

# QAM调制解调的仿真实现（b测）

2021年05月10日

组队ID：3212

姓名	学号	学院
张帅豪	18030100101	计算机科学与技术学院
赵宇轩	18030100103	计算机科学与技术学院
贾景元	18130500232	软件工程

教师指导评语：

成 绩

测试教师：

年 月 日

## 实验报告内容基本要求及参考格式

- 一、题目要求
- 二、原理分析
- 三、需求方案
- 四、实验结果
- 五、附录代码

## 一、题目要求

任务：使用MATLAB软件，实现对QAM系统调制与解调过程的仿真，然后分析系统的可靠性。

- (1) 对原始信号分别进行4QAM和16QAM调制，画出星座图；
- (2) 采用高斯白噪声信道传输信号，画出信噪比为18dB时，4QAM和16QAM的接收信号星座图；
- (3) 画出两种调制方式的眼图；
- (4) 解调接收信号，分别绘制4QAM和16QAM的误码率曲线图，并与理论值进行对比；

## 二、原理分析

### • QAM概念

正交振幅调制（QAM）是用两个独立的基带数字信号对两个互相正交的同频载波进行抑制在播的双边带调制，利用这种已调信号在同一带宽内频谱正交的性质来实现两路并行的数字信息传输。它是把MASK与MPSK两种结合到一起的调制技术，使得带宽得到双倍拓展。

AM：振幅和相位联合键控的调制方式。

信号的一个码元可表示为：

$$s_k(t) = A_k \cos(\omega_0 t + \theta_k), kT < t \leq (k+1)T$$

将上式展开：

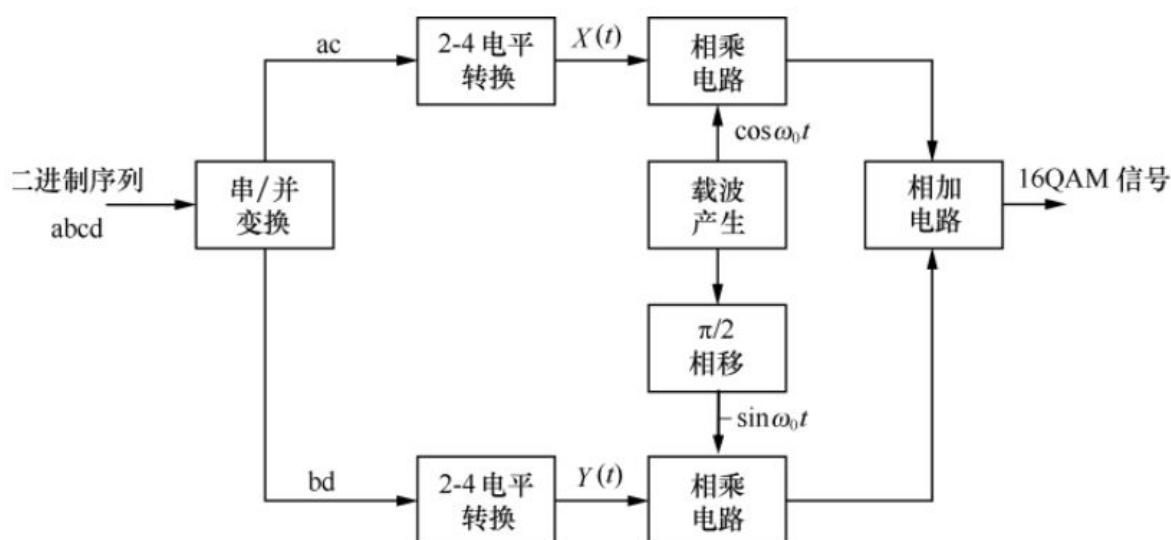
$$s_k(t) = A_k \cos\theta_k \cos\omega_0 t - A_k \sin\theta_k \sin\omega_0 t$$

