

《信号与系统》实验课教程

陈淑珍 邹炼
武汉大学电子信息学院

实验一 波形合成与分解

1. 实验目的

在理论学习的基础上，通过本实验熟悉信号的合成、分解原理，了解信号频谱的含义，加深对傅里叶变换性质和作用的理解。

2. 实验原理

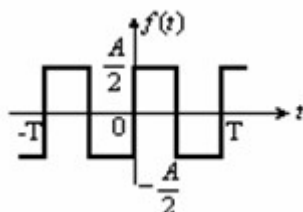
根据傅里叶分析的原理，任何周期信号都可以用一组三角函数 $\{\sin(n\omega_0 t); \cos(n\omega_0 t)\}$ 的组合表示，即：

$$x(t) = a_0 + a_1 \cos(\omega_0 t) + b_1 \sin(\omega_0 t) + a_2 \cos(2\omega_0 t) + b_2 \sin(2\omega_0 t)$$

即可以用一组正弦波和余弦波来合成任意形状的周期信号。

3. 实验内容

(1) 方波的合成



图示方波是一个奇谐信号，由傅里叶级数可知，它是由无穷个奇次谐波分量合成的，本实验用图形的方式来表示它的合成。方波信号可以分解为：

$$x(t) = \frac{2A}{\pi} \sum_{n=1}^{\infty} \sin(2\pi n f_0 t) \cdot \frac{1}{n}, n = 1, 3, 5, 7, 9, \dots$$

用前 5 项谐波近似合成 50Hz，幅值为 3 的方波，写出实验步骤。

a. 只考察从 $t = 0$ s 到 $t = 10$ s 这段时间内的信号。

b. 画出基波分量 $y(t) = \sin(t)$ 。

c. 将三次谐波加到基波之上，并画出结果，并显示。

$$y(t) = \sin(t) + \sin(3*t)/3$$

d. 再将一次、三次、五次、七次和九次谐波加在一起。

$$y(t) = \sin(t) + \sin(3*t)/3 + \sin(5*t)/5 + \sin(7*t)/7 + \sin(9*t)/9$$

e. 合并从基波到十九次谐波的各奇次谐波分量。

f. 将上述波形分别画在一幅图中，可以看出它们逼近方波的过程。注意“吉布斯现象”。周期信号傅里叶级数在信号的连续点收于该信号，在不连续点收敛于信号左右极限的平均值。如果我们用周期信号傅里叶级数的部分和来近似周期信号，在不连续点附近将会出现起伏和超量。在实际中，如果应用这种近似，就应该选择足够大的 N ，以保证这些起伏拥有的能量可以忽略。

(2) 设计谐波合成三角波的实验，写出实验步骤，并完成实验。

(3) 设计分析方波、三角波频谱的分析实验，写出实验步骤，并完成实验（并比较二者频谱的特点）。

4. 实验报告要求

简述实验目的及原理，按实验步骤附上相应的信号波形曲线，总结实验得出的主要结论。

实验二 信号的抽样和内插

1. 实验目的

熟悉信号采样过程，并通过本实验观察欠采样时信号频谱的混迭现象，了解采样前后信号频谱的变化，加深对采样定理的理解，掌握采样频率的确定方法。

2. 实验内容和原理

模拟信号经过 A/D 变换转换为数字信号的过程称之为采样，信号采样后其频谱产生了周期延拓，每隔一个采样频率 f_s ，重复出现一次。为保证采样后信号的频谱形状不失真，采样频率必须大于信号中最高频率成份的两倍，这称之为采样定理。

