

注意事项

现代通信原理实验分为软件实验和硬件实验。软件实验主要使用 matlab 编程，信息学院实验教学中心 202 房间，206 房间会提供上机环境，使用的是 matlab2018b 正版软件。每位同学需在助教要求下运行并通过自己的程序，检查通过后登记为平时成绩。硬件实验使用软件定义无线电设备 PLUTO，利用射频和无线通信的基础知识，在 ISM 频段创建信号并在 RF 上播放，实现点对点无线通信系统。

二班助教：何瑞浪，rl666@mail.ustc.edu.cn，手机：18756073608，学号：SA19006020

张淙皓，zch0115@mail.ustc.edu.cn，手机：17305699830，学号：SA20006089

一班助教：葛晓凯，gxm1225@mail.ustc.edu.cn，手机：13665695569，学号：BA20006022

侯振杰，jchzj@mail.ustc.edu.cn，手机：15056036177，学号：BA20006007

注：本实验指导参考了网上的部分资料，仅供本实验室用作教学之用。谢谢。

《现代通信原理》实验指导书(上)

信息学院

2021-3

目 录

实验一 MATLAB 基础知识	1
1.1 MATLAB 基础知识	1
1.1.1 MATLAB 程序设计语言简介	1
1.1.2 MATLAB 界面及帮助	1
1.2 MATLAB 基本运算	3
1.2.1 MATLAB 内部特殊变量和常数	3
1.2.2 变量类型	3
1.2.3 内存变量管理	3
1.2.4 MATLAB 常用数学函数	4
1.2.5 MATLAB 矩阵生成	4
1.2.6 MATLAB 矩阵运算	5
1.2.7 MATLAB 中的矩阵分析	7
1.3 MATLAB 程序设计	7
1.3.1 M 文件	7
1.3.2 程序控制结构	8
1.4 计算结果可视化	13
1.5 确知信号分析	16
1.5.1 周期信号的傅里叶级数	16
1.5.2 信号的傅里叶变换及其反变换	18
实验二（2 小班） 连续波 MATLAB 实现	23
2.1 模拟调制	23
2.2 信道加性高斯白噪声	24
2.3 AM 调制解调的 MATLAB 实现	25
实验二（1 小班） 模拟信号的数字传输	35
2.1 脉冲编码调制	35
2.2 低通抽样定理	35
2.3 均匀量化原理	36
2.4 非均匀量化	38
实验三 眼图	42
3.1 基本原理	42
实验四 调制解调（BPSK，QPSK，信噪比）	46
4.1 调制解调原理	46
4.2 调制解调的 MATLAB 实现（参考）	48
实验五 信道编解码（）	49
5.1 信道编解码原理	错误!未定义书签。
5.2 调制解调注意事项:	错误!未定义书签。

5.3 调制解调的 MATLAB 实现（参考）	错误!未定义书签。
实验六 FSK 传输实验	50
一、实验目的	51
二、实验仪器	51
三、实验原理	51
四、实验步骤	52
五、实验报告	52
附录 实验系统概述	71
电路组成概述	72
用户操作界面使用说明	76
实验一报告 MATLAB 基础实验	80
实验二（2 小班） 连续波 MATLAB 实现	82
实验二（1 小班） 模拟信号的数字传输	82
实验三 眼图（基带调制）	84
实验五 信道编解码	8

实验一 MATLAB 基础知识

本章目标

- 了解 MATLAB 程序设计语言的基本特点，熟悉 MATLAB 软件运行环境
- 掌握创建、保存、打开 m 文件及函数的方法
- 掌握变量等有关概念，具备初步的将一般数学问题转化为对应的计算机模型并进行处理的能力

进行处理的能力

- 掌握二维平面图形的绘制方法，能够使用这些方法进行常用的数据可视化处理
- 理解周期信号的傅里叶级数展开的物理意义
- 掌握信号的傅里叶变换及其反变换

1.1 MATLAB 基础知识

1.1.1 MATLAB 程序设计语言简介

MATLAB, Matrix Laboratory 的缩写, 是由 MathWorks 公司开发的一套用于科学与工程计算的可视化高性能语言, 具有强大的矩阵运算能力。与大家常用的 Fortran 和 C 等高级语言相比, MATLAB 的语法规则更简单, 更贴近人的思维方式, 被称为“草稿纸式的语言”。MATLAB 软件主要由主包、仿真系统 (simulink) 和工具箱 (toolbox) 三大部分组成。

1.1.2 MATLAB 界面及帮助

MATLAB 基本界面如图 1-1 所示, 命令窗口包含标题栏、菜单栏、工具栏、命令行区、状态栏、垂直和水平波动条等区域。

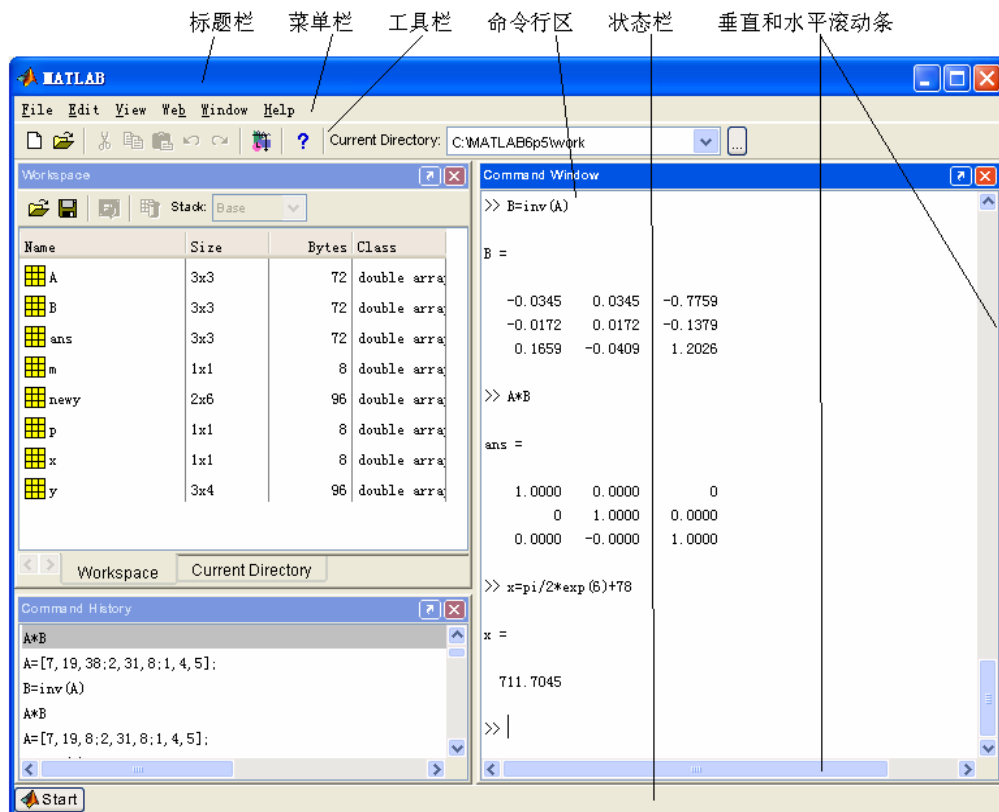


图 1-1 MATLAB 基本界面

（1）菜单栏

在 MATLAB 主窗口的菜单栏，共包含 File、Edit、View、Web、Window 和 Help 6 个菜单项。

- **File 菜单项**：File 菜单项实现有关文件的操作。
- **Edit 菜单项**：Edit 菜单项用于命令窗口的编辑操作。
- **View 菜单项**：View 菜单项用于设置 MATLAB 集成环境的显示方式。
- **Web 菜单项**：Web 菜单项用于设置 MATLAB 的 Web 操作。
- **Window 菜单项**：主窗口菜单栏上的 Window 菜单，只包含一个子菜单 Close all，用于关闭所有打开的编辑器窗口，包括 M-file、Figure、Model 和 GUI 窗口。
- **Help 菜单项**：Help 菜单项用于提供帮助信息。

（2）工具栏

MATLAB 主窗口的工具栏共提供了 10 个命令按钮。这些命令按钮均有对应的菜单命令，但比菜单命令使用起来更快捷、方便。

（3）命令行区

MATLAB 按以下顺序对输入命令进行解释：

- 检查它是否是工作空间中的变量，实则显示变量内容。
- 检查它是否是嵌入函数，是则运行之。
- 检查它是否是子函数。
- 检查它是否是私有函数。
- 检查它是否是位于 MATLAB 搜索路径范围内的函数文件或脚本文件。

MATLAB 里有以下几种方法可获得帮助：

- （1）帮助命令（help）是查询函数相关信息的最直接方式，信息会直接显示在命令窗口中。

键入 `help sin`，会显示 `sin` 相关信息。

(2) `lookfor` 命令可以从键入的关键字列出所有相关的题材，和 `help` 相比，`lookfor` 覆盖范围更广，可查找到某个主题所有词组或短语。

(3) 帮助窗口 (`help window`) 提供与帮助命令相同的信息，但帮助窗口界面更为方便直接。

(4) 帮助桌面 (`help desk`) 通过在命令窗口中选择帮助菜单的“`help desk`”选项或键入 `helpdesk` 命令即可进入帮助桌面。

(5) 在线帮助页是帮助桌面的在线帮助均有相应的 PDF 格式文件。

(6) Mathworks 网站，对于连接入 Internet 的用户通过 Mathworks 公司的网站 www.mathworks.com 询问有关问题。

1.2 MATLAB 基本运算

1.2.1 MATLAB 内部特殊变量和常数

MATLAB 内部有很多变量和常数，用来表达特殊含义，常用的有：

- 变量 `ans`：指示当前未定义变量名的答案。
- 常数 `eps`：表示浮点相对精度，其值是从 1.0 到下一个最大浮点数之间的差值。
- 常数 `Inf`：表示无穷大。
- 虚数单位 `i` 或 `j`：表示不定型值，是由 0/0 运算产生。
- 常数 `pi`：表示圆周率 π 。

1.2.2 变量类型

(1) 变量命名规则

MATLAB 中对变量的命名应遵循以下规则：

- 变量名可以由字母、数字和下划线混合组成，但必须以字母开头。
- 字符长度不能大于 31。
- 变量命名区分大小写。

(2) 局部变量和全局变量

局部变量是指那些每个函数体内自己定义的，不能从其他函数和 MATLAB 工作空间访问的变量。

全局变量是指用关键字“`global`”声明的变量。全局变量名应尽量大写，并能反映它本身的含义。如果需要在工作空间和几个函数中都能访问一个全局变量，必须在工作空间和几个函数中都声明该变量是全局的。

1.2.3 内存变量管理

(1) 内存变量的删除与修改

`clear` 命令用于删除 MATLAB 工作空间中的变量。

`clc` 命令用于清空 MATLAB 命令窗口中的变量。

`who` 和 `whos` 这两个命令用于显示在 MATLAB 工作空间中已经驻留的变量名清单。`who` 命令只显示出驻留变量的名称，`whos` 在给出变量名的同时，还给出它们的大小、所占字节数及数据类型等信息。

(2) 内存变量文件

利用 MAT 文件可以把当前 MATLAB 工作空间中的一些有用变量长久地保留下来，扩展名是 `.mat`。MAT 文件的生成和装入由 `save` 和 `load` 命令来完成。常用格式为：

`save` 文件名 [变量名表] [-append][-ascii]

`load` 文件名 [变量名表] [-ascii]

1.2.4 MATLAB 常用数学函数

MATLAB 提供了许多数学函数，函数的自变量规定为矩阵变量，运算法则是将函数逐项作用于矩阵的元素上，因而运算的结果是一个与自变量同维数的矩阵。函数使用说明：

- 三角函数以弧度为单位计算。
- `abs` 函数可以求实数的绝对值、复数的模、字符串的 ASCII 码值。
- 用于取整的函数有 `fix`、`floor`、`ceil`、`round`，要注意它们的区别。
- `rem` 与 `mod` 函数的区别。`rem(x,y)`和 `mod(x,y)`要求 `x,y` 必须为相同大小的实矩阵或为标量。

1.2.5 MATLAB 矩阵生成

MATLAB 具有强大的矩阵运算和数据处理功能，对矩阵的处理必须遵从代数规则。

(1) 矩阵的建立

- 直接输入法

最简单的建立矩阵的方法是从键盘直接输入矩阵的元素。具体方法如下：将矩阵的元素用方括号括起来，按矩阵行的顺序输入各元素，同一行的各元素之间用空格或逗号分隔，不同行的元素之间用分号分隔。

- 利用 M 文件建立矩阵

对于比较大且比较复杂的矩阵，可以为它专门建立一个 M 文件。下面通过一个简单例子来说明如何利用 M 文件创建矩阵。

- 利用冒号表达式建立一个向量

冒号表达式可以产生一个行向量，一般格式是：`e1:e2:e3`，其中 `e1` 为初始值，`e2` 为步长，`e3` 为终止值。

在 MATLAB 中，还可以用 `linspace` 函数产生行向量。其调用格式为：`linspace(a,b,n)`，其中 `a` 和 `b` 是生成向量的第一个和最后一个元素，`n` 是元素总数。显然，`linspace(a,b,n)`与 `a:(b-a)/(n-1):b` 等价。

- 建立大矩阵

大矩阵可由方括号中的小矩阵或向量建立起来。

(2) 矩阵的拆分

- 矩阵元素

通过下标引用矩阵的元素，例如

```
A=[1,2,3;4,5,6];
```

```
A(1,2)
```

```
ans=2
```

采用矩阵元素的序号来引用矩阵元素。矩阵元素的序号就是相应元素在内存中的排列顺序。在 MATLAB 中，矩阵元素按列存储，先第一列，再第二列，依次类推。例如

```
A=[1,2,3;4,5,6];
```

```
A(3)
```

```
ans=2
```

显然，序号(Index)与下标(Subscript)是一一对应的，以 $m \times n$ 矩阵 `A` 为例，矩阵元素 `A(i,j)` 的序号为 $(j-1)*m+i$ 。其相互转换关系也可利用 `sub2ind` 和 `ind2sub` 函数求得。

- 矩阵拆分

1) 利用冒号表达式获得子矩阵

① `A(:,j)`表示取 `A` 矩阵的第 `j` 列全部元素；`A(i,:)`表示 `A` 矩阵第 `i` 行的全部元素；`A(i,j)`表示取 `A` 矩阵第 `i` 行、第 `j` 列的元素。

② $A(i:i+m,:)$ 表示取 A 矩阵第 $i \sim i+m$ 行的全部元素; $A(:,k:k+m)$ 表示取 A 矩阵第 $k \sim k+m$ 列的全部元素, $A(i:i+m,k:k+m)$ 表示取 A 矩阵第 $i \sim i+m$ 行内, 并在第 $k \sim k+m$ 列中的所有元素。

此外, 还可利用一般向量和 end 运算符来表示矩阵下标, 从而获得子矩阵。end 表示某一维的末尾元素下标。

2) 利用空矩阵删除矩阵的元素

在 MATLAB 中, 定义[]为空矩阵。给变量 X 赋空矩阵的语句为 $X=[]$ 。注意, $X=[]$ 与 clear X 不同, clear 是将 X 从工作空间中删除, 而空矩阵则存在于工作空间中, 只是维数为 0。

(3) 特殊矩阵

常用的产生通用特殊矩阵的函数有:

zeros: 产生全 0 矩阵(零矩阵)。

ones: 产生全 1 矩阵(幺矩阵)。

eye: 产生单位矩阵。

rand: 产生 0~1 间均匀分布的随机矩阵。

randn: 产生均值为 0, 方差为 1 的标准正态分布随机矩阵。

例 1-1 分别建立 3×3 、 3×2 和与矩阵 A 同样大小的零矩阵。

(1) 建立一个 3×3 零矩阵。

zeros(3)

(2) 建立一个 3×2 零矩阵。

zeros(3,2)

(3) 设 A 为 2×3 矩阵, 则可以用 zeros(size(A))建立一个与矩阵 A 同样大小零矩阵。

$A=[1\ 2\ 3;4\ 5\ 6]$; %产生一个 2×3 阶矩阵 A

zeros(size(A)) %产生一个与矩阵 A 同样大小的零矩阵

例 1-2 建立随机矩阵: (1) 在区间[20,50]内均匀分布的 5 阶随机矩阵。(2) 均值为 0.6、方差为 0.1 的 5 阶正态分布随机矩阵。

命令如下:

F 此外, 常用的函数还有 reshape(A,m,n), 它在矩阵总元素保持不变的前提下, 将矩阵 A 重新排成 $m \times n$ 的二维矩阵。

1.2.6 MATLAB 矩阵运算

(1) 算术运算

● 基本算术运算

MATLAB 的基本算术运算有: +(加)、-(减)、*(乘)、/(右除)、\ (左除)、^(乘方)。注意, 运算是在矩阵意义下进行的, 单个数据的算术运算只是一种特例。

1) 矩阵加减运算

假定有两个矩阵 A 和 B, 则可以由 $A+B$ 和 $A-B$ 实现矩阵的加减运算。运算规则是: 若 A 和 B 矩阵的维数相同, 则可以执行矩阵的加减运算, A 和 B 矩阵的相应元素相加减。如果 A 与 B 的维数不相同, 则 MATLAB 将给出错误信息, 提示用户两个矩阵的维数不匹配。

2) 矩阵乘法

假定有两个矩阵 A 和 B, 若 A 为 $m \times n$ 矩阵, B 为 $n \times p$ 矩阵, 则 $C=A*B$ 为 $m \times p$ 矩阵。

3) 矩阵除法

在 MATLAB 中, 有两种矩阵除法运算: \和/, 分别表示左除和右除。如果 A 矩阵是非奇异方阵, 则 $A \setminus B$ 和 B/A 运算可以实现。 $A \setminus B$ 等效于 A 的逆左乘 B 矩阵, 也就是 $\text{inv}(A)*B$,

而 B/A 等效于 A 矩阵的逆右乘 B 矩阵，也就是 $B \cdot \text{inv}(A)$ 。对于含有标量的运算，两种除法运算的结果相同，如 $3/4$ 和 $4/3$ 有相同的值，都等于 0.75 。又如，设 $a=[10.5, 25]$ ，则 $a/5=5 \backslash a=[2.1000 \ 5.0000]$ 。对于矩阵来说，左除和右除表示两种不同的除数矩阵和被除数矩阵的关系。对于矩阵运算，一般 $A \backslash B \neq B/A$ 。

4) 矩阵的乘方

一个矩阵的乘方运算可以表示成 A^x ，要求 A 为方阵， x 为标量。

● 点运算

在 MATLAB 中，有一种特殊的运算，因为其运算符是在有关算术运算符前面加点，所以叫点运算。点运算符有 $.*$ 、 $./$ 、 $.\$ 和 $.^$ 。两矩阵进行点运算是指它们的对应元素进行相关运算，要求两矩阵的维参数相同。

(2) 关系运算

MATLAB 提供了 6 种关系运算符： $<$ (小于)、 \leq (小于或等于)、 $>$ (大于)、 \geq (大于或等于)、 $==$ (等于)、 \sim (不等于)。它们的含义不难理解，但要注意其书写方法与数学中的不等式符号不尽相同。关系运算符的运算法则为：

- 当两个比较量是标量时，直接比较两数的大小。若关系成立，关系表达式结果为 1，否则为 0。
- 当参与比较的量是两个维数相同的矩阵时，比较是对两矩阵相同位置的元素按标量关系运算规则逐个进行，并给出元素比较结果。最终的关系运算的结果是一个维数与原矩阵相同的矩阵，它的元素由 0 或 1 组成。
- 当参与比较的一个是标量，而另一个是矩阵时，则把标量与矩阵的每一个元素按标量关系运算规则逐个比较，并给出元素比较结果。最终的关系运算的结果是一个维数与原矩阵相同的矩阵，它的元素由 0 或 1 组成。

例 1-3 产生 5 阶随机方阵 A ，其元素为 $[10, 90]$ 区间的随机整数，然后判断 A 的元素是否能被 3 整除。

(1) 生成 5 阶随机方阵 A 。

$A = \text{fix}((90-10+1) \cdot \text{rand}(5)) + 10$

(2) 判断 A 的元素是否可以被 3 整除。

$P = \text{rem}(A, 3) == 0$

其中， $\text{rem}(A, 3)$ 是矩阵 A 的每个元素除以 3 的余数矩阵。此时，0 被扩展为与 A 同维数的零矩阵， P 是进行等于($==$)比较的结果矩阵。

(3) 逻辑运算

MATLAB 提供了 3 种逻辑运算符： $\&$ (与)、 $|$ (或)和 \sim (非)。逻辑运算的运算法则为：

- 在逻辑运算中，确认非零元素为真，用 1 表示，零元素为假，用 0 表示。
- 设参与逻辑运算的是两个标量 a 和 b ，那么，
 $a \& b$ a, b 全为非零时，运算结果为 1，否则为 0。
 $a | b$ a, b 中只要有一个非零，运算结果为 1。
 $\sim a$ 当 a 是零时，运算结果为 1；当 a 非零时，运算结果为 0。
- 若参与逻辑运算的是两个同维矩阵，那么运算将对矩阵相同位置上的元素按标量规则逐个进行。最终运算结果是一个与原矩阵同维的矩阵，其元素由 1 或 0 组成。
- 若参与逻辑运算的一个是标量，一个是矩阵，那么运算将在标量与矩阵中的每个元素之间按标量规则逐个进行。最终运算结果是一个与矩阵同维的矩阵，其元素由 1 或 0 组成。
- 逻辑非是单目运算符，也服从矩阵运算规则。在算术、关系、逻辑运算中，算术运算优先级最高，逻辑运算优先级最低。

例 1-4 建立矩阵 A，然后找出大于 4 的元素的位置。

```
A=[4,-65,-54,0,6;56,0,67,-45,0]
find(A>4)
```

1.2.7 MATLAB 中的矩阵分析

(1) 矩阵的转置

转置运算符是单撇号'。

(2) 矩阵的旋转

利用函数 `rot90(A,k)` 将矩阵 A 旋转 90° 的 k 倍，当 k 为 1 时可省略。

(3) 矩阵的逆

对于一个方阵 A，如果存在一个与其同阶的方阵 B，使得： $A \cdot B = B \cdot A = I$ (I 为单位矩阵) 则称 B 为 A 的逆矩阵，当然，A 也是 B 的逆矩阵。求一个矩阵的逆是一件非常烦琐的工作，容易出错，但在 MATLAB 中，求一个矩阵的逆非常容易。求方阵 A 的逆矩阵可调用函数 `inv(A)`。

(4) 方阵的行列式

把一个方阵看作一个行列式，并对其按行列式的规则求值，这个值就称为矩阵所对应的行列式的值。在 MATLAB 中，求方阵 A 所对应的行列式的值的函数是 `det(A)`。

(5) 矩阵的秩

矩阵线性无关的行数与列数称为矩阵的秩。在 MATLAB 中，求矩阵秩的函数是 `rank(A)`。

1.3 MATLAB 程序设计

1.3.1 M 文件

用 MATLAB 语言编写的程序，称为 M 文件。M 文件可以根据调用方式的不同分为两类：命令文件(Script File)和函数文件(Function File)。M 文件是一个文本文件，它可以用任何编辑程序来建立和编辑，而一般常用且最为方便的是使用 MATLAB 提供的文本编辑器。

(1) 建立新的 M 文件

启动 MATLAB 文本编辑器有 3 种方法：

- 菜单操作。从 MATLAB 主窗口的 File 菜单中选择 New 菜单项，再选择 M-file 命令，屏幕上将出现 MATLAB 文本编辑器窗口。
- 命令操作。在 MATLAB 命令窗口输入命令 `edit`，启动 MATLAB 文本编辑器后，输入 M 文件的内容并存盘。
- 命令按钮操作。单击 MATLAB 主窗口工具栏上的 New M-File 命令按钮，启动 MATLAB 文本编辑器后，输入 M 文件的内容并存盘。

(2) 打开已有的 M 文件

打开已有的 M 文件，也有 3 种方法：

- 菜单操作。从 MATLAB 主窗口的 File 菜单中选择 Open 命令，则屏幕出现 Open 对话框，在 Open 对话框中选中所需打开的 M 文件。在文档窗口可以对打开的 M 文件进行编辑修改，编辑完成后，将 M 文件存盘。
- 命令操作。在 MATLAB 命令窗口输入命令：`edit 文件名`，则打开指定的 M 文件。
- 命令按钮操作。单击 MATLAB 主窗口工具栏上的 Open File 命令按钮，再从弹出的对话框中选择所需打开的 M 文件。