令 $X_k = A_k \cos\theta_k, Y_k = -A_k \sin\theta_k$ ，则信号变为：

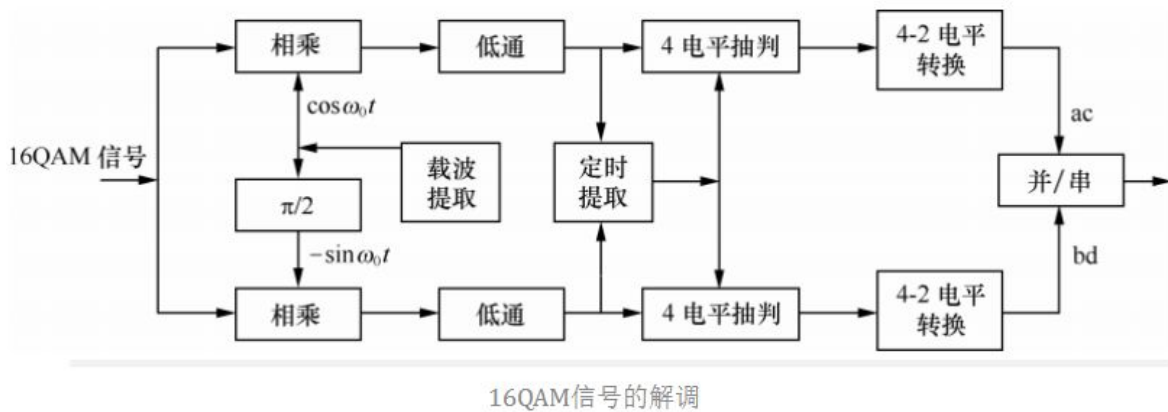
$$s_k(t) = X_k \cos\omega_0 t + Y_k \sin\omega_0 t$$

$X_k$ 和 $Y_k$ 是由振幅和相位决定的取离散值的变量。每一个码元可以看作是二个载波正交的振幅键控信号之和。

### • 原理图（以16QAM为例）



16QAM信号的生成



### 三、需求方案

(1) 对原始信号分别进行4QAM和16QAM调制，画出星座图；

分析：

原始信号：我们创建10000\*log2(QAM)个随机信号（0 or 1）将其转换为（0---(QAM-1)）的值

eg: 01 ---> 1 10--->2

0100-->4 1111-->15

```
1 M = input('输入qam');
2 m=log2(M);
3 n=10000; %Random number.
4 x1=randi([0,1],m*n,1); %random bit stream.
5
6 [m1,n1] = size(x1);
7 fprintf('原始信号1 %d行%d列 最小值%d 最大值%d\n',m1,n1,min(x1),max(x1));
8 x2=reshape(x1,m,length(x1)/m);
9 [m2,n2] = size(x2);
10 fprintf('原始信号2 %d行%d列\n',m2,n2);
11 x3=bi2de(x2,'left-msb');%将二进制随机序列转换为M进制序列
12 [m3,n3] = size(x3);
13 fprintf('原始信号3 %d行%d列 最小值%d 最大值%d 平均值%f\n',m3,n3,min(x3),max(x3),mean(x3));
```

4QAM	16QAM
命令行窗口 输入qam>> 4 原始信号1 20000行1列 最小值0 最大值1 原始信号2 2行10000列 原始信号3 10000行1列 最小值0 最大值3 平均值1.500500 4QAM调制 可视化星座图	命令行窗口 输入qam>> 16 原始信号1 40000行1列 最小值0 最大值1 原始信号2 4行10000列 原始信号3 10000行1列 最小值0 最大值15 平均值7.468000 16QAM调制 可视化星座图

QAM调制:使用qammod函数，调制根据M确定

```
1 % (1) 对原始信号分别进行4QAM和16QAM调制，画出星座图
2 fprintf('%dQAM调制 可视化星座图\n',M);
3 y=qammod(x1,M,'bin','InputType','bit');
```

