

**2025年B级实验能力达标测试 实验报告**

**实验名称 QAM调制解调的仿真实现**

姓名 方子康 学号 22009200766

成 绩

同作者 刘惠新22009200114

同作者 高梓轩22009200100

组号：5990

实验日期 2025 年 5 月 7 日

|  |
| --- |
| 指导教师评语：  实验目的：□明确 □基本明确 □目的不明  实验方案：□正确 □基本正确 □有待完善  实验结果：□正确 □基本正确 □存在错误  结果分析：□完善 □基本完善 □无分析或存在错误  实验报告格式：□格式规范，图表清晰 □ 格式基本规范  □格式不规范  指导教师：  年 月 日 |

**一、实验任务**

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。

**二、实验内容**

（1）对原始信号分别进行8QAM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，8QAM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8QAM和16QAM的误码率曲线图，并与理论值进行对比；

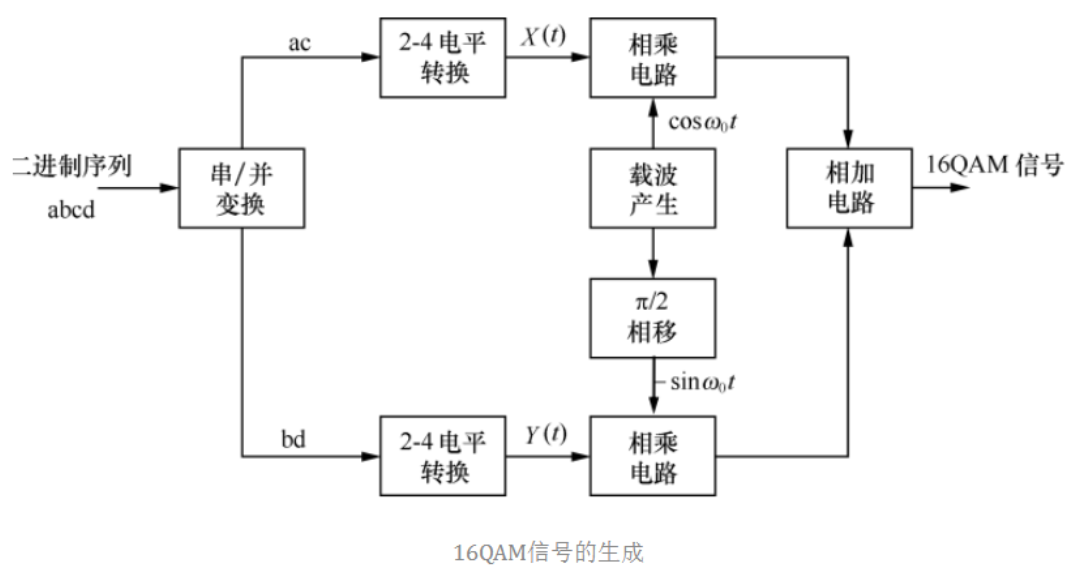
（5）提交详细的设计报告和实验报告。

**三、实验原理**

QAM（正交幅度调制）是一种高效的数字调制方式，它通过改变载波信号的幅度和相位来传输数字信息。QAM 结合了幅度调制（AM）和相位调制（PM）的特点，能够在有限的带宽内传输更多的信息。

**QAM 调制原理**

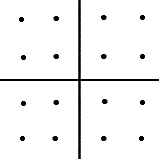
正交幅度调制（QAM）信号采用了两个正交载波和，每一个载波都被一个独立的信息比特序列所调制,将输入的比特流按照每个符号的比特数进行分组。每组对应不同的幅度和相位，用于调制载波信号。调制公式可以表示为：