例 1-5 分别建立命令文件和函数文件，将华氏温度 f 转换为摄氏温度 c。

程序 1:

首先建立命令文件并以文件名 f2c.m 存盘。

```
clear;          %清除工作空间中的变量
f=input('Input Fahrenheit temperature: ');
c=5*(f-32)/9
```

然后在 MATLAB 的命令窗口中输入 f2c, 将会执行该命令文件, 执行情况为:

```
Input Fahrenheit temperature: 73
c =
    22.7778
```

程序 2:

首先建立函数文件 f2c.m。

```
function c=f2c(f)
c=5*(f-32)/9
```

然后在 MATLAB 的命令窗口调用该函数文件。

```
clear;
y=input('Input Fahrenheit temperature: ');
x=f2c(y)
```

输出情况为:

```
Input Fahrenheit temperature: 70
c =
    21.1111
x =
    21.1111
```

1.3.2 程序控制结构

(1) 顺序结构

- 数据的输入, 该函数的调用格式为: `A=input(提示信息, 选项)`;

例 1-5 想输入一个人的姓名, 可采用命令:

```
xm=input('What''s your name?','s'); %采用's'选项, 则允许用户输入一个字符串。
```

- 数据的输出, 其调用格式为: `disp(输出项)`;
- 程序的暂停可以使用 `pause` 函数, 其调用格式为: `pause(延迟秒数)`; 如果省略延迟时间, 直接使用 `pause`, 则将暂停程序, 直到用户按任一键后程序继续执行。若要强行中止程序的运行可使用 `Ctrl+C` 命令。

(2) 选择结构

- if 语句

在 MATLAB 中, if 语句有 3 种格式。

1) 单分支 if 语句:

```
if 条件
    语句组
end
```

当条件成立时, 则执行语句组, 执行完之后继续执行 if 语句的后继语句, 若条件不成立, 则直接执行 if 语句的后继语句。

2) 双分支 if 语句:

```
if 条件
    语句组 1
else
```

语句组 2

end

当条件成立时，执行语句组 1，否则执行语句组 2，语句组 1 或语句组 2 执行后，再执行 if 语句的后继语句。

例 1-6 计算分段函数的值。

程序如下：

```
x=input('请输入 x 的值:');
if x<=0
    y=(x+sqrt(pi))/exp(2);
else
    y=log(x+sqrt(1+x*x))/2;
end
y
```

3) 多分支 if 语句：

```
if 条件 1
    语句组 1
elseif 条件 2
    语句组 2
.....
elseif 条件 m
    语句组 m
else
    语句组 n
end
```

语句用于实现多分支选择结构。

例 1-7 输入一个字符，若为大写字母，则输出其对应的小写字母；若为小写字母，则输出其对应的大写字母；若为数字字符则输出其对应的数值，若为其他字符则原样输出。

```
c=input('请输入一个字符','s');
if c>='A' & c<='Z'
    disp(setstr(abs(c)+abs('a')-abs('A')));
elseif c>='a' & c<='z'
    disp(setstr(abs(c)-abs('a')+abs('A')));
elseif c>='0' & c<='9'
    disp(abs(c)-abs('0'));
else
    disp(c);
end
```

● switch 语句

switch 语句根据表达式的取值不同，分别执行不同的语句，其语句格式为：

```
switch 表达式
case 表达式 1
    语句组 1
case 表达式 2
    语句组 2
```

```

.....
case 表达式 m
    语句组 m
otherwise
    语句组 n
end

```

当表达式的值等于表达式 1 的值时，执行语句组 1，当表达式的值等于表达式 2 的值时，执行语句组 2，...，当表达式的值等于表达式 m 的值时，执行语句组 m，当表达式的值不等于 case 所列的表达式的值时，执行语句组 n。当任意一个分支的语句执行完后，直接执行 switch 语句的下一句。

例 1-8 某商场对顾客所购买的商品实行打折销售，标准如下(商品价格用 price 来表示)：

price<200	没有折扣
200≤price<500	3%折扣
500≤price<1000	5%折扣
1000≤price<2500	8%折扣
2500≤price<5000	10%折扣
5000≤price	14%折扣

输入所售商品的价格，求其实际销售价格。

程序如下：

```

price=input('请输入商品价格');
switch fix(price/100)
    case {0,1}                %价格小于 200
        rate=0;
    case {2,3,4}              %价格大于等于 200 但小于 500
        rate=3/100;
    case num2cell(5:9)         %价格大于等于 500 但小于 1000
        rate=5/100;
    case num2cell(10:24)       %价格大于等于 1000 但小于 2500
        rate=8/100;
    case num2cell(25:49)       %价格大于等于 2500 但小于 5000
        rate=10/100;
    otherwise                  %价格大于等于 5000
        rate=14/100;
end
price=price*(1-rate) %输出商品实际销售价格

```

- try 语句
语句格式为：

```

try
    语句组 1
catch
    语句组 2
end

```

try 语句先试探性执行语句组 1，如果语句组 1 在执行过程中出现错误，则将错误信息赋给保留的 lasterr 变量，并转去执行语句组 2。

例 1-9 矩阵乘法运算要求两矩阵的维数相容，否则会出错。先求两矩阵的乘积，若出错，则自动转去求两矩阵的点乘。

程序如下：

```
A=[1,2,3;4,5,6]; B=[7,8,9;10,11,12];
try
    C=A*B;
catch
    C=A.*B;
end
C
lasterr          %显示出错原因
```

(3) 循环结构

- for 语句

for 语句的格式为：

```
for 循环变量=表达式 1:表达式 2:表达式 3
    循环体语句
```

```
end
```

其中表达式 1 的值为循环变量的初值，表达式 2 的值为步长，表达式 3 的值为循环变量的终值。步长为 1 时，表达式 2 可以省略。

例 1-10 一个三位整数各位数字的立方和等于该数本身则称该数为水仙花数。输出全部水仙花数。

程序如下：

```
for m=100:999
    m1=fix(m/100);          %求 m 的百位数字
    m2=rem(fix(m/10),10);   %求 m 的十位数字
    m3=rem(m,10);          %求 m 的个位数字
    if m==m1*m1*m1+m2*m2*m2+m3*m3*m3
        disp(m)
    end
end
```

- while 语句

while 语句的一般格式为：

```
while (条件)
    循环体语句
```

```
end
```

其执行过程为：若条件成立，则执行循环体语句，执行后再判断条件是否成立，如果不成立则跳出循环。

例 1-11 从键盘输入若干个数，当输入 0 时结束输入，求这些数的平均值和它们之和。

程序如下：

```
sum=0;
cnt=0;
val=input('Enter a number (end in 0):');
while (val~=0)
    sum=sum+val;
```

```

    cnt=cnt+1;
    val=input('Enter a number (end in 0):');
end
if (cnt > 0)
    sum
    mean=sum/cnt
end

```

- **break 语句和 continue 语句**

break 语句用于终止循环的执行。当在循环体内执行到该语句时，程序将跳出循环，继续执行循环语句的下一语句。

continue 语句控制跳过循环体中的某些语句。当在循环体内执行到该语句时，程序将跳过循环体中所有剩下的语句，继续下一次循环。

例 1-12 求[100, 200]之间第一个能被 21 整除的整数。

程序如下：

```

for n=100:200
    if rem(n,21)~=0
        continue
    end
    break
end

```

(4) 函数文件

- **函数文件的基本结构**

函数文件由 **function** 语句引导，其基本结构为：

```

function 输出形参表=函数名(输入形参表)
注释说明部分
函数体语句

```

其中以 **function** 开头的一行为引导行，表示该 M 文件是一个函数文件。函数名的命名规则与变量名相同。输入形参为函数的输入参数，输出形参为函数的输出参数。当输出形参多于一个时，则应该用方括号括起来。

- **函数调用**

函数调用的一般格式是：

```

[输出实参表]=函数名(输入实参表)

```

要注意的是，函数调用时各实参出现的顺序、个数，应与函数定义时形参的顺序、个数一致，否则会出错。函数调用时，先将实参传递给相应的形参，从而实现参数传递，然后再执行函数的功能。

例 1-13 利用函数的递归调用，求 $n!$ 。

$n!$ 本身就是以递归的形式定义的：显然，求 $n!$ 需要求 $(n-1)!$ ，这时可采用递归调用。递归调用函数文件 **factor.m** 如下：

```

function f=factor(n)
if n<=1
    f=1;
else
    f=factor(n-1)*n;    %递归调用求(n-1)!
end

```


1.4 计算结果可视化

MATLAB 在数据可视化方面的表现能力很强。它的图形处理能力不仅功能强大，而且充分考虑了不同层次用户的不同需求，系统具有两个层次的绘图指令：一个层次是直接对图形句柄进行操作的底层绘图指令；另一层次是在底层指令基础上建立的高层绘图指令。常用的 MATLAB 绘图语句有 figure、plot、subplot、stem 等，图形修饰语具有 title、axis、text 等。

(1) figure 语句

figure 有两种用法。当只有一句 figure 命令时，程序会创建一个新的图形窗口，并返回一个整数型的窗口编号。当采用 figure(n) 时，表示将第 n 个图形窗口作为当前的图形窗口，将其显示在所有窗口的最前面。如果该图形窗口不存在，则新建一个窗口，并赋以编号 n。

(2) plot 语句

线形绘图函数。用法为 plot(x,y,'s')。参数 x 为横轴变量，y 为纵轴变量，s 用以控制图形的基本特征如颜色、粗细等，通常可以省略，常用方法如表 2-1 所示。

表 2-1 plot 命令的参数及其含义

参数	含义	参数	含义	参数	含义
y	黄色	.	点	-	实线
m	紫色	o	圆	:	虚线
c	青色	x	打叉	-.	点划线
r	红色	+	加号	--	破折线
g	绿色	*	星号	^	向上三角形
b	蓝色	s	正方形	<	向左三角形
w	白色	d	菱形	>	向右三角形
k	黑色	v	向下三角形	p	五角星形

(3) subplot 语句

subplot(m,n,i) 是分割显示图形窗口命令，它把一个图形窗口分为 m 行 n 列共 $m \times n$ 个小窗口，并指定第 i 个小窗口为当前窗口。

(4) 二维统计分析图

在 MATLAB 中，二维统计分析图形很多，常见的有条形图、阶梯图、杆图和填充图等，所采用的函数分别是：

```
bar(x,y,选项)
stairs(x,y,选项)
stem(x,y,选项)
fill(x1,y1,选项 1,x2,y2,选项 2,...)
```

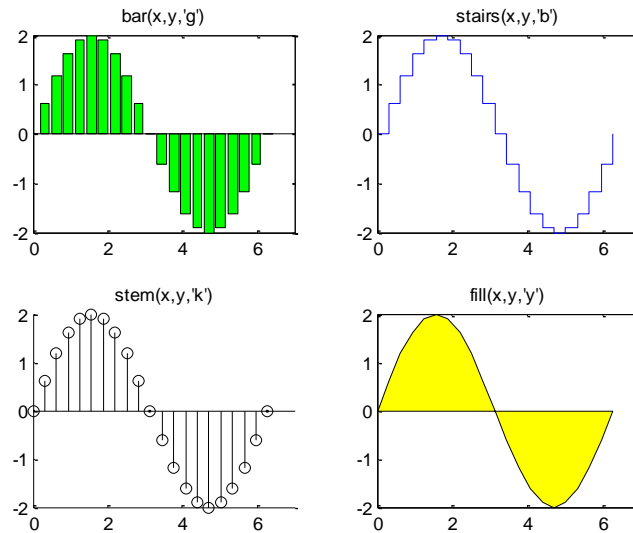
例 2-1 分别以条形图、阶梯图、杆图和填充图形式绘制曲线 $y=2\sin(x)$ 。

程序如下：

```
x=0:pi/10:2*pi;
y=2*sin(x);
subplot(2,2,1);bar(x,y,'g');
title('bar(x,y,\'g\')');axis([0,7,-2,2]);
subplot(2,2,2);stairs(x,y,'b');
title('stairs(x,y,\'b\')');axis([0,7,-2,2]);
```

```
subplot(2,2,3);stem(x,y,'k');
title('stem(x,y,''k'')');axis([0,7,-2,2]);
subplot(2,2,4);fill(x,y,'y');
title('fill(x,y,''y'')');axis([0,7,-2,2]);
```

仿真结果：



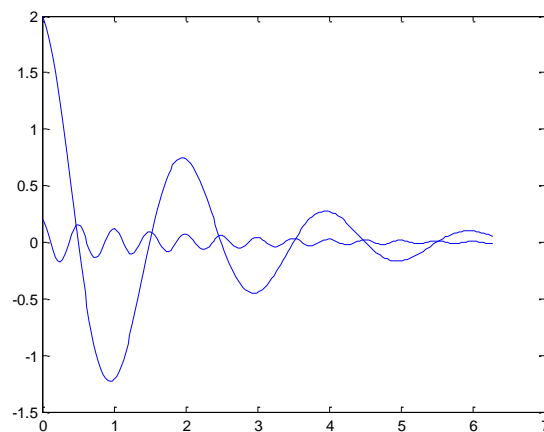
(5) 图形保持

`hold on/off` 命令控制是保持原有图形还是刷新原有图形，不带参数的 `hold` 命令在两种状态之间进行切换。

例 2-2 采用图形保持，在同一坐标内绘制曲线 $y_1=0.2e^{-0.5x}\cos(4\pi x)$ 和 $y_2=2e^{-0.5x}\cos(\pi x)$ 。程序如下：

```
x=0:pi/100:2*pi;
y1=0.2*exp(-0.5*x).*cos(4*pi*x);
plot(x,y1)
hold on
y2=2*exp(-0.5*x).*cos(pi*x);
plot(x,y2);
hold off
```

仿真结果：



(6) 绘图修饰命令

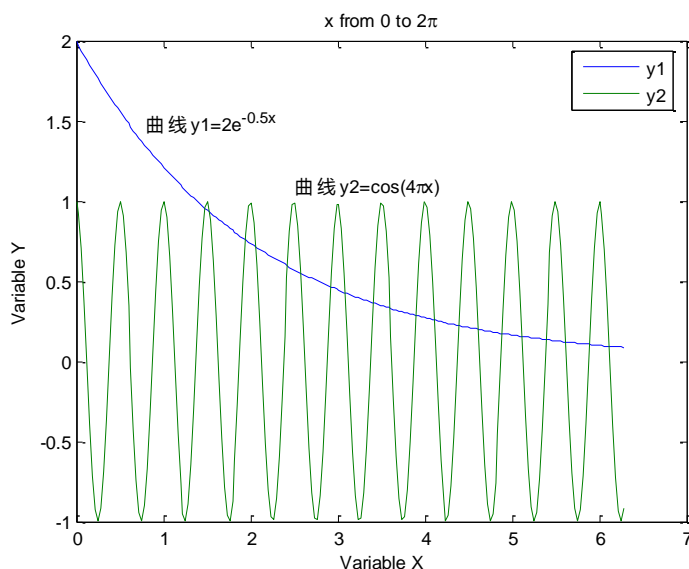
`title(图形名称)`
`xlabel(x 轴说明)`
`ylabel(y 轴说明)`
`text(x,y,图形说明)`
`legend(图例 1,图例 2,...)`

例 2-3 在 $0 \leq x \leq 2\pi$ 区间内，绘制曲线 $y_1=2e^{-0.5x}$ 和 $y_2=\cos(4\pi x)$ ，并给图形添加图形标注。

程序如下：

```
x=0:pi/100:2*pi;y1=2*exp(-0.5*x);y2=cos(4*pi*x);
plot(x,y1,x,y2)
title('x from 0 to 2{\pi}');           %加图形标题
xlabel('Variable X');                 %加 X 轴说明
ylabel('Variable Y');                 %加 Y 轴说明
text(0.8,1.5,'曲线 y1=2e^{-0.5x}'); %在指定位置添加图形说明
text(2.5,1.1,'曲线 y2=cos(4{\pi}x)');
legend('y1','y2')                    %加图例
```

仿真结果：



(7) 坐标控制

`axis` 函数的调用格式为：

`axis([xmin xmax ymin ymax zmin zmax])`

`axis` 函数功能丰富，常用的格式还有：

`axis equal`：纵、横坐标轴采用等长刻度。

`axis square`：产生正方形坐标系(缺省为矩形)。

`axis auto`：使用缺省设置。

`axis off`：取消坐标轴。

`axis on`：显示坐标轴。

给坐标加网格线用 `grid` 命令来控制。`grid on/off` 命令控制是画还是不画网格线，不带参数的 `grid` 命令在两种状态之间进行切换。

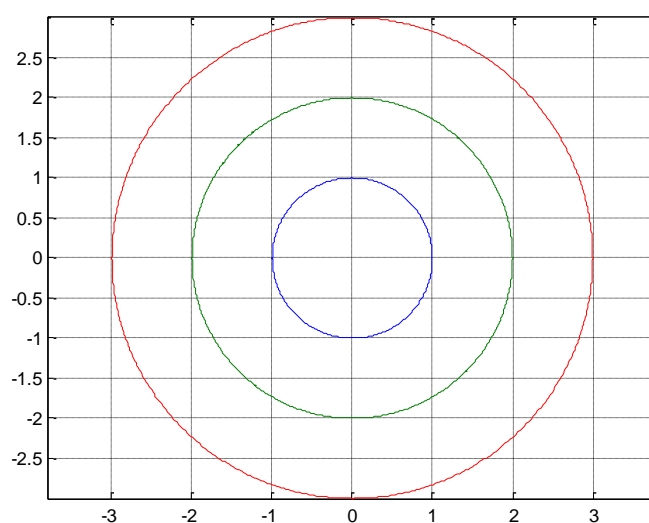
给坐标加边框用 `box` 命令来控制。`box on/off` 命令控制是加还是不加边框线，不带参数

的 `box` 命令在两种状态之间进行切换。

例 2-4 在同一坐标中，可以绘制 3 个同心圆，并加坐标控制。

程序如下：

```
t=0:0.01:2*pi;  
x=exp(i*t);  
y=[x;2*x;3*x]';  
plot(y)  
grid on;           %加网格线  
box on;            %加坐标边框  
axis equal         %坐标轴采用等刻度  
仿真结果：
```



1.5 确知信号分析

1.5.1 周期信号的傅里叶级数

(1) 基本原理

若一周期信号 $f(t) = f(t + kT)$ ，其中 k 为整数， T 成为信号的周期。若周期信号在一个周期内可积，则可通过傅立叶级数对该信号进行展开。其傅立叶展开式如 (2-1) 式所示：

$$f(t) = \sum_{n=-\infty}^{\infty} F_n e^{j2\pi n f_s t} \quad (2-1)$$

其中, $F_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-j2\pi n f_s t} dt$, T 为信号周期; $f_s = 1/T$ 为信号的基波; F_n 为

傅立叶展开系数, 其物理意义为频率分量 $n f_s$ 的幅度和相位。

式 2-1 表明: 信号可以展开成一系列频率为 $f_s = 1/T$ 的整数倍的正弦、余弦信号的加权叠加, 其中相应频率分量的加权系数即为 F_n , 因此可以用周期信号的傅立叶展开来重构该周期信号, 其逼近程度与展开式的项数有关。

(2) 举例

设周期信号一个周期的波形为 $f(t) = \begin{cases} 1, & 0 \leq t \leq T/2 \\ -1, & T/2 < t \leq T \end{cases}$, 求该信号傅里叶级数展开式,

并用 MATLAB 画出傅里叶级数展开后的波形, 并通过展开式项数的变化考察其对 $f(t)$ 的逼近程度, 考察其物理意义。

解:

$$\begin{aligned} F_n &= \frac{1}{T} \int_0^T f(t) e^{-j2\pi n f_s t} dt \\ &= \frac{1}{T} \left(\int_0^{T/2} e^{-j2\pi n f_s t} dt - \int_{T/2}^T e^{-j2\pi n f_s t} dt \right) \\ &= \frac{1}{T} \left(\frac{e^{-j\pi n} - 1}{-j2\pi n f_s} - \frac{1 - e^{-j\pi n}}{-j2\pi n f_s} \right) \\ &= \text{sinc}(n/2) e^{-jn\pi/2} \end{aligned}$$

注: $\text{sinc}(x) = \sin \pi x / \pi x = \text{sa}(\pi x)$

源代码:

```
clear all;
```

```
N=20;%取展开式的项数为 2N+1 项
```

```
%可以改为N=input('input N:')
```

```
T=1;%周期为 1
```

```
fs=1/T;
```

```
N_sample=128;%为了画波形, 设置每个周期的采样点数
```

```
dt=1/N_sample;%时间分辨率
```

```
t=0:dt:10*T-dt;%取 10 个周期
```

```
n=-N:N;
```

```
Fn=sinc(n/2).*exp(-j*n*pi/2);%求傅立叶系数
```

```
Fn(N+1)=0;%当 n=0 时, 代入 Fn 得 Fn=0, 由于数组的序号是从 1 开始的, 即 n=-N 时对%  
应 Fn(1), n=0 时对%应 Fn(n+1), 即 n=N 时对%应 Fn(2N+1)
```

```
ft=zeros(1,length(t));%建立一个全零数组, 其长度和原始信号长度相同, 用来存放由傅里  
叶%展开恢复的信号
```

```
for m=-N:N;%一共 2N+1 项累加。
```

```
ft=ft+Fn(m+N+1)*exp(j*2*pi*m*fs*t);%Fn 是一个数组, 而 MATLAB 中数组中元素的序%
```

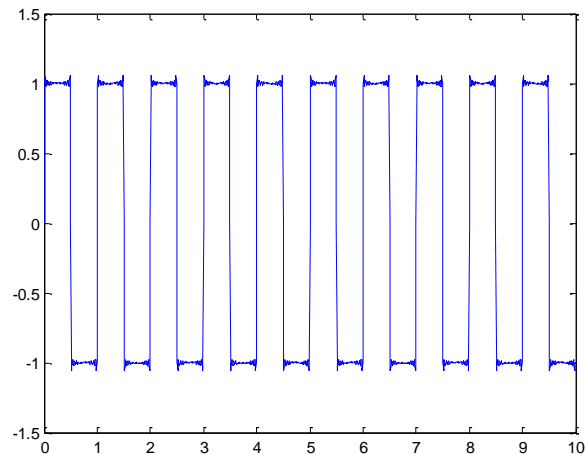
号是从 1 开始的, 故 F_n 序号是从 1 开始的, 到 $2N+1$ 结束, 该语句中体现为 $F_n(m+N+1)$
%而当 $n=0$ 时, $F_n=0$, 在数组中的位置为第 $N+1$ 个元素, 故令 $F_n(N+1)=0$

end

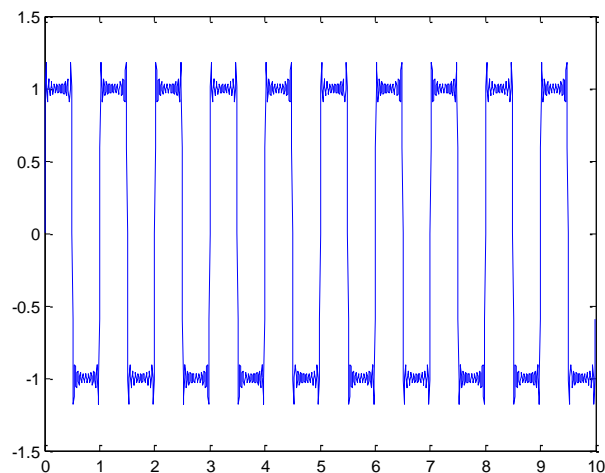
plot(t,ft)

仿真结果:

$N=100$ 时,



$N=20$ 时,



可以看出: 用周期信号的傅立叶展开来重构该周期信号, 其逼近程度与展开式的项数有关。

1.5.2 信号的傅里叶变换及其反变换

(1) 基本原理

对于非周期信号 $s(t)$, 满足绝对可积的条件下, 可利用傅里叶变换对其进行频域分析。

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi ft} dt, \quad s(t) = \int_{-\infty}^{\infty} S(f) e^{j2\pi ft} df$$