星座图

```

1 scatterplot(y);%MQAM星座图
2 text(real(y)+0.1,imag(y),dec2bin(x3));
3 hold on

```

(2) 采用高斯白噪声信道传输信号，画出信噪比为18dB时，4QAM和16QAM的接收信号星座图；

添加信噪比18db的噪声：使用awgn函数

```

1 y_noisy=awgn(y,18);

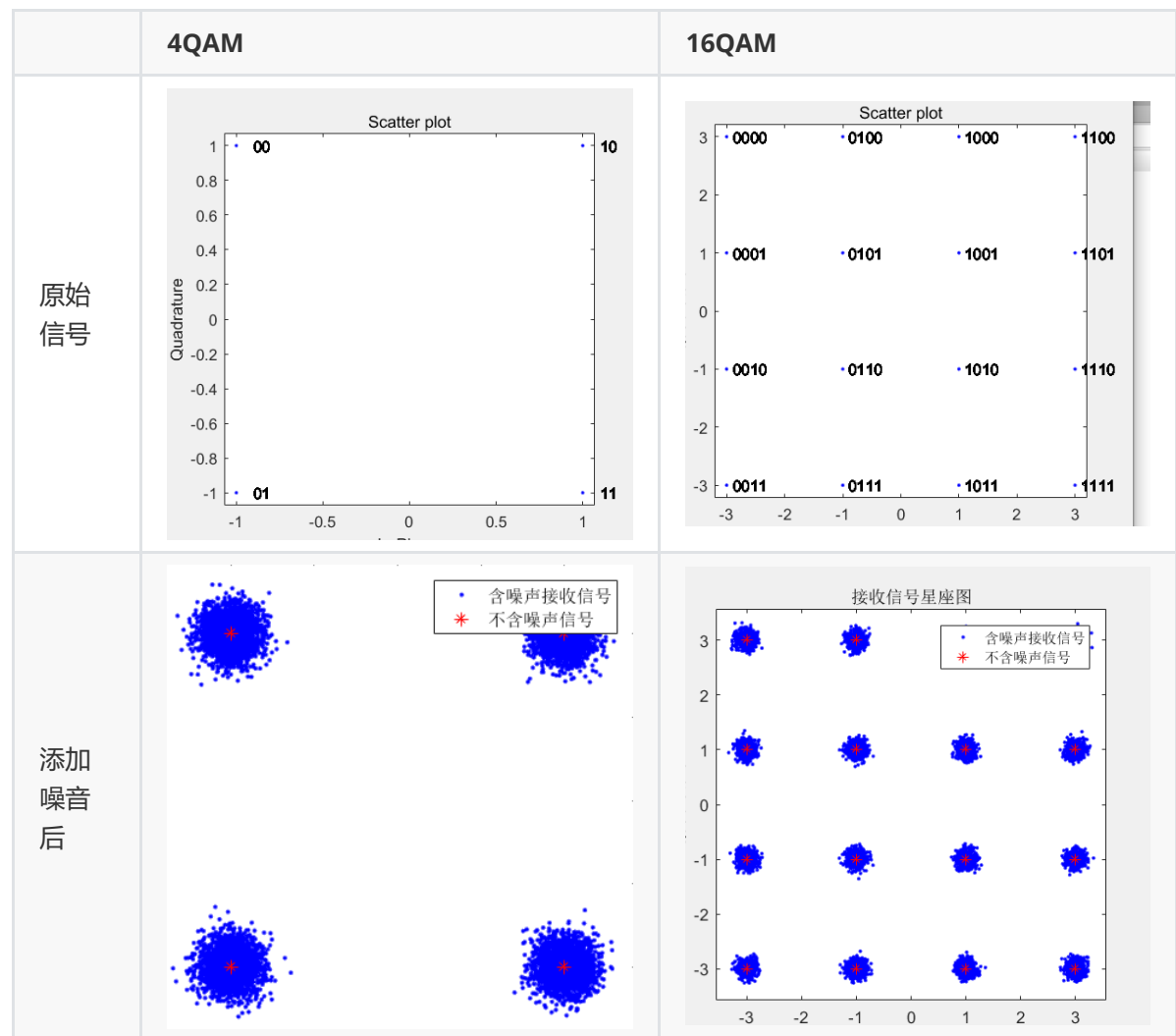
```

