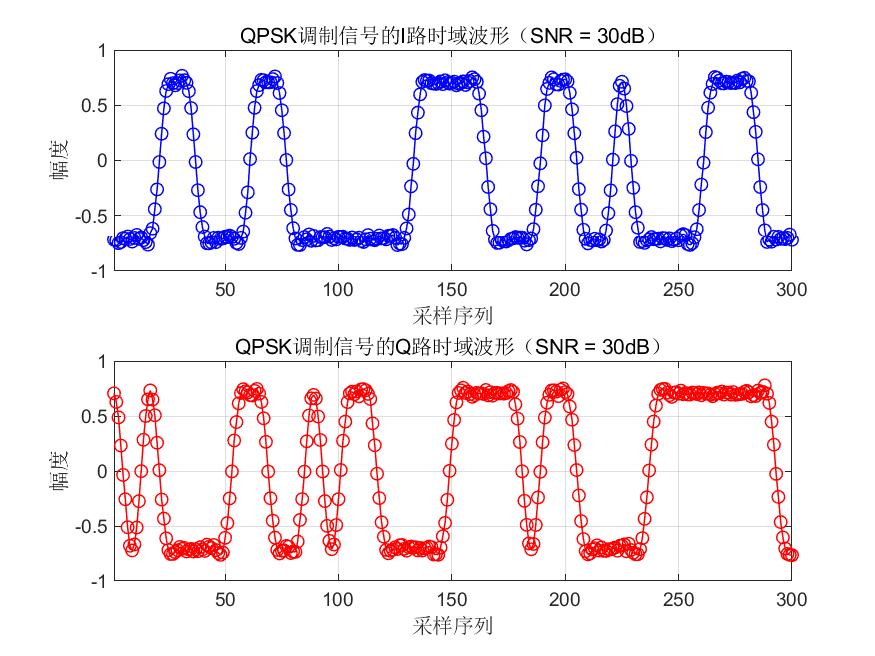
3.3 比较不同信噪比下的QPSK功率谱



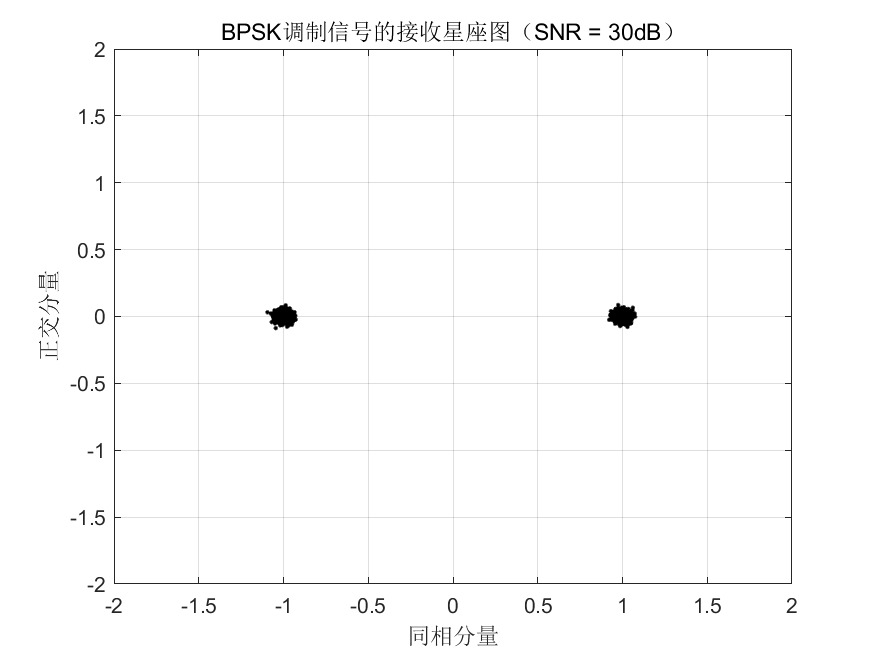


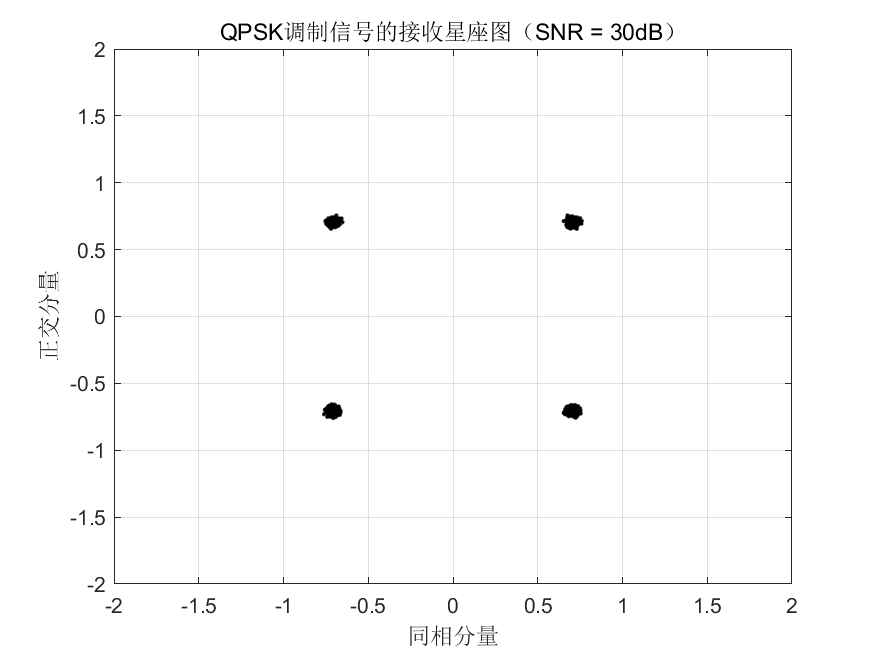