其中, $S(f)$ 称为信号 $s(t)$ 傅里叶变换, 表示了该信号的频谱特性。

在数字信号处理中, 需要利用离散傅立叶变换 (DFT) 计算信号的傅里叶变换, 现在

考察一下信号 $s(t)$ 的傅里叶变换与其离散傅立叶变换之间的关系。

将信号 $s(t)$ 按照时域均匀抽样定理进行等间隔抽样后，得到序列 $\{s_n, n=0,1,2,\dots,N-1\}$ ， $s_n = s(n\Delta t)$ ，其中， Δt 为抽样间隔，则由数字信号处理的知识可知，序列 s_n 的离散傅立叶变换为

$$S_k = \sum_{n=0}^{N-1} s_n e^{-j\frac{2\pi}{N}nk} \quad (k=0,1,2,\dots,N-1)$$

其中， N 为采样点数。

而 $s(t)$ 在一段时间 $[0, T]$ 内的傅立叶变换为

$$\begin{aligned} S(f) &= \int_0^T s(t) e^{-j2\pi ft} dt \\ &= \lim_{N \rightarrow \infty} \sum_{n=0}^{N-1} s(n\Delta t) e^{-j2\pi fn\Delta t} \Delta t \\ &\stackrel{\Delta t = T/N}{=} \lim_{N \rightarrow \infty} \frac{T}{N} \sum_{n=0}^{N-1} s(n\Delta t) e^{-j\frac{2\pi}{N}nfT} \\ &\stackrel{\text{注意到 } s(n\Delta t) = s_n}{=} \lim_{N \rightarrow \infty} \frac{T}{N} \sum_{n=0}^{N-1} s_n e^{-j\frac{2\pi}{N}nfT} \end{aligned}$$

得到 $s(t)$ 在一段时间 $[0, T]$ 内的傅立叶变换是连续谱 $S(f)$ ，而对 $s(t)$ 进行离散傅立叶变换得到的是离散谱 S_k ，为了比较它们之间的关系，对 $S(f)$ 也进行等间隔抽样，且抽样间隔为 $\Delta f = 1/T$ ，即其频率分辨率，则在频率范围 $[0, (N-1)\Delta f]$ 内，

$$\begin{aligned} S(f) &= S(k\Delta f) = \lim_{N \rightarrow \infty} \frac{T}{N} \sum_{n=0}^{N-1} s_n e^{-j\frac{2\pi}{N}nfT} \\ &= \lim_{N \rightarrow \infty} \frac{T}{N} \sum_{n=0}^{N-1} s_n e^{-j\frac{2\pi}{N}nk} \\ &= \lim_{N \rightarrow \infty} \frac{T}{N} S_k \quad (k=0,1,2,\dots,N-1) \end{aligned}$$

可以看到， $s(t)$ 的离散傅里叶变换与 $s(t)$ 在一段时间 $[0, T]$ 内的傅立叶变换 $S(f)$ 的抽样 $S(k\Delta f)$ 成正比。由于 N 点离散傅里叶变换具有 $S_k = S_{k+mN}$ 的性质，故信号 $s(t)$ 连续谱的负半轴部分可以通过对 S_k 的平移得到。

需要注意的是信号 $s(t)$ 的离散傅立叶变换只和信号 $s(t)$ 在一段时间 $[0, T]$ 内的傅立叶变换有关，而由公式 2-1， $s(t)$ 的频谱是在时间 $[-\infty, \infty]$ 上得到的。所以上述计算所得到的并不是真正的信号频谱，而是信号加了一个时间窗后的频谱。当信号 $s(t)$ 是随时间衰减的或是时限信号，只要时间窗足够长，可以通过这种方法获得信号的近似频谱。因此，用

DFT 计算的信号频谱精度依赖于信号、抽样的时间间隔和时间窗的大小。一般情况下，对于时限信号，在抽样时间间隔小，即抽样频率高的情况下能获得较为精确的信号频谱。

计算信号的离散傅里叶变换在数字信号处理中有一种高效算法，即快速傅里叶变换 FFT，Matlab 中也有专门的工具，下面简要介绍：

`fft(x)`, `x` 是离散信号，或对模拟信号取样后的离散值。

`ifft(x)`, `x` 是对信号进行快速傅里叶变换后的离散谱。

源代码一：

利用 `fft,fftshift` 定义函数 T2F 计算信号的傅立叶变换

`function [f,sf]=T2F(t,st)%`该子函数需要两个参数 `t` 和 `st`。

`%t—`离散时间；`st—`离散信号

`dt=t(2)-t(1);%` 时间分辨率

`T=t(end);`

`df=1/T;%`频率分辨率

`N=length(st);%`离散傅立叶变换长度

`f=-N/2*df:df:N/2*df-df;%`设定频谱区间，注意要关于原点对称，共有 `N` 个点，包括 0 点，%故要减去一个 `df`

`sf=fft(st);`

`sf=T/N*fftshift(sf);%`信号的频谱与离散傅立叶变换之间的关系，`fftshift(x)`是将信号%的频谱 `x` 进行移位，与原点对称。

源代码二：

利用 `ifft,fftshift` 定义函数 T2F 计算信号的傅立叶反变换

`function [t,st]= F2T (f,sf)`

`%f` 离散的频率；`sf—`信号的频谱

`df=f(2)-f(1);%`频率分辨率

`Fmx=f(end)-f(1)+df;%`频率区间长度

`dt=1/Fmx;%`已知频率区间长度时，求时间分辨率，由前面频率分辨率公式 $\Delta f=df=1/T$,

`%T=dt*N`，得到 $\Delta f=df=1/(dt*N)$ ，故 `dt=1/(df*N)=1/Fmx`，即时间分辨率

`N=length(sf);`

`T=dt*N;%`信号持续时间

`t=0:dt:T-dt;`

`%`离散傅立叶反变换，是 T2F 的逆过程

`sff=fftshift(sf);%`把对称的频谱进行平移，平移后同 T2F 中的 `sf`

`st=Fmx*ifft(sff); %由于 T2F 中求信号频谱在 DFT 基础上乘了一个因子 T/N，反变换求信号时要乘以其倒数即 N/T=1/dt，正好等于 Fmx。`

(2) 举例

设非周期信号 $s(t) = \begin{cases} 1, & 0 \leq t \leq T/2 \\ -1, & T/2 < t \leq T \end{cases}$ ，求该信号的傅里叶变换，用 MATLAB 画出傅里

叶变换后的频谱，并对频谱进行反变换，画出 $s(t)$ 的波形。

解：

$$\begin{aligned}
 S(f) &= \int_0^{\frac{T}{2}} e^{-j2\pi ft} dt - \int_{\frac{T}{2}}^T e^{-j2\pi ft} dt \\
 &= \frac{e^{-j\pi fT} - 1}{-j2\pi f} - \frac{e^{-j2\pi fT} - e^{-j\pi fT}}{-j2\pi f} \\
 &= \frac{1 - e^{-j\pi fT}}{j2\pi f} - \frac{e^{-j\pi fT} - e^{-j2\pi fT}}{j2\pi f} = \frac{(1 - e^{-j\pi fT})^2}{j2\pi f} \\
 &= \frac{\left[e^{-j\frac{1}{2}\pi fT} \left(e^{j\frac{1}{2}\pi fT} - e^{-j\frac{1}{2}\pi fT} \right) \right]^2}{j2\pi f} \\
 &= e^{-j\pi fT} \frac{-4 \sin^2\left(\frac{1}{2}\pi fT\right)}{j2\pi f} = e^{-j\pi fT} \frac{\sin^2\left(\frac{1}{2}\pi fT\right)}{\left(\frac{1}{2}\pi fT\right)^2} j \frac{\pi f}{2} T^2 \\
 &= j \frac{\pi f}{2} T^2 e^{-j\pi fT} \sin^2(fT/2)
 \end{aligned}$$

主程序：

`clear all`

`T=1;`

`N_sample=128;%为了画波形，设置每个周期的采样点数`

`dt=1/N_sample;%时间分辨率`

`t=0:dt:T-dt;`

`st=[ones(1,N_sample/2),-ones(1,N_sample/2)];%依据 T 将信号离散化`

`subplot(311);plot(t,st);axis([0 1 -2 2]);xlabel('t');ylabel('s(t)');`

`subplot(312);`

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

`plot(f,abs(sf));hold on;%画出 sf 的幅度谱，不含相位`

`axis([-10 10 0 1]);`

`xlabel('f');ylabel('|S(f)|');`

`sff=T^2*j*pi*f*0.5.*exp(-j*2*pi*f*T).*sinc(f*T*0.5).*sinc(f*T*0.5);%依据傅里叶变换求%`

信号频谱

```
plot(f,abs(sff),'r-')
```

```
[t,st]= F2T (f,sf);%进行离散傅立叶反变换，求原始信号
```

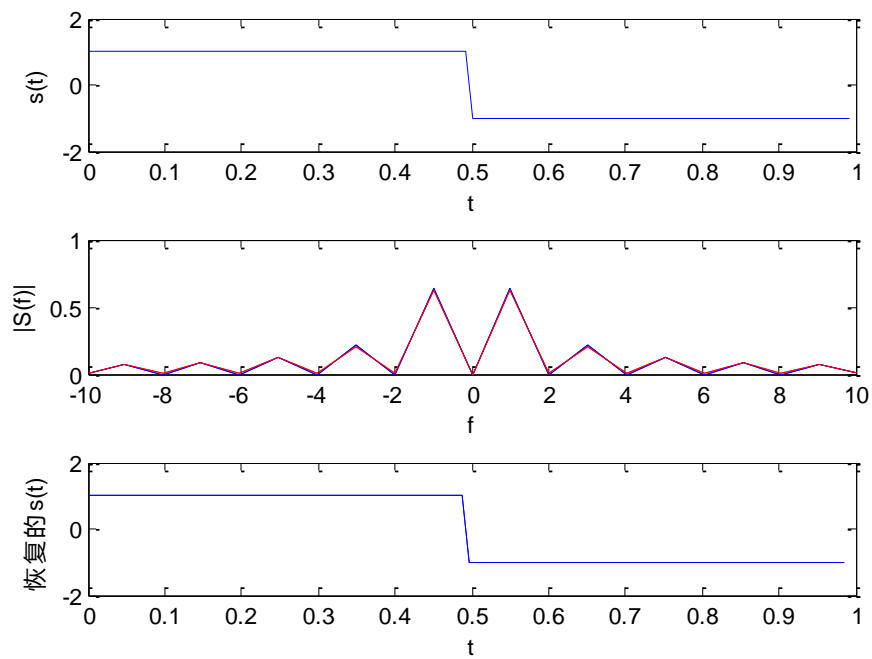
```
subplot(313);
```

```
axis([0 1 -2 2]);
```

```
xlabel('t');ylabel('恢复的 s(t)');
```

```
plot(t,st);hold on;
```

仿真结果：



实验二（2 小班戴旭初老师） 连续波 MATLAB 实现

本章目标

- 掌握线性模拟调制信号的波形及产生方法；
- 掌握线性模拟调制信号的频谱特点；
- 掌握线性模拟调制信号的解调方法；
- 掌握线性模拟调制系统的 MATLAB 仿真实现。

2.1 模拟调制

模拟调制包括幅度调制（DSB，SSB，AM）和相角调制（频率和相位调制），其中幅度调制（线性调制）是正弦载波的幅度随着调制信号而改变的调制方案。若调制信号 $m(t)$ ，频谱为 $M(f)$ ，带宽为 $B = f_m$ ，SSB 调制的带宽为 B ，DSB 和 AM 调制的带宽为 $2B$ 。VSB-AM 的带宽在 $B-2B$ 区间内。

AM 调制与 DSB 调制在许多方面十分相似，唯一区别在于 AM 调制用 $[A + m(t)]$ 代替 DSB 的 $m(t)$ 。AM 调制信号可表示为

$$s_{AM}(t) = [A + m(t)] \cos 2\pi f_c t$$

该调制信号的频谱可表示为

$$S_{AM}(f) = \frac{A_c}{2} [\delta(f + f_c) + \delta(f - f_c) + aM(f + f_c) + aM(f - f_c)]$$

AM 调制器模型如图 3-1 所示

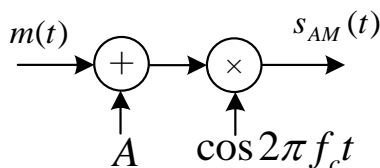


图 3-1 AM 调制器模型

假设被调信号 $m(t)$ 是零均值信号，则调制信号的功率为 $\overline{m^2(t)}$ ，则

$$P_{AM} = \frac{A^2 + \overline{m^2(t)}}{2} = P_c + P_s$$

为调制信号平均功率， P_s 为边带功率， P_c 为载波功率。调制信号中用于发送信息的功率和总功率的比值称为调制效率，

$$\eta_{AM} = \frac{P_s}{P_c + P_s} = \frac{\overline{m^2(t)}/2}{A^2/2 + \overline{m^2(t)}/2}$$

解调器输入信噪比定义为

$$r_i = \frac{S_i}{N_i} = \frac{S_i}{n_0 B_{BPF}},$$

B_{BPF} 为理想带通滤波器带宽，在理想信道中，当带通滤波器幅频特性为常数 1 时，

$$\frac{S_i}{N_i} = \frac{P_s}{n_0 B_{BPF}}, \quad n_0 = \frac{P_s}{r_i B_{BPF}},$$

即可由解调器输入信噪比计算出信道噪声的单边带功率谱密度，而当系统抽样速率为 f_s 时，产生的高斯白噪声带宽为 $f_s/2$ ，由此可计算出信道中高斯白噪声的平均功率，即方差 $\sigma_n^2 = n_0 f_s/2$ ，可以产生信道中所叠加的高斯白噪声。

2.2 信道加性高斯白噪声

信道中加性高斯白噪声功率由于其均值为 0，故其方差 $\sigma^2 = n_0 B$ 即为其平均功率，其中 n_0 为高斯白噪声单边带功率谱密度； B 为信道带宽。在图 1 中，信道噪声的功率谱密度图可以看出，当 $f_s=16\text{Hz}$ 时，叠加于信道的高斯噪声带宽为 8Hz，信道中的加性高斯白噪声通过带宽为 2Hz，幅度为 1 的理想带通滤波器后，输出的窄带噪声的平均功率即为相干解调器输入噪声的平均功率 N_i ，其功率谱密度不变，仍为 n_0 。可以看出，

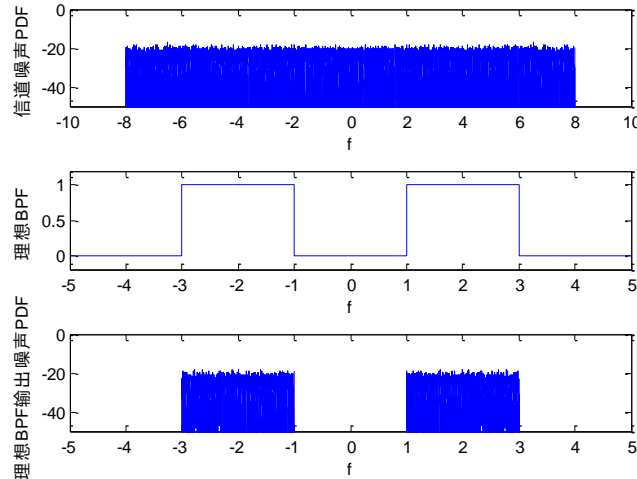


图 1 系统采样频率为 16Hz 时的噪声

N_i 与信道加性高斯白噪声功率 σ^2 之间有一定的关系，其共同点是其功率谱密度相同，从图 1 也可以观察出来，为 -20dB。因此，在理想通信系统中，利用已给解调器输入信噪比及已调信号功率和带宽，可以计算出 n_0 ，从而算出信道加性高斯白噪声的方差，由于其均值为 0，故该方差为其平均功率，利用它可以生成信道加性高斯白噪声。转换关系为：

$$r = \frac{S_i}{N_i} = \frac{S_i}{n_0 B_{BPF}}, \quad n_0 = \frac{S_i}{r B_{BPF}}, \quad \text{因此 } \sigma^2 = \frac{n_0 f_s}{2} = \frac{S_i f_s}{2 r B_{BPF}},$$

$$\sigma = \sqrt{\frac{S_i f_s}{2 r B_{BPF}}} \quad \text{或} \quad \sigma^2 = \frac{E f_s}{4 r}, \quad \sigma = \sqrt{\frac{E f_s}{4 r}}, \quad E \text{ 是接收信号平均能量。}$$

2.3 AM 调制解调的 MATLAB 实现

例：调制信号为

$$m(t) = \begin{cases} 1, & 0 \leq t \leq t_0/3 \\ -2, & t_0/3 < t \leq 2t_0/3, \\ 0, & \text{其它} \end{cases}$$

利用 AM 调制方式调制载波 $c(t) = \cos 2\pi f_c t$ ，假设 $t_0 = 0.15s$, $f_c = 250Hz$, 直流分量为 3,

采用相干方式解调实现 AM 信号的调制解调。

主程序

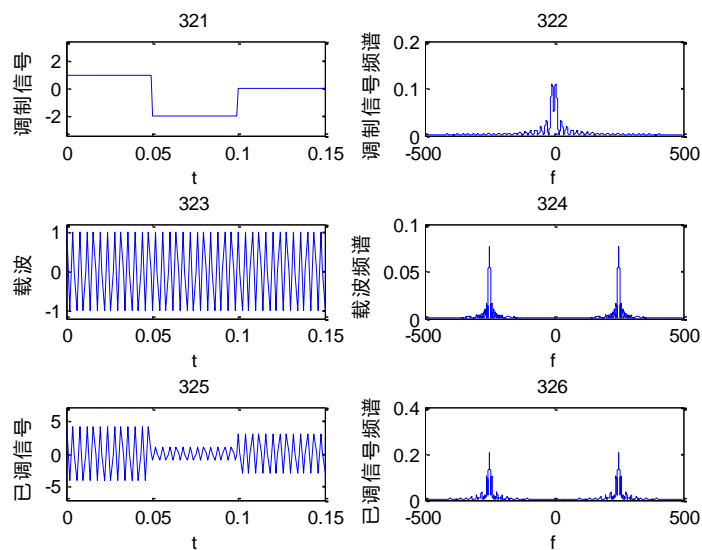
```
% AM调制解调
clear all;close all;echo on
%-----系统仿真参数
A=3;    %直流分量
fc=250;  %载波频率 (Hz)
t0=0.15;%信号时长
snr=25;  %解调器输入信噪比dB
dt=0.001% 系统时域采样间隔
fs=1/dt;%系统采样频率
df = 0.2;  %所需的频率分辨率
t=0:dt:t0;

Lt=length(t);%仿真过程中，信号长度
snr_lin = 10^(snr/10); %解调器输入信噪比
%-----画出调制信号波形及频谱
% 产生模拟调制信号
m=[ones(1, t0/(3*dt)), -2*ones(1, t0/(3*dt)), zeros(1, t0/(3*dt)+1)];
L=2*min(m);
R=2*max(abs(m))+A;
pause%画出调制信号波形及频谱
clf
figure(1); subplot(321);
plot(t,m(1:length(t)));% 画出调制信号波形
axis([0 t0 -R/2 R/2]);
xlabel('t');
ylabel('调制信号');
subplot(322);
```

```

[M,m,df1,f]=T2F(m,dt,df,fs);%求出调制信号频谱
[Bw_eq]=signalband(M,df,t0);%求出信号等效带宽
f_start=fc-Bw_eq;
f_cutoff=fc+Bw_eq;
plot(f,fftshift(abs(M)))% 画出调制信号频谱
xlabel('f');ylabel('调制信号频谱');
pause %画出载波及频谱
subplot(323);
c=cos(2*pi*fc*t);%载波
plot(t,c);
axis([0 t0 -1.2 1.2]);
xlabel('t');ylabel('载波');
subplot(324)% 载波频谱
[C,c,df1,f]=T2F(c,dt,df,fs);
plot(f,fftshift(abs(C)))% 画出载波频谱
xlabel('f');ylabel('载波频谱');
pause% 画已调信号及其频谱
subplot(325)% 画已调信号
u=(A+m(1:Lt)).*c(1:Lt);%已调信号
plot(t,u);% 画出已调信号波形
axis([0 t0 -R R]);
xlabel('t');ylabel('已调信号');
subplot(326);
[U,u,df1,f]=T2F(u,dt,df,fs);
plot(f,fftshift(abs(U)))% 画出已调信号频谱
xlabel('f');ylabel('已调信号频谱');
%-----该图为figure(1)-----%

```



```

%-----将已调信号送入信道

```

```

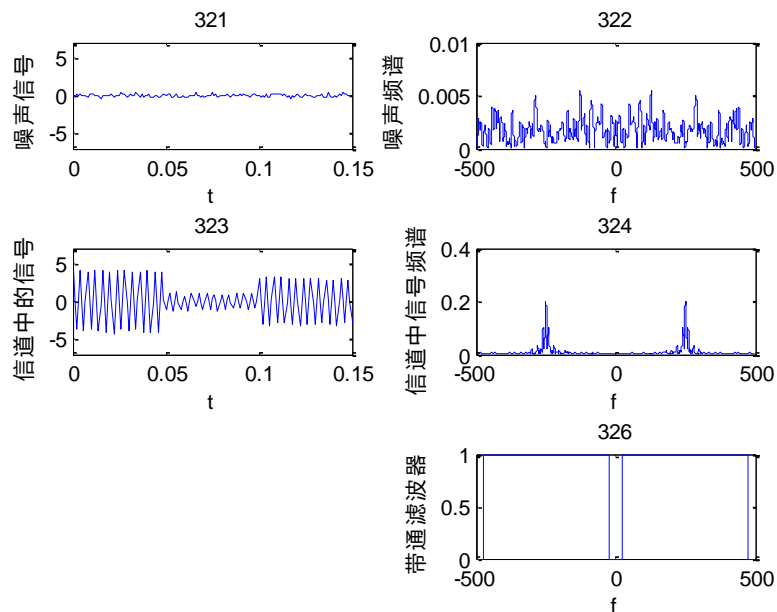
%先根据所给信噪比产生高斯白噪声

```

```

signal_power = power_x(u(1:Lt)); %已调信号的平均功率
noise_power=(signal_power*fs)/(snr_lin*4* Bw_eq);%求出噪声方差（噪声均值为0）
noise_std = sqrt(noise_power); %噪声标准偏差
noise = noise_std*randn(1,Lt); %产生噪声
pause%画出信道高斯白噪声波形及频谱，此时，噪声已实现，为确知信号，可求其频谱
figure(2); subplot(321);
plot(t,noise);% 画出噪声波形
axis([0 t0 -R R]);
xlabel('t');ylabel('噪声信号');
subplot(322);
[noisef,noise,df1,f]=T2F(noise,dt,df,fs);%噪声频谱
plot(f,fftshift(abs(noisef)))% 画出噪声频谱
xlabel('f');ylabel('噪声频谱');
pause%画出叠加了噪声的已调信号波形及频谱
sam=u(1:Lt)+noise(1:Lt);%叠加了噪声的已调信号
subplot(323); %画出叠加了噪声的已调信号波形
plot(t,sam);
axis([0 t0 -R R]);
xlabel('t');
ylabel('信道中的信号');
subplot(324);
[samf,sam,df1,f]=T2F(sam,dt,df,fs);%求出叠加了噪声的已调信号频谱
plot(f,fftshift(abs(samf)))% 画出叠加了噪声的已调信号频谱
xlabel('f');
ylabel('信道中信号频谱');
[H,f]=bp_f(length(sam),f_start,f_cutoff,df1,fs,1);%求带通滤波器
subplot(326);
plot(f,fftshift(abs(H)))% 画出带通滤波器
xlabel('f');ylabel('带通滤波器');
%-----该图为figure(2)-----%