添加噪声的接收信号图

```

1 h=scatterplot(y_noisy,1,0,'b. ');
2 hold on;
3 scatterplot(y,1,0,'r*',h);
4 title('接收信号星座图');
5 legend('含噪声接收信号','不含噪声信号');
6 hold on;

```



(3) 画出两种调制方式的眼图

使用eyediagram函数

```

1 eyediagram(y,2);
2 eyediagram(y_noisy,2);

```

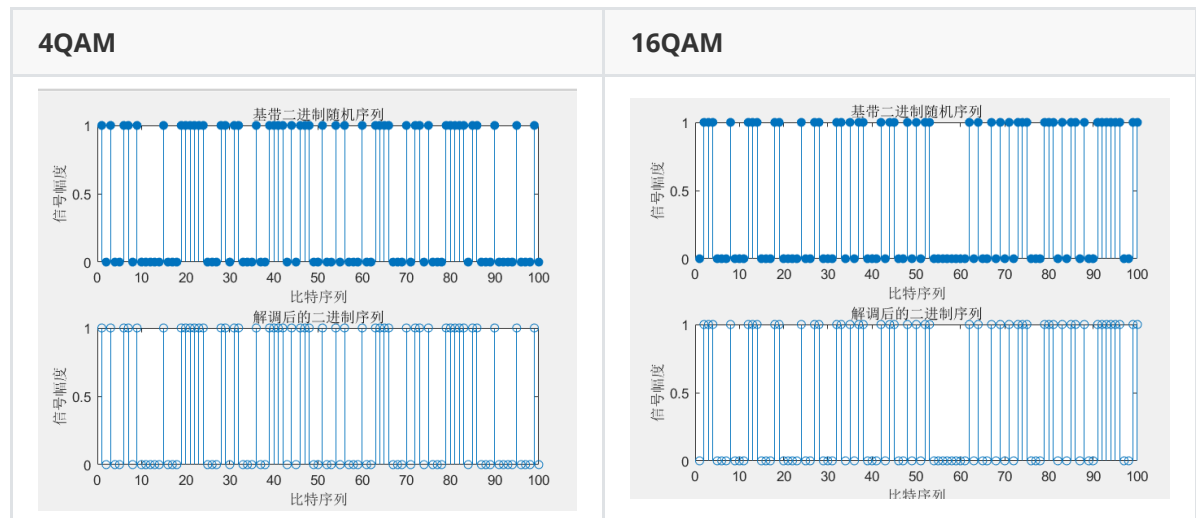
(4) 解调接收信号，分别绘制4QAM和16QAM的误码率曲线图，并与理论值进行对比；

解调信号：使用qamdemod函数

```

1 y_deqam1=qamdemod(y,M,'bin','outputType','bit');%解调信号
2 figure
3 subplot(2,1,1);
4 stem(x1(1:100),'filled'); %基带信号前100位
5 title('基带二进制随机序列');
6 xlabel('比特序列');ylabel('信号幅度');
7 subplot(2,1,2);
8 stem(y_deqam1(1:100)); %画出相应的经过QAM调制解调后的信号的前100
   位
9 title('解调后的二进制序列');
10 xlabel('比特序列');ylabel('信号幅度');

```



分析：可以看出解调后出错的比特很少。

理论误码率：使用berawgn函数， format long可以提高数据精度

```

1 SNR_in_dB1=0:1:24;
2 format long;
3 berQ = berawgn(SNR_in_dB1,'qam',M);

