# 通信原理第三次实验

## 一、实验目的

1. 掌握BPSK调制解调原理；

2. 掌握QPSK调制解调原理；

3. 掌握QAM调制解调原理；

4. 理解噪声对BPSK，QPSK，QAM相干解调性能的影响；

## 二、实验内容

1. 用MATLAB产生256bit独立等概率的二进制信源，码元持续时间为1s，仿真信号在AWGN信道中的BPSK调制解调过程，其中载波幅值为1，频率为20Hz，判决门限为0。

（1）当解调器输入信噪比为0dB时，分别画出调制信号、BPSK已调信号、低通滤波器输出信号和解调信号；

（2）画出解调器输入信噪比在-15dB到5dB时的接收机理论误码率曲线与实际误码率曲线(20次以上平均)；

2. 用MATLAB产生256bit独立等概率的二进制信源，码元持续时间为1s，仿真信号在AWGN信道中的QPSK调制解调过程，其中载波幅值为1，频率为20Hz，判决门限为0。

（1）当解调器输入信噪比为0dB时，分别画出调制信号、QPSK已调信号、低通滤波器输出信号和解调信号；

（2）画出解调器输入信噪比在-15dB到5dB时的接收机理论误码率曲线与实际误码率曲线(20次以上平均)；

3. 用MATLAB产生256bit独立等概率的二进制信源，码元持续时间为1s，仿真信号在AWGN信道中的16QAM调制解调过程，其中载波幅值为1或2，频率为20Hz，判决门限为0。

（1）当解调器输入信噪比为0dB时，分别画出调制信号、16QAM星座图、16QAM已调信号、低通滤波器输出信号和解调信号；

（2）画出解调器输入信噪比在-15dB到5dB时的接收机理论误码率曲线与实际误码率曲线(20次以上平均)；

## 三、实验程序

**用到的低通滤波器 lowpass**

function [ x1 ] = lowpass( fc,fs,signal )

%LOWPASS 低通滤波器，采用巴特沃斯滤波器

% fc =是截至频率， fs=是采样频率，sigal 是时域信号

[b,a]=butter(3,fc/(fs/2),'low'); % 四阶的巴特沃斯低通滤波

x1=filter(b,a,signal);

end

**用到的带通滤波器 bandpass**

function [x1] = bandpass( fl,fh,fs,signal )

%LOWPASS 低通滤波器，采用巴特沃斯滤波器

% fl =是带通滤波器的低频， fh是带通滤波器的高频，fs是采样频率 sigal 是时域信号 ，输出 x1也为时域信号

[b,a]=butter(10,[fl,fh]/(fs/2),'bandpass'); % 四阶的巴特沃斯低通滤波

x1=filter(b,a,signal);

end

**实验一（1）**

**clear all;**

**%--------------------------**

**%产生256 bit的数据**

**bitlist(256)=0;**

**for i=1 :256**

**x=rand(1);**

**bit=0;**

**if x>0.5**

**bit=1;**

**else**

**bit=-1;**

**end**

**bitlist(i)=bit;**

**end**

**%-------------------------------------**

**%生成256bit的数字信号，采样频率 1/dt**

**dt=0.001; %参数可以调，调细一点，取值更加合理，跑的更慢**

**t=0:dt:256-dt;**

**xt=zeros(1,length(t));**

**%对于每1秒中的取值赋值**

**for i=1 :256**

**if bitlist(i)==1**

**xt(1+(i-1)\*(1/dt):i\*(1/dt) )=1;**

**else**

**xt(1+(i-1)\*(1/dt):i\*(1/dt) )=-1;**

**end**

**end**

**subplot(2,2,1);**

**plot(t,xt);**

**title('调制信号');**

**xlabel('t/s');**

**ylabel('st/V');**

**axis([0,20,-1.2,1.2]);**

**%--------------------**

**%对载波进行调制**

**carrier=cos(2\*pi\*20\*t); %频率20**

**%调制之后**

**st=xt.\*carrier;**

**subplot(2,2,2);**

**plot(t,st);**

**axis([0,5,-1.2,1.2]);**

**title('BPSK已调信号');**

**xlabel('t/s');**

**ylabel('st/V');**

**%--------------------------**

**%模拟AWGN信道，并且通过带通滤波器**

**%信噪比为0，A=1 噪声功率是1/2**

**P=1/2;**

**st=st+sqrt(P\*(1/dt/2/4)).\*randn(1,length(t)); %模拟信道的噪声 ， fs/2=50 ， B=4 所以功率是50/4**

**[f,sf]=T2F(t,st);**

**[f,sf]=ideal\_bandpassing(f,sf,18,22); %通过带通滤波器**

**[t,st]=F2T(f,sf);**

**%----------------**

**%进行解调，通过低通滤波器， 因为数字基带信号的频率是 0-fB fB设为2**

**st=st.\*carrier;**

**[f,sf]=T2F(t,st);**

**[f,sf]=ideal\_bandpassing(f,sf,0,2);**

**[t,st]=F2T(f,sf);**

**subplot(2,2,3);**

**plot(t,st);**

**axis([0,20,-2,2]);**

**title('低通滤波器的输出');**

**xlabel('t/s');**

**ylabel('st/V');**

**%---------------------------------**

**%定时抽样得到比特序列，抽样的点是时间点的中点**

**newblist(256)=0;**

**for i =1:256**

**if st(1/dt\*1/2+(1/dt)\*(i-1))>0**

**newblist(i)=1;**

**else**

**newblist(i)=-1; %可以是0，也可以是1**

**end**

**end**

**%------------------------------**

**%通过比特序列，生成解调的数字信号**

**xt\_after=zeros(1,length(t));**

**for i=1 :256**

**if newblist(i)==1**

**xt\_after(1+(i-1)\*(1/dt):i\*(1/dt) )=1;**

**else**

**xt\_after(1+(i-1)\*(1/dt):i\*(1/dt) )=-1;**

**end**

**end**

**subplot(2,2,4);**

**plot(t,xt);**

**title('解调信号');**

**axis([0,20,-1.2,1.2]);**

**xlabel('t/s');**

**ylabel('st/V');**

**%---------------------------**

**实验一（2）**

**clear all;**

**r\_db=-15:0.1:5;**

**r\_pow=db2pow(r\_db);**

**%---------------------------**

**%通过公式计算误码率， 得到理论误码曲线**

**Pe\_pow=0.5\*erfc(sqrt(r\_pow));**

**Pe\_db=pow2db(Pe\_pow);**

**subplot(2,1,1);**

**plot(r\_db,Pe\_db);**

**xlabel('r/db');**

**ylabel('Pe/db');**

**title('2PSK理想的误码率曲线');**

**%------------------------------------**

**%通过模拟函数计算误码率， 得到理论误码曲线**

**sim\_Pe=zeros(1,length(r\_db));**

**for i=1:length(r\_db)**

**sim\_Pe(i)=sim\_q1(r\_db(i)); %输入r/db ,得到对应信噪比的误码率**

**end**

**subplot(2,1,2);**

**plot(r\_db,sim\_Pe);**

**xlabel('r/db');**

**ylabel('Pe/db');**

**title('2PSK仿真的误码率曲线');**

**axis([-15,5,-30,0]);**

**%-------------------------------**

# Sim\_q1函数

function [ Pe] = sim\_q1(r)  
%SIM\_Q1 输入信噪比r/db，得到一次2PSK的误码率,Pe/db  
% 此处显示详细说明  
%产生256 bit的数据  
sim\_times=20; %模拟次数  
allwrongbit=0; %总错误bit  
bitlen=256;  
r=db2pow(r); %输入  
for i1=1:sim\_times  
 %-------------------  
 %生成256比特序列  
 bitlist(bitlen)=0;  
 for i=1 :bitlen  
 x=rand(1);  
 bit=0;  
 if x>0.5  
 bit=1;  
 else  
 bit=-1;  
 end  
 bitlist(i)=bit;  
 end  
  