3.4 比较不同信噪比下的QPSK波形

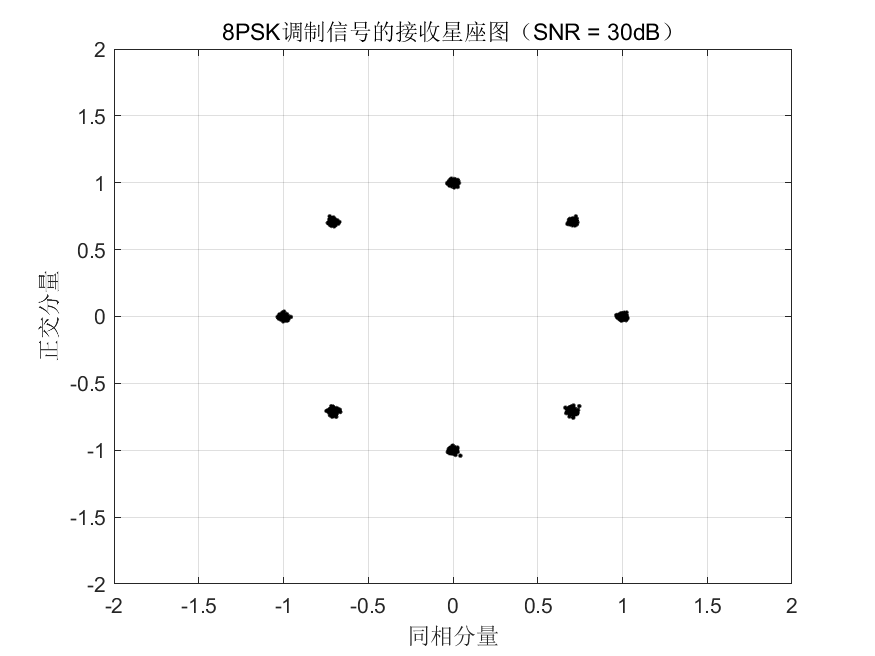


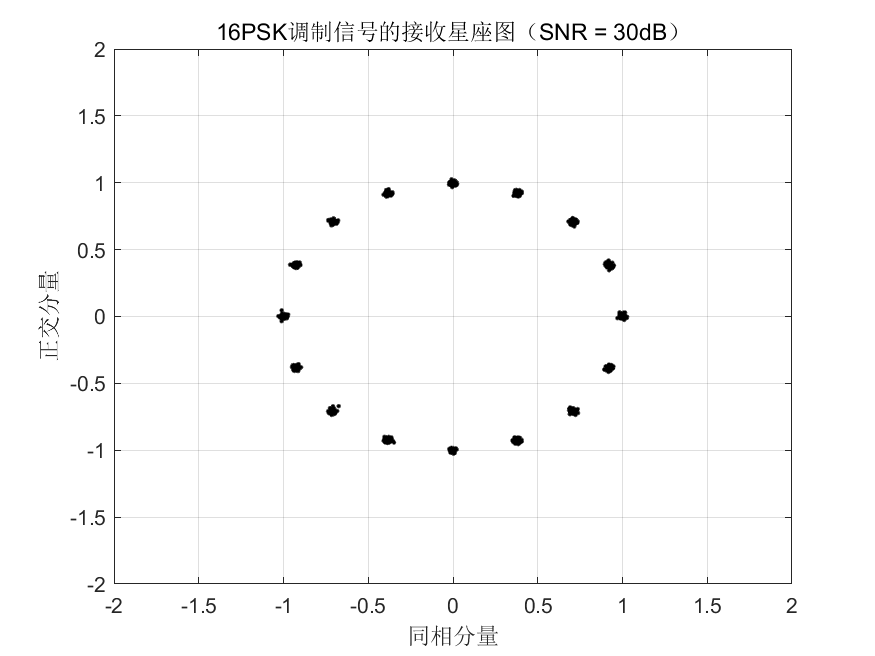


3.5 比较不同进制PSK调制的星座图（在相同的信噪比30dB下比较）