```

实际误码率:查看每个信噪比的噪音下，解调后的比特码流与初始的差别

```

1 NR_in_dB=0:1:24; %AWGN信道信噪比;
2 for j=1:length(SNR_in_dB)
3     format long;
4     y_noise = awgn(y,SNR_in_dB(j));%加入不同强度的高斯白噪声;
5     y_output = qamdemod(y_noise,M,'bin','outputType','bit'); %对已调信号进行解
   调
6     num=0;
7     for i=1:length(x1)
8         if (y_output(i)~=x1(i))
9             num=num+1;
10        end
11    end

```

```
12
13     [number, ratio] = biterr(x1,y_output);
14     Pe0(j) = num/length(y_output);
15     Pe(j) = number/length(y_output);
16 end
17
```

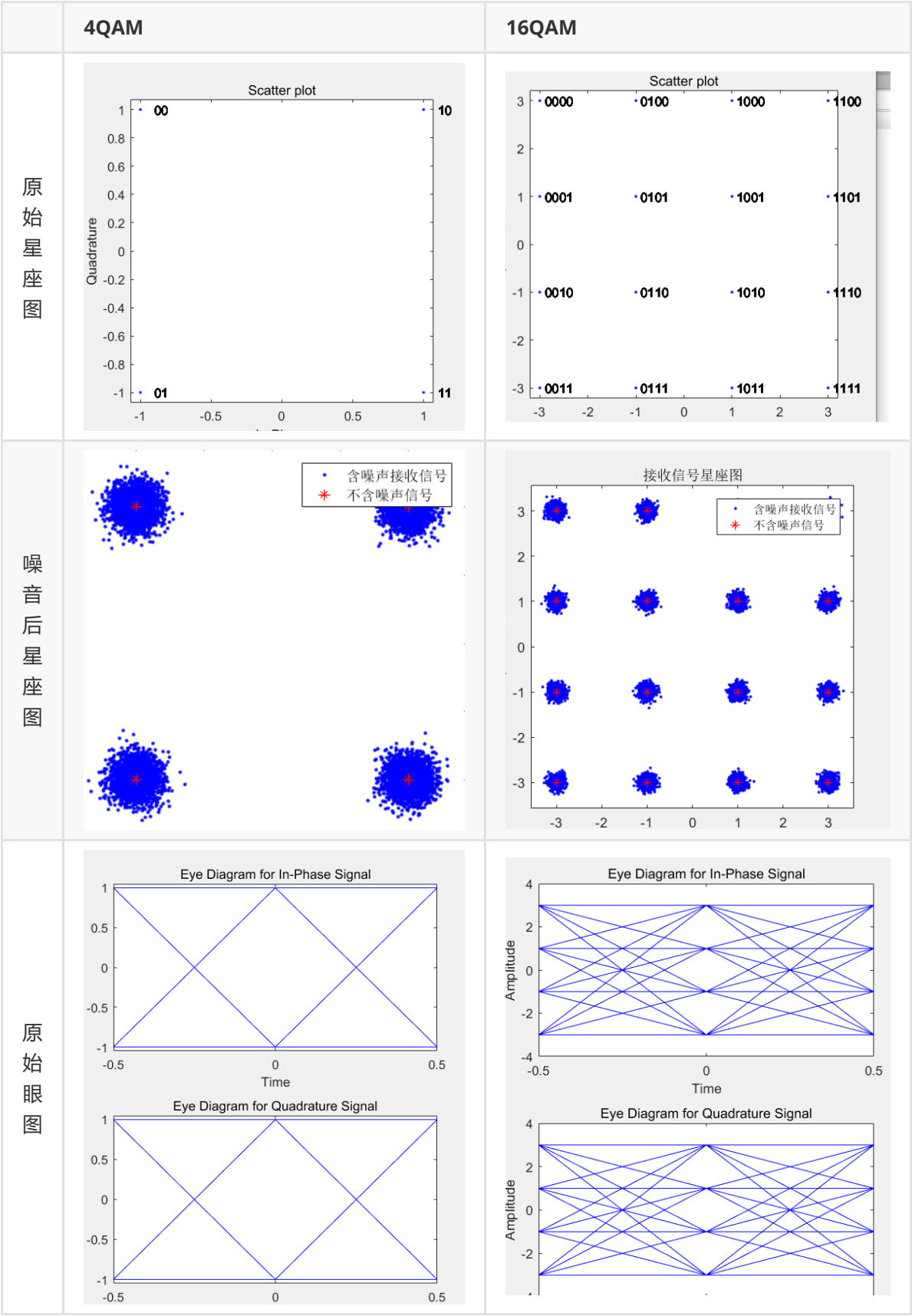
(实际误码率)使用了for循环寻找和biterr查找两种方法对比，结果如下：

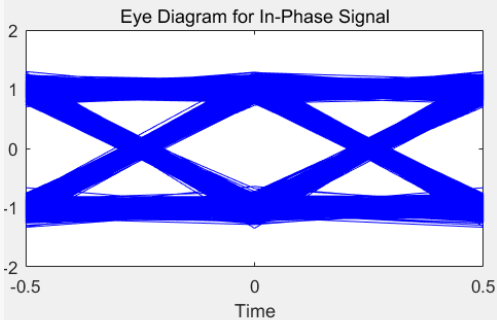
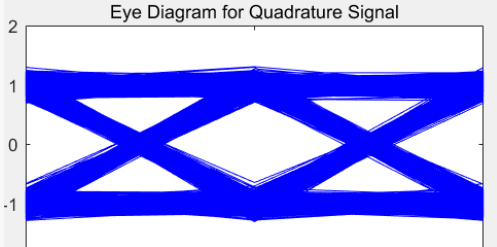
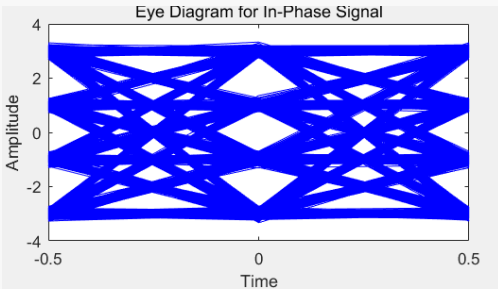
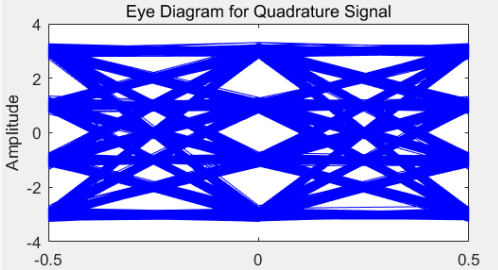
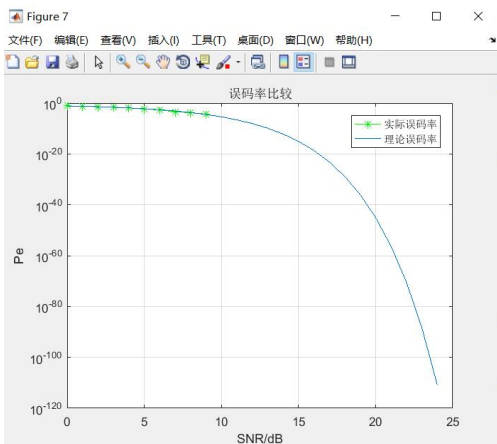
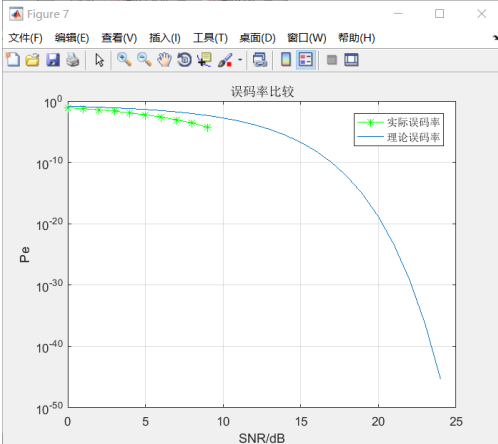
4QAM	16QAM
4QAM调制 可视化星座图	16QAM调制 可视化星座图
信噪比: 0 实际误码率:0.080350 0.080350	信噪比: 0 实际误码率:0.079250 0.079250
信噪比: 1 实际误码率:0.057700 0.057700	信噪比: 1 实际误码率:0.053250 0.053250
信噪比: 2 实际误码率:0.035800 0.035800	信噪比: 2 实际误码率:0.037575 0.037575