 %---------------------------  
 %生成数字信号  
  
 dt=0.01; %参数可以调，调细一点，取值更加合理，跑的更慢  
 t=0:dt:bitlen-dt;  
 xt=zeros(1,length(t));  
 %对于每1秒中的取值赋值  
 for i=1 :bitlen  
 if bitlist(i)==1  
 xt(1+(i-1)\*(1/dt):i\*(1/dt) )=1;  
 else  
 xt(1+(i-1)\*(1/dt):i\*(1/dt) )=-1;  
 end  
 end  
 %--------------------------------  
 %生成载波并调制载波  
 carrier=cos(2\*pi\*20\*t);  
 st=xt.\*carrier;  
  
 %信噪比为0，A=1 噪声功率是1/2  
  
 %------------  
 %模拟AWGN信号，加上高斯白噪声，通过带通滤波器  
 P=0.5/r;  
 st=st+sqrt(P\*(50/4)).\*randn(1,length(t)); %模拟信道的噪声 ， fs/2=50 ， B=4 所以功率是50/4  
 st=bandpass(18,22,1/dt,st);  
 %------------------------------  
 %用载波进行解调，并通过低通滤波器，  
 st=st.\*carrier;  
 st=lowpass(2,1/dt,st);  
 %----------------------------  
 %定时抽样  
 newblist(bitlen)=0;  
 for i =1:bitlen  
 if st((1/dt)\*(i)-(1/5)\*(1/dt))>0  
 newblist(i)=1;  
 else  
 newblist(i)=-1; %可以是0，也可以是1  
 end  
 end  
 %------------------------------  
 %计算错误比特数  
 wrongbit=0;  
 for i=1:bitlen  
 if bitlist(i)~=newblist(i)  
 wrongbit=wrongbit+1;  
 end  
 end  
 allwrongbit=allwrongbit+wrongbit;  
end  
%---------------------------------  
%计算总误码率  
Pe=allwrongbit/(bitlen\*sim\_times);  
Pe=pow2db(Pe);  
end

# 实验二（1）

clear all;  
%------------------------------  
%产生256 bit的数据  
bitlist(256)=0;  
for i=1 :256  
 x=rand(1);  
 bit=0;  
 if x>0.5  
 bit=1;  
 else  
 bit=-1;  
 end  
 bitlist(i)=bit;  
end  
%--------------------------------------  
%生成数字信号，持续周期1s  
dt=0.001; %参数可以调，调细一点，取值更加合理，跑的更慢  
t=0:dt:128-dt; %128个码元的时间  
t1=0:dt:256-dt; %256个比特的时间  
xt=zeros(1,length(t1));  
for i=1 :256  
 if bitlist(i)==1  
 xt(1+(i-1)\*(1/dt):i\*(1/dt) )=1;  
 else  
 xt(1+(i-1)\*(1/dt):i\*(1/dt) )=0;  
 end  
end  
subplot(3,2,1);  
plot(t1,xt);  
title('调制信号(256bit串)');  
axis([0,20,-0.5,1.2]);  
grid;  
  
%-----------------------------  
%对于每1秒中的取值赋值  
st=zeros(1,length(t));  
for i=1 :128  
 %串并转换  
 a=bitlist(2\*i-1);  
 b=bitlist(2\*i);  
 %QPSK调制  
 %1+(i-1)\*(1/dt):i\*(1/dt) 就是 i-1 到 i时间切片的索引，比如i=1 就是0-1时间的索引  
 st(1+(i-1)\*(1/dt):i\*(1/dt) )=a\*cos(2\*pi\*20\*t(1+(i-1)\*(1/dt):i\*(1/dt)))-b\*sin(2\*pi\*20\*t(1+(i-1)\*(1/dt):i\*(1/dt)));  
  
end  
subplot(3,2,2);  
plot(t,st);  
title('QPSK调制信号');  
xlabel('t/s');  
ylabel('st/V');  
axis([0,2,-1.5,1.5]);  
  
%------------------------------  
%模拟AWGN信道，添加高斯白噪声，并通过带通滤波器  
%信噪比为0db，A=sqrt(2) 噪声功率是1 (因为信号的功率是 1,信噪比是1，)  
 P=1;  
 st=st+sqrt(P\*(1/dt/2/4)).\*randn(1,length(t)); %模拟信道的噪声的功率由解调器的信号功率推出 ， fs/2=50 ， B=4 所以功率是50/4  
 [f,sf]=T2F(t,st);  
 [f,sf]=ideal\_bandpassing(f,sf,18,22);  
 [t,st]=F2T(f,sf);  
%----------------------------------------  
%用两个载波进行解调，分为上下之路  
st\_up=st.\*cos(2\*pi\*20\*t);  
st\_low=st.\*-sin(2\*pi\*20\*t);  
[f,sf\_up]=T2F(t,st\_up);  
[f,sf\_low]=T2F(t,st\_low);  
%-------------------------------------  
%低通滤波器， 因为数字基带信号的频率是 0-fB  
[f,sf\_up]=ideal\_bandpassing(f,sf\_up,0,2);  
[f,sf\_low]=ideal\_bandpassing(f,sf\_low,0,2);  
[t,st\_up]=F2T(f,sf\_up);  
[t,st\_low]=F2T(f,sf\_low);  
  
subplot(3,2,3);  
plot(t,st\_up);  
axis([0,10,-2,2]);  
title('QPSK上支路低通滤波器的输出');  
xlabel('t/s');  
ylabel('st/V');  
  
subplot(3,2,4);  
plot(t,st\_low);  
axis([0,10,-2,2]);  
title('QPSK下支路低通滤波器的输出');  
xlabel('t/s');  
ylabel('st/V');  
%------------------------  
%对每一个支路进行抽样判决  
newblist(256)=0;  
for i =1:128  
 if st\_up(1/dt\*1/2+(1/dt)\*(i-1))>0  
 a=1;  
 else  
 a=-1;  
 end  
 if st\_low(1/dt\*1/2+(1/dt)\*(i-1))>0  
 b=1;  
 else  
 b=-1;  
 end  
 %------  
 %并串转换，生成解调后的比特序列  
 newblist(2\*i-1)=a;  
 newblist(2\*i)=b;  
end  
%-----------------  
%生成解调后的数字信号  
xt\_after=zeros(1,length(t1));  
for i=1 :256  
 if newblist(i)==1  
 xt\_after(1+(i-1)\*(1/dt):i\*(1/dt) )=1;  
 else  
 xt\_after(1+(i-1)\*(1/dt):i\*(1/dt) )=0;  
 end  
end  
subplot(3,2,5);  
plot(t1,xt\_after);  
title('相干解调信号');  
axis([0,20,-0.5,1.2]);  
xlabel('t/s');  
ylabel('st/V');  
grid;