其中，​ 和 分别是同相分量和正交分量的幅度，是载波频率。调制原理框图如下：

**为了建立从比特到符号的映射**QAM 将多个比特组合成一个符号，8QAM 将 3 个比特组合成一个符号，16QAM 将 4 个比特组合成一个符号。每个符号对应一个特定的幅度和相位组合，这些组合在星座图上表示为不同的点。

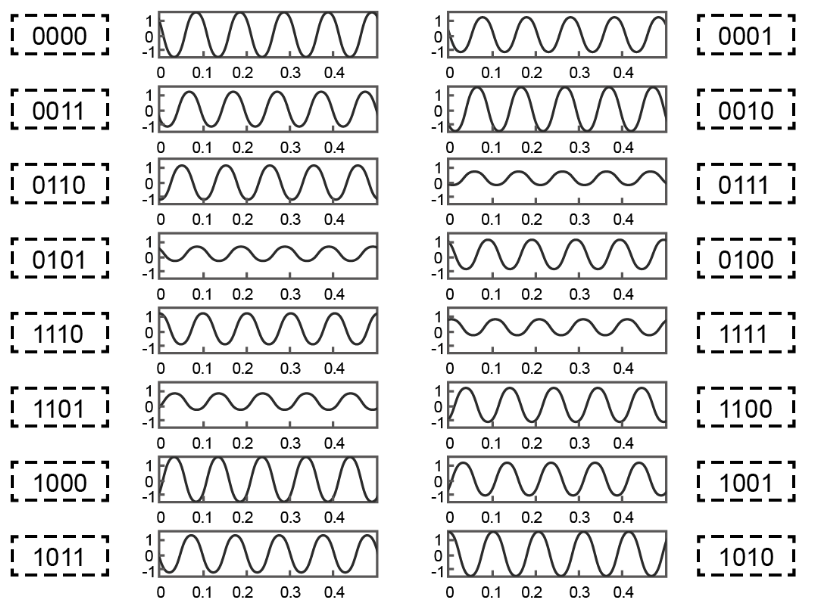
**星座图**

星座图是 QAM 调制的核心，它直观地展示了每个符号对应的幅度和相位信息。横轴通常表示同相分量（In-phase，I），纵轴表示正交分量（Quadrature，Q）。例如，8QAM 的星座图上有 8 个点，每个点对应不同的幅度和相位组合；16QAM 的星座图上有 16 个点，各符号通过格雷码进行编码。



M=16QAM信号星座图

16QAM通过QAM调制可得到16个不同的波形，分别代表0000，0001....这也意味着一共有16种符号，一个符号可以传递4 bit信息。



**QAM 解调原理**

**（1）信号接收**

接收端收到的信号通常会受到噪声和干扰的影响。解调的过程是从接收到的信号中提取出原始的比特信息。

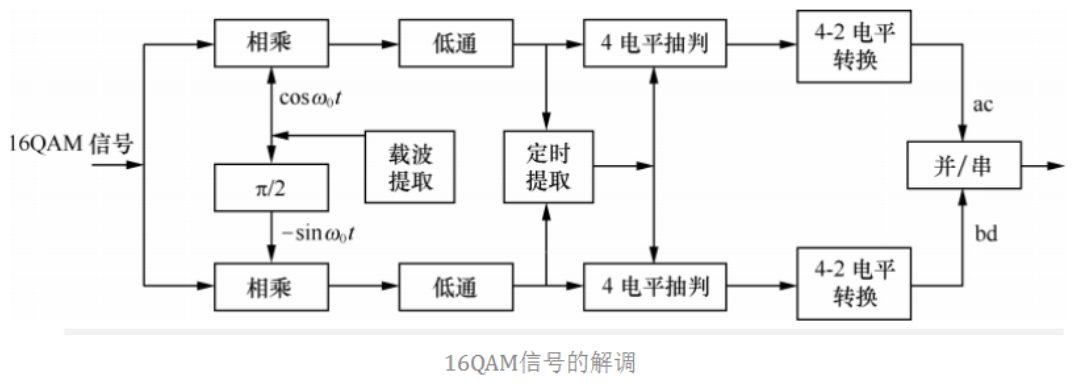
**（2）下变频**

将接收到的射频（RF）信号下变频到基带信号。这通常通过与本地振荡器产生的同相（I）和正交（Q）载波信号进行混频来实现。

**（3）符号检测**

对下变频后的基带信号进行采样和量化，将其映射到星座图上的最近点。这一步骤通常使用判决技术来确定最接近的星座点。例如，在 8QAM 中，接收信号的同相和正交分量会被量化到最近的幅度和相位组合，从而确定对应的符号。