信噪比: 3 实际误码率:0.021650 0.021650	信噪比: 3 实际误码率:0.023375 0.023375
信噪比: 4 实际误码率:0.012750 0.012750	信噪比: 4 实际误码率:0.012725 0.012725
信噪比: 5 实际误码率:0.006200 0.006200	信噪比: 5 实际误码率:0.005675 0.005675
信噪比: 6 实际误码率:0.002650 0.002650	信噪比: 6 实际误码率:0.002700 0.002700
信噪比: 7 实际误码率:0.000500 0.000500	信噪比: 7 实际误码率:0.001050 0.001050
信噪比: 8 实际误码率:0.000100 0.000100	信噪比: 8 实际误码率:0.000175 0.000175
信噪比: 9 实际误码率:0.000050 0.000050	信噪比: 9 实际误码率:0.000025 0.000025
信噪比: 10 实际误码率:0.000000 0.000000	信噪比: 10 实际误码率:0.000050 0.000050
信噪比: 11 实际误码率:0.000000 0.000000	信噪比: 11 实际误码率:0.000000 0.000000
信噪比: 12 实际误码率:0.000000 0.000000	信噪比: 12 实际误码率:0.000000 0.000000
信噪比: 13 实际误码率:0.000000 0.000000	信噪比: 13 实际误码率:0.000000 0.000000
信噪比: 14 实际误码率:0.000000 0.000000	信噪比: 14 实际误码率:0.000000 0.000000
信噪比: 15 实际误码率:0.000000 0.000000	信噪比: 15 实际误码率:0.000000 0.000000
信噪比: 16 实际误码率:0.000000 0.000000	信噪比: 16 实际误码率:0.000000 0.000000
信噪比: 17 实际误码率:0.000000 0.000000	信噪比: 17 实际误码率:0.000000 0.000000
信噪比: 18 实际误码率:0.000000 0.000000	信噪比: 18 实际误码率:0.000000 0.000000
信噪比: 19 实际误码率:0.000000 0.000000	信噪比: 19 实际误码率:0.000000 0.000000
信噪比: 20 实际误码率:0.000000 0.000000	信噪比: 20 实际误码率:0.000000 0.000000