```



```
%-----经过带通滤波器
pause%经过理想带通滤波器后的信号及其频谱
DEM = H.*samf;      %滤波器输出的频谱
[dem]=F2T(DEM, fs); %滤波器的输出波形

figure(3)
subplot(321)%经过理想带通滤波器后的信号波形
plot(t, dem(1:Lt))%画出经过理想带通滤波器后的信号波形
axis([0 t0 -R R]);
xlabel('t');
ylabel('理想BPF输出信号');
[demf, dem, df1, f]=T2F(dem(1:Lt), dt, df, fs); %求经过理想带通滤波器后信号频谱
subplot(322)
plot(f, fftshift(abs(demf))); % 画出经过理想带通滤波器后信号频谱
xlabel('f');
ylabel('理想BPF输出信号频谱');

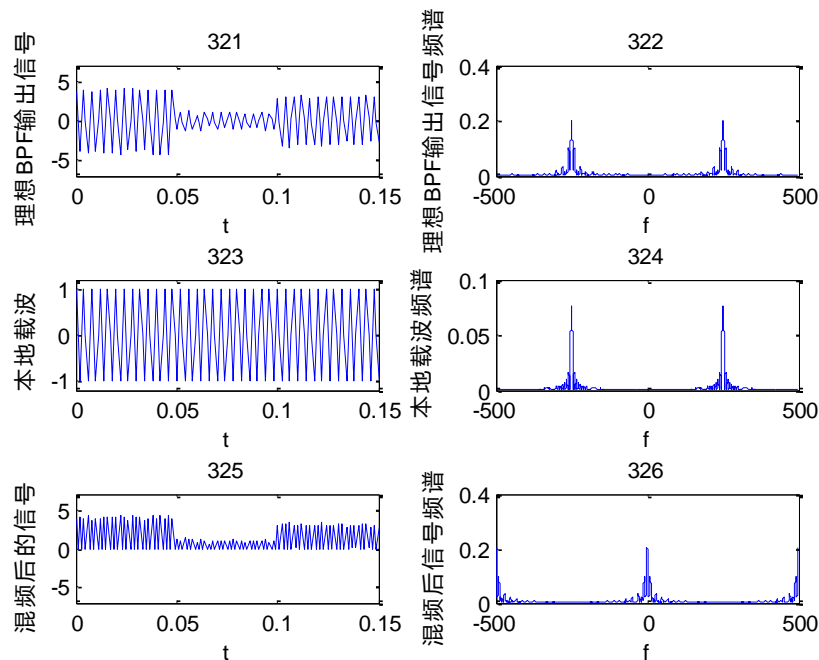
%-----和本地载波相乘，即混频

pause%混频后的信号，先画本地载波及其频谱
subplot(323)
plot(t, c(1:Lt));
axis([0 t0 -1.2 1.2]);
xlabel('t');
ylabel('本地载波');
subplot(324)% 载波频谱
[C, c, df1, f]=T2F(c(1:Lt), dt, df, fs);
plot(f, fftshift(abs(C))) % 画出载波频谱
```

```

xlabel('f');
ylabel('本地载波频谱');
pause%再画混频后信号及其频谱
der=dem(1:Lt).*c(1:Lt);%混频
subplot(325)%画出混频后的信号
plot(t,der);
axis([0 t0 -R R]);
xlabel('t');
ylabel('混频后的信号');
subplot(326)
[derf,der,df1,f]=T2F(der,dt,df,fs);%求混频后的信号频谱
plot(f,fftshift(abs(derf)))%画出混频后的信号的频谱
xlabel('f');
ylabel('混频后信号频谱');
%-----该图为figure(3)-----%

```



```

%-----经过低通滤波器
pause%画出理想低通滤波器

figure(4)
[LPF,f]=lp_f(length(der),Bw_eq,df1,fs,2);%求低通滤波器
subplot(322)
plot(f,fftshift(abs(LPf)))% 画出理想低通滤波器
xlabel('f');
ylabel('理想LPF');
pause%混频信号经理想低通滤波器后的频谱及波形

```

```
DM = LPF.*derf; %理想低通滤波器输出的频谱
```

```
[dm]=F2T(DM, fs); %滤波器的输出波形
```

```
subplot(323)
```

```
plot(t, dm(1:Lt)); %画出经过低通滤波器后的解调出的波形
```

```
axis([0 t0 -R R]);
```

```
xlabel('t');
```

```
ylabel('LPF输出信号');
```

```
subplot(324)
```

```
[dmf, dm, df1, f]=T2F(dm(1:Lt), dt, df, fs); %求LPF输出信号的频谱
```

```
plot(f, fftshift(dmf)); %画出LPF输出信号的频谱
```

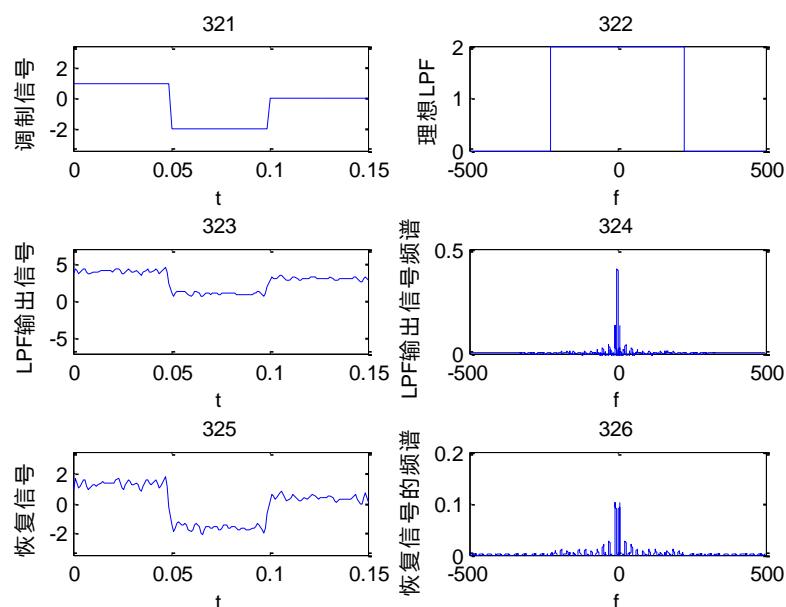
```
xlabel('f');
```

```
ylabel('LPF输出信号频谱');
```

```
axis([-fs/2 fs/2 0 0.5]);
```

```
%-----去除解调信号中的直流分量
```

```
%-----该图为 figure(4)-----%
```



```
pause %去除解调信号中的直流分量
```

```
dmd=dm(1:Lt)-mean(dm(1:Lt));
```

```
subplot(325)
```

```
plot(t, dmd); %画出恢复信号(去除直流分量)
```

```
axis([0 t0 -R/2 R/2]);
```

```
xlabel('t');
```

```
ylabel('恢复信号');
```

```
[dmdf, dmd, df1, f]=T2F(dmd, dt, df, fs); %求恢复信号的频谱
```

```
subplot(326)
```

```
plot(f, fftshift(dmdf)); %画出恢复信号的频谱
```

```

xlabel('f');
ylabel('恢复信号的频谱');
axis([-fs/2 fs/2 0 0.2]);
subplot(321);
plot(t,m(1:Lt));% 画出调制信号波形
axis([0 t0 -R/2 R/2]);
xlabel('t');
ylabel('调制信号');

```

4.子函数

● 序列的傅立叶变换

```

function [M,m,df]=fftseq(m,ts,df)
%各参数含义与子函数T2F中的完全相同，完成
fs = 1/ts;
if nargin ==2
    n1 =0;
else
    n1 = fs/df;
end
n2 = length(m);
n = 2^(max(nextpow2(n1),nextpow2(n2)));
M = fft(m,n);
m = [m,zeros(1,n-n2)];
df = fs/n;

```

● 计算信号功率

```

function p=power_x(x)
%x:输入信号
%p:返回信号的x功率
p=(norm(x).^2)./length(x);

```

● 信号从频域转换到时域

```

function [m]=F2T(M,fs)

%-----输入参数

%M: 信号的频谱
%fs:系统采样频率
%-----输出(返回)参数
%m:傅里叶逆变换后的信号，注意其长度为2的整数次幂，利用其画波形时，要注意选取m的一部分，选取长度和所给时间序列t的长度要一致，plot(t,m(1:length(t))),否则会出错。
m = real(ifft(M))*fs;

```

● 信号从时域转换到频域

```

function [M,m,df1,f]=T2F(m,ts,df,fs)

```

%-----输入参数

%m: 信号

%ts: 系统时域采样间隔

%df: 所需的频率分辨率

%fs: 系统采样频率

%-----输出(返回)参数

%M: 傅里叶变换后的频谱序列

%m: 输入信号参与过傅里叶变换后对应的序列, 需要注意的是, 该序列与输入信号m的区别, 其长度是不一样的, 输入的m长度不一定是2的整数次幂, 而傅里叶变换要求输入信号长度为2的整数次幂, 故傅里叶变换前需对输入的m信号进行补零, 其长度有所增加, 故输出参数中的m为补零后的输入信号, 其长度与输入参数m不一样, 但与M, f长度是一样的, 并且, 其与时间序列t所对应的序列m(1:length(t))与输入参数中的m是一致的。

%df1: 返回的频率分辨率

%f: 与M相对应的频率序列

[M, m, df1]=fftseq(m, ts, df);

f = [0:df1:df1*(length(m)-1)] -fs/2; %频率向量

M=M/fs;

● 带通滤波器

Function[H, f]=bp_f(n, f_start, f_cutoff, df1, fs, p)

%带通滤波器函数 输入设计的滤波器参数, 产生带通滤波器频率特性函数H和频率向量f

%-----输入参数

%n 带通滤波器的输入信号长度

%f_start 通带起始频率

%f_cutoff 带通滤波器的截止频率

%df1 频率分辨率

%fs 抽样频率

%p 滤波器幅度

%-----输出(返回)参数

%H 带通滤波器频率响应

%f 频率向量

%设计滤波器

n_cutoff = floor(f_cutoff/df1);

n_start = floor(f_start/df1);

f = [0:df1:df1*(n-1)] -fs/2; %频率向量

H = zeros(size(f));

H(n_start+1:n_cutoff) = p*ones(1, n_cutoff-n_start);

H(length(f) - n_cutoff+1:length(f)-n_start) = p*ones(1, n_cutoff-n_start);

● 低通滤波器

function [H, f]=lp_f(n, f_cutoff, df1, fs, p)

%低通滤波器函数 输入设计的滤波器参数, 产生低通滤波器频率特性函数H和频率向量f

```

%-----输入参数

%n  低通滤波器的输入信号长度
%f_cutoff  低通滤波器的截止频率
%df1      频率分辨率
%fs       抽样频率
%p       滤波器幅度
%-----输出(返回)参数
%H  低通滤波器频率响应
%f  频率向量
n_cutoff = floor(f_cutoff/df1);    %设计滤波器
f = [0:df1:df1*(n-1)] -fs/2;    %频率向量
H = zeros(size(f));
H(1:n_cutoff) = p*ones(1,n_cutoff);
H(length(f) - n_cutoff+1:length(f)) = p*ones(1,n_cutoff);

```

● 计算信号有效带宽

```

function [Bw_eq]=signalband(sf, df, T)
%计算信号等效带宽
%sf: 信号频谱
%df:频率分辨率
%T: 信号持续时间
sf_max=max(abs(sf));
Bw_eq=sum(abs(sf).^2)*df/T/sf_max.^2

```

2.4 实验内容

1. 产生调幅(AM)信号，并观察对其时频域特征

编写函数 mod_AM 用于产生 AM 调制信号，其中输入参数为被调制信号 x，调制系数 k，载波 fc，输出参数为调制信号 y。

$$y(t) = A_c[1 + kx(t)]\cos(2\pi f_c t)$$

2. 产生调频(FM)信号，并观察对其时频域特征

编写函数 mod_FM 用于产生 FM 调制信号，其中输入参数为被调制信号 x，调制系数 b，载波频率 fc，输出参数为调制信号 y。

$$y(t) = A_c \cos[2\pi f_c t + b \int_0^t x(\tau) d\tau]$$

3. 产生调相(PM)信号，并观察对其时频域特征

编写函数 `mod_PM` 用于产生 PM 调制信号，其中输入参数为被调制信号 x ，调制系数 k ，载波频率 f_c ，输出参数为调制信号 y 。

$$y(t) = A_c \cos[2\pi f_c t + kx(t)]$$

4. 解调 AM 调制信号

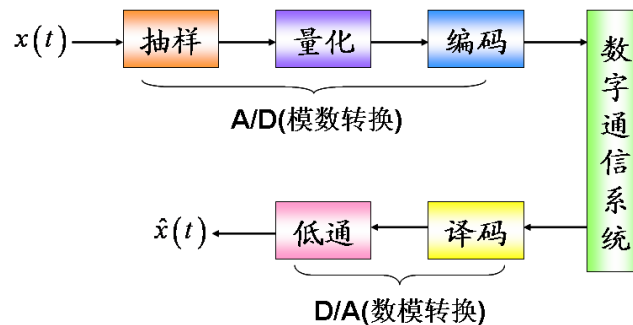
编写函数 `dem_AM` 用于解调 AM 调制信号，其中输入参数为调制信号 y ，调制系数 k ，载波 f_c ，输出参数为调制信号 x 。

实验二（1 小班周武旻老师） 模拟信号的数字传输

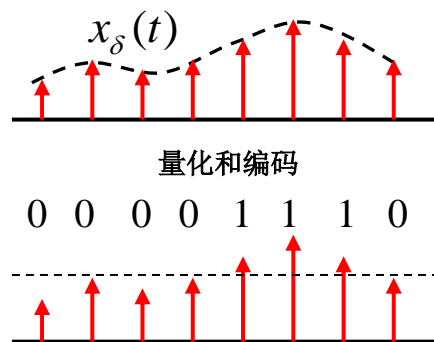
本章目标

- 掌握低通信号抽样定理
- 理解 13 折线 A 率逐次比较型 PCM 编码仿真的思想
- 掌握 13 折线 A 律逐次比较型 PCM 编，译码原理

2.1 脉冲编码调制



抽样:



2.2 低通抽样定理

一频带限制在 $(0, f_H)$ 赫内的时间连续信号 $m(t)$ ，若以 $f_s \geq 2f_H$ 速率对 $m(t)$ 等间隔

$T_s = 1/f_s \leq 1/2f_H$ 抽样，则 $m(t)$ 将被所得抽样函数 $m_s(t)$ 完全确定。

例1. 设低通信号

$$x(t) = 0.1 \cos(0.15\pi t) + 1.5 \sin 2.5\pi t + 0.5 \cos 4\pi t$$

(1)画出该低通信号的波形

(2)画出抽样速率为fs=4Hz的抽样序列

(3)抽样序列恢复出原始信号

%低通抽样定理

clear all;

close all;

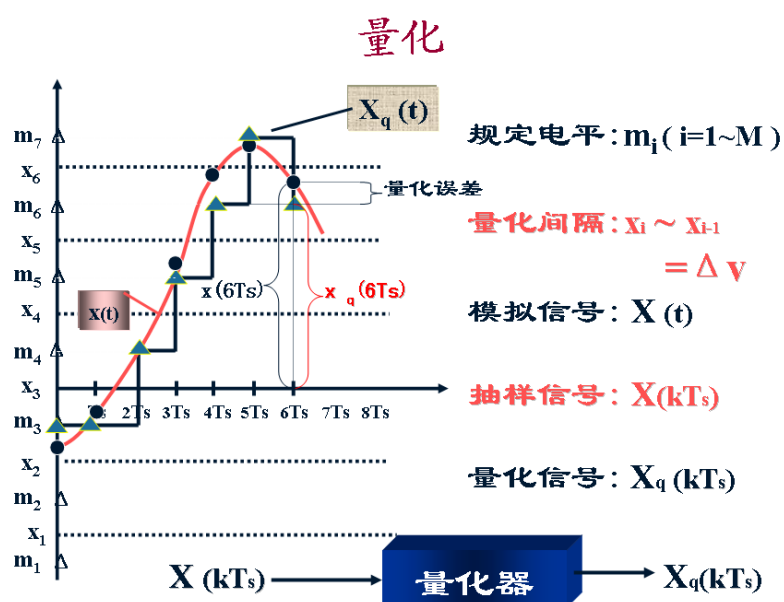
dt=0.01;

```

t=0:dt:10;
xt=0.1*cos(0.15*pi*t)+1.5*sin(2.5*pi*t)+0.5*cos(4*pi*t);
[f,xf]=T2F(t,xt);
%抽样信号，抽样频率为4Hz
fs=4;
sdt=1/fs;
t1=0:sdt:10;
st=0.1*cos(0.15*pi*t1)+1.5*sin(2.5*pi*t1)+0.5*cos(4*pi*t1);
[f1,sf]=T2F(t1,st);
%恢复原始信号
t2=-50:dt:50;
gt=sinc(fs*t2);
stt=sigexpand(st,sdt/dt);
xt_t=conv(stt,gt);
figure(1);
subplot(311);
plot(t,xt);
title('原始信号');
subplot(312);
plot(t1,st);
title('抽样信号');
subplot(313);
t3=-50:dt:60+sdt-dt;
plot(t3,xt_t);
title('抽样信号恢复');
axis([0 10 -4 4]);

```

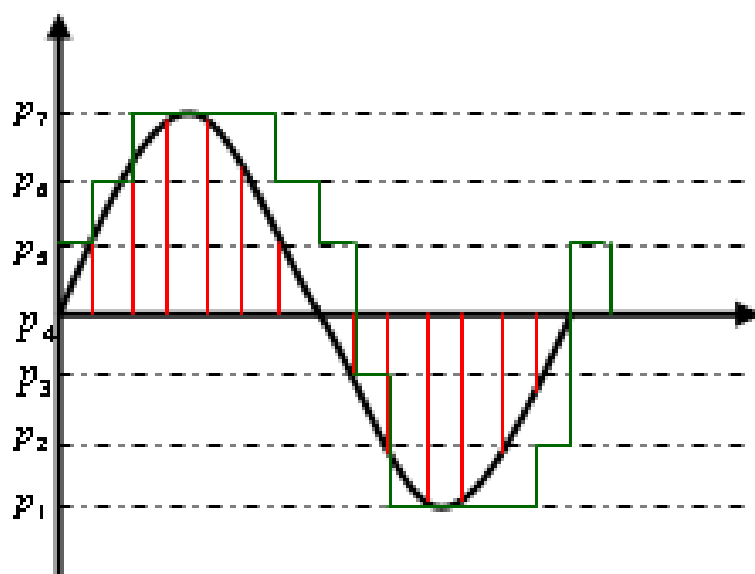
2.3 均匀量化原理



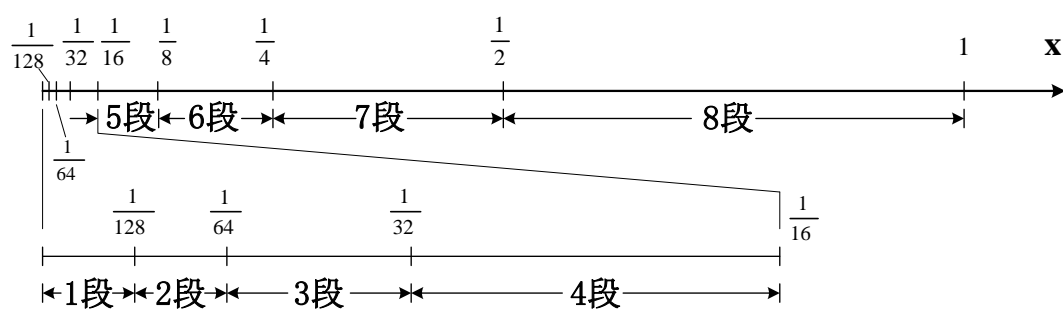
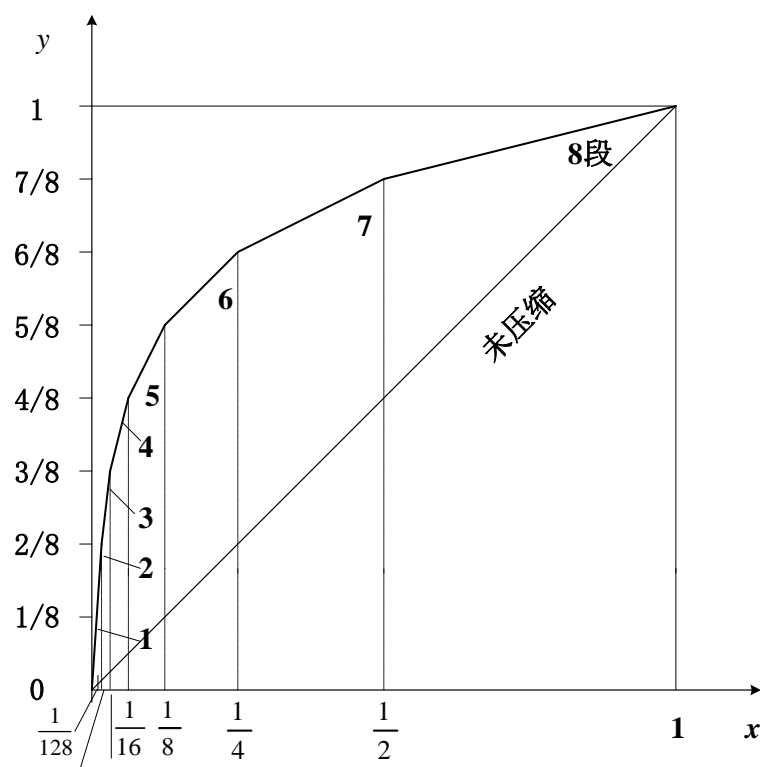
例2. 均匀量化程序

输入参数：f—待量化的值；V—从+V到-V量化；L—量化的级数。

```
function h=junyun(f,V,L)
n=length(f);%抽样序列的长度
t=2*V/L;%量化区间的宽度
p=zeros(1,L+1);
for i=1:L+1
    p(i)=-V+(i-1)*t;
end        %计算量化电平值
for i=1:n
    if f(i)>V
        h(i)=V;
    end
    if f(i)<=-V
        h(i)=-V;
    end        %处理过载情况
flag=0;
    for j=2:L/2+1
        if flag==0
            if f(i)<p(j)
                h(i)=p(j-1);
                flag=1;
            end
        end
    end        %处理小于0的抽样值
    for j=L/2+2:L+1
        if flag==0
            if f(i)<p(j)
                h(i)=p(j);
                flag=1;
            end
        end
    end        %处理大于0的抽样值
end
均匀量化程序调用
x=[0:0.004:4*pi];
y=sin(x);
w=junyun(y,1,32);
plot(x,y,x,w);
x=[0:0.004:4*pi];
y=sin(x);
w=junyun(y,1,8);
plot(x,y,x,w);
```



2.4 非均匀量化



信号小时， Δ 小，信号大时， Δ 大。一般语音信号，信号幅度小出现的概率大，信号幅度大出现的概率小。通过非均匀量化，使得平均信噪比增大。

2. 回忆 13 折线 A 律 PCM 的非线性编码方法。

极性码 段落码 段内码
C1 C2C3C4 C5C6C7C8

- 将量化区间 $[a, b]$ 分为 4096 个小段
- 正半轴 2048 个小段，负半轴 2048 个小段
- 每个小段用 Δ 表示

段落 编码	区间 范围/ Δ	量化 间隔/ Δ	量化 区间/ Δ	译码器 量化输出/ Δ	PCM 编码
000	[0, 16)	1	[0, 1) [1, 2) [2, 3) ... [15, 16)	1 2 3 ... 15	1 000 0000 1 000 0001 1 000 0010 ... 1 000 1111
001	[16, 32)	1	[16, 17) [17, 18) [18, 19) ... [31, 32)	16 17 18 ... 31	1 001 0000 1 001 0001 1 001 0010 ... 1 001 1111
010	[32, 64)	2	[32, 34) [34, 36) [36, 38) ... [62, 64)	33 35 37 ... 63	1 010 0000 1 010 0001 1 010 0010 ... 1 010 1111
011	[64, 128)	4	[64, 68) [68, 72) [72, 76) ... [124, 128)	66 70 74 ... 126	1 011 0000 1 011 0001 1 011 0010 ... 1 011 1111
100	[128, 256)	8	[128, 136) [136, 144) [144, 152) ... [248, 256)	134 140 148 ... 252	1 100 0000 1 100 0001 1 100 0010 ... 1 100 1111
101	[256, 512)	16	[256, 272) [272, 288) [288, 304) ... [496, 512)	264 280 296 ... 504	1 101 0000 1 101 0001 1 101 0010 ... 1 101 1111
110	[512, 1024)	32	[512, 544) [544, 576) [576, 608) ... [992, 1024)	528 560 592 ... 1008	1 110 0000 1 110 0001 1 110 0010 ... 1 110 1111
111	[1024, 2048)	64	[1024, 1088) [1088, 1152) [1152, 1216) ... [1984, 2048)	1056 1120 1184 ... 2016	1 111 0000 1 111 0001 1 111 0010 ... 1 111 1111