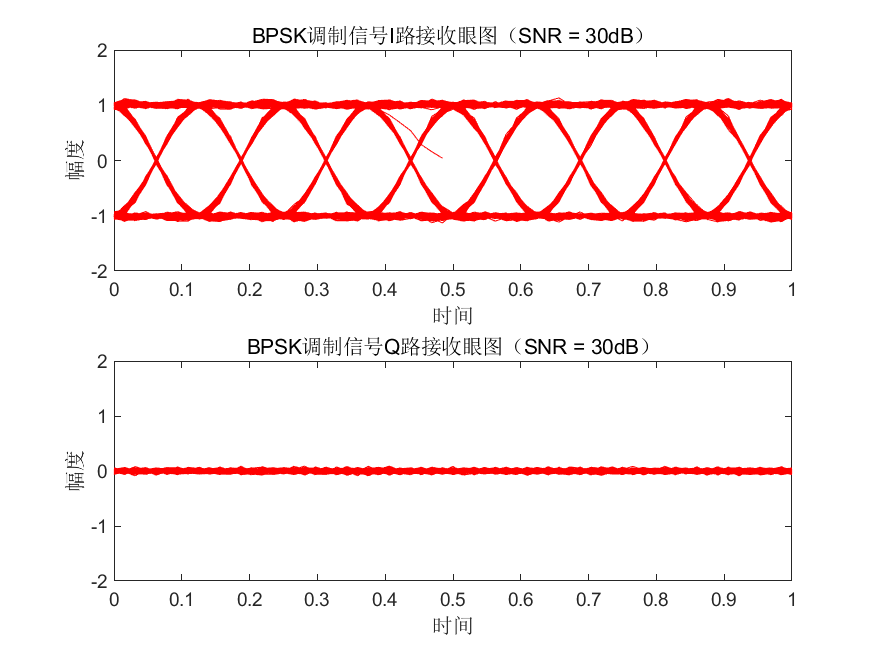


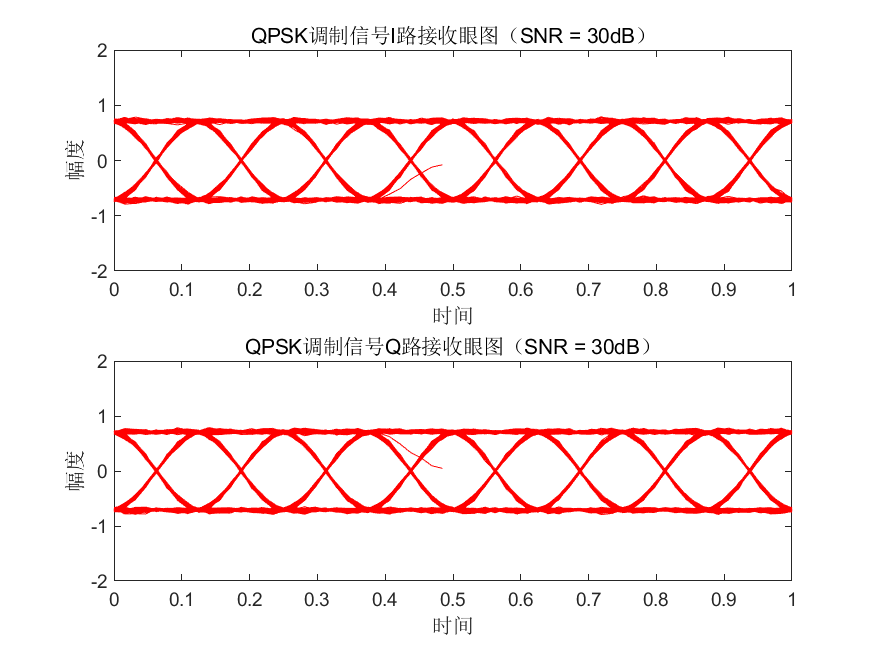


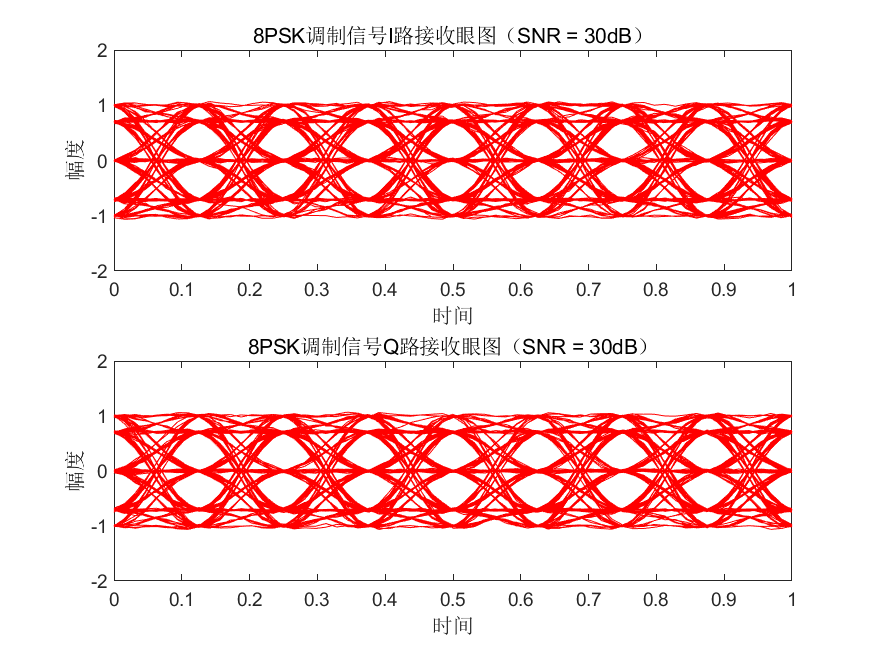