误码率曲线图：

```
1 figure();
2 semilogy(SNR_in_dB,Pe,'green*-');
3 hold on;
4 semilogy(SNR_in_dB1,berQ);
5 title('误码率比较');
6 legend('实际误码率','理论误码率');
7 hold on;
8 grid on;
```

四、实验结果



	4QAM	16QAM
噪声后眼图	 	 
误码率曲线		

## 五、附录代码

```

1  clear;
2  clc;
3  close all;
4  %生成随机信号并转换为0-(M-1) %
5  M = input('输入qam');
6  m=log2(M);
7  n=10000; %Random number.
8  x1=randi([0,1],m*n,1); %random bit stream.
9  [m1,n1] = size(x1);
10 fprintf('原始信号1 %d行%d列 最小值%d 最大值%d\n',m1,n1,min(x1),max(x1));
11 x2=reshape(x1,m,length(x1)/m);
12 [m2,n2] = size(x2);
13 fprintf('原始信号2 %d行%d列\n',m2,n2);
14 x3=bi2de(x2','left-msb');%将二进制随机序列转换为M进制序列
15 [m3,n3] = size(x3);
16 fprintf('原始信号3 %d行%d列 最小值%d 最大值%d 平均值%f\n',m3,n3,min(x3),max(x3),mean(x3));
17 % (1) 对原始信号分别进行4QAM和16QAM调制, 画出星座图