# 实验二（2）

**clear all;**

**r\_db=-15:0.5:5;**

**r\_pow=db2pow(r\_db);**

**%通过公式计算理论的信噪比和误码率的函数**

**Pe\_pow=1-(1-0.5\*erfc(sqrt(r\_pow/2) ) ).^2;**

**Pe\_db=pow2db(Pe\_pow);**

**subplot(2,1,1);**

**plot(r\_db,Pe\_db);**

**xlabel('r/db');**

**ylabel('Pe/db');**

**title('QPSK理想的误码率曲线');**

**%调用 sim\_q2 得到仿真的误码率**

**sim\_Pe=zeros(1,length(r\_db));**

**for i=1:length(r\_db)**

**sim\_Pe(i)=sim\_q2(r\_db(i));**

**end**

**subplot(2,1,2);**

**plot(r\_db,sim\_Pe);**

**xlabel('r/db');**

**ylabel('Pe/db');**

**title('QPSK仿真的误码率曲线');**

**axis([-15,5,-15,0]);**

# sim\_q2函数

function [ Pe] = sim\_q2(r)  
%SIM\_Q2 输入信噪比r/db，得到一次QPSK的误码率,Pe/db  
% 此处显示详细说明  
%产生256 bit的数据  
sim\_times=20; %模拟次数  
allwrongbit=0; %总共模拟的错误比特数  
r=db2pow(r);  
bitlen=256; %比特序列长度  
baudlen=bitlen/2; %码元数  
fs=100; %采样频率  
dt=1/fs; %参数可以调，调细一点，取值更加合理，跑的更慢  
t=0:dt:baudlen-dt; %128个码元的时间  
t1=0:dt:bitlen-dt; %256个比特的时间  
for i1=1:sim\_times  
 %-------------------------------  
 %得到256位比特序列  
 bitlist(bitlen)=0;  
 for i=1 :bitlen  
 x=rand(1);  
 if x>0.5  
 bit=1;  
 else  
 bit=-1;  
 end  
 bitlist(i)=bit;  
 end  
 %----------------------------------  
 %进行QPSK调制  
 st=zeros(1,length(t));  
 for i=1 :baudlen  
 %串并转换  
 a=bitlist(2\*i-1);  
 b=bitlist(2\*i);  
 %1+(i-1)\*(1/dt):i\*(1/dt) 就是 i-1 到 i时间切片的索引，比如i=1 就是0-1时间的索引  
 st(1+(i-1)\*(1/dt):i\*(1/dt) )=a\*cos(2\*pi\*20\*t(1+(i-1)\*(1/dt):i\*(1/dt)))-b\*sin(2\*pi\*20\*t(1+(i-1)\*(1/dt):i\*(1/dt)));  
  
 end  
  
 %模拟AWGN信道，添加高斯白噪声，并通过带通滤波器  
 %信噪比为r db，A=sqrt(2) 噪声功率是1  
 P=1/r;  
 st=st+sqrt(P\*(1/dt/2/4)).\*randn(1,length(t)); %模拟信道的噪声的功率由解调器的信号功率推出 ， fs/2=50 ， B=4 所以功率是50/4  
 st=bandpass(18,22,1/dt,st);  
  
 %----------------------------------------  
 %用两个载波进行解调，分为上下之路  
 st\_up=st.\*cos(2\*pi\*20\*t);  
 st\_low=st.\*-sin(2\*pi\*20\*t);  
 %-------------------------------------  
 %低通滤波器， 因为数字基带信号的频率是 0-fB  
 st\_up=lowpass(2,1/dt,st\_up); %0-2 Hz的低通滤波器  
 st\_low=lowpass(2,1/dt,st\_low);  
  
 %------------------------  
 %对每一个支路进行抽样判决  
 newblist(bitlen)=0;  
 for i =1:baudlen  
 if st\_up((1/dt)\*(i)-(1/5)\*(1/dt) )>0  
 a=1;  
 else  
 a=-1;  
 end  
 if st\_low((1/dt)\*(i)-(1/5)\*(1/dt))>0  
 b=1;  
 else  
 b=-1;  
 end  
 %并串转换  
 newblist(2\*i-1)=a;  
 newblist(2\*i)=b;  
 end  
 %计算一次模拟的错误比特  
 wrongbit=0;  
 for i=1:bitlen  
 if bitlist(i)~=newblist(i)  
 wrongbit=wrongbit+1;  
 end  
 end  
 allwrongbit=allwrongbit+wrongbit;  
end  
  
 %计算总误码率  
Pe=allwrongbit/(bitlen\*sim\_times);  
Pe=pow2db(Pe);  
end

# 实验三（1）

clear all;  
%---------------------  
%产生256 bit的数据  
bitlist(256)=0;  
for i=1 :256  
 x=rand(1);  
 bit=0;  
 if x>0.5  
 bit=1;  
 else  
 bit=0;  
 end  
 bitlist(i)=bit;  
end  
dt=0.001; %参数可以调，调细一点，取值更加合理，跑的更慢  
t=0:dt:64-dt; %64个码元的时间  
t1=0:dt:256-dt; %256个比特的时间  
%-------------------  
%生成256比特数字信号  
xt=zeros(1,length(t1));  
for i=1 :256  
 if bitlist(i)==1  
 xt(1+(i-1)\*(1/dt):i\*(1/dt) )=1;  
 else  
 xt(1+(i-1)\*(1/dt):i\*(1/dt) )=0;  
 end  
end  
subplot(3,2,1);  
plot(t1,xt);  
axis([0,20,-0.5,1.2]);  
title('调制信号(256bit串)');  
grid;  
%---------------------------  
%16QAM调制  
st=zeros(1,length(t));  
for i=1 :64  
 %串并转换  
 a=bitlist(4\*i-3);  
 b=bitlist(4\*i-2);  
 c=bitlist(4\*i-1);  
 d=bitlist(4\*i);  
 ac=a\*2+c;  
 bd=b\*2+d;  
 %ac的二四电平转换，编码为格雷码  
 if ac==2 % 10  
 Ak=2;  
 elseif ac==3 %11  
 Ak=1;  
 elseif ac==1 %01  
 Ak=-1;  
 else %00  
 Ak=-2;  
 end  
 %bd的二四电平转换，编码为格雷码  
 if bd==2 % 10  
 Bk=2;  
 elseif bd==3 %11  
 Bk=1;  
 elseif bd==1 %01  
 Bk=-1;  
 else %00  
 Bk=-2;  
 end  
 %生成16QAM信号  
 st(1+(i-1)\*(1/dt):i\*(1/dt) )=Ak\*cos(2\*pi\*20\*t(1+(i-1)\*(1/dt):i\*(1/dt)))-Bk\*sin(2\*pi\*20\*t(1+(i-1)\*(1/dt):i\*(1/dt)));  
  
end  
subplot(3,2,2);  
plot(t,st);  
title('16QAM调制信号');  
xlabel('t/s');  
ylabel('st/V');  
axis([0,2,-3,3]);  
var(st);  
%---------------------------  
%模拟AWGN信号，添加高斯白噪声，并通过带通滤波器  
%信噪比为0db， ,信号功率 是2.5 ，所以噪声功率是2.5  
 P=2.5;  
 st=st+sqrt(P\*(1/dt/2/4)).\*randn(1,length(t)); %模拟信道的噪声的功率由解调器的信号功率推出 ， fs/2=50 ， B=4 所以功率是50/4  
 [f,sf]=T2F(t,st);  
 [f,sf]=ideal\_bandpassing(f,sf,18,22);  
 [t,st]=F2T(f,sf);  
 %----------------------  
%用两个载波对16QAM进行解调，得到上下支路的信号  
st\_up=st.\*cos(2\*pi\*20\*t)\*2;  
st\_low=st.\*-sin(2\*pi\*20\*t)\*2;  
%-------------------  
%通过低通滤波器，因为数字基带信号的频率是 0-fB  
[f,sf\_up]=T2F(t,st\_up);  
[f,sf\_low]=T2F(t,st\_low);  
[f,sf\_up]=ideal\_bandpassing(f,sf\_up,0,2);  
[f,sf\_low]=ideal\_bandpassing(f,sf\_low,0,2);  
[t,st\_up]=F2T(f,sf\_up);  
[t,st\_low]=F2T(f,sf\_low);  
  
  
subplot(3,2,3);  
plot(t,st\_up);  
xlim([0,64]);  
title('16QAM上支路低通滤波器的输出');  
xlabel('t/s');  
ylabel('st/V');  
  
subplot(3,2,4);  
plot(t,st\_low);  
xlim([0,64]);  
title('16QAM下支路低通滤波器的输出');  
xlabel('t/s');  
ylabel('st/V');  
  