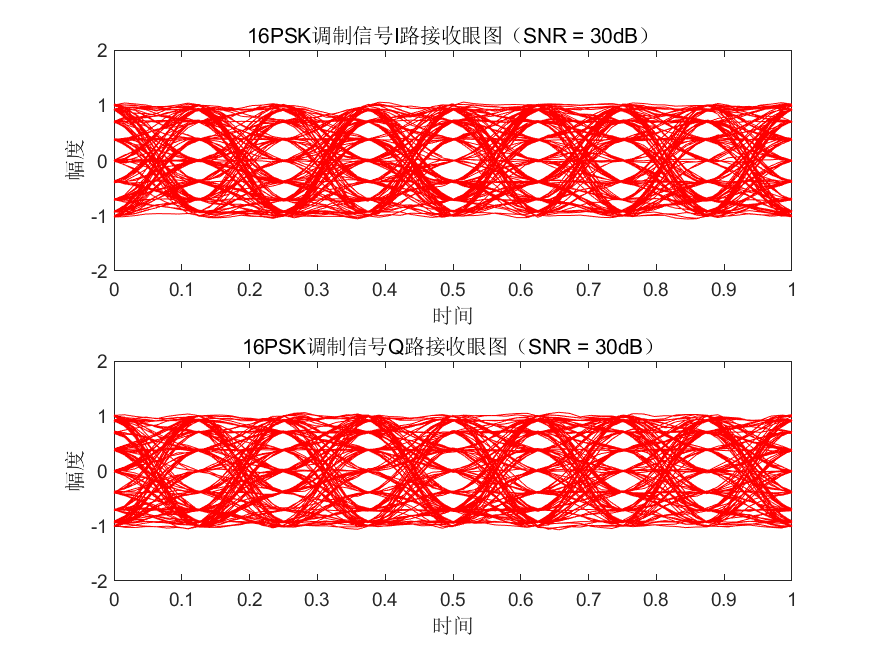


3.6 比较不同进制PSK调制的眼图（在相同的信噪比20dB下比较）

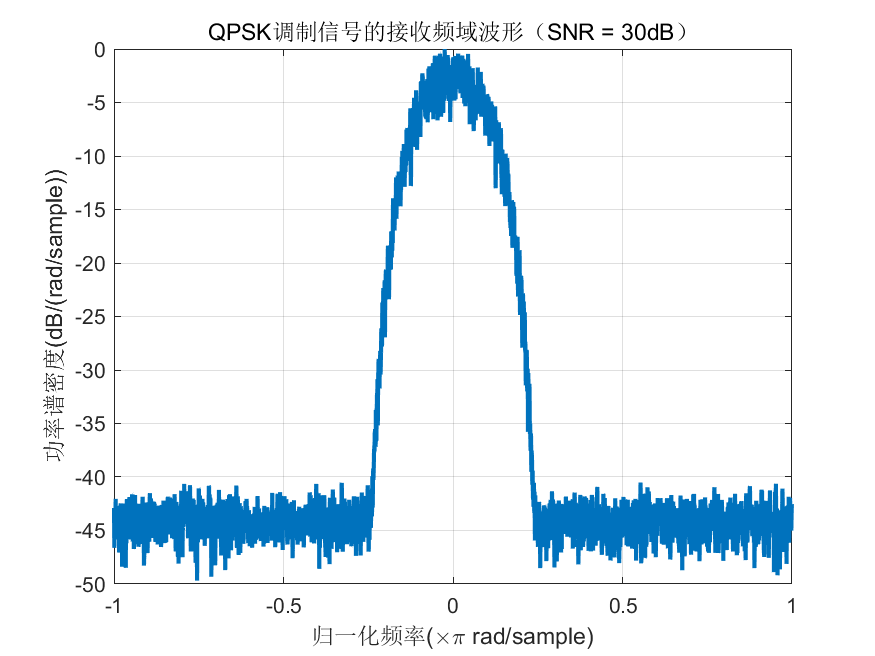
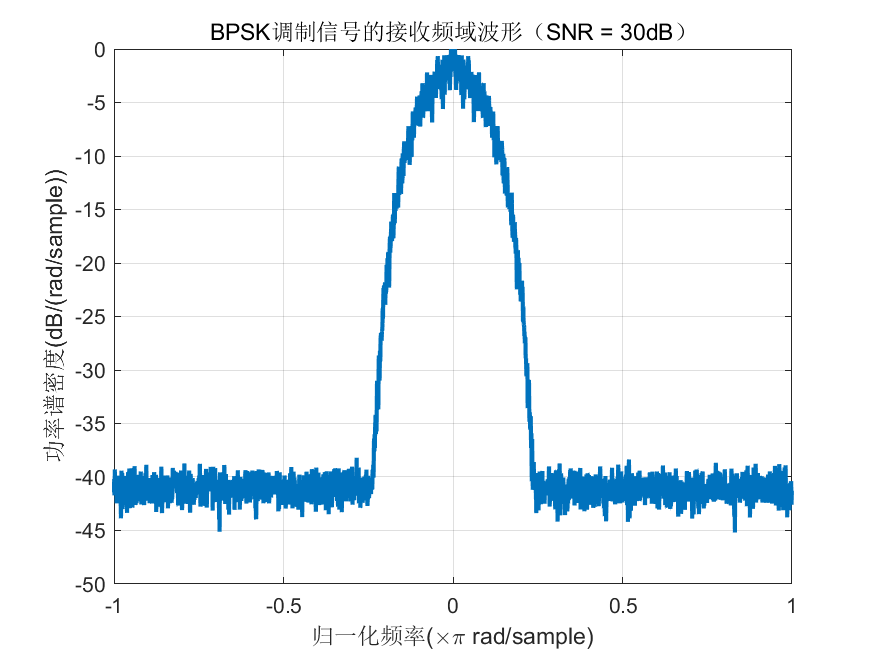