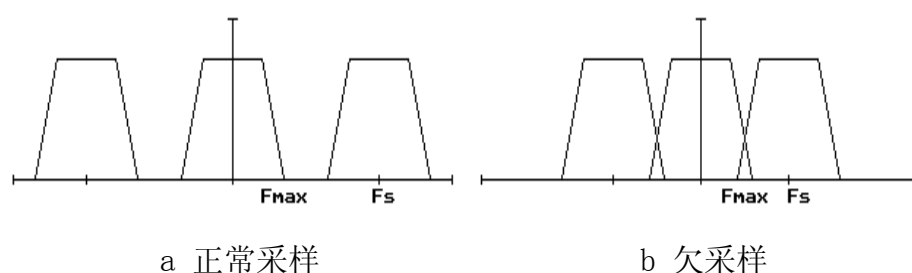


图 1. 采样信号的频混现象

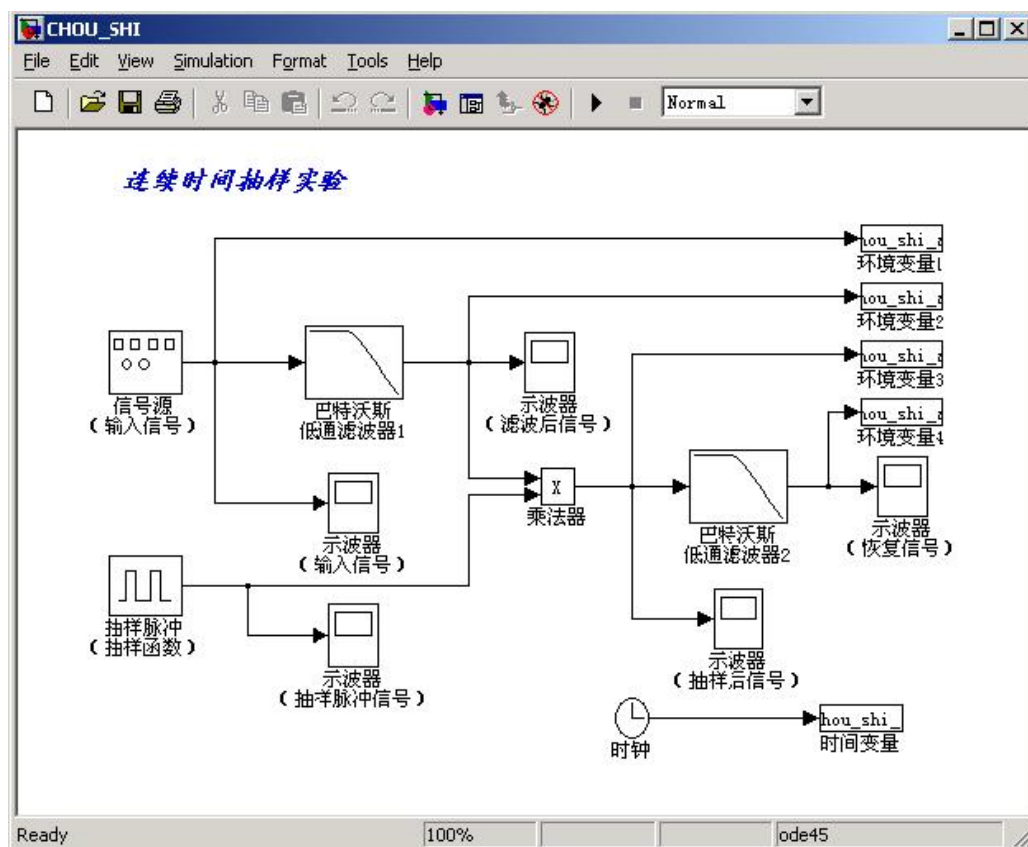
实验内容为设计一模拟信号：

$$x(t) = 3\sin(2\pi ft), \quad f = 6\text{Hz}$$

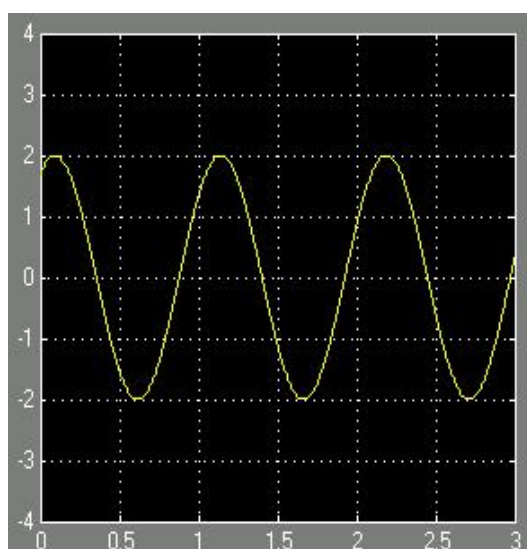
对采样频率 f_s 为正常采样和欠采样时两种情况进行分析，观察欠采样时信号频谱的混迭现象。