**（4）比特恢复**

将检测到的符号转换回对应的比特序列。例如，8QAM 的每个符号对应 3 个比特，16QAM 的每个符号对应 4 个比特。解调的原理框图如下：

**QAM的误码率性能**

矩形QAM信号星座最突出的优点就是容易产生PAM信号可直接

加到两个正交载波相位上，此外它们还便于解调。

对于M＝下的矩形信号星座图（k为偶数），QAM信号星座图与正交载波上的两个PAM信号是等价的，这两个信号中的每一个上都有个信号点。因为相位正交分量上的信号能被相干判决极好的分离，所以易于通过PAM的误码率确定QAM的误码率。M进制QAM系统正确判决的概率是



式中是进制PAM系统的误码率，该PAM系统具有等价QAM系统的每一个正交信号中的一半平均功率。通过适当调整M进制PAM系统的误码率，可得



其中是每个符号的平均信噪比。因此M进制QAM的误码率为

)

**四、实验方案与步骤**

**1.信号生成与调制**

**符号生成与格雷编码映射**：通过 randi 生成均匀分布的随机符号序列（8QAM 符号范围 0-7，16QAM 符号范围 0-15）。使用预定义的格雷映射表（grayMap8/grayMap16）将原始符号转换为格雷编码符号，确保相邻星座点仅 1 比特差异（降低误符号时的比特错误数）。

%% 8QAM

txSym8 = randi([0, M-1], 1, nSym); % 生成8进制符号序列

graySym8 = grayMap8(txSym8 + 1); % 格雷编码映射(MATLAB索引从1开始)

%%16QAM

txSym16 = randi([0, N-1], 1, nSym);

graySym16 = grayMap16(txSym16 + 1);

**QAM 调制**：调用 qammod 函数将格雷编码符号映射到复平面星座点（8QAM 生成 8 个星座点，16QAM 生成 16 个星座点）。

txSig8 = qammod(graySym8, M); % 执行QAM调制，输出复数星座点(点在星座图的位置)

txSig16 = qammod(graySym16, N);

**2.信号质量可视化（眼图与星座图）**

**眼图与星座图绘制**：通过 eyediagram 观察调制信号的时域质量，通过 scatterplot 观察星座点的分布

% 4QAM可视化

eyediagram(txSig8, 2); % 绘制眼图，2表示每符号2个采样点

scatterplot(txSig8); % 绘制星座图观察调制效果

pow8 = norm(txSig8)^2 / nSym; % 计算信号平均功率：范数平方/符号数

% 16QAM可视化

eyediagram(txSig16, 2); % 眼图（对比与4QAM的差异）

scatterplot(txSig16); % 16QAM星座图（观察16个星座点分布）

pow16 = norm(txSig16)^2 / nSym; % 计算功率

**3.信道噪声模拟**

**计算噪声标准差**：根据信号平均功率（pow8/pow16）和当前信噪比（snrLinear），计算高斯噪声的标准差（sigma8/sigma16）

% 噪声功率计算（实部虚部独立，总噪声功率=sigma^2 \* 2）

sigma8 = sqrt(pow8 / (2 \* snrLinear(i))); % 8QAM噪声标准差：sqrt(符号功率/(2\*SNR))

sigma16 = sqrt(pow16 / (2 \* snrLinear(i))); % 16QAM噪声标准差

**添加加性高斯白噪声（AWGN）**：向调制信号叠加实部、虚部独立的高斯噪声（rxSig4/rxSig16），模拟实际信道的噪声干扰。

% 加性高斯白噪声信道模拟

rxSig8 = txSig8 + sigma8\*(randn(1,nSym) + 1i\*randn(1,nSym));%复噪声：实虚独立同分布

rxSig16 = txSig16 + sigma16\*(randn(1,nSym) + 1i\*randn(1,nSym));