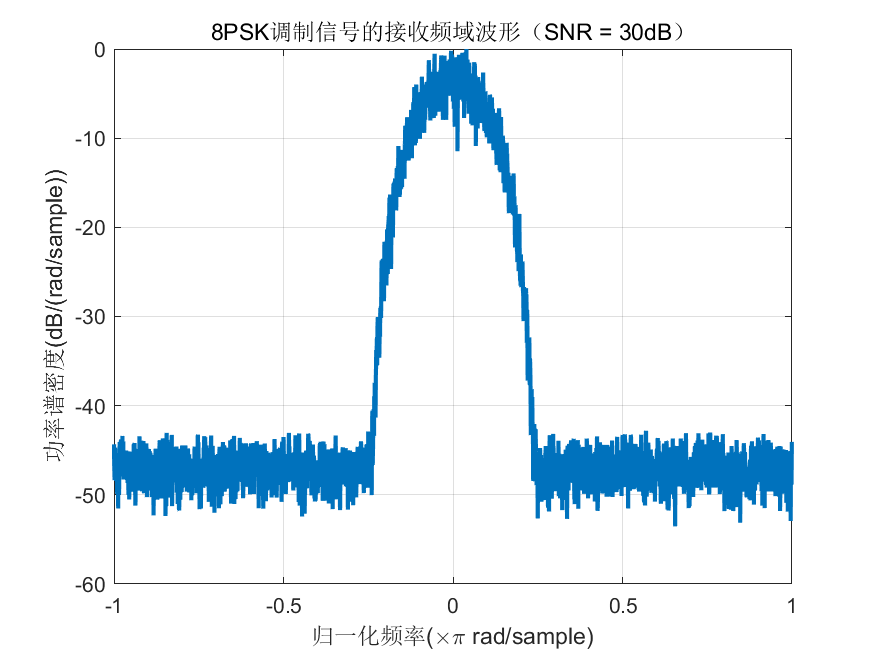


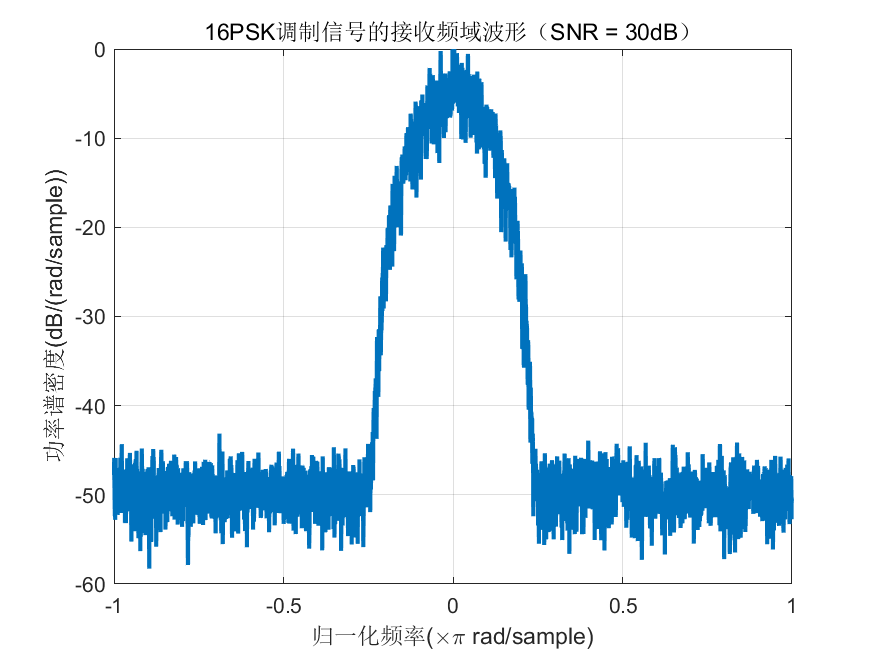




3.7 比较不同进制PSK调制的功率谱（在相同的信噪比30dB下比较）

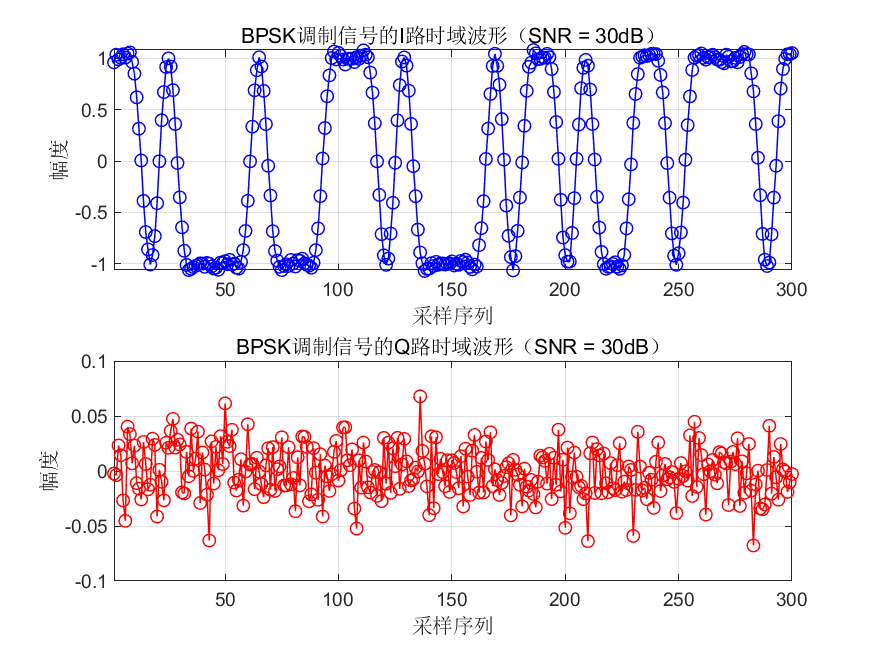


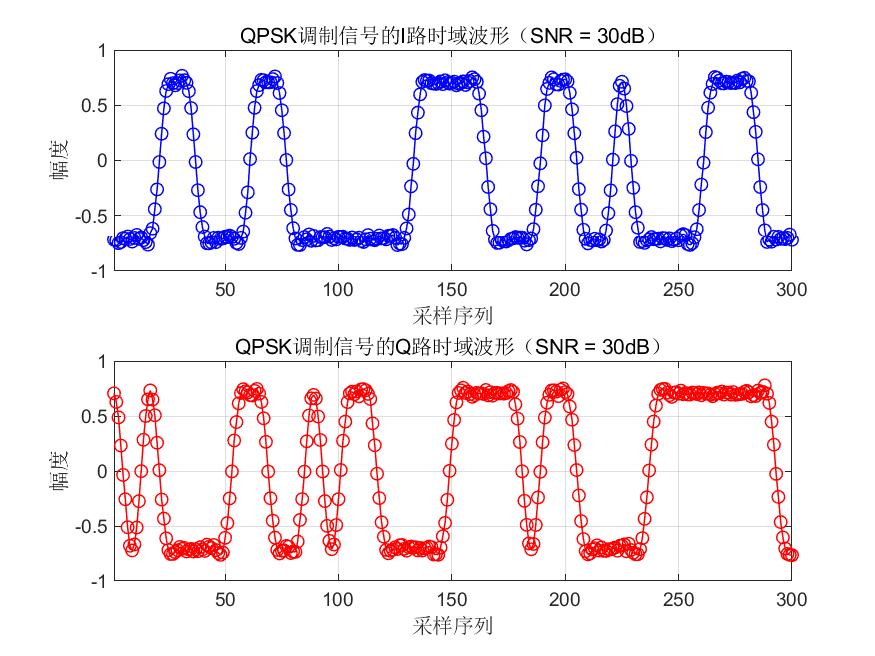


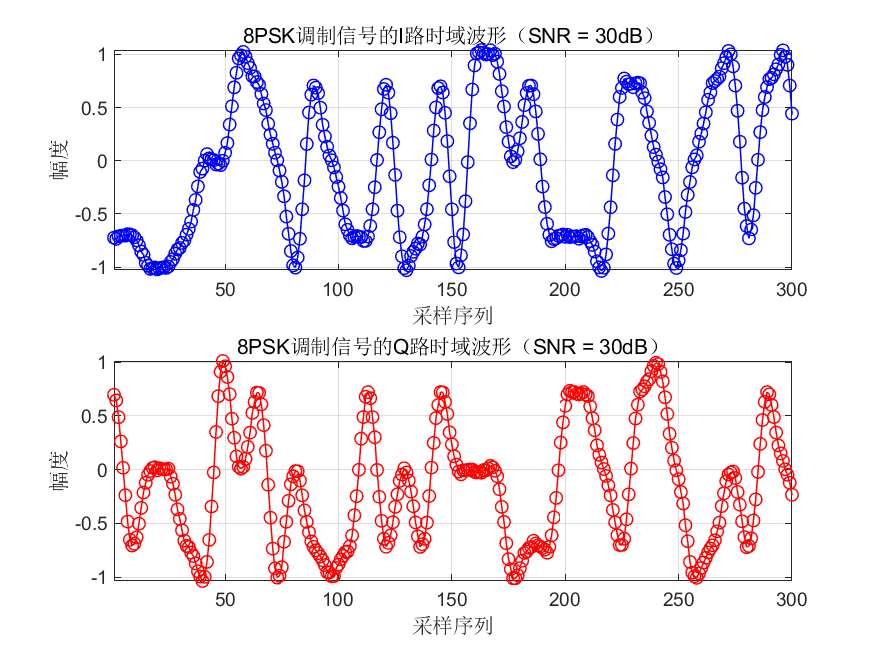


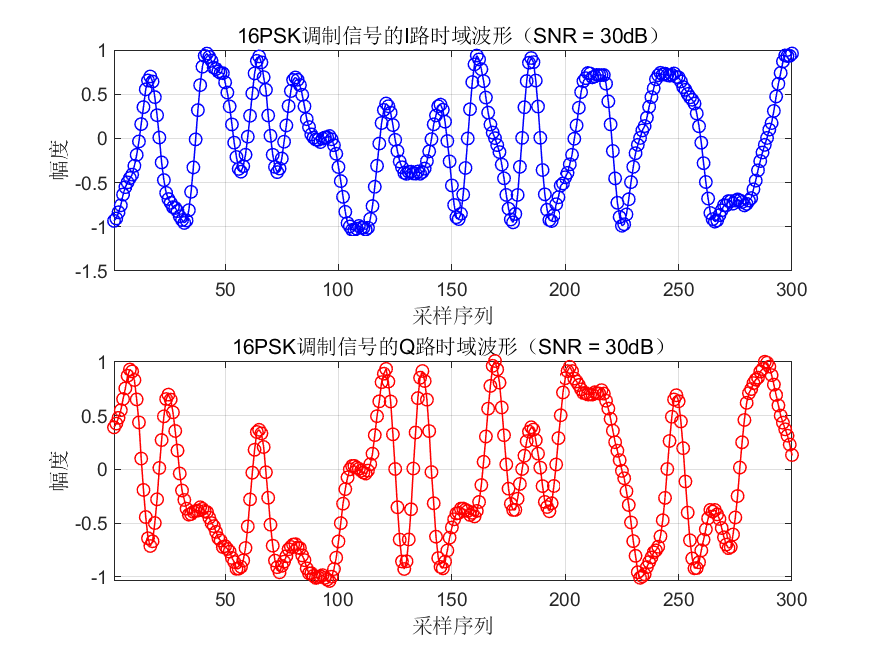
在相同的符号宽度上，不同的调制方式所占用的带宽是相同的。

3.7 比较不同进制PSK调制的时域波形（在相同的信噪比30dB下比较）



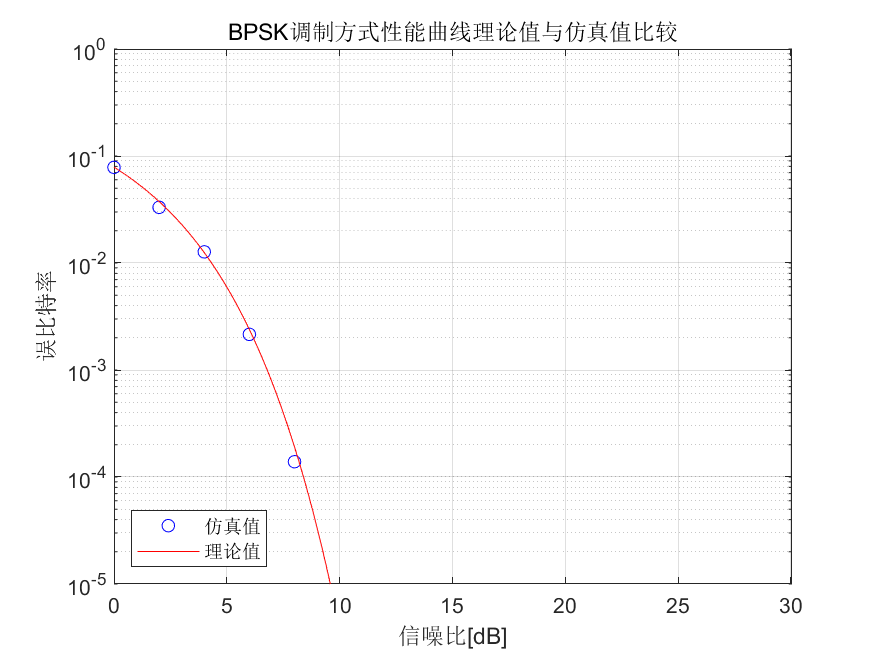


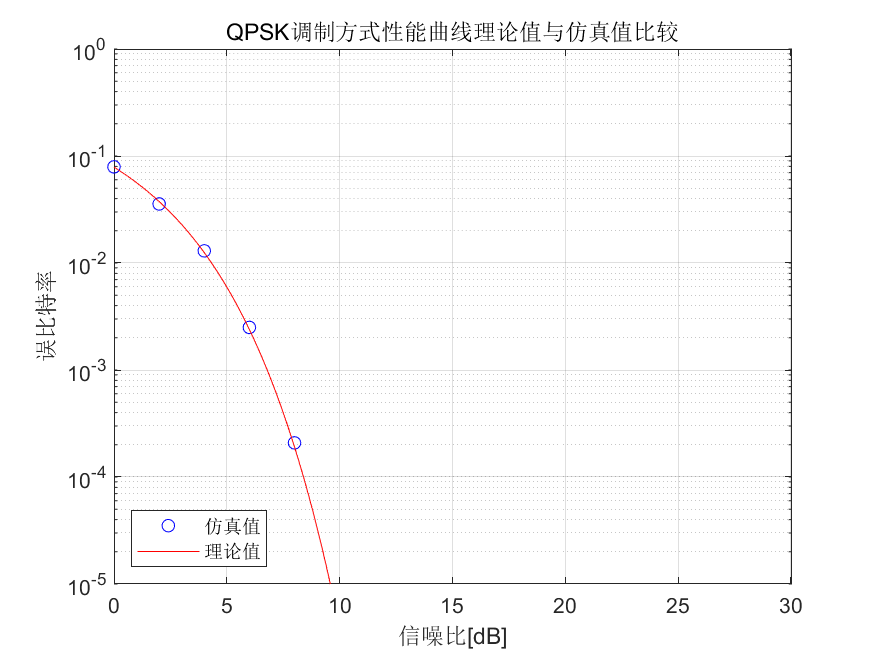


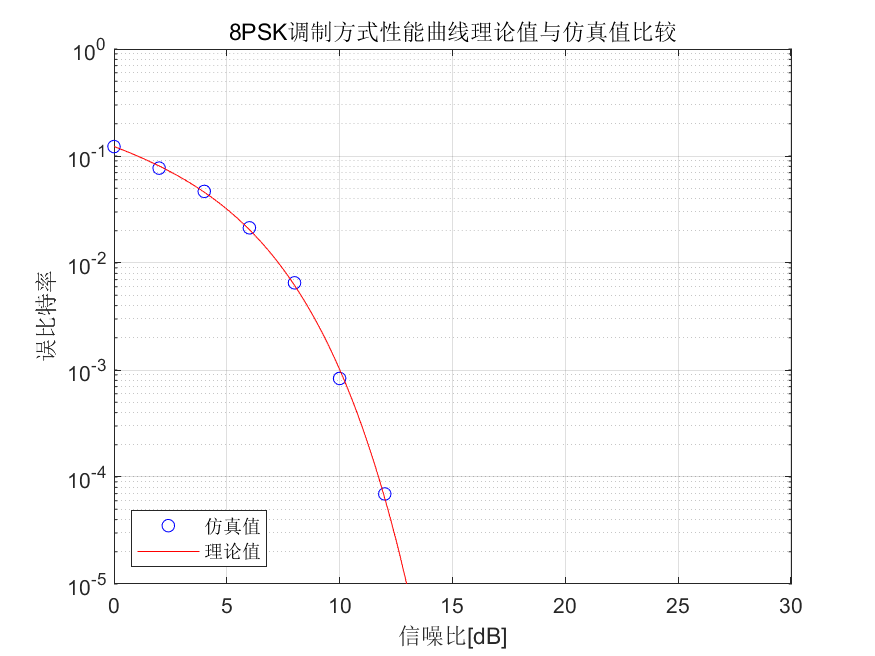


进制数目越大，抖动越大。

3.5 比较不同进制PSK调制的误比特率









观察仿真图可以发现，不同进制的PSK调制理论值和仿真值完全吻合，验证了仿真的正确性，

其次，随着信噪比的提高，系统的误比特率不断降低，表明提高信道传输质量，可以有效改善系统高性能

再次，BPSK和QPSK调制的误比特率完全一致，符合理论预期。