例 3. 采用 13 折线 A 律编译码电路，设最小量化级为 1Δ ，已知抽样脉冲值为 $+635\Delta$

1) 试求此时编码器输出码组，并计算接收端译码后的量化误差；

2) 写出对应七位码（不包含极性码）的均匀量化 11 位码

%输入 x 参数为 0—2048 的样值，输出 out 为 8 位二进制码

```
Function [out]=pcm_encode(x)
n=length(x);
for i=1:n
%编写极性码
    if x(i)>0
        out(i,1)=1;
    else
        out(i,1)=0;
    end
%编写段落码并计算量化间隔和量化起始电平
    If 0<=abs(x(i))&abs(x(i))<16
%段落码
        out(i,2)=0;out(i,3)=0;out(i,4)=0;
%量化间隔
        step=1;
%起始电平
        st=0;
    elseif 16<=abs(x(i))&abs(x(i))<32
        out(i,2)=0;out(i,3)=0;out(i,4)=1;step=1;st=16;
    elseif 32<=abs(x(i))&abs(x(i))<64
        out(i,2)=0;out(i,3)=1;out(i,4)=0;step=2;st=32;
    elseif 64<=abs(x(i))&abs(x(i))<128
        out(i,2)=0;out(i,3)=1;out(i,4)=1;step=4;st=64;
    elseif 128<=abs(x(i))&abs(x(i))<256
        out(i,2)=1;out(i,3)=0;out(i,4)=0;step=8;st=128;
    elseif 256<=abs(x(i))&abs(x(i))<512
        out(i,2)=1;out(i,3)=0;out(i,4)=1;step=16;st=256;
    elseif 512<=abs(x(i))&abs(x(i))<1024
        out(i,2)=1;out(i,3)=1;out(i,4)=0;step=32;st=512;
    else
        out(i,2)=1;out(i,3)=1;out(i,4)=1;step=64;st=1024;
    end
    if (abs(x(i))>=2048)
%处理过载现象
        out(i,2:8)=[1 1 1 1 1 1 1];
    else
%计算段内码
        tmp=floor((abs(x(i))-st)/step);
%十进制转二进制数，%如果不减48，最后4位是ASCII数
```

```

        % 49 48 48 48
        t=dec2bin(tmp,4)-48;
        out(i,5:8)=t(1:4);
    end
end
out=reshape(out',1,8*n);
13 折线译码函数:
%输入 in 为 8 位二进制码, (−v, +v) 为量化区间
function [out]=pcm_decode(in,v)
n=length(in);
in=reshape(in',8,n/8)';
slot(1)=0;
slot(2)=16;
slot(3)=32;
slot(4)=64;
slot(5)=128;
slot(6)=256;
slot(7)=512;
slot(8)=1024;
step(1)=1;
step(2)=1;
step(3)=2;
step(4)=4;
step(5)=8;
step(6)=16;
step(7)=32;
step(8)=64;
for i=1:n/8
    ss=2*in(i,1)-1;
    tmp=in(i,2)*4+in(i,3)*2+in(i,4)+1;
    st=slot(tmp);
    dt=(in(i,5)*8+in(i,6)*4+in(i,7)*2+in(i,8))*step(tmp)+0.5*step(tmp);
    out(i)=ss*(st+dt)/2048*v;
end

```

实验三 眼图

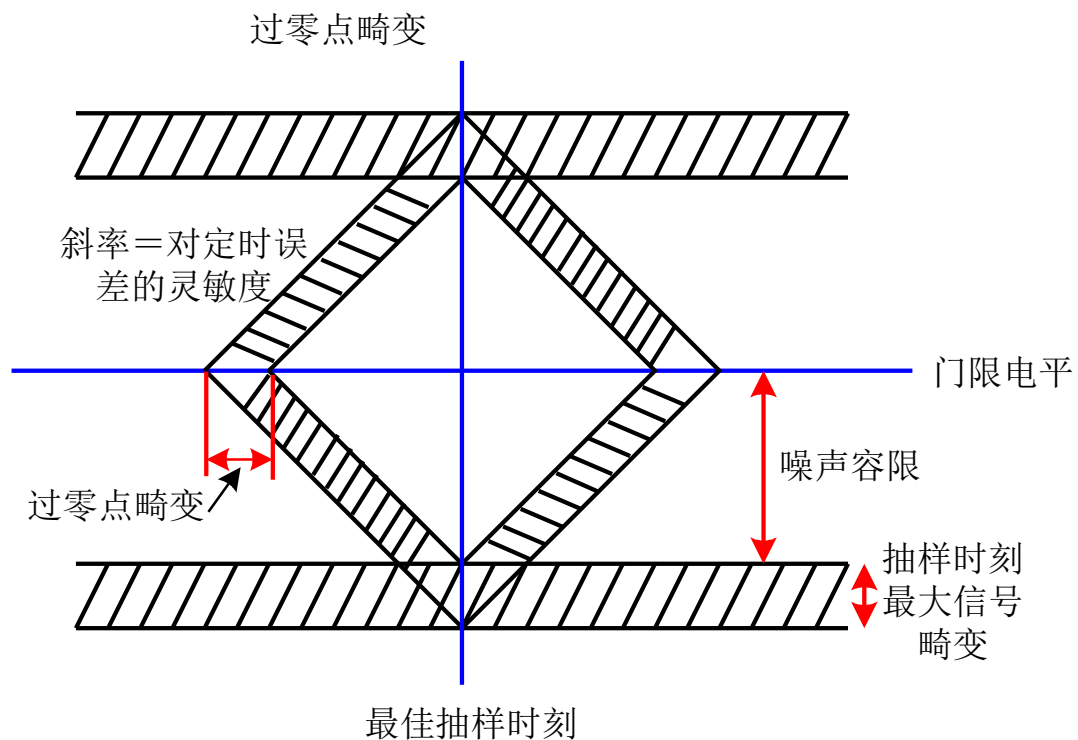
本章目标

- 掌握生成基带调制信号的方法
- 掌握生成眼图的方法
- 掌握分析眼图的分析方法

3.1 基本原理

在实际系统中，完全消除码间串扰是十分困难的，而码间串扰对误码率的影响目前尚无法找到数学上便于处理的统计规律，还不能进行准确计算。为了衡量基带传输系统的性能优劣，在实验室中，通常用示波器观察接收信号波形的方法来分析码间串扰和噪声对系统性能的影响，这就是眼图分析法。如果将输入波形输入示波器的 Y 轴，并且当示波器的水平扫描周期和码元定时同步时，在示波器上显示的图形很象人的眼睛，因此被称为眼图。

二进制信号传输时的眼图只有一只“眼睛”，当传输三元码时，会显示两只“眼睛”。眼图是由各段码元波形叠加而成的，眼图中央的垂直线表示最佳抽样时刻，位于两峰值中间的水平线是判决门限电平。在无码间串扰和噪声的理想情况下，波形无失真，“眼”开启得最大。当有码间串扰时，波形失真，引起“眼”部分闭合。若再加上噪声的影响，则使眼图的线条变得模糊，“眼”开启得小了，因此，“眼”张开的大小表示了失真的程度。由此可知，眼图能直观地表明码间串扰和噪声的影响，可评价一个基带传输系统性能的优劣。另外也可以用此图形对接收滤波器的特性加以调整，以减小码间串扰和改善系统的传输性能。



通常眼图可以用上图所示的图形来描述。由此图可以看出：

- 1) 眼图张开的宽度决定了接收波形可以不受串扰影响而抽样再生的时间间隔。显然，最佳抽样时刻应选在眼睛张开最大的时刻。

-
- 2) 眼图斜边的斜率，表示系统对定时抖动（或误差）的灵敏度，斜边越陡，系统对定时抖动越敏感。
 - 3) 眼图左（右）角阴影部分的水平宽度表示信号零点的变化范围，称为零点失真量，在许多接收设备中，定时信息是由信号零点位置来提取的，对于这种设备零点失真量很重要。
 - 4) 在抽样时刻，阴影区的垂直宽度表示最大信号失真量。
 - 5) 在抽样时刻上、下两阴影区间隔的一半是最小噪声容限，噪声瞬时值超过它就有可能发生错误判决；
 - 6) 横轴对应判决门限电平。

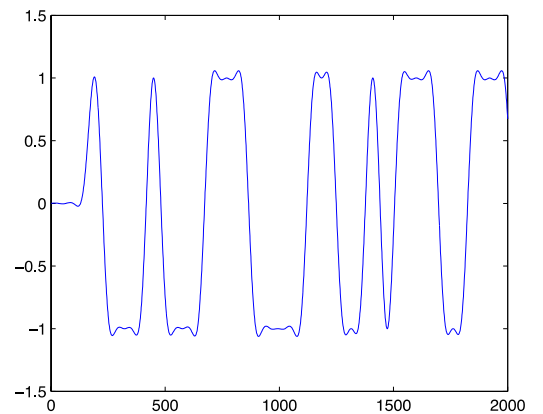
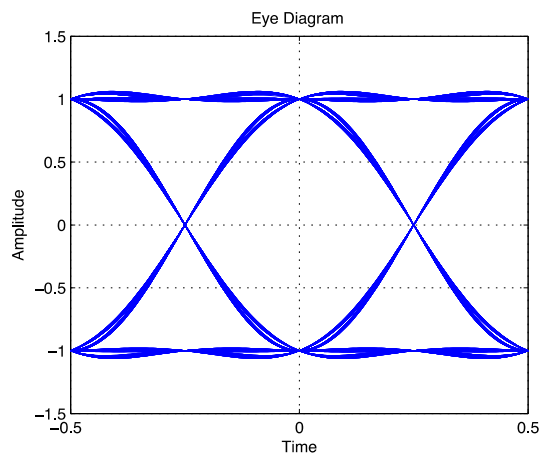
3.2 眼图的产生

基本步骤：

- 1) 产生随机符号序列；
- 2) 将符号序列通过等效信道，包括信道噪声、码间串扰和脉冲成形等，生成接收波形；
- 3) 仿真面向波形的信道过程：如定时误差、带外噪声等；
- 4) 绘制眼图

参考代码：

```
1) 无噪声，无码间串扰，2PAM 调制，等效信道脉冲响应为升余弦滚降成形脉冲
Fs=64; Fd=1; N=200;
sym=[-1;1];
alpha=0.5;
iS=randint(N,1, [1,length(sym)]);
S=sym(iS);
H=rcosine(Fd, Fs, 'fir', alpha);
S1=upsample(S, Fs/Fd);
X=filter(H, 1, S1);
plot(X(1:2000)); % 下图左
Y=X(Fs*4+1:end-Fs*4);
eyediagram(Y, Fs*2); % 下图右
```



Pluto 部分参考代码

%接收机相关参数设置

`fc = 1e9;`

`rx=sdrxx('Pluto','SamplesPerFrame',Fs*N,'CenterFrequency',fc,'OutputDataType','double','ShowAdvancedProperties',1,'BISTLoopbackMode','Disabled')`

%发送机相关参数设置

`tx = sdrtx('Pluto','CenterFrequency',fc);`

%数据的发送

`tx.transmitRepeat(X);`

`tx.Gain = -30;`

%数据的接收

`rxdata=rx();`

2) 码间串扰模型

S 为无码间串扰符号序列，S1 为存在码间串扰符号序列。

...

`ch=[1;-0.2];`

`ch=ch/norm(ch);`

`S1=filter(ch, 1, S);`

...

3.3 实验内容

必做部分：

分别产生符合下列要求的眼图，输入调制方式为 2PAM 和 4PAM，符号率为 1Hz，等效信道滤波器为升余弦滚降成形滤波器(缺省滚降系数为 0.5):

- 1) 无噪声和 20dB 加性高斯白噪声

-
- 2) 升余弦滚降系数分别为 0.5 和 1
 - 3) 受到码间串扰影响，信道参数可设为：ch=[0.9806; -0.1961];
 - 4) 利用 Pluto 实现基带信号在真实信道传输，并在接收端接收信号，观察其眼图

提高部分：

- 1) 不使用 matlab 自带的 eyediagram 函数，自己编写程序绘制眼图（此处需添加对绘制眼图原理解释）

project:

分析比较上述不同参数和信道情况下的眼图。

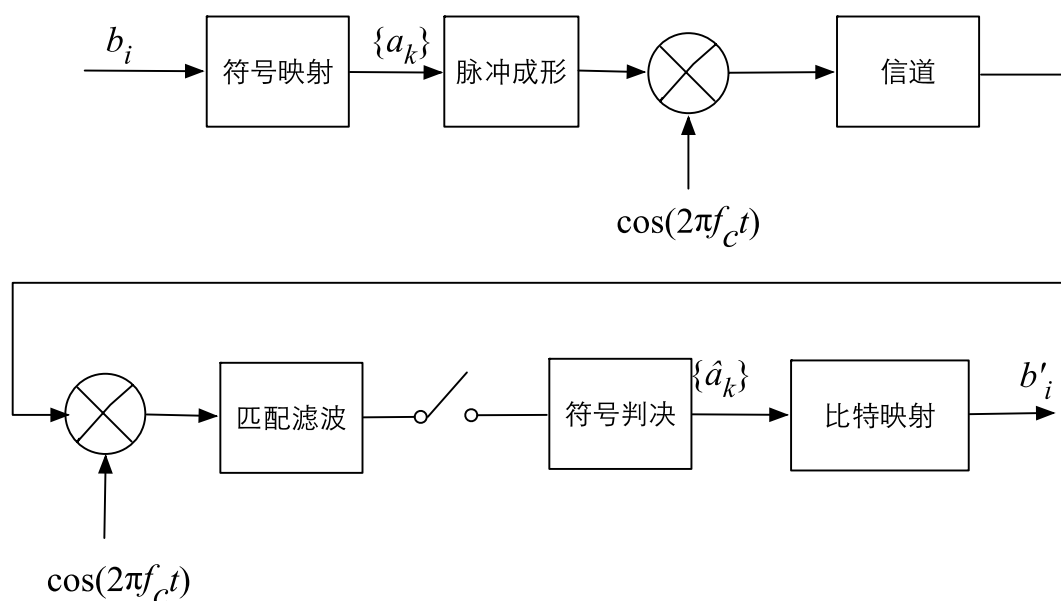
实验四 调制解调（BPSK, QPSK, 信噪比）

本章目标

- 掌握数字频带传输系统调制解调的仿真过程
- 掌握数字频带传输系统误码率仿真分析方法

4.1 调制解调原理

本章以 BPSK 为例，仿真说明数字频带传输的整个过程



假定：信道为加性高斯白噪声信道，其均值为 0、方差为 σ^2 ，采用矩形成形；发射端 BPSK 调制信号为：

$$s(t) = \begin{cases} A \cos(2\pi f_c t) & b_k = "1" \\ -A \cos(2\pi f_c t) & b_k = "0" \end{cases} \quad kT \leq t < (k+1)T$$

经信道传输，接收端输入信号为：

$$y_d(t) = s(t) + n(t)$$

经相干解调，匹配滤波，定时恢复后输出：

$$x_k = \begin{cases} A + n_k & b_k = "1" \\ -A + n_k & b_k = "0" \end{cases}$$

当 1, 0 独立等概出现时, BPSK 系统的最佳判决门限电平 $U_d^* = 0$ 。故判决规则为在取样时刻的判决值大于 0, 判 1, 小于 0, 判 0。

增加: `rectpulse(t,w)`可产生高度为 1、宽度为 w、关于 $t=0$ 对称的矩形脉冲信号, 该函数的横坐标范围由向量 t 决定, 是以 $t=0$ 为中心向左右扩展。

`randi (n,m)`产生的是一个 $n*m$ 维的矩阵, 矩阵的元素或者是 0 或者是 1, 是随机的。如果想产生一个范围的数, 可以设置一个区间, 如 `randint(2,3,[1 6])`, 就是产生一个 $2*3$ 随机矩阵, 这个矩阵的元素是区间[1 6]的随机数。

1, `rand` 生成均匀分布的伪随机数。分布在 (0~1) 之间

主要语法: `rand(m,n)`生成 m 行 n 列的均匀分布的伪随机数

`rand(m,n,'double')`生成指定精度的均匀分布的伪随机数, 参数还可以是 'single', `rand(RandStream,m,n)`利用指定的 `RandStream`(我理解为随机种子)生成伪随机数

2, `randn` 生成标准正态分布的伪随机数 (均值为 0, 方差为 1)

主要语法: 和上面一样

3, `randi` 生成均匀分布的伪随机整数

主要语法: `randi (iMax)` 在开区间 (0, iMax) 生成均匀分布的伪随机整数

`randi (iMax, m, n)` 在开区间 (0, iMax) 生成 $m \times n$ 型随机矩阵

`r = randi([iMin,iMax],m,n)`在开区间 (iMin, iMax) 生成 $m \times n$ 型随机矩阵

4, `randsrc`

无参数形式, 随机输出 -1 或 1;

`randsrc(m,n);`

`randsrc(m);`

输出 $m \times n$ 阶或 $m \times m$ 阶矩阵, 元素为随机出现的 -1 或 1, 概率为 1/2;

而 `rand` 是随机产生 0——1 中某一数呢,

例如

```
>> rand(2,3)
```

```
ans =
```

```
    0.9501    0.6068    0.8913
    0.2311    0.4860    0.7621
```

```
>> rand(3)
```

```
ans =
```

```
    0.4565    0.4447    0.9218
    0.0185    0.6154    0.7382
```

```
clear all;
close all;
echo on
%-----系统仿真参数
fc=2; %载波频率 (Hz)
snr=5; %信噪比dB
fs=16;% 系统采样率
N=1000; % 码元数
Ts=1; % 码元宽度
B=1/Ts; % 波特率
ts=1/fs; % 系统采样间隔
t=0:ts:N*Ts-ts;
t=t.';
Lt=length(t);
N_sample=Ts*fs;
%-----画出调制信号波形及功率谱
% 产生二进制信源
B=randi(1,N);
sym=[-1;1];
A=sym(B+1);
P=sum(abs(sym).^2)/length(sym);
std_v=sqrt(P/10^(snr/10))
A1=A+randn(size(A))*std_v;
S=rectpulse(A1, N_sample);
X=S.*cos(2*pi*fc*t);

figure(1)

plot(t(1:200),X(1:200));% 画出调制信号波形

r=X.*cos(2*pi*fc*t)

Sr=zeros(1,N);
H=rectpulse(1, N_sample);
for k=1:N
    Sr(k)=sum(r((k-1)*N_sample+1:k*N_sample).*rectpulse(1, ...
        N_sample));
end
Br=Sr>0;
```

```
errbit=sum(Br~=B);  
ber=errbit/N; % 误码率
```

4.3 实验要求

- 1) 分别编写 BPSK 与 QPSK 调制解调系统的 Matlab 仿真程序，要求：
 - 发送滤波器与接收滤波器均为根升余弦滚降滤波器；
 - 信道噪声为加性高斯白噪声
- 2) 绘制 BPSK 与 QPSK 调制下的误码率与信噪比曲线图，并与理论曲线进行对比

实验五 信道编译码 ()

本章目标

- 掌握数字频带传输系统调制解调的仿真过程
- 掌握数字频带传输系统误码率仿真分析方法

5.1 实验目的

1. 使用 MATLAB 进行卷积码编/译码器的仿真。
2. 熟练掌握 MATLAB 软件、语句。
3. 了解卷积码编/译码器的原理、知识。

5.2 实验要求

1. 编写源程序、准备测试数据。
2. 在 MATLAB 环境下完成程序的编辑、编译、运行，获得程序结果。如果结果有误，应找出原因，并设法更正之。

5.3 实验原理

(一) 卷积码编码器

1. 连接表示

卷积码由 3 个整数 n ， k ， N 描述。 k/n 也表示编码效率（每编码比特所含的信息量）；但 n 与线性分组码中的含义不同，不再表示分组或码子长度； N 称为约束长度，表示在编码移位寄存器中 k 元组的级数。卷积码不同于分组码的一个重要特征就是编码器的记忆性，即卷积码编码过程中产生的 n 元组，不仅是当前输入 k 元组的函数，而且还是前面 $N-1$ 个输入 k 元组的函数。实际情况下， n 和 k 经常取较小的值，而通过 N 的变化来控制编码的能力和复杂性。

下面以 1 卷中的卷积码编码器为例介绍卷积码编码器。该图表示一个约束长度 $K=3$ 的 $(2, 1)$ 卷积译码器，模 2 加法器的数目为 $n=2$ ，因此，编码效率 $k/n=1/2$ 。在每个输入比特时间上，1 位信息比特移入寄存器最左端的一级，同时将寄存器中原有比特均右移一级，接着便交替采样两个模 2 加法器，得到的码元就是与该输入比特相对应的分支字。对每一个输入信号比特都重复上述采样过程。

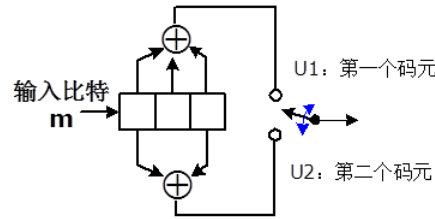


图 1 卷积码编码器（编码效率 $1/2$ ， $K=3$ ）

用于描述反馈移位寄存器实现循环码时所使用的生成多项式也可用户描述卷积码编码器的连接。应用 n 个生成多项式描述编码的移位寄存器与模 2 加法器的连接方式， n 个生成多项式分别对应 n 个模 2 加法器，每个生成多项式不超过 $K-1$ 阶。仍以 1 卷中的编码器为例，用生成多项式 $\mathbf{g}_1(X)$ 代表上方连接， $\mathbf{g}_2(X)$ 代表下方连接，则有：

$$\mathbf{g}_1(X) = 1 + X + X^2$$

$$\mathbf{g}_2(X) = 1 + X^2$$

多项式中的最低阶项对应于寄存器的输入级。输出序列根据如下方式求得：

$$\mathbf{U}(X) = \mathbf{m}(X)\mathbf{g}_1(X) \text{ 与 } \mathbf{m}(X)\mathbf{g}_2(X) \text{ 交织}$$

其中 \mathbf{m} 表示输入的信息矢量。

2. 状态图

卷积编码器属于有限状态机的器件。“有限”表明状态机制只有有限个不同的状态。有限状态机的状态可以用设备的当前输入和最少的信息量，来预测设备的输出。状态提供了有关过去序列过程及一组将来可能输出序列的限制，下一状态总是受到前一状态的限制。将编码器在时刻 t_i 的状态定义为 $X_i = m_{i-1}, m_{i-2}, \dots, m_{i-K+1}$ 。

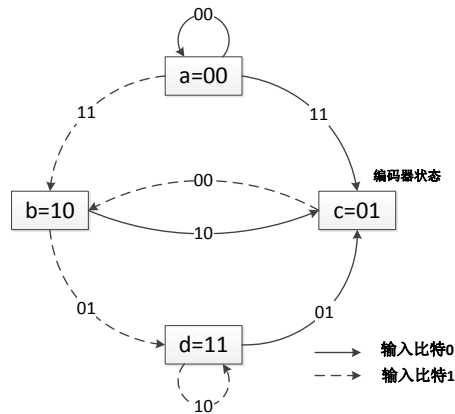


图 2 状态转移图

如 2 状所示，方框内的状态表示寄存器最右端 $N-1$ 级的内容，状态间的路径表示由此状态转移时的输出分支字。对应于两种可能的输入 bit，从每个状态出发只有两种转移。

3. 编码器网格图

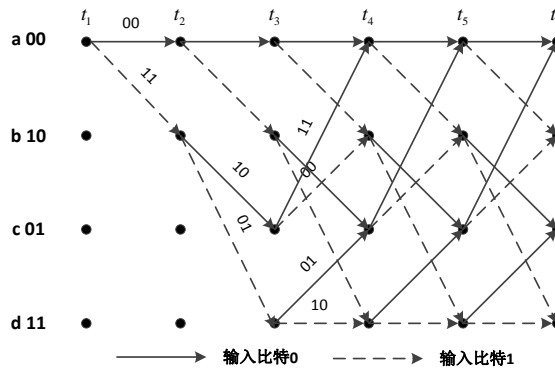


图 3 编码器网格图

虽然状态图完全地描述了编码器的特性，但由于没有表示时间过程，所以采用状态图跟踪编码器的状态转移很不方便。树状图在状态图的基础上增加了时间尺度。每个相继输入信息比特的编码过程可表述为从左向右经过树状图，每条数值代表一个输出分支字。树状图上增加的时间尺度是我们可以动态地描述输入序列的编码过程。但由于树状图的规模增长很快，因而只适于序列中分支子数目较小的情况。