%--------------------------------------  
%对每一个支路进行抽样判决，并进行42电平转换，最后进行串并转换  
newblist(256)=0;  
for i =1:64  
 %抽样判决  
 ak=st\_up(1/dt\*1/2+(1/dt)\*(i-1));  
 bk=st\_low(1/dt\*1/2+(1/dt)\*(i-1));  
 %上支路4，2转换  
 if ak>1.5  
 a=1; c=0;  
 elseif ak>0  
 a=1;c=1;  
 elseif ak>-1.5  
 a=0;c=1;  
 else  
 a=0;c=0;  
 end  
 %下支路4，2转换  
 if bk>1.5  
 b=1; d=0;  
 elseif bk>0  
 b=1;d=1;  
 elseif bk>-1.5  
 b=0;d=1;  
 else  
 b=0;d=0;  
 end  
 %进行并串转换，生成解调后的数字序列  
 newblist(4\*i-3)=a;  
 newblist(4\*i-2)=b;  
 newblist(4\*i-1)=c;  
 newblist(4\*i)=d;  
  
end  
%---------  
%重新生成数字信号  
xt\_after=zeros(1,length(t1));  
for i=1 :256  
 if newblist(i)==1  
 xt\_after(1+(i-1)\*(1/dt):i\*(1/dt) )=1;  
 else  
 xt\_after(1+(i-1)\*(1/dt):i\*(1/dt) )=0;  
 end  
end  
  
subplot(3,2,5);  
plot(t1,xt\_after);  
title('相干解调信号');  
axis([0,20,-0.5,1.2]);  
xlabel('t/s');  
ylabel('st/V');  
grid;

# 实验三（2）

clear all;

r\_db=-15:0.1:15;

r\_pow=db2pow(r\_db);

%通过公式计算信噪比与误码率的关系，得到理论曲线

Pe\_pow=1-(1-0.75\*erfc(sqrt(r\_pow/10) ) ).^2;

Pe\_db=pow2db(Pe\_pow);

subplot(2,1,1);

plot(r\_db,Pe\_db);

xlabel('r/db');

ylabel('Pe/db');

title('16QAM理想的误码率曲线');

%通过仿真结果，生成信噪比和误码率的曲线

sim\_Pe=zeros(1,length(r\_db));

for i=1:length(r\_db)

sim\_Pe(i)=**sim\_q3**(r\_db(i));

end

subplot(2,1,2);

plot(r\_db,sim\_Pe);

xlabel('r/db');

ylabel('Pe/db');

title('16QAM仿真的误码率曲线');

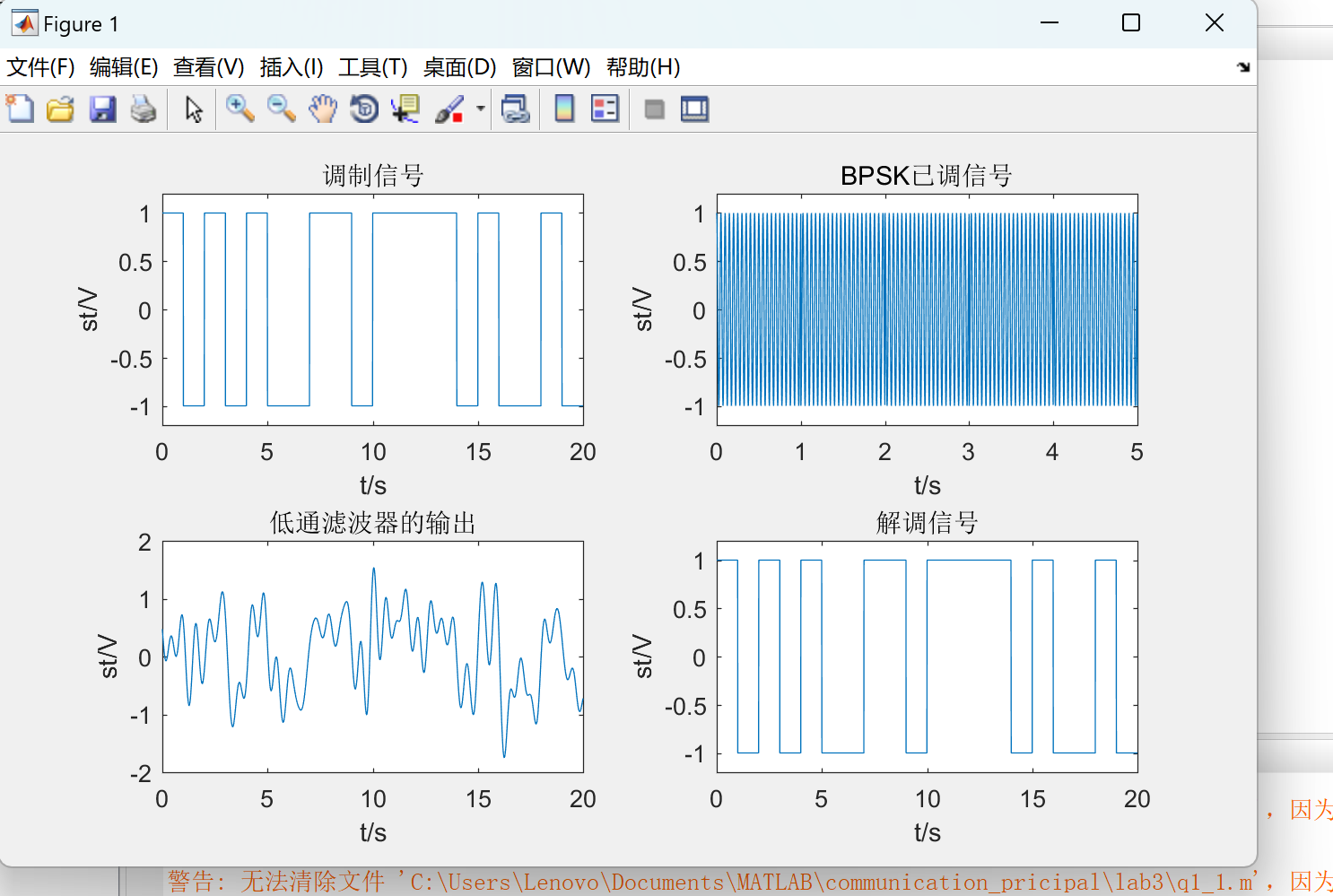
axis([-15,15,-20,0]);