**4.信号解调与符号恢复**

**QAM 解调与格雷逆映射**：调用 qamdemod 函数对带噪声信号进行硬判决解调，恢复符号序列（rxSym8/rxSym16）。使用格雷映射表将解调符号转换回原始符号（decSym8/decSym16），消除格雷编码的影响。

% QAM解调

rxSym8 = qamdemod(rxSig8, M); % 8QAM解调，返回0-7的整数

rxSym16 = qamdemod(rxSig16, N); % 16QAM解调，返回0-15的整数

% 格雷逆映射（解格雷编码）

decSym8 = grayMap8(rxSym8 + 1); % +1索引调整，映射回原始数据

decSym16 = grayMap16(rxSym16 + 1);

**5.误码率计算与理论验证**

**误比特率（BER）**：通过 biterr 函数对比原始符号与解调符号，结合每个符号的比特数（log2(M)/log2(N)）计算 BER。

**误符号率（SER）**：通过 symerr 函数直接对比原始符号与解调符号，统计符号错误比例。

**理论值计算**：基于 QAM 理论误码率公式（如serTheory8/berTheory8），计算 8QAM/16QAM 的理论 SER 和 BER，用于验证仿真结果的准确性。

[~, ber8(i)] = biterr(txSym8, decSym8, log2(M)); % 比特错误率，log2(M)指定比特数

[~, ser8(i)] = symerr(txSym8, decSym8); % 符号错误率（直接比较符号）

[~, ber16(i)] = biterr(txSym16, decSym16, log2(N)); % 16QAM BER

[~, ser16(i)] = symerr(txSym16, decSym16); % 16QAM SER

%%理论值计算

% 理论误码率计算（4QAM=QPSK）

p8 = 2\*(1 - 1/Q4) \* qfunc(sqrt(3\*snrLinear/(M-1))) % 符号错误率公式：2\*(1-1/sqrt(M))\*Q(sqrt(3\*SNR/(M-1)))

serTheory8 = 1 - (1 – p8).^2; % 正确概率平方反推总错误率

berTheory8 = serTheory8 / log2(M); % 近似关系：BER ≈ SER / 比特数

完整代码  
clc; clear all; close all;

% =============== 参数定义 ===============

nSym = 5e3; % 符号数量

M = 8; % 8QAM调制阶数，每个符号携带log2(8)=3比特信息

N = 16; % 16QAM调制阶数，每个符号携带log2(16)=4比特信息

grayMap8 = [0,1,3,2,6,7,5,4]; % 8QAM格雷映射表

grayMap16 = [0 1 3 2 4 5 7 6 12 13 15 14 8 9 11 10]; % 16QAM格雷映射，确保相邻星座点仅1比特差异

snrRange = 0:15; % 信噪比扫描范围(0-15dB)，步长1

snrSingle = 16; % 单独测试信噪比，用于眼图等可视化分析

snrLinear = 10.^(snrRange/10); % 将dB值转为SNR信噪比，用于理论计算

% =============== 8QAM调制 ===============

% 生成阶段

txSym8 = randi([0, M-1], 1, nSym); % 生成8进制符号序列

graySym8 = grayMap8(txSym8 + 1); % 格雷编码映射(MATLAB索引从1开始)

txSig8 = qammod(graySym8, M); % 执行QAM调制，输出复数星座点(点在星座图的位置)

% 可视化

eyediagram(txSig8, 2); % 绘制眼图，2表示每符号2个采样点

scatterplot(txSig8); % 绘制星座图观察调制效果

pow8 = norm(txSig8)^2 / nSym; % 计算信号平均功率：范数平方/符号数

% =============== 16QAM调制 ===============

% 生成阶段（流程同8QAM）

txSym16 = randi([0, N-1], 1, nSym);

graySym16 = grayMap16(txSym16 + 1);

txSig16 = qammod(graySym16, N);

% 可视化

eyediagram(txSig16, 2); % 眼图（对比与8QAM的差异）

scatterplot(txSig16); % 16QAM星座图（观察16个星座点分布）

pow16 = norm(txSig16)^2 / nSym; % 计算功率

% =============== 信道仿真 ===============

% 预分配存储数组 全0行向量，长16

ber8 = zeros(1, length(snrRange)); % 8QAM误比特率(BER)

ser8 = zeros(1, length(snrRange)); % 8QAM误符号率(SER)

ber16 = zeros(1, length(snrRange)); % 16QAM误比特率

ser16 = zeros(1, length(snrRange)); % 16QAM误符号率

% 循环遍历所有信噪比

for i = 1:length(snrRange)