我们采用移位寄存器的 4 种可能状态来标注树图的各个节点， $a=00, b=01, c=10, d=11$ 。树结构的第一次分支在时刻 t_1 ，产生一对节点，记为 a, b ；在后继的各个分支处，节点数翻倍。第二次分支在时刻 t_2 ，生成 4 个节点，记为 a, b, c, d ；第三次分支后共有 8 个节点。网格图利用了结构上的重复性，从而能够更加方便地描述编码器。

(二) 维特比译码算法

维特比译码算法由维特比在 1967 年提出。维特比算法的实质是最大似然译码，但

它利用了编码网格图的特殊结构，从而降低了计算的复杂性。该算法包括计算网格图上在时刻 t_i 到达各个状态的路径和接受序列之间的相似度，或者说距离。维特比算法考虑的是，去除不可能成为最大似然选择对象的网格图上的路径，即如果有两条路径到达同一状态，则具有最佳量度的路径被选中，成为幸存路径。对所有状态都将进行这样的选路操作，译码器不断在网格图上深入，通过去除可能性最小的路径实现判决。

网格图中每个时刻 t_i 上有 2^{K-1} 个状态，这里的 K 是约束长度，每种状态都可经两条路径到达。维特比译码包括计算到达每个状态的两条路径的路径量度，并舍弃其中一条路径。在时刻 t_i ，算法对 2^{K-1} 个状态（节点）都进行上述计算，然后进入时刻 t_{i+1} ，并重复上述过程。在一个给定的时刻，各状态的幸存路径量度就是该状态在该时刻的状态量度。

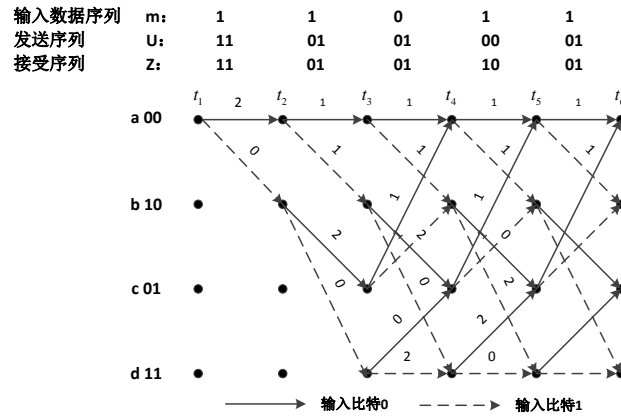


图 4 译码器网格图

5.4 实验内容

在 MATLAB 上设计一个 $(2, 1, 3)$ 卷积编码器和对应的采用维特比译码算法的译码器。编码器的生成多项式为：

$$\mathbf{g}_1(X) = 1 + X + X^2$$

$$\mathbf{g}_2(X) = 1 + X^2$$

将编码器的输出经过一个高斯白噪声信道的结果作为译码器的输入，观察比较译码器输出和编码器输入，了解卷积码的容错性，并计算译码结果的误比特率。基本流程如

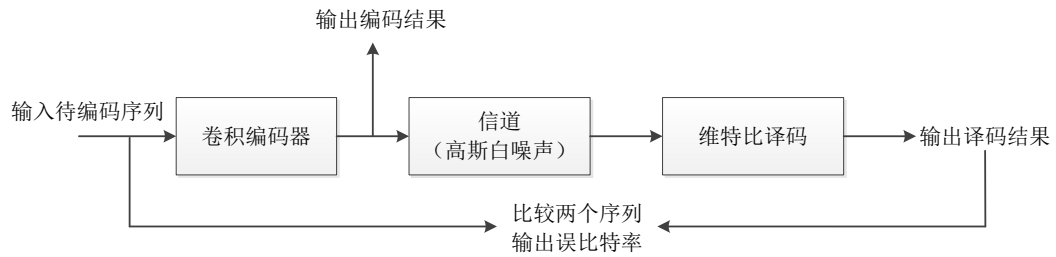


图 5 基所示。实验报告要求附加程序代码，并对代码中每个函数模块的实验方式和功能进行简要的说明。

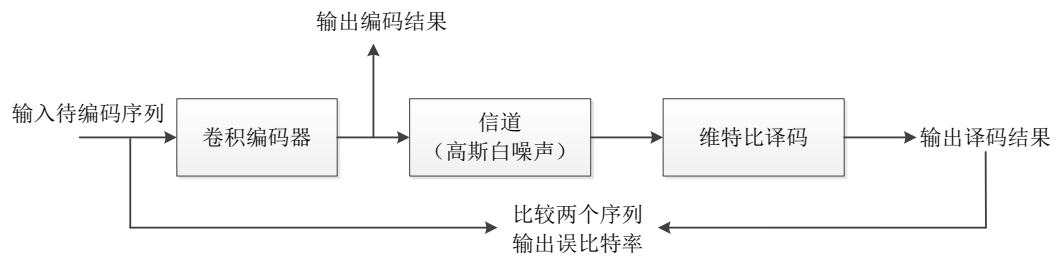


图 5 基本流程

实验参考：

可参考如下方式实现将编码结果通过高斯白噪声信道

$\text{Guass_noise} = \text{average} + \sqrt{p} * \text{randn}(1, 2*L);$ % L 为待编码信号序列长度，经过 1/2 编码器，序列长度变为 $2L$ ；其中 average 为白噪声的均值(这里均值为 0)； p 为噪声的方差；

$\text{code_out} = \text{code_out} + \text{Guass_noise};$

for $i = 1:2*L$

if $\text{code_out}(i) \geq 0.5$

$\text{code_out}(i) = 1;$

else

$\text{code_out}(i) = 0;$

end

end

以下为一段维特比算法解码器的实现函数，作为思路参考

function $\text{decoder_output} = \text{viterbi_hard}(y, L)$

global $G;$

$n = \text{size}(G, 1);$

$K = \text{size}(G, 2);$

$\text{number_of_states} = 2^{(K-1)};$

%-----生成各分支的输出-----

for $j = 0:\text{number_of_states}-1$

for $t = 0:1$

$[\text{next_state}, \text{memory_contents}] = \text{next_state_fun}(j, t, K);$

$\text{input}(j+1, \text{next_state}+1) = t;$

$\text{branch_output} = \text{rem}(\text{memory_contents} * G', 2);$

$\text{nextstate}(j+1, t+1) = \text{next_state};$

```

        output(j+1,t+1)=bin2deci(branch_output);
    end
end
%-----
metric_of_states=zeros(1,number_of_states);           % 各状态的度量
metricmetric_of_states_c=zeros(number_of_states,2);    % 各状态两个输入的度量
length_seq=length(y)/n;                                % 符号个数
decoder_output=zeros(1,length_seq-K+1);               % 解码输出
channel_output_matrix=reshape(y,n,length_seq);         % 将解调输出的比特按符号排列
survivor_state=zeros(number_of_states,length_seq+1);   % 留存路径
input_of_state=zeros(number_of_states,length_seq+1,2); % 汇聚到各状态的分支对应的
输入 state_sequence=zeros(1,length_seq+1);
count=zeros(1,number_of_states);

for i=1:length_seq-K+1
    for j=0:number_of_states-1
        for t=0:1
            binary_output=deci2bin(output(j+1,t+1),n); % 将各分支的输出转换为 2 进制
            branch_metric=Hamming_dis(channel_output_matrix(:,i)',binary_output); % 计算
            分支度量
            count(nextstate(j+1,t+1)+1)=count(nextstate(j+1,t+1)+1)+1;
            metric_of_states_c(nextstate(j+1,t+1)+1,count(nextstate(j+1,t+1)+1))=metric_of_st
            ates(j+1)+branch_metric; % 计算累积度量（加）
            input_of_state(nextstate(j+1,t+1)+1,.,count(nextstate(j+1,t+1)+1))=survivor_state(j
            +1,.); % 该分支所在路径的对应的输入
            input_of_state(nextstate(j+1,t+1)+1,i,count(nextstate(j+1,t+1)+1))=t;
        end;
    end;
end;
%-----比较汇聚到同一状态的两条路径，选取距离较小的-----
for j=0:number_of_states-1
    if metric_of_states_c(j+1,1)>=metric_of_states_c(j+1,2)
        metric_of_states(j+1)=metric_of_states_c(j+1,2);
        survivor_state(j+1,:)=input_of_state(j+1,.,2);
    else
        metric_of_states(j+1)=metric_of_states_c(j+1,1);
        survivor_state(j+1,:)=input_of_state(j+1,.,1);
    end;
end;
count=zeros(1,number_of_states);
%-----截短输出-----
if i>L
    [min_metric,location]=min(metric_of_states);
    decoder_output(i-L)=survivor_state(location,i-L);
end
end
end

```

```
    end;
end
%-----最后 L 个比特译码输出-----
[min_metric,location]=min(metric_of_states);
decoder_output(length_seq-K+1-L+1:length_seq-K+1)=survivor_state(location,length_seq-K+1-L
+1:length_seq-K+1);
```

在最后译码结果出来后，将原始信号和译码输出的还原信号进行比较，计算误比特率。

实验六 FSK 传输实验

一、实验目的

- 1、熟悉 FSK 调制和解调基本工作原理
- 2、掌握 FSK 数据传输过程
- 3、掌握 FSK 正交调制的基本工作原理与实现方法

二、实验仪器

- | | |
|---------------------|----|
| 1、ZH7001 通信原理综合实验系统 | 一台 |
| 2、40MHz 双踪数字存储示波器 | 一台 |

三、实验原理

（一）FSK 调制

在二进制频移键控中，幅度恒定不变的载波信号的频率随着输入码流的变化而切换（称为高音和低音，代表二进制的 1 和 0）。通常，FSK 信号的表达式为：

$$S_{FSK} = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c + 2\pi \Delta f)t \quad 0 \leq t \leq T_b \quad (\text{二进制 1})$$

$$S_{FSK} = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c - 2\pi \Delta f)t \quad 0 \leq t \leq T_b \quad (\text{二进制 0})$$

其中 $2\pi \Delta f$ 代表信号载波的恒定偏移。

产生 FSK 信号最简单的方法是根据输入的数据比特是 0 还是 1，在两个独立的振荡器中切换。采用这种方法产生的波形在切换的时刻相位是不连续的，因此这种 FSK 信号称为不连续 FSK 信号。不连续的 FSK 信号表达式为：

$$S_{FSK} = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_H t + \theta_1) \quad 0 \leq t \leq T_b \quad (\text{二进制 } 1)$$

$$S_{FSK} = \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_L t + \theta_2) \quad 0 \leq t \leq T_b \quad (\text{二进制 } 0)$$

其实现如图 4.1.1 所示：

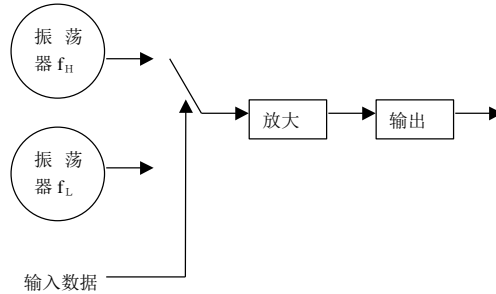


图 4.1.1 非连续相位 FSK 的调制框图

由于相位的不连续会造成频谱扩展，这种 FSK 的调制方式在传统的通信设备中采用较多。随着数字处理技术的不断发展，越来越多地采用连续相位 FSK 调制技术。

目前较常用产生 FSK 信号的方法是，首先产生 FSK 基带信号，利用基带信号对单一载波振荡器进行频率调制。因此，FSK 可表示如下：

$$\begin{aligned} S_{FSK}(t) &= \sqrt{\frac{2E_b}{T_b}} \cos[2\pi f_c t + \theta(t)] \\ &= \sqrt{\frac{2E_b}{T_b}} \cos[2\pi f_c t + 2\pi k \int_{-\infty}^t m(t) dt] \end{aligned}$$

应当注意，尽管调制波形 $m(t)$ 在比特转换时不连续，但相位函数 $\theta(t)$ 是与 $m(t)$ 的积分成比例的，因而是连续的，其相应波形如图 4.1.2 所示：

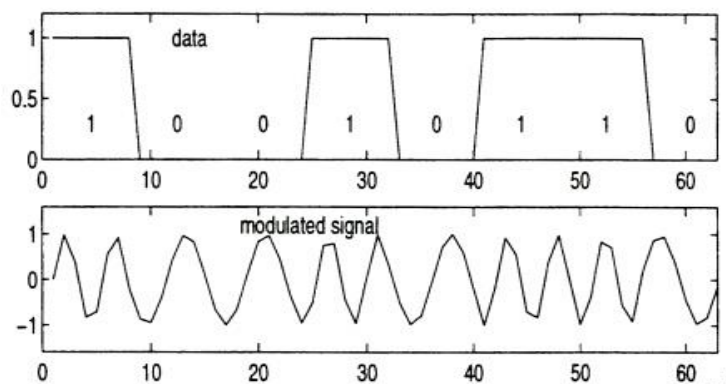


图 4.1.2 连续相位 FSK 的调制信号

由于 FSK 信号的复包络是调制信号 $m(t)$ 的非线性函数，确定一个 FSK 信号的频谱通常是相当困难的，经常采用实时平均测量的方法。二进制 FSK 信号的功谱密度由离散频率分量 f_c 、 $f_c + n\Delta f$ 、 $f_c - n\Delta f$ 组成，其中 n 为整数。相位连续的 FSK 信号的功率谱密度函数最终按照频率偏移的负四次幂衰减。如果相位不连续，功率谱密度函数按照频率偏移的负二次幂衰减。

FSK 的信号频谱如图 4.1.3 所示。

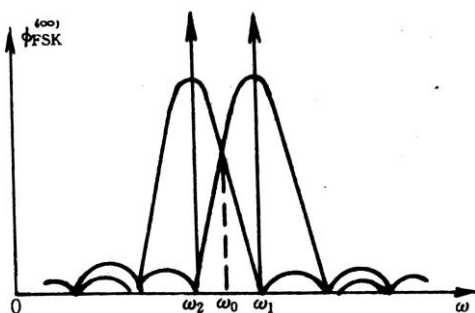


图 4.1.3 FSK 的信号频谱

FSK 信号的传输带宽 B_r ，由 Carson 公式给出：

$$B_r = 2\Delta f + 2B$$

其中 B 为数字基带信号的带宽。假设信号带宽限制在主瓣范围，矩形脉冲信号的带宽 $B = R$ 。因此，FSK 的传输带宽变为：

$$B_r = 2(\Delta f + R)$$

如果采用升余弦脉冲滤波器，传输带宽减为：

$$B_r = 2 \Delta f + (1 + \alpha) R$$

其中 α 为滤波器的滚降因子。

在通信原理综合实验系统中，FSK 的调制方案如下：

FSK 信号：

$$s(t) = \cos(\omega_0 t + 2\pi f_i \cdot t)$$

其中：

$$f_i = \begin{cases} f_1 & \text{当输入码为1} \\ f_2 & \text{当输入码为0} \end{cases}$$

因而有：

$$\begin{aligned} s(t) &= \cos \omega_0 t \cos 2\pi f_i \cdot t - \sin \omega_0 t \sin 2\pi f_i \cdot t \\ &= \cos \omega_0 t \cos \theta(t) - \sin \omega_0 t \sin \theta(t) \end{aligned}$$

其中：

$$\theta(t) = 2\pi f_c t + 2\pi K \int_{-\infty}^t m(t) dt$$

如果进行量化处理，采样速率为 f_s ，周期为 T_s ，有下式成立：

$$\begin{aligned} \theta(n) &= \theta(n-1) + 2\pi f_c T_s + 2\pi K m(n) T_s \\ &= \theta(n-1) + 2\pi T_s [f_s + K m(n)] \\ &= \theta(n-1) + 2\pi f_i T_s \end{aligned}$$

按照上述原理，FSK 正交调制器的实现为如图 4.1-4 结构：

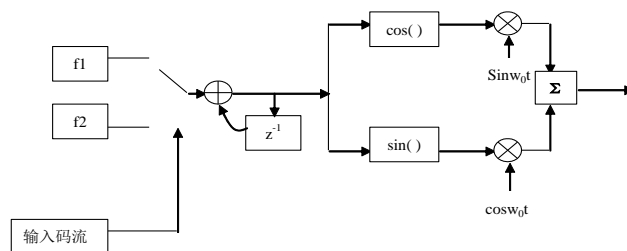


图 4.1.4 FSK 正交调制器结构图

如果发送 0 码，则相位累加器在前一码元结束时相位 $\theta(n)$ 基础上，在每个抽样到达时

刻相位累加 $2\pi f_1 T_s$ ，直到该信号码元结束；如发送 1 码，则相位累加器在上一码元结束时的相位 $\theta(n)$ 基础上，在每个抽样到达时刻相位累加 $2\pi f_2 T_s$ ，直到该码元结束。

在通信信道 FSK 模式的基带信号中传号采用 f_H 频率，空号采用 f_L 频率。在 FSK 模式下，不采用汉明纠错编译码技术。调制器提供的数据源有：

- 1、外部数据输入：可来自同步数据接口、异步数据接口和 m 序列；
- 2、全 1 码：可测试传号时的发送频率（高）；
- 3、全 0 码：可测试空号时的发送频率（低）；
- 4、0/1 码：0101...交替码型，用作一般测试；
- 5、特殊码序列：周期为 7 的码序列，以便于常规示波器进行观察；
- 6、m 序列：用于对通道性能进行测试；

FSK 调制器基带处理结构如图 4.1.5 所示：

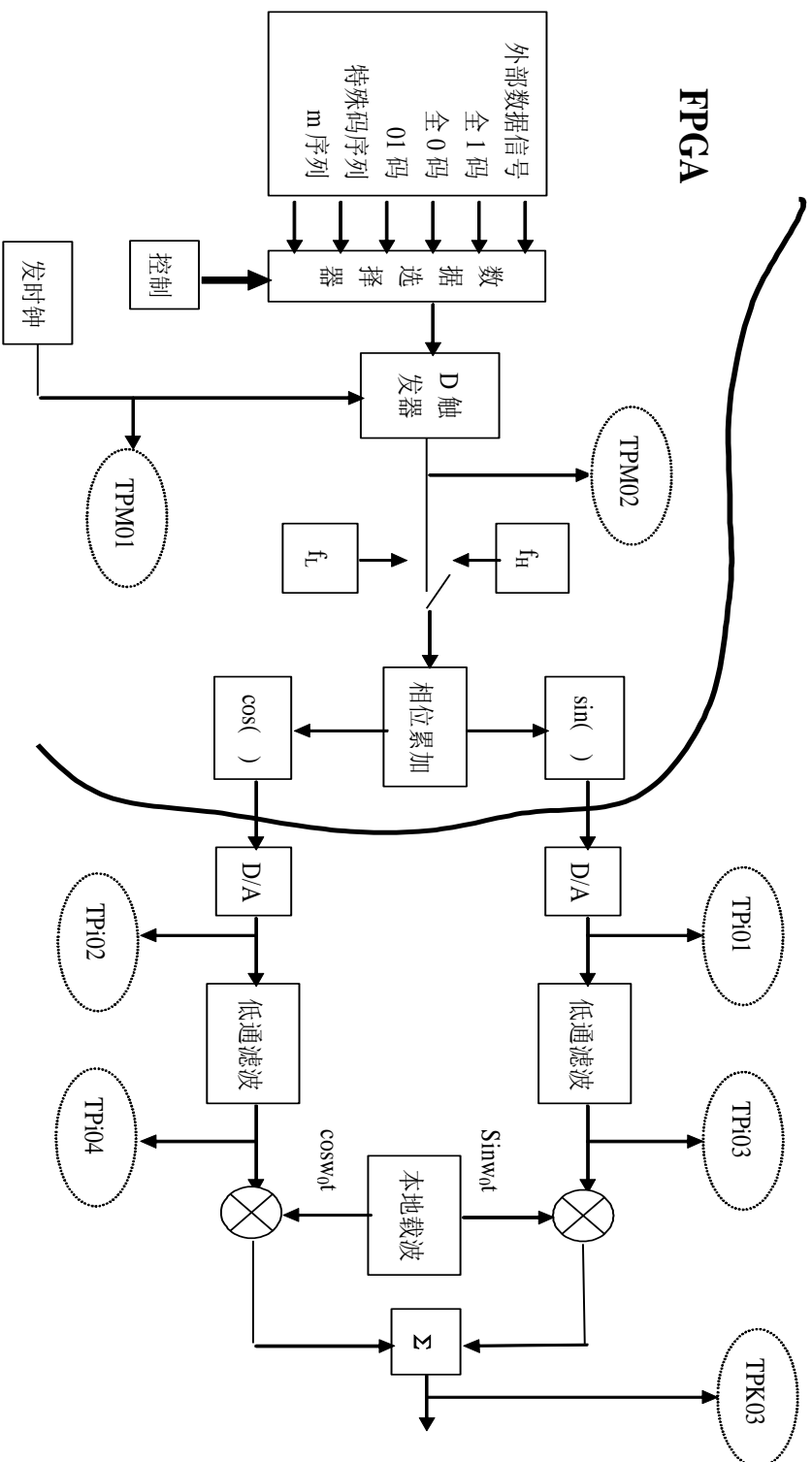


图 4.1.5 FSK 调制器基带处理结构示意图

(二) FSK 解调

对于 FSK 信号的解调方式很多：相干解调、滤波非相干解调、正交相乘非相干解调。

1、FSK 相干解调

FSK 相干解调要求恢复出传号频率 (f_H) 与空号频率 (f_L)，恢复出的载波信号分别与接收的 FSK 中频信号相乘，然后分别在一个码元内积分，将积分之后的结果进行相减，如果差值大于 0 则当前接收信号判为 1，否则判为 0。相干 FSK 解调框图如图 4.1.6 所示：

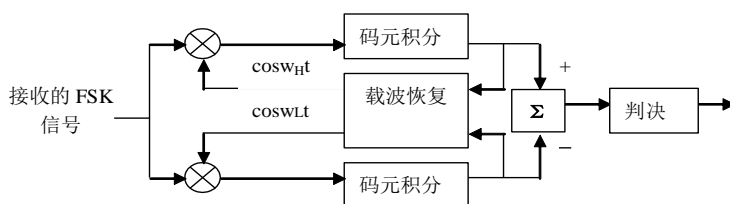


图 4.1.6 相干 FSK 的解调框图

相干 FSK 解调器是在加性高斯白噪声信道下的最佳接收，其误码率为：

$$P_e = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

相干 FSK 解调在加性高斯白噪声下具有较好的性能，但在其它信道特性下情况则不完全相同，例如在无线衰落信道下，其性能较差，一般采用非相干解调方案。

2、FSK 滤波非相干解调

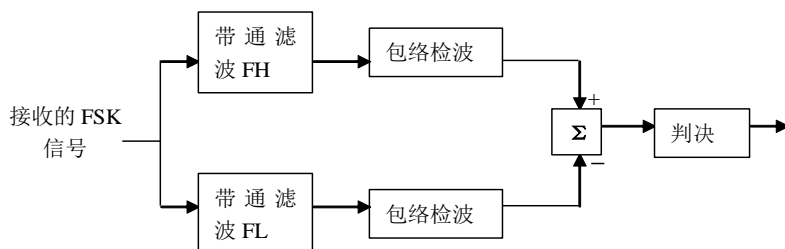


图 4.1.7 非相干 FSK 接收机的方框图

对于 FSK 的非相干解调一般采用滤波非相干解调，如图 4.1.7 所示。输入的 FSK 中频信号分别经过中心频率为 f_H 、 f_L 的带通滤波器，然后分别经过包络检波，包络检波的输