```



```

18 fprintf('%dQAM调制 可视化星座图\n',M);
19 y=qammod(x1,M,'bin','InputType','bit');
20 scatterplot(y);%MQAM星座图
21 text(real(y)+0.1,imag(y),dec2bin(x3));
22 hold on
23 % (2) 采用高斯白噪声信道传输信号，画出信噪比为18dB时，
24 % 4QAM和16QAM的接收信号星座图：
25 y_noisy=awgn(y,18);
26 h=scatterplot(y_noisy,1,0,'b. ');
27 hold on;
28 scatterplot(y,1,0,'r*',h);
29 title('接收信号星座图');
30 legend('含噪声接收信号','不含噪声信号');
31 hold on;
32 % (3) 画出两种调制方式的眼图：
33 eyediagram(y,2);
34 eyediagram(y_noisy,2);
35 %基带信号和解调信号图示%
36 y_deqam1=qamdemod(y,M,'bin','OutputType','bit');%解调信号
37 figure
38 subplot(2,1,1);
39 stem(x1(1:100),'filled'); %基带信号前100位
40 title('基带二进制随机序列');
41 xlabel('比特序列');ylabel('信号幅度');
42 subplot(2,1,2);
43 stem(y_deqam1(1:100)); %画出相应的经过QAM调制解调后的信号的前100
44 %位
45 title('解调后的二进制序列');
46 xlabel('比特序列');ylabel('信号幅度');
47 %基带信号和解调信号图示%
48 y_deqam2=qamdemod(y_noisy,M,'bin','OutputType','bit');%解调信号
49 figure
50 subplot(2,1,1);
51 stem(x1(1:100),'filled'); %基带信号前100位
52 title('基带二进制随机序列');
53 xlabel('比特序列');ylabel('信号幅度');
54 subplot(2,1,2);
55 stem(y_deqam2(1:100)); %画出相应的经过QAM调制解调后的信号的前100
56 %位
57 title('解调后的二进制序列');
58 xlabel('比特序列');ylabel('信号幅度');
59 %实际误差值%
60 SNR_in_dB=0:1:24; %AWGN信道信噪比;
61 for j=1:length(SNR_in_dB)
62     format long;
63     y_noise = awgn(y,SNR_in_dB(j));%加入不同强度的高斯白噪声;
64     y_output = qamdemod(y_noise,M,'bin','OutputType','bit'); %对已调信号进行解
65     调
66     num=0;
67     for i=1:length(x1)
68         if (y_output(i)~=x1(i))
69             num=num+1;
70         end
71     end
72     [number, ratio] = biterr(x1,y_output);
73     Pe0(j) = num/length(y_output);

```

```

72     Pe(j) = number/length(y_output); %t通过错误码元数与总码元数之比求误码率，不同j
    是不同信噪比下的误码
73 end
74 %理论误差值%
75 SNR_in_dB1=0:1:24;
76 format long;
77 berQ = berawgn(SNR_in_dB1,'qam',M);
78
79 for j=1:length(SNR_in_dB)
80     fprintf('信噪比: %d    实际误码率:%f    %f    \n',SNR_in_dB(j),Pe(j),Pe0(j));
81 end
82 %画图
83 figure();
84 semilogy(SNR_in_dB,Pe,'green*-');
85 hold on;
86 semilogy(SNR_in_dB1,berQ);
87 title('误码率比较');
88 legend('实际误码率','理论误码率');
89 hold on;
90 grid on;
91 xlabel('SNR/dB');
92 ylabel('Pe1');

```