% 噪声功率计算（实部虚部独立，总噪声功率=sigma^2 \* 2）

sigma8 = sqrt(pow8 / (2 \* snrLinear(i))); % 8QAM噪声标准差：sqrt(符号功率/(2\*SNR))

sigma16 = sqrt(pow16 / (2 \* snrLinear(i))); % 16QAM噪声标准差

% 加性高斯白噪声信道模拟

rxSig8 = txSig8 + sigma8\*(randn(1,nSym) + 1i\*randn(1,nSym)); % 复噪声：实虚独立同分布

rxSig16 = txSig16 + sigma16\*(randn(1,nSym) + 1i\*randn(1,nSym));

% QAM解调

rxSym8 = qamdemod(rxSig8, M); % 8QAM解调，返回0-7的整数

rxSym16 = qamdemod(rxSig16, N); % 16QAM解调，返回0-15的整数

% 格雷逆映射（解格雷编码）

decSym8 = grayMap8(rxSym8 + 1); % +1索引调整，映射回原始数据

decSym16 = grayMap16(rxSym16 + 1);

[~, ber8(i)] = biterr(txSym8, decSym8, log2(M)); % 比特错误率，log2(M)指定比特数

[~, ser8(i)] = symerr(txSym8, decSym8); % 符号错误率（直接比较符号）

[~, ber16(i)] = biterr(txSym16, decSym16, log2(N)); % 16QAM BER

[~, ser16(i)] = symerr(txSym16, decSym16); % 16QAM SER

end

% =============== 8QAM噪声分析 ===============

% 使用awgn函数分别加噪（验证两种加噪方法等效性）

rxReal8 = awgn(real(txSig8), snrSingle); % 实部加噪，snrSingle指定信噪比(dB)

rxImag8 = awgn(imag(txSig8), snrSingle); % 虚部加噪

rxNoise8 = complex(rxReal8, rxImag8); % 重构复信号

% 可视化加噪效果

scatterplot(rxNoise8); % 显示加噪后的星座图

eyediagram(rxNoise8, 2); % 眼图观察噪声影响

% 理论误码率计算（8QAM=QPSK）

p8 = 2\*(1 - 1/sqrt(M)) \* qfunc(sqrt(3\*snrLinear/(M-1))); % 符号错误率公式：2\*(1-1/sqrt(M))\*Q(sqrt(3\*SNR/(M-1)))

serTheory8 = 1 - (1 - p8).^2; % 正确概率平方反推总错误率

berTheory8 = serTheory8 / log2(M); % 近似关系：BER ≈ SER / 比特数

% =============== 16QAM噪声分析 ===============

% 加噪过程同8QAM

rxReal16 = awgn(real(txSig16), snrSingle);

rxImag16 = awgn(imag(txSig16), snrSingle);

rxNoise16 = complex(rxReal16, rxImag16);

% 可视化

scatterplot(rxNoise16);

eyediagram(rxNoise16, 2);

% 理论计算（16QAM）

p16 = 2\*(1 - 1/sqrt(N)) \* qfunc(sqrt(3\*snrLinear/(N-1))); % 符号错误率公式

serTheory16 = 1 - (1 - p16).^2;

berTheory16 = serTheory16 / log2(N);

% =============== 绘图 ===============

% 8QAM性能对比图

figure()

semilogy(snrRange, ber8, "o", snrRange, ser8, "\*", snrRange, serTheory8, "-", snrRange, berTheory8, "-");

title("8QAM误码率性能");

xlabel("Es/N0 (dB)");

ylabel("误码率");

legend("BER仿真", "SER仿真", "SER理论", "BER理论", 'Location','best');

grid on;

% 8QAM与16QAM SER理论对比

figure()

semilogy(snrRange, serTheory8, 'o', snrRange, serTheory16, 'o');

title('8QAM vs 16QAM性能对比');

grid on;

xlabel('Es/N0 (dB)');

ylabel('SER');

legend('8QAM理论', '16QAM理论', 'Location','best');