出在 $t=kT_b$ 时抽样（其中 k 为整数），并且将这些值进行比较。根据包络检波器输出的大小，比较器判决数据比特是 1 还是 0。

使用非相干检测时 FSK 系统的平均误码率为：

$$P_e = \frac{1}{2} \exp\left(-\frac{E_b}{2N_0}\right)$$

在 高斯白噪声信道环境下 FSK 滤波非相干解调性能较相干 FSK 的性能要差，但在无线衰落环境下，FSK 滤波非相干解调却表现出较好的稳健性。

FSK 滤波非相干解调方法一般采用模拟方法来实现，该方法不太适合对 FSK 的数字化解调。对于 FSK 的数字化实现方法一般采用正交相乘方法加以实现。

3、FSK 的正交相乘非相干解调

FSK 的正交相乘非相干解调框图如图 4.1.8 所示：

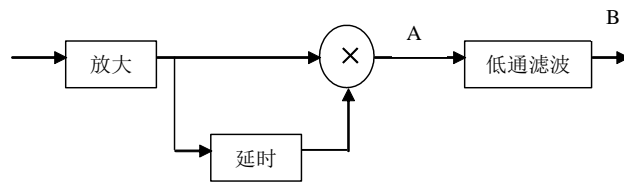


图 4.1.8 FSK 正交相乘非相干解调示意图

输入的信号为

$$R(t) = \cos(w_c t \pm \Delta w \cdot t)$$

传号频率为： $w_c + \Delta w$

空号频率为： $w_c - \Delta w$

在上图中，延时信号为：

$$R'(t) = \cos[(w_c \pm \Delta w) \cdot (t - \tau)]$$

其中 τ 为延时量。

相乘之后的结果为：

$$\begin{aligned}
 2R(t) \cdot R'(t) &= 2\cos(w_c \pm \Delta w) \cdot t * \cos[(w_c \pm \Delta w) \cdot (t - \tau)] \\
 &= \cos[2(w_c \pm \Delta w) \cdot t - (w_c \pm \Delta w) \cdot \tau] + \cos(w_c \pm \Delta w) \cdot \tau
 \end{aligned}$$

在上式中，第一项经过低通滤波器之后可以滤除。当 $w_c \cdot \tau = \pi/2$ 时，上式可简化为：

$$2R(t) \cdot R'(t) \approx \sin(\pm \Delta w) \cdot \tau = \pm \sin \Delta w \tau$$

因而经过积分器（低通滤波器）之后，输出信号大小为： $\pm T_b \sin \Delta w \tau$ ，从而实现了 FSK 的正交相乘非相干解调。

AB 两点的波形如图 4.1.9 所示：

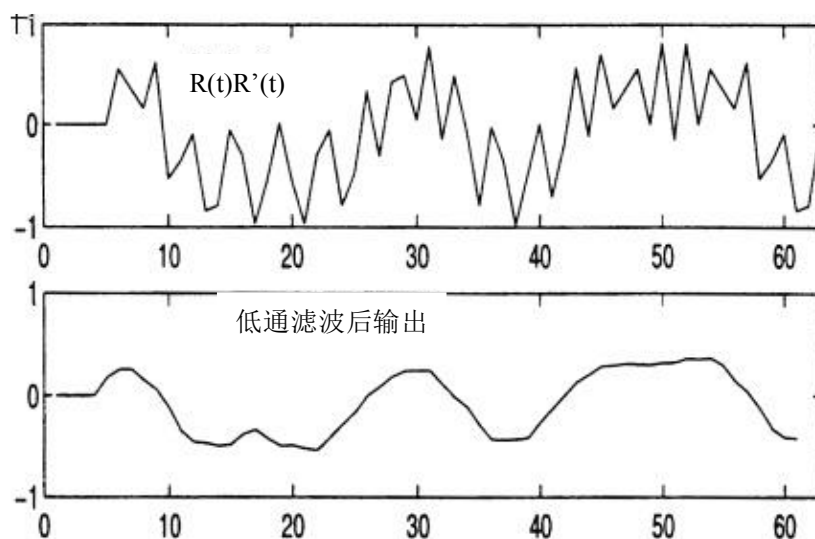


图 4.1.9 差分解调波形

在 FSK 中位定时的恢复见 BPSK 解调方式。

通信原理实验的 FSK 模式中，采样速率为 96KHz 的采样速率（每一个比特采 16 个点），FSK 基带信号的载频为 24KHz，因而在 DSP 处理过程中，延时取 1 个样值。

FSK 的解调框图如图 4.1.10 所示：

注意：FSK 信号首先要和接收端的两个本地正交载波相乘，然后分别通过低通滤波器到达 TPJ05 和 TPJ06。图 4.1.10 中仅画出低通后的电路。

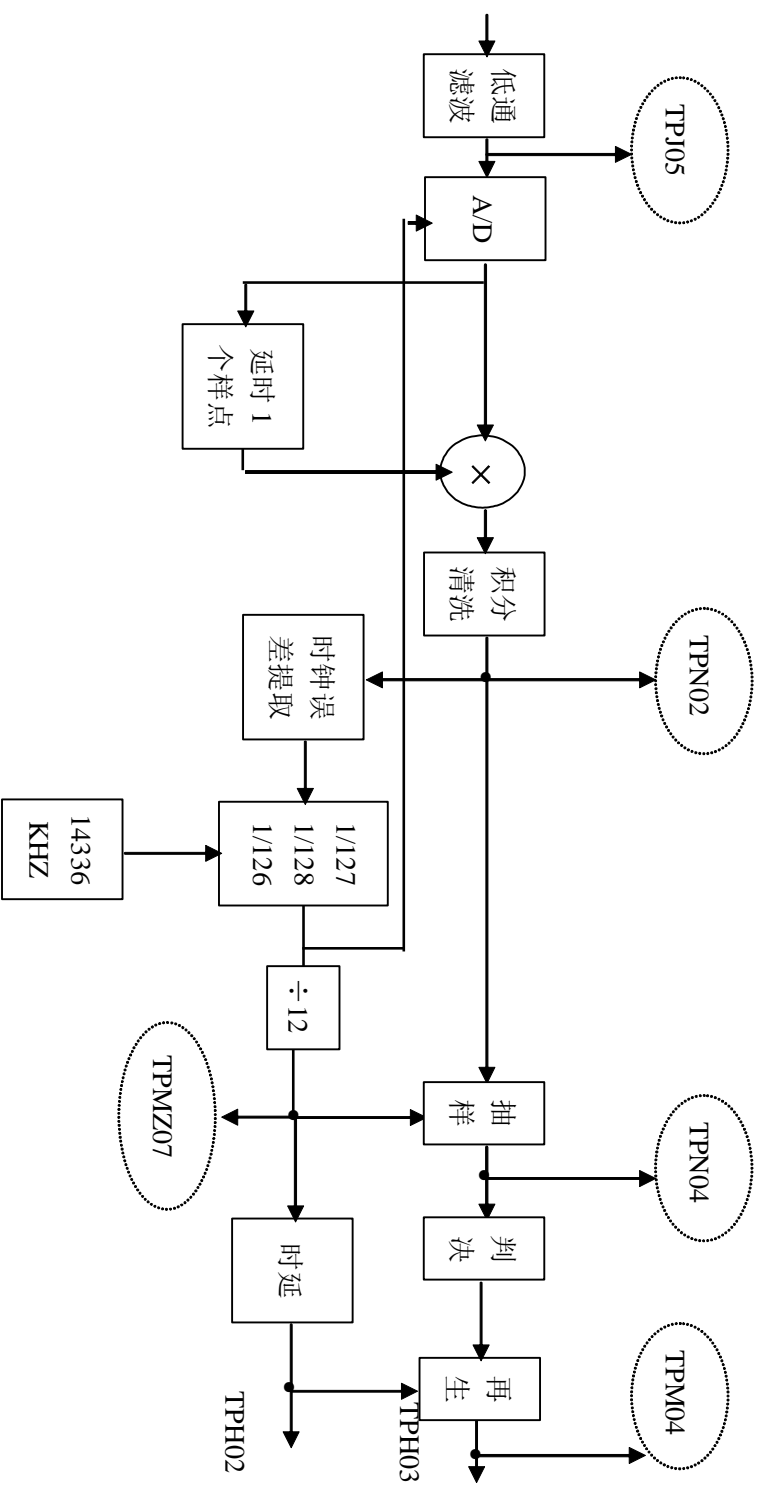


图 4.1.10 FSK 的解调方框图

四、实验步骤

测试前检查：首先将通信原理综合实验系统调制方式设置成“FSK 传输系统”；用示波器测量 TPMZ07 测试点的信号，如果有脉冲波形，说明实验系统已正常工作；如果没有脉冲波形，则需按面板上的复位按钮重新对硬件进行初始化。

（一）FSK 调制

1. FSK 基带信号观测

（1）TPi03 是基带 FSK 波形（D/A 模块内）。通过菜单选择为 1 码输入数据信号，观测 TPi03 信号波形，测量其基带信号周期。

（2）通过菜单选择为 0 码输入数据信号，观测 TPi03 信号波形，测量其基带信号周期。将测量结果与 1 码比较。

2. 发端同相支路和正交支路信号时域波形观测

TPi03 和 TPi04 分别是基带 FSK 输出信号的同相支路和正交支路信号。测量两信号的时域信号波形时将输入全 1 码（或全 0 码），测量其两信号是否满足正交关系。

思考：产生两个正交信号去调制的目的。

3. 发端同相支路和正交支路信号的李沙育（x-y）波形观测

将示波器设置在（x-y）方式，可从相平面上观察 TPi03 和 TPi04 的正交性，其李沙育应为一个圆。

通过菜单选择在不同的输入码型下进行测量。

4. 连续相位 FSK 调制基带信号观测

（1）TPM02 是发送数据信号（DSP+FPGA 模块左下脚），TPi03 是基带 FSK 波形。

测量时，通过菜单选择为 0/1 码输入数据信号，并以 TPM02 作为同步信号。观测 TPM02 与 TPi03 点波形应有明确的信号对应关系。并且，在码元的切换点发送波形的相位连续。

思考：非连续相位 FSK 调制在码元切换点的相位是如何的。

（2）通过菜单选择为特殊序列码输入数据信号，重复上述测量步骤。记录测量结果。

5. FSK 调制中频信号波形和频谱的观测

在 FSK 正交调制方式中，必须采用 FSK 的同相支路与正交支路信号；不然如果只采

一路同相 FSK 信号进行调制，会产生两个 FSK 频谱信号，这需在后面采用较复杂的中频窄带滤波器，如图 4.1.12 所示：

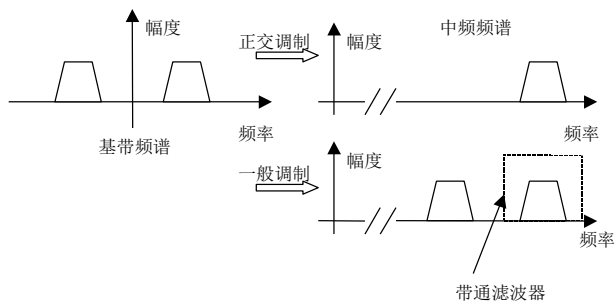


图 4.1.12 FSK 的频谱调制过程

- (1) 利用 SYSTEMVIEW 仿真电路,对测试点 TPK03 的 FSK 调制中频信号观测点进行观察，并进行频谱的测量和分析。
- (2) 将正交调制输入信号中的一路基带调制信号断开，重复上述测量步骤，观测频谱信号的变化，分析变化原因。

(二) FSK 解调

1. 解调基带 FSK 信号观测

首先用中频电缆连结 KO02 和 JL02，建立中频自环（自发自收）。

测量 FSK 解调基带信号测试点 TPJ05 的波形，观测时仍用发送数据（TPM02）做同步，比较其两者的对应关系。

- (1) 通过菜单选择为 1 码（或 0 码）输入数据信号，观测 TPJ05 信号波形，测量其信号周期。
- (2) 通过菜单选择为 0/1 码（或特殊码）输入数据信号，观测 TPJ05 信号波形。

2. 解调基带信号的李沙育（x-y）波形观测

将示波器设置在（x-y）方式，从相平面上观察 TPJ05 和 TPJ06 的李沙育波形。

- (1) 通过菜单选择为 1 码（或 0 码）输入数据信号，仔细观测其李沙育信号波形。
- (2) 通过菜单选择为 0/1 码（或特殊码）输入数据信号，仔细观测李沙育信号波形。

3. 接收位同步信号相位抖动观测

用发送时钟 TPM01（DSP+FPGA 模块左下脚）信号作同步，选择不同的测试码序列测量接收时钟 TPMZ07（DSP 芯片左端）的抖动情况。

思考：为什么在全 0 或全 1 码下观察不到位定时的抖动？

4. 解调器位定时恢复与最佳抽样判决点波形观测

TPMZ07 为接收端 DSP 调整之后的最佳判决抽样时刻。选择输入测试数据为 m 序列，用示波器同时观察 TPMZ07（观察时以此信号作同步）和观察抽样判决点 TPN04 波形（抽样判决点信号）的之间的相位关系。

5. 位定时锁定和位定时调整观测

TPMZ07 为接收端恢复时钟，它与发端时钟（TPM01）具有明确的相位关系。

- （1）在输入测试数据为 m 序列时，用示波器同时观察 TPM01（观察时以此信号作同步）和 TPMZ07（收端最佳判决时刻）之间的相位关系。
- （2）不断按确认键，此时仅对 DSP 位定时环路初始化，让环路重新调整锁定，观察 TPMZ07 的调整过程和锁定后的相位关系。
- （3）在测试数据为全 1 或全 0 码时重复该实验，并解释原因。断开 JL02 接收中频环路，在没有接收信号的情况下重复上述步骤实验，观测 TPM01 和 TPMZ07 之间的相位关系，并解释测量结果的原因。

五、实验报告

- 1、FSK 正交调制方式与传统的一般 FSK 调制方式有什么区别？其有哪些特点？
- 2、TPi03 和 TPi04 两信号具有何关系？
- 3、画出各测量点的工作波形；

附录 实验系统概述

电路组成概述

通信原理综合实验系统主要由下列功能模块组成：

1. 显示控制模块
2. FPGA 初始化模块
3. 信道接口模块
4. DSP+FPGA 处理模块
5. D/A 模块
6. 中频调制模块
7. 中频解调模块
8. A/D 模块
9. 测试模块
10. 汉明编码模块（含加扰模块）
11. 汉明译码模块（含解扰模块）
12. 噪声模块
13. 电话接口（1、2）模块
14. DTMF（1、2）模块
15. PAM 模块
16. ADPCM（1、2）模块
17. CVSD 发模块
18. CVSD 收模块
19. 帧传输复接模块
20. 帧传输解复接模块
21. AMI/HDB3 码模块
22. CMI 编码模块

-
23. CMI 译码模块
 24. 模拟锁相环模块
 25. 数字锁相环模块
 26. 计算机接口模块
 27. 同步数据接口模块
 28. 函数信号产生器模块

硬件平台按功能模块划分，其电路布局见图 1.2.1 所示。对于每一个模块，在 PCB 板上均由白色线条将其明显分割开来，每个测试模块都能单独开设实验。

交流电源插座和电源开关在机箱的后部，直流电源模块在电路板的下面，它主要完成交流~220V 到+5V、+12V、-12V 的直流变换，给整个硬件平台供电。

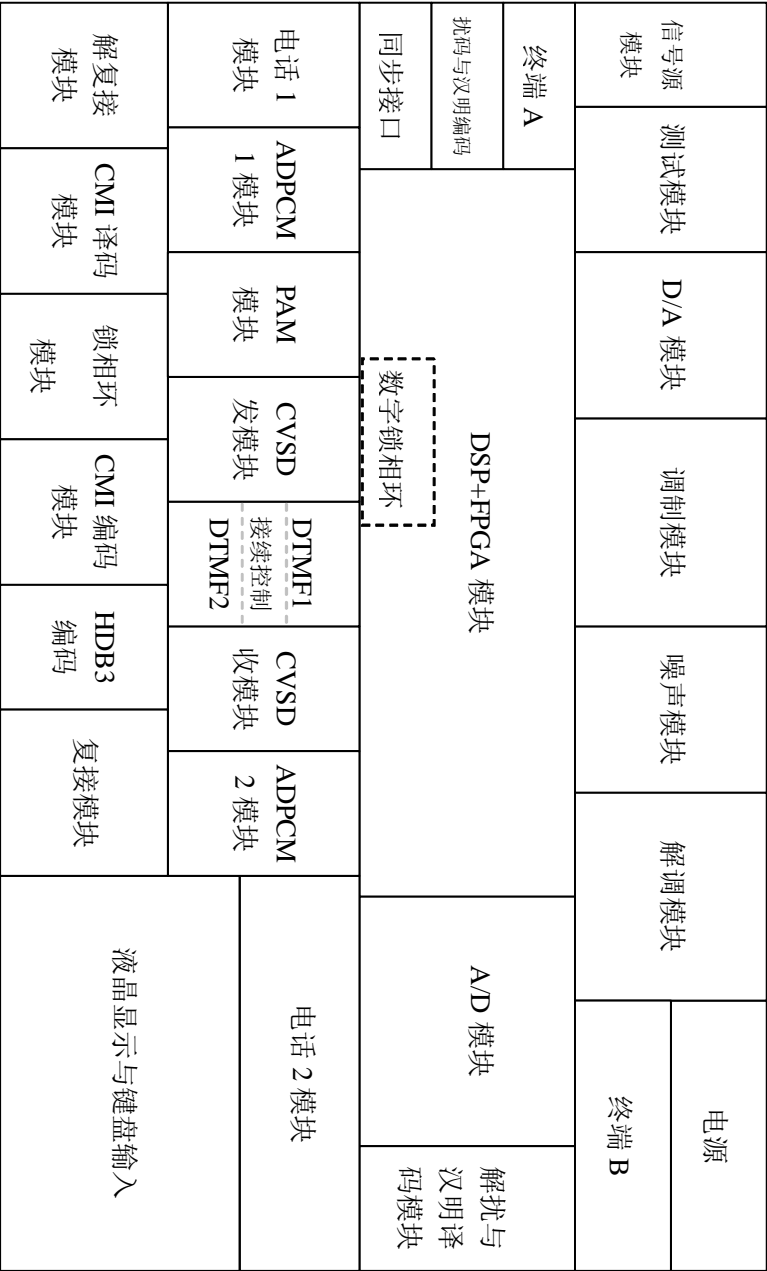


图 1.2.1 通信原理综合实验系统布局图

各模块之间的系统连接见图 1.2.3 所示。由图可以看出，在系统中通信双方的传输信道是不对称的。

从用户电话 1 向用户电话 2 的信号支路是以无线信道传输技术为主，信号流程为：用户电话接口 1→话音编码 1→汉明纠错编码→信道调制→加噪信道→信道解调→汉明译码→话音解码 2→用户电话接口 2。

从用户电话 2 向用户电话 1 的信号支路是以有线信道传输技术为主，信号流程为：用户电话接口 2→话音编码 2→信道复接→线路编码（HDB3/CMI）→线路译码→信道解复接→话音解码 1→用户电话接口 1。

在每一个模块中，都有测试点与测试插座对应信号点的定义。

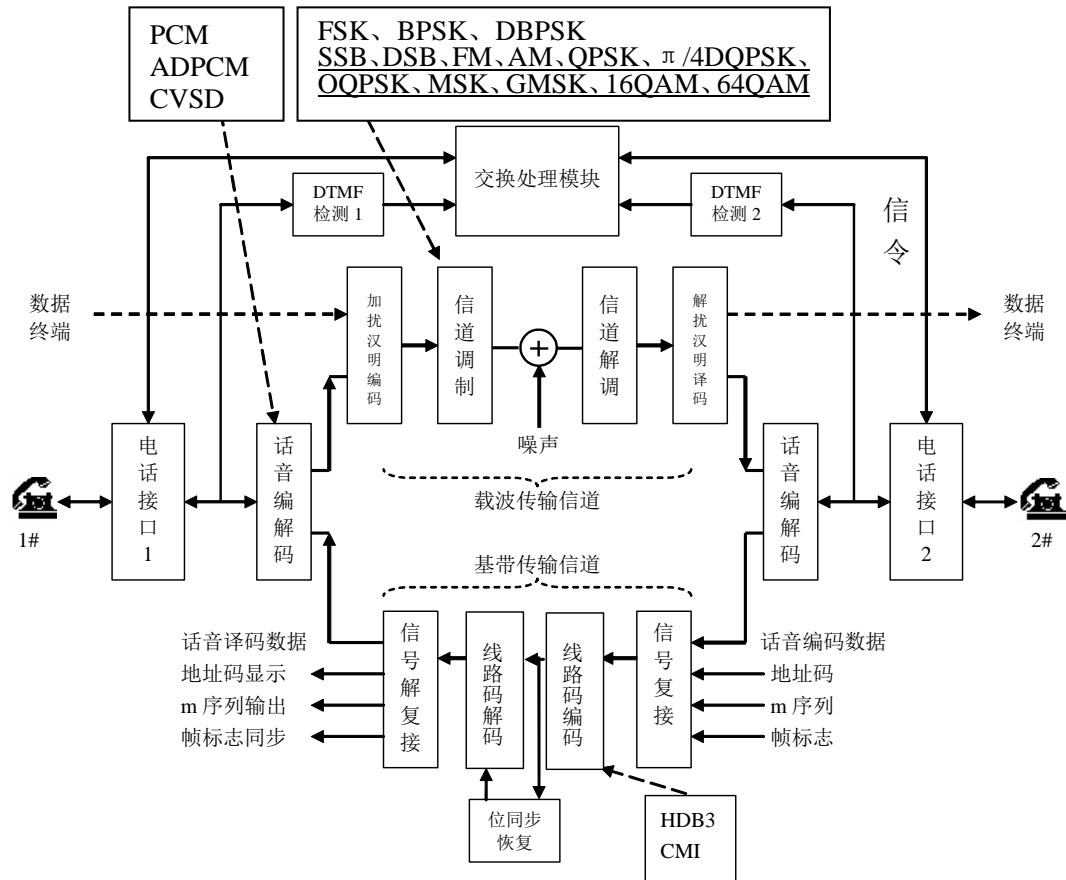


图 1.2.3 各电路测试模块间连接框图

用户操作界面使用说明

● 实验设备加电

交流电源插座和电源开关在机箱的后部。

在加电之后，系统按照上次关机前选择的模式进行初始化，在这期间 DSP+FPGA 模块中的初始化灯（DV01）熄灭。当初始化完成之后，初始化灯变亮。在这之后大约经过 5 秒钟，即自动完成相应模式参数的设置。

在初始化过程中，液晶显示器显示：

通信原理实验

完成初始化之后，液晶显示器显示：

调制方式选择

● 小键盘及液晶显示器菜单的使用

在通信原理综合实验系统中，各模块的功能实现，需初始化不同的 FPGA 程序与数字信号处理 DSP 程序，并对它们进行一定的管理。这些都是通过小键盘和液晶显示器操作界面进行选择的。

液晶显示器操作界面以菜单的方式进行显示，通过小键盘的上、下、左、右键进行菜单的选择。左、右键用来在子菜单之间进行切换，上、下键用来在菜单项中进行选择，最后用“确认”键进行确认。已选定的模式或参数在对应的菜单项上被打上勾，否则显示一个小手。