3. 实验内容

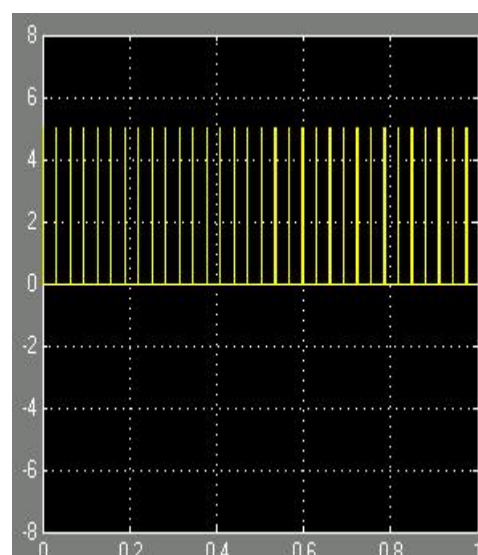
- (1) 熟悉 MATLAB 中 simulink 的用法。
- (2) 根据下图提示是完成信号 $x(t)$ 的抽样和内插试验仿真设计。



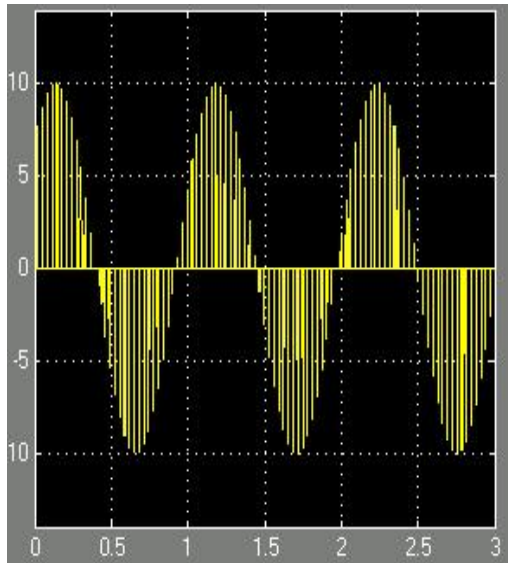
* 运行仿真后各器件的波形如下：



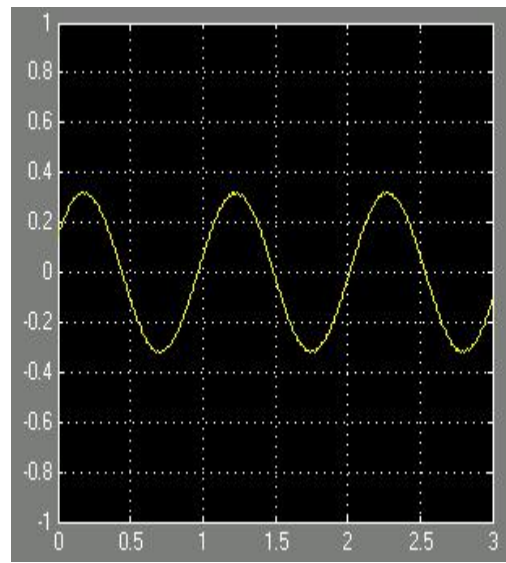
信号源的波形



抽样脉冲的波形



抽样后信号的波形



恢复以后信号的波形

(3) 改变信号源的波形、抽样脉冲的频率，将正弦信号换成方波、三角波后重复实验步骤，思考采样频率如何选择的问题。

4. 实验报告要求

简述实验目的及原理，按实验步骤附上相应的信号波形和频谱曲线，说明采样频率变化对信号时域和频域特性的影响，总结实验得出的主要结论。

实验三 二阶状态轨迹的显示

1. 实验目的

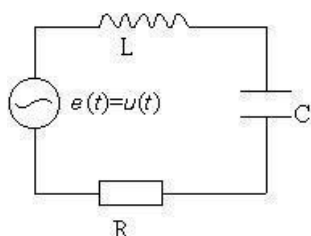
观察 RLC 电路的状态轨迹；观察过阻尼，欠阻尼，临界阻尼情况下系统的单位阶跃响应；明确用示波器显示二阶系统的状态轨迹的原理与显示李沙育图形完全一样。