# sim\_q3函数

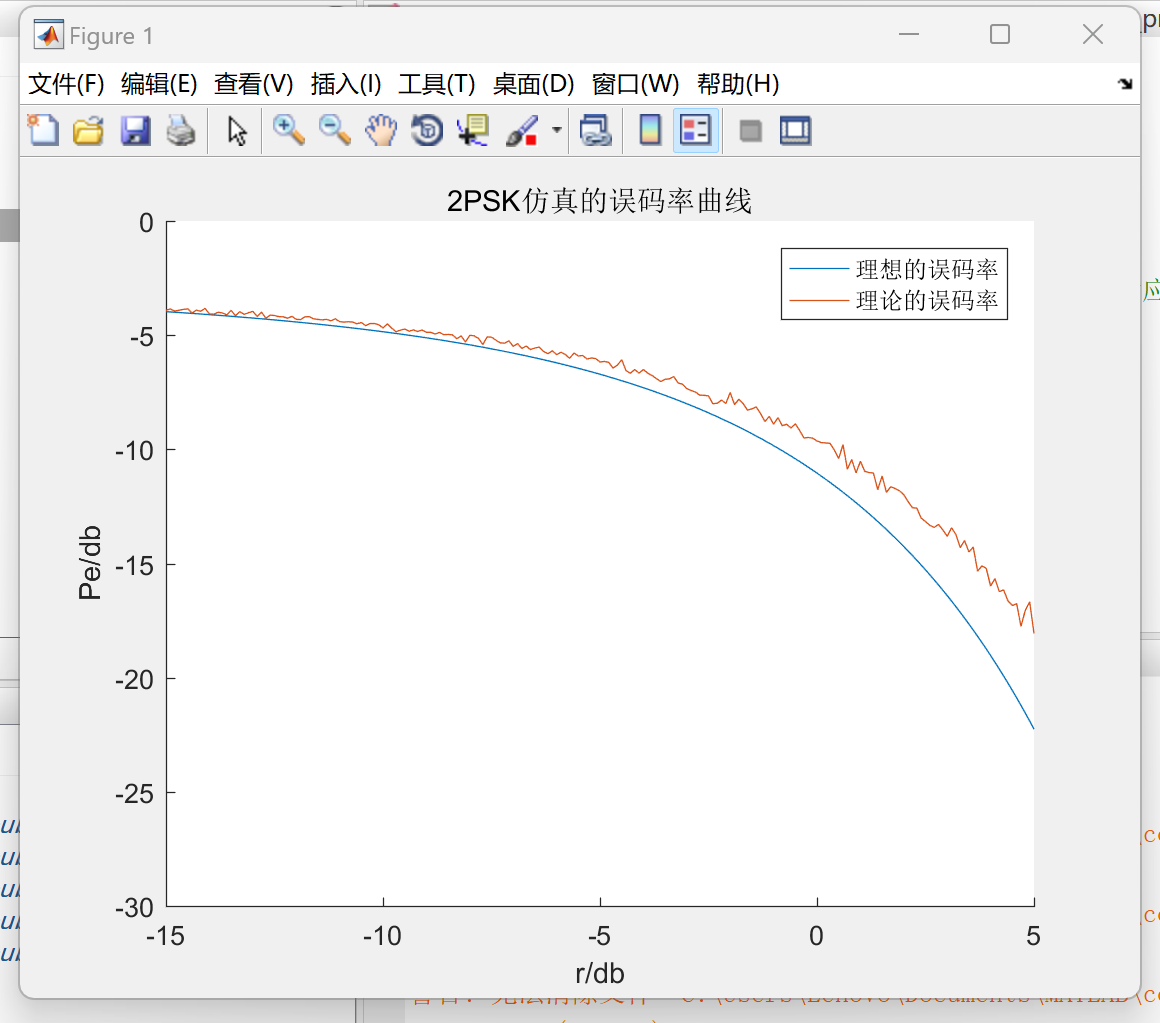
function [ Pe] = sim\_q3(r)  
%SIM\_Q1 输入信噪比r/db，得到一次16QAM的误码率,Pe/db  
% 此处显示详细说明  
%产生256 bit的数据  
sim\_times=100; %模拟次数  
allwrongbit=0; %总错误比特  
r=db2pow(r);  
bitlen=256;  
baudlen=bitlen/4;  
fs=80; %对数字信号的采样频率  
dt=1/fs; %参数可以调，调细一点，取值更加合理，跑的更慢  
t=0:dt:baudlen-dt; %64个码元的时间  
t1=0:dt:bitlen-dt; %256个比特的时间  
for i1=1:sim\_times  
 %---------------------------  
 %得到256比特序列  
 bitlist(bitlen)=0;  
 for i=1 :bitlen  
 x=rand(1);  
 if x>0.5  
 bit=1;  
 else  
 bit=0;  
 end  
 bitlist(i)=bit;  
 end  
 %--------------------------  
 %进行16QAM调制  
 st=zeros(1,length(t));  
 for i=1 :baudlen  
 %串并转换  
 a=bitlist(4\*i-3);  
 b=bitlist(4\*i-2);  
 c=bitlist(4\*i-1);  
 d=bitlist(4\*i);  
 ac=a\*2+c;  
 bd=b\*2+d;  
 %ac的二四电平转换，编码为格雷码  
 if ac==2 % 10  
 Ak=2;  
 elseif ac==3 %11  
 Ak=1;  
 elseif ac==1 %01  
 Ak=-1;  
 else %00  
 Ak=-2;  
 end  
 %bd的二四电平转换，编码为格雷码  
 if bd==2 % 10  
 Bk=2;  
 elseif bd==3 %11  
 Bk=1;  
 elseif bd==1 %01  
 Bk=-1;  
 else %00  
 Bk=-2;  
 end  
 %生成16QAM信号  
 st(1+(i-1)\*(1/dt):i\*(1/dt) )=Ak\*cos(2\*pi\*20\*t(1+(i-1)\*(1/dt):i\*(1/dt)))-Bk\*sin(2\*pi\*20\*t(1+(i-1)\*(1/dt):i\*(1/dt)));  
  
 end  
 %----------------------------------------------  
 %模拟AWGN信号，添加高斯白噪声，并通过带通滤波器  
 %信噪比为r ,信号功率是2.5 ，所以噪声功率是2.5/r  
 P=2.5/r;  
 st=st+sqrt(P\*(1/dt/2/4)).\*randn(1,length(t)); %模拟信道的噪声的功率由解调器的信号功率推出  
 st=bandpass(18,22,1/dt,st);  
 %-------------------------------------  
 %用两个载波对16QAM进行解调，得到上下支路的信号  
 st\_up=st.\*cos(2\*pi\*20\*t)\*2;  
 st\_low=st.\*-sin(2\*pi\*20\*t)\*2;  
 %经过低通滤波器  
 st\_up=lowpass(2,1/dt,st\_up); %0-2 Hz的低通滤波器  
 st\_low=lowpass(2,1/dt,st\_low);  
 %----------------------------  
  
 %--------------------------------------  
 %对每一个支路进行抽样判决，并进行42电平转换，最后进行串并转换  
 newblist(bitlen)=0;  
 for i =1:baudlen  
 %%抽样判决  
 ak=st\_up((1/dt)\*(i));  
 bk=st\_low((1/dt)\*(i));  
 %上支路4，2转换  
 if ak>1.5  
 a=1; c=0;  
 elseif ak>0  
 a=1;c=1;  
 elseif ak>-1.5  
 a=0;c=1;  
 else  
 a=0;c=0;  
 end  
 %下支路4，2转换  
 if bk>1.5  
 b=1; d=0;  
 elseif bk>0  
 b=1;d=1;  
 elseif bk>-1.5  
 b=0;d=1;  
 else  
 b=0;d=0;  
 end  
 % 进行并串转换，生成解调后的数字序列  
 newblist(4\*i-3)=a;  
 newblist(4\*i-2)=b;  
 newblist(4\*i-1)=c;  
 newblist(4\*i)=d;  
 end  
 wrongbit=0;  
 %计算一次模拟的错误比特数  
 for i=1:bitlen  
 if bitlist(i)~=newblist(i)  
 wrongbit=wrongbit+1;  
 end  
 end  
 allwrongbit=allwrongbit+wrongbit;  
end  
  
%计算模拟 sim\_times 次数的平均误码率  
Pe=allwrongbit/(bitlen\*sim\_times);  
Pe=pow2db(Pe);  
end