菜单的组成如下所示：

子菜单一：**调制方式选择**

菜单项 1：**FSK 传输系统**

菜单项 2：**BPSK 传输系统**

菜单项 3：**DBPSK 传输系统**

子菜单二: 调制器输入信号

菜单项 1: 外部数据信号

菜单项 2: 全 1 码

菜单项 3: 全 0 码

菜单项 4: 0 / 1 码

菜单项 5: 特殊码序列

菜单项 6: m 序列

子菜单三: 工作方式选择

菜单项 1: 匹配滤波

菜单项 2: PCM

菜单项 3: ADPCM

子菜单四: 增强调制选择

菜单项 1: AM

菜单项 2: FM

菜单项 3: QPSK

菜单项 4: OQPSK

菜单项 5: PI4QPSK

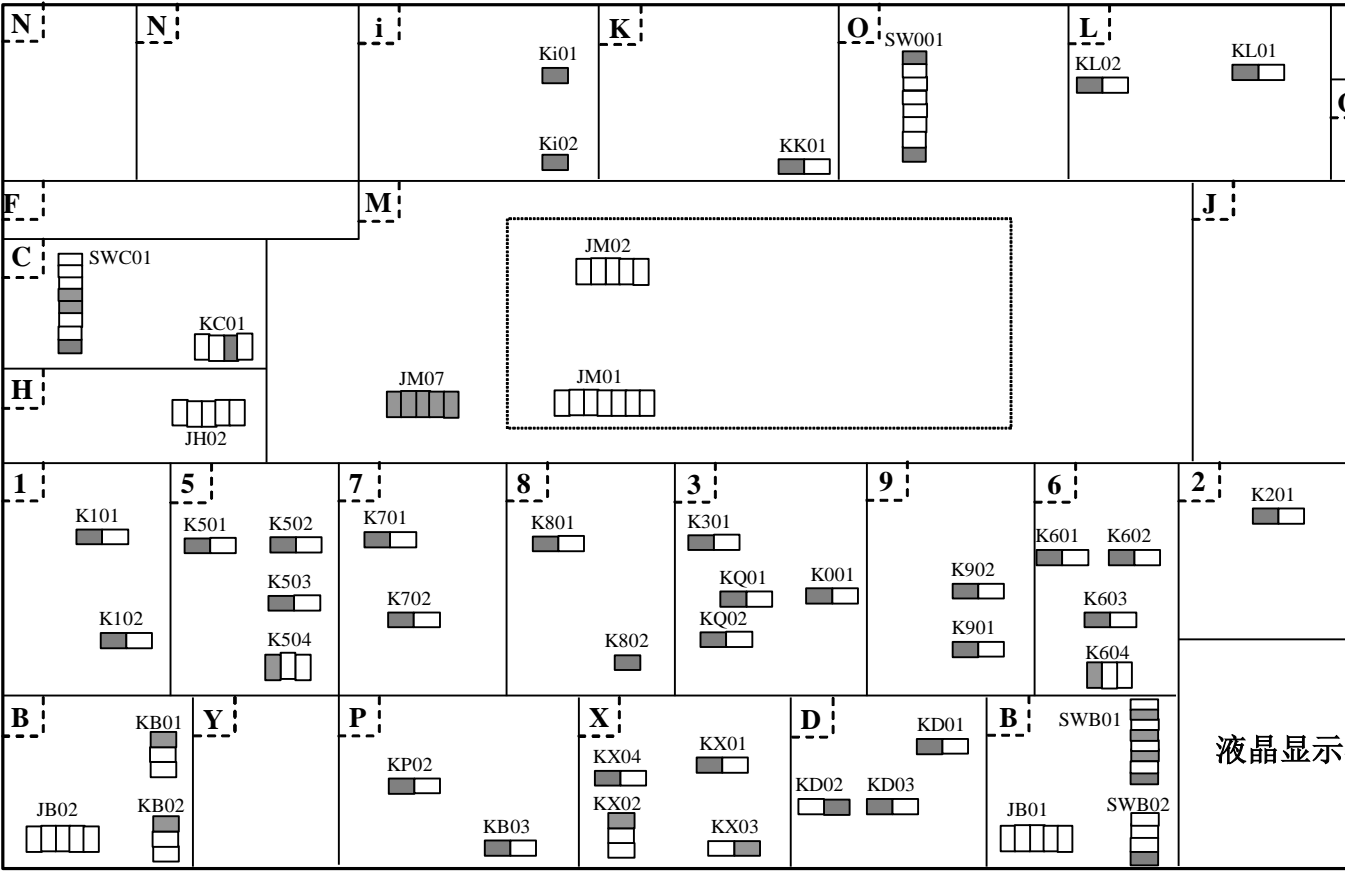
菜单项 6: MSK

菜单项 7: GMSK

菜单项 8: 16QAM

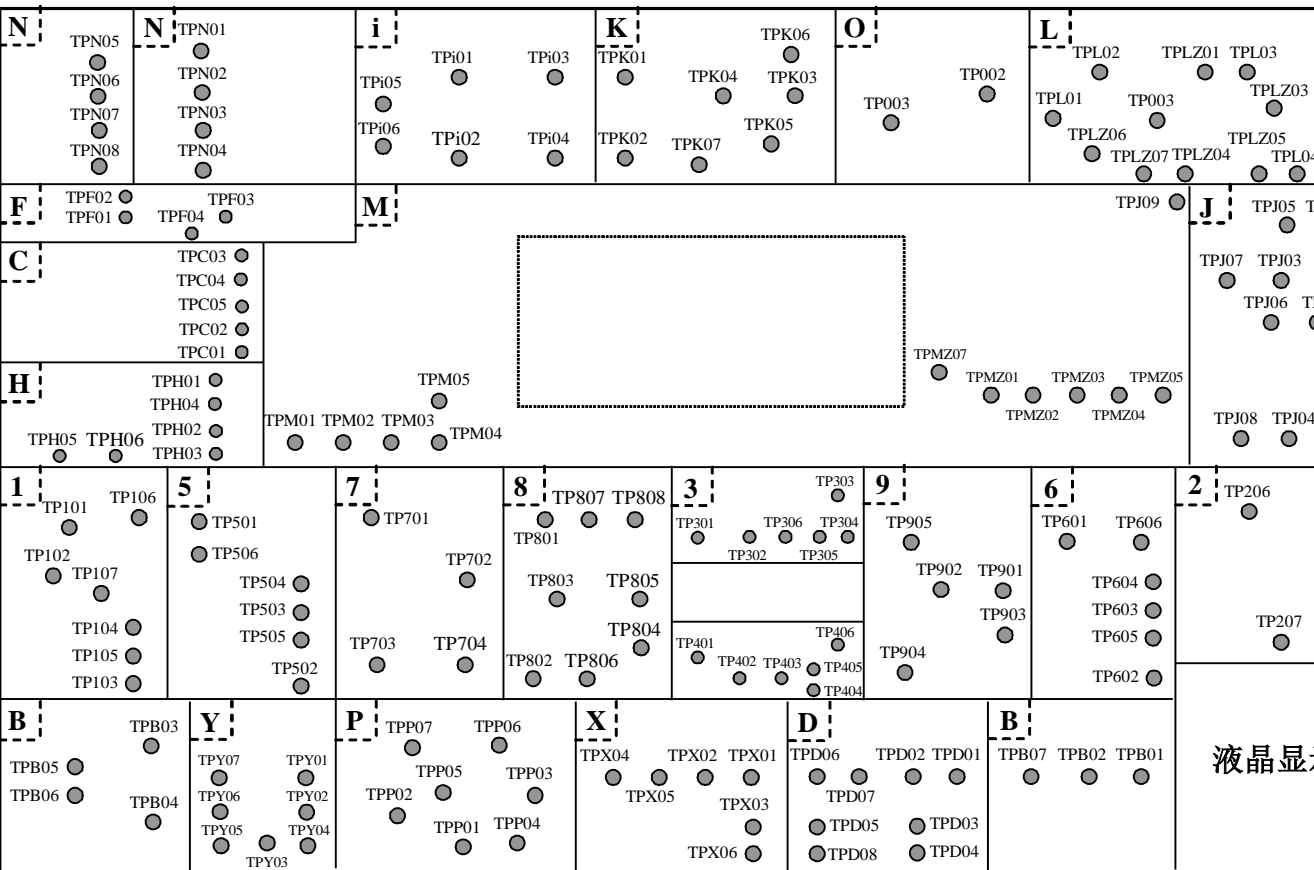
菜单项 9: 64QAM

ZH7001 跳线器默认位置状态图



注： 虚线框内字母为测试点 TPX0Y 中的 X
深色的表示插入跳线帽，白色的表示未插入

ZH7001 测试孔默认位置状态图



《现代通信原理》实验指导书(下)

实验题目和实验报告

- 1, 实验报告必须独立完成;
- 2, 实验报告中预习部分请在实验前完成。

上课时间: _____ 学年第 _____ 学期

院 系: _____

学 号: _____

姓 名: _____

上课教师: _____

实验指导教师: _____

实验课程成绩: _____

信息学院

2013-09

实验一报告 MATLAB 基础实验(1 次试验)

一、实验目的

了解 MATLAB 程序设计语言的基本特点, 熟悉 MATLAB 软件运行环境

掌握创建、保存、打开 m 文件及函数的方法

掌握二维平面图形的绘制方法, 能够使用这些方法进行常用的数据可视化处理

理解周期信号的傅里叶级数展开的物理意义

掌握信号的傅里叶变换及其反变换

二、实验原理

掌握变量等有关概念, 具备初步的将一般数学问题转化为对应的计算机模型并进行处理的能力

1. 周期信号的傅里叶级数

若一周期信号 $f(t) = f(t + kT)$, 其中 k 为整数, T 成为信号的周期。若周期信号在一个周期内可积, 则可通过傅里叶级数对该信号进行展开。其傅里叶展开式如下:

$$f(t) = \sum_{n=-\infty}^{\infty} F_n e^{j2\pi n f_s t}, \quad F_n = \frac{1}{T} \int_{-T/2}^{T/2} f(t) e^{-j2\pi n f_s t} dt$$

其中, T 为信号最小周期; $f_s = 1/T$ 为信号的基波; F_n 为傅里叶展开系数, 其物理意义为频率分量 $n f_s$ 的幅度和相位。

2. 信号的傅里叶变换及其反变换

对于非周期信号 $s(t)$, 满足绝对可积的条件下, 可利用傅里叶变换对其进行频域分析。

$$S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi f t} dt, \quad s(t) = \int_{-\infty}^{\infty} S(f) e^{j2\pi f t} df$$

其中, $S(f)$ 称为信号 $s(t)$ 傅里叶变换, 表示了该信号的频谱特性。

三、实验内容及步骤

1. 在 Command Window 里面计算

① $(3+5+8) \div 5 \times 10$;

② $\sin(3\pi) \div \sqrt{9/5}$;

③ $A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 8 \end{bmatrix}, B = \begin{bmatrix} 7 & 8 & 8 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \end{bmatrix}$, 计算: $C = A \times B, D = A + B, A \setminus C, C / B$;

④ $A = \begin{bmatrix} 3 & 1.2 & 4 \\ 7.5 & 6.6 & 3.1 \\ 5.4 & 3.4 & 6.1 \end{bmatrix}$, 求 $A', A^{-1}, |A|$;

⑤ $Z = \begin{bmatrix} 1+2i & 3+4i \\ 5+6i & 7+8i \end{bmatrix}$, 输入复数矩阵;

$$\begin{bmatrix} 1 & 2 & \cdots & 50 \\ 2 & 3 & \cdots & 51 \\ \vdots & \vdots & \vdots & \vdots \\ 50 & 51 & \cdots & 99 \end{bmatrix}$$

2.建立.m 文件, 用 for 循环语句生成 50×50 的矩阵 A: 将 A 矩阵进行水平和垂直翻转得到矩阵 B 和 C。将 A 矩阵的前 10 行, 10 列变成 0 并赋值给 D。

3.建立.m 文件, 随机产生一个 50×50 的矩阵, 元素值为从 0 到 255, 要求用 0 和 255 对该矩阵进行标记, 元素值大于 128 的标记为 255, 元素值小于 128 的标记为 0。

4.产生一个均值为 2.4 方差为 0.2 大小为 3×4 的随机矩阵。

5. 假设 N=12。对于 M=4, 5, 7, 10, 在 $0 \leq n \leq 2N-1$ 区间上画出 $x_M[n] = \sin\left(\frac{2\pi M n}{N}\right)$, 并添上适当标注。用 plot 和 stem 分别绘制该信号, 并比较。

6. 设周期信号一个周期[0, T]的波形为 $s(t) = \begin{cases} 1, & 0 \leq t \leq T/2 \\ 0, & T/2 < t \leq T \end{cases}$, 其中 T=1。求该信号傅里叶级数展开式, 并用 MATLAB 画出傅里叶级数展开后的波形, 并通过展开式项数的变化考察其对 $s(t)$ 的逼近程度, 考察其物理意义。

7. 设非周期信号 $s(t) = \begin{cases} 1, & 0 \leq t \leq 1/2 \\ 0, & 1/2 < t \leq 1 \end{cases}$, 求该信号的傅里叶变换, MATLAB 画出傅里叶变换后的频谱, 并对频谱进行反变换, 画出 $s(t)$ 的波形。

三、预习报告: (原理, 流程图, 编程思想, 基本程序)

四、实验报告: (调试好的程序, 实验结果及分析)

实验二 (2 小班) 连续波 MATLAB 实现

一、实验目的

二、实验原理

三、实验内容 (见实验指导)

四、预习报告: (原理, 流程图, 编程思想, 基本程序)

五、实验报告: (调试好的程序, 实验结果及分析)

实验二 (1 小班) 模拟信号的数字传输

一、实验目的

掌握低通信号抽样定理

理解 13 折线 A 率逐次比较型 PCM 编码仿真的思想

二、实验原理

1. 低通信号的抽样定理

一频带限制在 $(0, f_H)$ 赫内的时间连续信号 $m(t)$ ，若以 $f_s \geq 2f_H$ 速率对 $m(t)$ 等间隔 $T_s = 1/f_s \leq 1/2f_H$ 抽样，则 $m(t)$ 将被所得抽样函数 $m_s(t)$ 完全确定。

2. 验证低通信号的抽样定理

抽样的过程是将输入的模拟信号与抽样信号相乘而得，通常抽样信号是一个周期为 T_s 的周期脉冲信号，抽样后得到的信号称为抽样序列。理想抽样信号定义如下：

$$\sigma_T(t) = \sum_n P(t - nT_s)$$

其中， $P(t) = \begin{cases} 1 & t = 0 \\ 0 & t \neq 0 \end{cases}, f_s = \frac{1}{T_s}$ 称为抽样速率。

因此抽样后的信号为

$$x_s(t) = x(t)\sigma_T(t) = \sum_{k=-\infty}^{\infty} x(kT_s)P(t - kT_s)$$

经带宽为 f_H 低通滤波器后可恢复抽样信号。

3. PCM 的基本原理如图 1 所示：

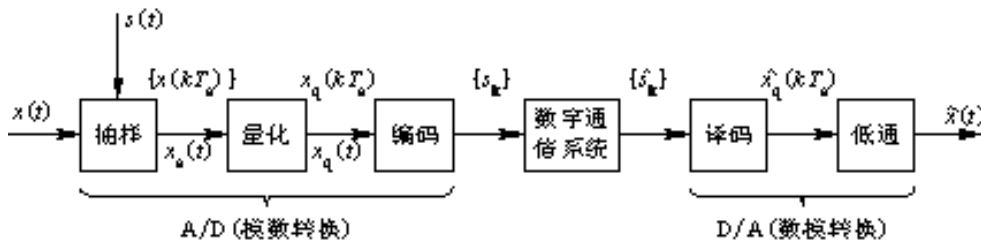


图 1 PCM 的基本原理

抽样是对模拟信号进行周期性的扫描，把时间上连续的信号变成时间上离散的信号。我们要求经过抽样的信号应包含原信号的所有信息，即能无失真地恢复出原模拟信号，抽样速率的下限由抽样定理确定。

量化是把经抽样得到的瞬时值进行幅度离散，即指定 Q 规定的电平，把抽样值用最接近的电平表示。

编码是用二进制码组表示有固定电平的量化值。实际上量化是在编码过程中同时完成的。

4. 逐次比较型 PCM 编码的产生原因：

①均匀量化时其量化信噪比随信号电平的减小而下降。产生这一现象的原因就是均匀量化时的量化级间隔 Δ 为固定值，而量化误差不管输入信号的大小均在 $(-\Delta/2, \Delta/2)$ 内变化。故大信号时量化信噪比大，小信号时量化信噪比小。对于语音信号来说，小信号出现的概率要大于大信号出现的概率，这就使平均信噪比下降。同时，为了满足一定的信噪比输出要求，输入信号应有一定范围(即动态范围)，由于小信号信噪比明显下降，也使输入信号范围减小。要改善小信号量化信噪比，可以采用量化间隔非均匀的方法，即非均匀量化。

②压扩特性曲线用 A 律压扩特性曲线，工程上使用 13 折线来近似 A 律压扩特性曲线，目的是为了使用逐次比较算法来提高运算速度。

5. 逐次比较型编码的编码方法

码位安排：

极性码	段落码	段内码
M1	M2M3M4	M5M6M7M8

其中：

- ① 第一位 M1 表示量化值的极性正负。M1=1 代表信号极性为正 M1=0 代表信号极性为负；后面 7 位分为段落码和段内码两部分,用于表示量化值的绝对值。
- ② M2M3M4 为段落码，分别对应表 1 段落编码
- ③ M5M6M7M8 为段内码，分别对应表 1 PCM 编码的后四位。

三、实验内容

1. 利用 MATLAB 软件验证低通抽样定理，若低通信号为

$$x(t) = 0.1\cos(0.15\pi t) + 0.5\cos(4\pi t)$$

①画出该低通信号的波形；

②画出抽样速率为 $f_s = 4Hz$ 的抽样序列，画出抽样后的信号波形；

③画出经低通滤波器恢复的波形。

2. 设输入一个样值 $x \in [-2048, +2048]$ ，对 x 进行 A 律 PCM 编码。要求编写成函数，该函数输入变量为样值，输出变量为 A 律 13 折线逐次比较的八位 PCM 编码，设码元宽度为 1，画出其波形。

四、预习报告：（原理，流程图，编程思想，基本程序）

五、实验报告：（调试好的程序，实验结果及分析）

实验三 眼图（基带调制）

一、实验目的

熟悉基带调制信号的产生方法；掌握眼图的生成方法和分析方法

二、实验原理

三、实验内容

分别产生符合下列要求的 BPSK 和 4ASK 调制波形的眼图，设符号率为 1Hz，要求调制波形经过升余弦滚降成形(缺省滚降系数为 0.5)：

- 5) 无噪声和 20dB 加性高斯白噪声
- 6) 升余弦滚降系数分别为 0.5 和 1
- 7) 受到码间串扰影响，信道参数可设为：ch=[0.9806; -0.1961];

四、预习报告：（原理，流程图，编程思想，基本程序）

五、实验报告：（调试好的程序，实验结果及分析）

实验四 调制解调（BPSK, QPSK, 信噪比）

一、实验目的

掌握数字频带传输系统调制解调的仿真过程

掌握数字频带传输系统误码率仿真分析方法

二、实验原理

数字频带信号通常也称为数字调制信号，其信号频谱通常是带通型的，适合于在带通型信道中传输。数字调制是将基带数字信号变换成适合带通型信道传输的一种信号处理方式，正如模拟通信一样，可以通过对基带信号的频谱搬移来适应信道特性，也可以采用频率调制、相位调制的方式来达到同样的目的。

1. 调制过程

1) 2ASK

如果将二进制码元“0”对应信号0，“1”对应信号 $A\cos 2\pi f_c t$ ，则2ASK信号可以写成如下表达式：

$$s_T(t) = \left\{ \sum_n a_n g(t - nT_s) \right\} A \cos 2\pi f_c t$$

$$a_n \in \{0,1\}, \quad g(t) = \begin{cases} 1 & 0 \leq t \leq T_s \\ 0 & \text{其他} \end{cases}$$

可以看到，上式是数字基带信号 $m(t) = \sum_n a_n g(t - nT_s)$ 经过DSB调制后形成的信号。其调制框图如图1所示：

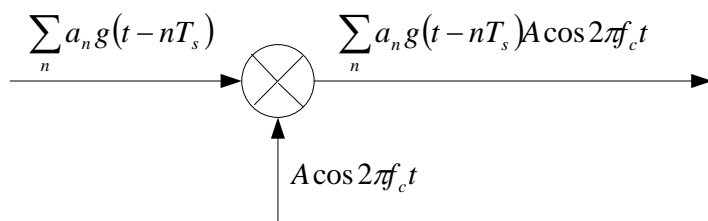


图1 2ASK信号调制框图

2ASK信号的功率谱密度为：

$$P_s(f) = \frac{A^2}{4} [P_m(f - f_c) + P_m(f + f_c)]$$

2) 2FSK

将二进制码元“0”对应载波 $A\cos 2\pi f_1 t$ ，“1”对应载波 $A\cos 2\pi f_2 t$ ，则形成2FSK信号，可以写成如下表达式：

$$s_T(t) = \sum_n \bar{a}_n g(t - nT_s) A \cos(2\pi f_1 t + \varphi_n) + \sum_n a_n g(t - nT_s) A \cos(2\pi f_2 t + \theta_n)$$

当 $a_n = 0$ 时，对应的传输信号频率为 f_1 ；当 $a_n = 1$ 时，对应的传输信号频率为 f_2 。上式中， φ_n 、

θ_n 是两个频率波的初相。2FSK也可以写成另外的形式如下：

$$s_T(t) = A \cos \left(2\pi f_c t + 2\pi h \sum_{n=-\infty}^{\infty} a_n g(t - nT_s) \right)$$

其中, $a_n \in \{+1, -1\}$, $f_c = (f_1 + f_2)/2$, $g(t) = \begin{cases} 1 & 0 \leq t \leq T_s \\ 0 & \text{其他} \end{cases}$, $h = |f_2 - f_1|$ 为频偏。

2FSK 信号也可以看成是两个不同载波的 2ASK 信号的叠加:

$$s_T(t) = s_1(t) \cos(\omega_1 t + \phi_1) + s_2(t) \cos(\omega_2 t + \phi_2)$$

当这两项不相关时 (如载波之间频率差足够大), 它的功率谱密度为:

$$P_s(f) = \frac{1}{4} [P_{s1}(f - f_1) + P_{s1}(f + f_1)] + \frac{1}{4} [P_{s2}(f - f_1) + P_{s2}(f + f_2)]$$

3) 2PSK

将二进制码元 “0” 对应相位为 π 的载波 $-A \cos 2\pi f_c t$, “1” 对应载波相位为 0 的载波 $A \cos 2\pi f_c t$, 则 2PSK 信号可以写成如下表达式:

$$s_T(t) = \left\{ \sum_n a_n g(t - nT_s) \right\} A \cos 2\pi f_c t$$

其中 $a_n \in \{+1, -1\}$, $g(t) = \begin{cases} 1 & 0 \leq t \leq T_s \\ 0 & \text{其他} \end{cases}$ 。其实现框图与 2ASK 相同, 只是输入是双极性的信号。

2PSK 信号的功率谱密度为:

$$P_s(f) = \frac{A^2}{4} [P_m(f - f_c) + P_m(f + f_c)]$$

2. 解调过程

数字频带通信系统相干解调基本框图如图 2 所示。

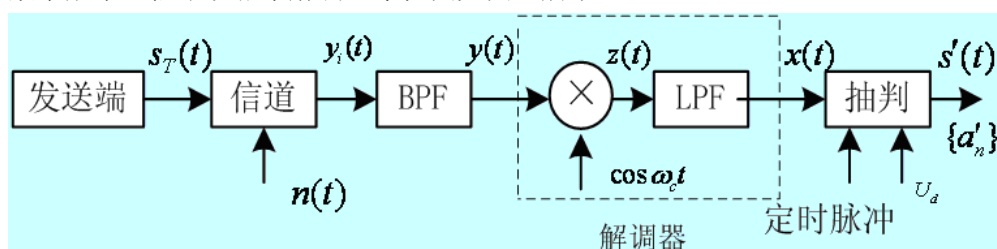


图 2 数字频带通信系统相干解调基本框图

3. 抗噪性能分析

二进制数字频带传输系统普通接收机误码率公式如表 1 所示。其中 r 为解调器输入端信噪比。

表 1 二进制数字频带传输系统普通接收机误码率公式

调制方式	相干解调	非相干解调
2ASK	$\frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{r}{4}} \right)$	$\frac{1}{2} e^{-r/4}$

2FSK	$\frac{1}{2} \operatorname{erfc}\left(\sqrt{\frac{r}{2}}\right)$	$\frac{1}{2} e^{-r/2}$
2PSK	$\frac{1}{2} \operatorname{erfc}(\sqrt{r})$	
2DPSK	$\operatorname{erfc}(\sqrt{r})$	$\frac{1}{2} e^{-r}$

三、实验内容

1.用 Matlab 产生独立等概的二进制信源。

① 画出 2ASK 调制解调中信号波形及其功率谱变化过程；

② 画出 2FSK 调制解调中信号波形及其功率谱变化过程；（设 $|f_1 - f_2| \gg \frac{1}{T_s}$ ）；

③ 画出 2DPSK 调制解调中信号波形及其功率谱。

三个题目任选其一，并在不同信噪比下对其进行抗噪声性能分析。

四、预习报告：（原理，流程图，编程思想，基本程序）

五、实验报告（调试好的程序、实验结果及及分析）

实验五 信道编解码

- 一、实验目的
- 二、实验原理
- 三、实验内容
- 四、预习报告：（原理，流程图，编程思想，基本程序）
- 五、实验报告：（调试好的程序，实验结果及分析）