最后，在相同信噪比下，进制数越高，得到的误比特率越大，而在相同的误比特率下，进制数越高，所需要的信噪比就越大，进制数越高，频带效率越大，表明，功率效率和频带效率就相互制约的。

代码部分：

clear all;

close all;

clc;

%% 参数设置

errNum = 1e4; %最大仿真差错数

maxNum = 1e4; %最大仿真比特数

SNRdBs = [0:2:20,30]; %信噪比

lineMarker = {'BPSK调制','QPSK调制','8PSK调制'};

Ms = [2 4 8];

for id\_M = 1:length(Ms); %调制阶数

close all;

M = Ms(id\_M);

linStr = lineMarker{id\_M};

dataSrc = randi([0 1],1,4800); %发送信源

if M == 2

phaIni = 0;

else

phaIni = pi/4;

end

%% QPSK 误比特率计算

pskModu = comm.PSKModulator(M,'BitInput',true,'PhaseOffset',phaIni,'SymbolMapping','Gray'); %QPSK调制

awgnChan = comm.AWGNChannel; %AWGN传输信道

pskDemodu = comm.PSKDemodulator( 'ModulationOrder', M, 'BitOutput', true, ...

'PhaseOffset', phaIni,'SymbolMapping','Gray'); %QPSK解调