## 四、实验结果

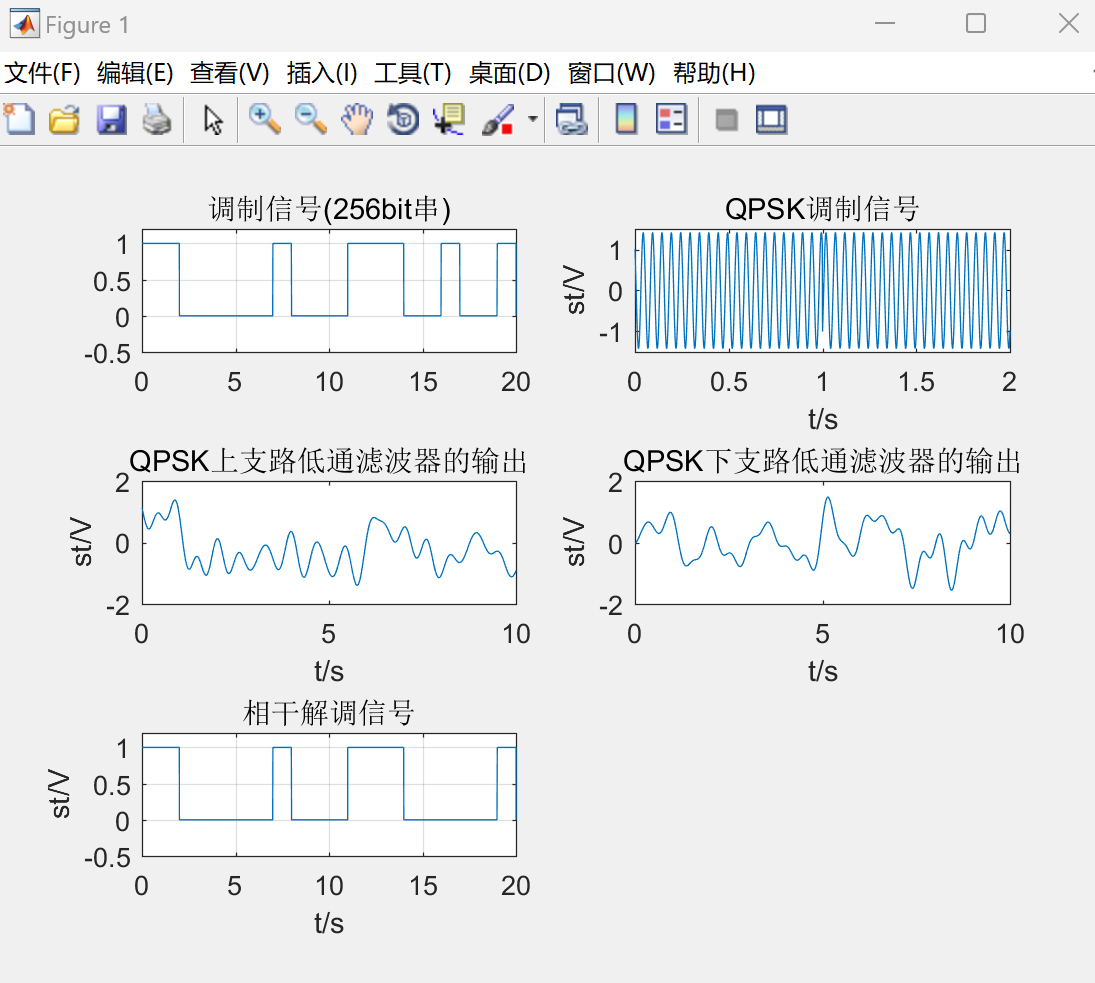
实验一（1）



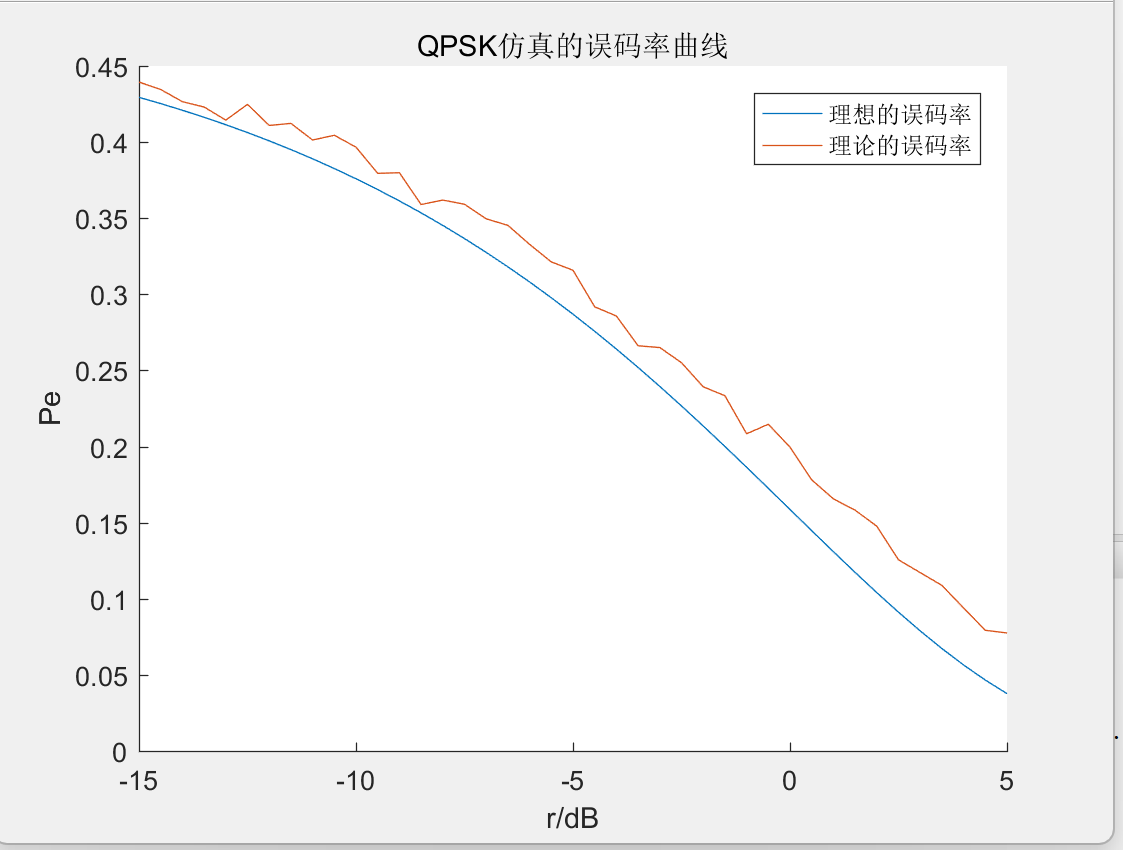
实验一（2） %模拟20次



实验二（1）

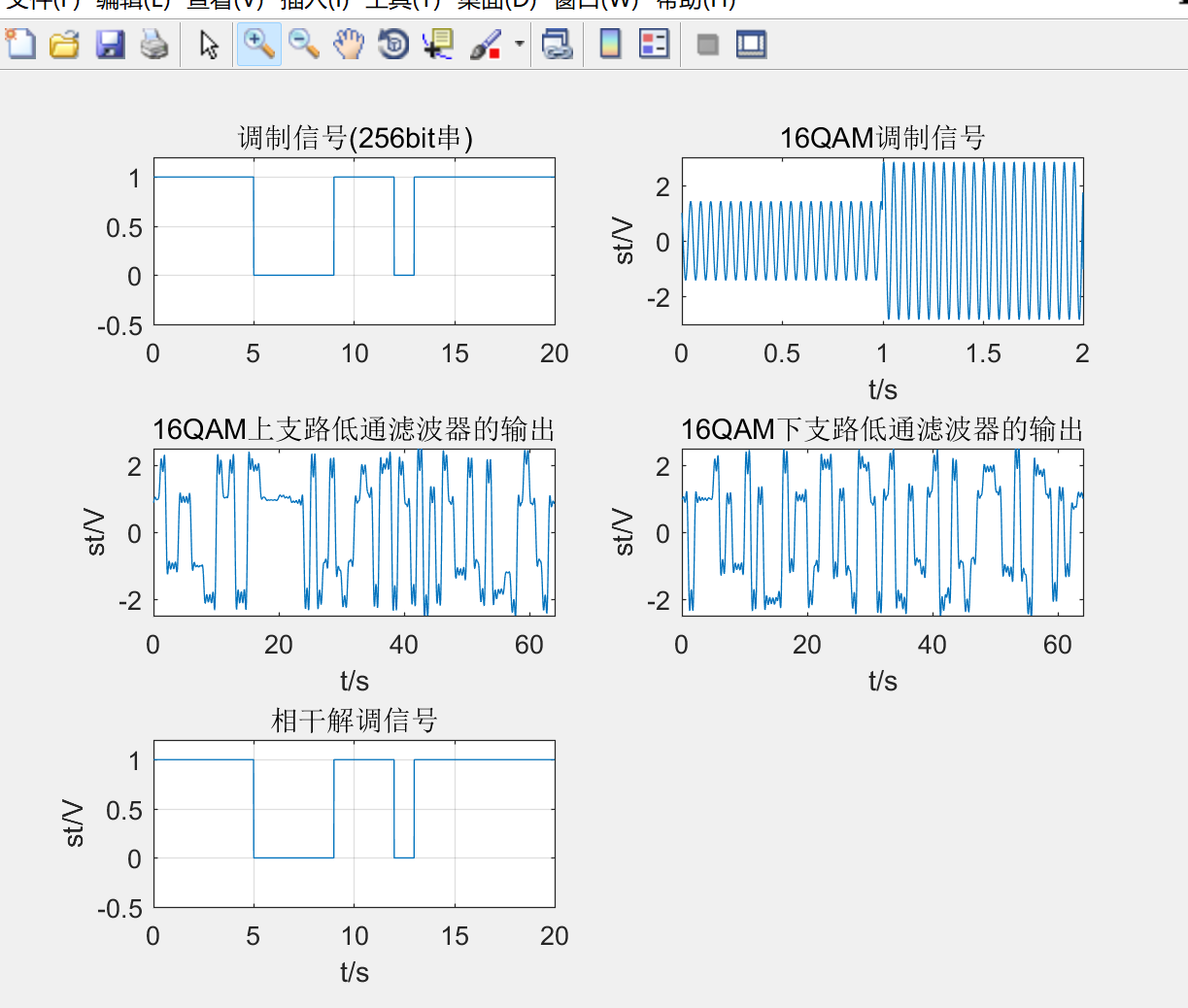


实验二（2） %模拟了20次

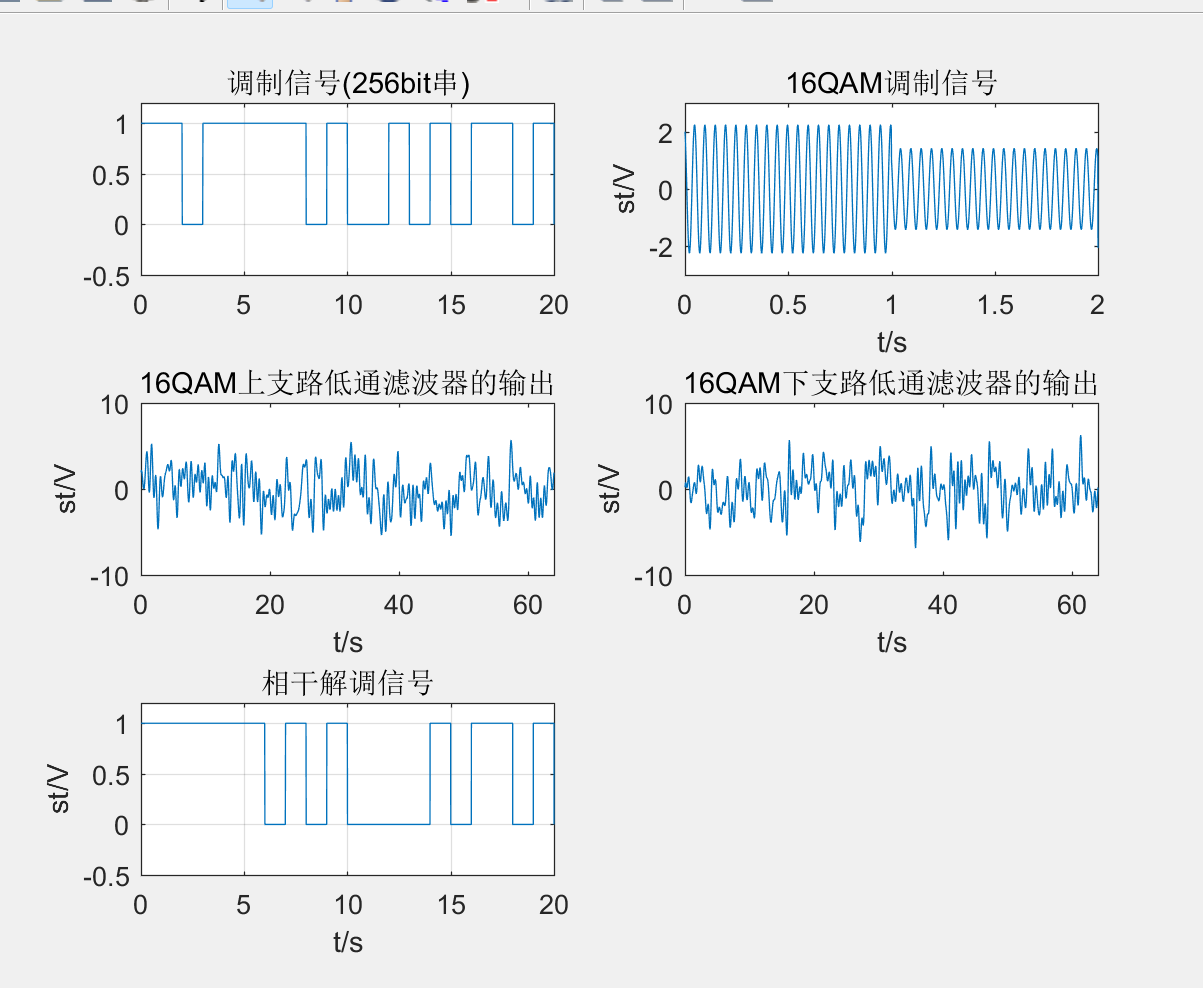


实验三（1）

无噪声的情况下



信噪比为0 db的情况



实验三（2）%模拟了100次

图表

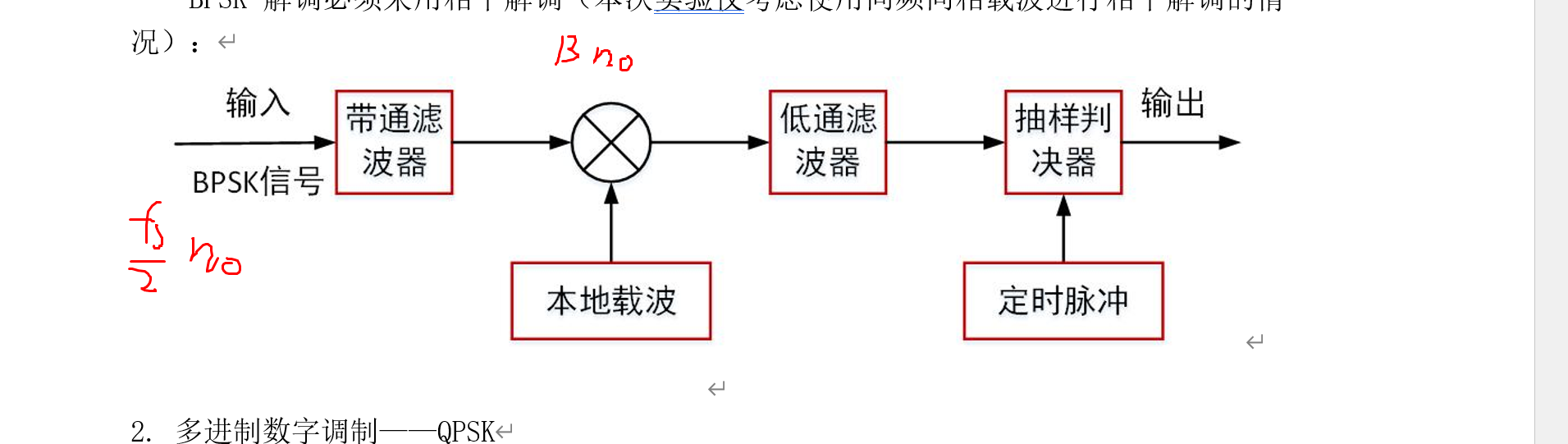
描述已自动生成

## 五、实验分析

1、这里对信号的噪声的添加做统一说明

，通过A（载波幅度），r（信噪比），可以求出高斯窄带白噪声的的功率 ,。

所以加入AWGN信道的高斯白噪声的功率是,fs是采样频率，B带通滤波器带宽



**2、对于不同信号的Ps求法**

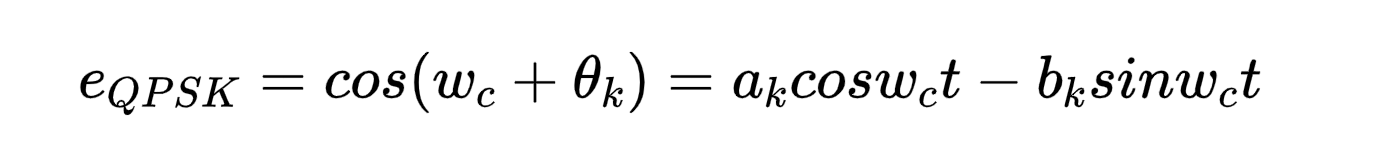
**因为要计算噪声的功率，首先要计算信号的功率，所以对不同信号的功率计算方法列举出来**