berCalc = comm.ErrorRate;%误比特率计算

ber\_psk = zeros(size(SNRdBs));%变量初始化

for id\_SNR = 1:length(SNRdBs)

k = log2(M);

SNR = SNRdBs(id\_SNR) + 10\*log10(k);%信噪比

awgnChan.EbNo = SNR;

numErrs = 0;

numBits = 0;

berOut=zeros(3,1);

while ((numErrs < errNum) && (numBits < maxNum))

% 发端

moduSym = step(pskModu, dataSrc'); %PSK Modulator

% 成型滤波

dataRat = 1;

sampRat = 8;

Rf = 1;

Dl = 8;

rrcFil = rcosine(dataRat,sampRat,'fir',Rf,Dl); %设计升余弦滤波器

shapedSig = rcosflt(moduSym,dataRat,sampRat,'filter',rrcFil);

shapedSig\_ = shapedSig(sampRat\*Dl+1:end-sampRat\*Dl);

% Channel

noisySig = step(awgnChan, shapedSig\_); % AWGN channel

% Receiver

demoduData = step(pskDemodu, noisySig(1:sampRat/dataRat:end)); %PSK Demodulator

berOut = step(berCalc, dataSrc', demoduData); % Update BER

numErrs = berOut(2);

numBits = berOut(3);

end

if mod(SNRdBs(id\_SNR),10) == 0

waveplot(noisySig,sampRat,dataRat,SNRdBs,id\_SNR,linStr)

end

% Compute BER

ber = berOut(1);

reset(berCalc);

ber\_psk(id\_SNR) = ber;%误比特率保存

end

figure(1)

SNRdBt = min(SNRdBs):0.1:20;

semilogy(SNRdBs,ber\_psk,'b o');hold on;

BERt = berawgn(SNRdBt,'psk',M,'nodiff');%理论值

semilogy(SNRdBt,BERt,'r');hold on; %仿真值

grid on;

axis([0 max(SNRdBs)+0.05 10^(-5) 10^(0)]);

xlabel('信噪比[dB]');ylabel('误比特率')

legend( '仿真值','理论值', 'location','southwest')

title(strcat(linStr,'方式性能曲线理论值与仿真值比较'))

saveas(gcf,strcat(linStr,'方式性能曲线理论值与仿真值比较.png'))

end