对于2PSK

信号表达式是

功率是

对于QPSK

信号表达式是,ak=1或-1，bk=1或-1  


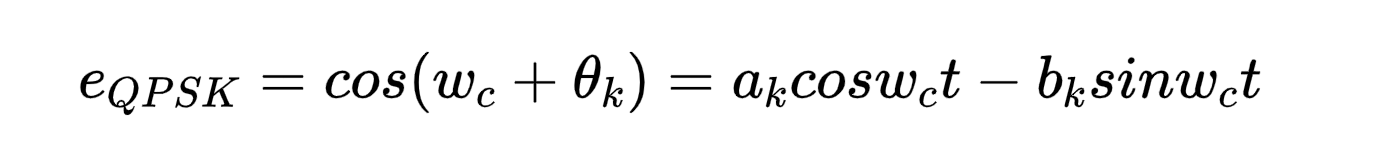
功率是:

文本

描述已自动生成

对于16QAM

3、QPSK的星座图



对于的一一关系

ak bk 调制的结果

1 1 文本, 信件

描述已自动生成

0 1 文本

描述已自动生成

0 0 文本

描述已自动生成

1 0 文本, 信件

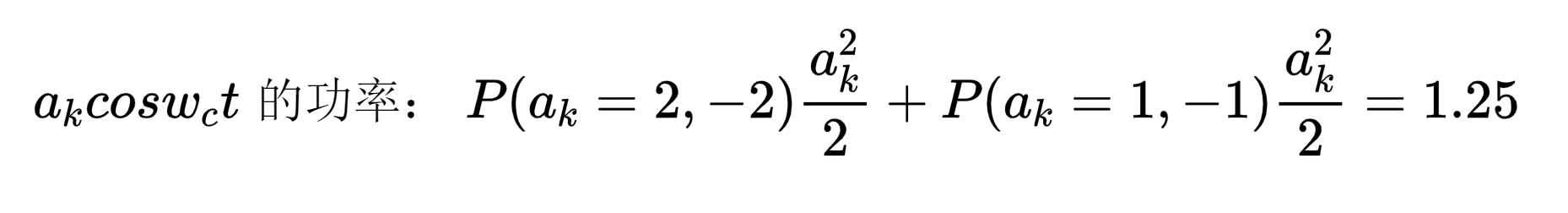
描述已自动生成

信号的表达式是 ak=-2，-1，1，2，bk=-2,-1,1,2

文本

中度可信度描述已自动生成

对于功率的计算



其中由于比特是0，1的随机比特，

所以P（ak=2,-2）概率是1/2，所以P（ak=1,-1）概率是1/2

对于右半部分同理，得到的最终Ps=2.5

QPSK的星座图

图表, 散点图

描述已自动生成

4、16QAM的星座图

16QAM调制的细节

文本

中度可信度描述已自动生成

ac 24电平转换 (满足格雷码)

ac比特值 ak的电压

10 2

11 1

01 -1

00 -2

bd 24电平转换 (满足格雷码)

bd比特值 bk的电压

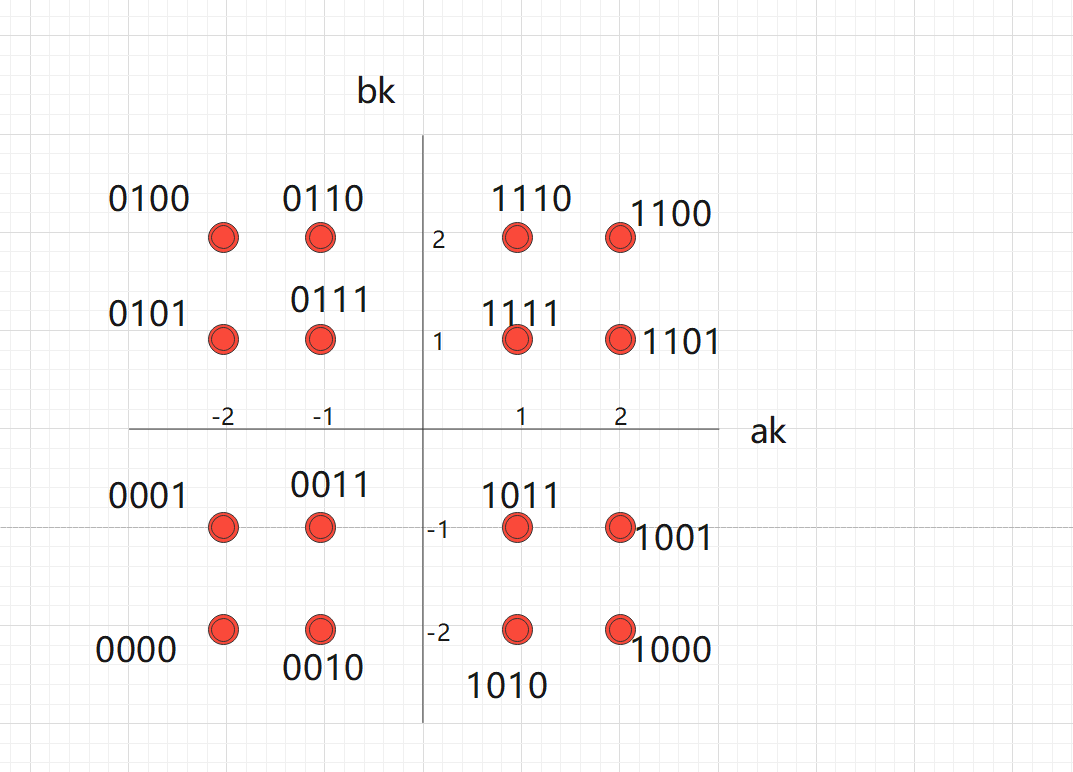
10 2

11 1

01 -1

00 -2

16QAM的星座图



5、理论误码率的公式

2PSK，（教材P200）

图示

中度可信度描述已自动生成

QPSK （教材P222）

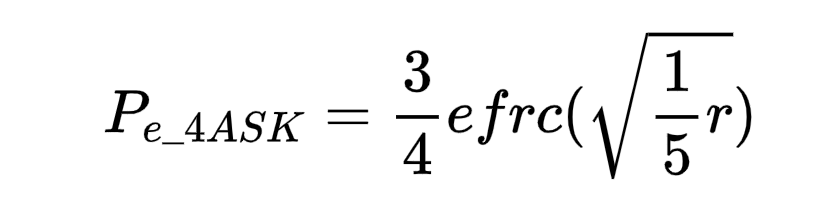
图示

中度可信度描述已自动生成

16QAM公式

本次实现16QAM相当于2个4ASK相加

4ASK的误码率公式（P217）：



16QAM的误码率计算方法是

首先计算两个4ASK都正确的概率，相乘得出16QAM正确的概率

再用1减去正确的概率，得到最终的误码率

由于r分为两个子路，所以对每一个支路来说r’=0.5r

最终的误码率理论公式