2. 实验原理

“状态变量”较确切的定义是能描述系统动态特性的一组最少量的数据。对于二阶系统，则可以用两个状态变量来表示，这两个状态变量所形成的空间称为状态空间。在状态空间中状态的端点随时间变化而描出的路径叫状态轨迹。因此状态轨迹点对应系统在不同时刻，不同条件下的状态，知道了某段时间内的状态轨迹，则系统在该时间内的变化过程也就知道了，所以二阶状态轨迹的描述方法是一种在几何平面上研究系统动态性能（包括稳定性在内）的方法。用计算机模拟二阶状态轨迹的显示，方法简单直观，且能很方便观察电路参数变化时，状态轨迹的变化规律。

3. 实验内容

下图是一个典型的离散二阶系统的模型：



对于二阶系统，可用两个状态变量 $i(t)$ 和 $u(t)$ 来表示，这两个状态变量所形成的空间称为状态空间。在状态空间中，状态变量随时间变化而描出的路径叫状态轨迹。本利用计算机模拟该系统的状态轨迹，使你可以很方便的观察电路参数变化时状态轨迹变化的规律。

状态方程：

$$\begin{bmatrix} \dot{i}_L(t) \\ \dot{v}_C(t) \end{bmatrix} = \begin{bmatrix} -R/L & -1/L \\ 1/C & 0 \end{bmatrix} \begin{bmatrix} i_L(t) \\ v_C(t) \end{bmatrix} + \begin{bmatrix} 1/L \\ 0 \end{bmatrix} u(t)$$

设 $\alpha = R/2L$ ， $\omega_0 = 1/\sqrt{LC}$

1. 当 $\alpha > \omega_0$ 时

特征方程 $s^2 + (R/L)s + 1/LC = 0$ 有两个不同实根。

$$\begin{cases} i_L(t) = (1/L)e^{-\alpha t} \text{sh}(\beta t) / \beta u(t) \\ v_C(t) = \frac{e^{-\alpha t}}{2\beta LC} [e^{-\alpha t} / (\alpha + \beta) - e^{-\alpha t} / (\alpha - \beta)] + 1 \end{cases}$$

其中 $\beta = \sqrt{\alpha^2 - \omega_0^2}$

2. 当 $\alpha = \omega_0$ 时，有两相同实根

$$\begin{cases} i_L(t) = (1/L)te^{-\alpha t} \\ v_C(t) = (1/LC\alpha^2)[1 - (\alpha t + 1)e^{-\alpha t}] \end{cases}$$

3. 当 $\alpha < \omega_0$ 时，有一对共轭复根

$$\begin{cases} i_L(t) = \varpi_1 L e^{-\alpha t} \sin \varpi_1 t \\ v_C(t) = 1 - e^{-\alpha t} (\cos \varpi_1 t + \sin \varpi_1 t / \alpha \varpi_1) \end{cases}$$

$\varpi_1 = \sqrt{\omega_0^2 - \alpha^2}$

4. 当 $R = 0$ 时

$$\begin{cases} i_L(t) = L\omega_0 \sin \omega_0 t \\ v_C(t) = 1 - \cos \omega_0 t \end{cases}$$

就是根据这四组方程的表达式 $i_L(t)$ ， $v_C(t)$ 编制程序，显示状态轨迹的。

4. 实验报告与练习

1. 简述用示波器显示李沙育图形的原理和示波器联接的方法。
2. 观察电路参数变化时，状态轨迹和的变化规律。
3. 用 MATLAB 解上述的微分方程。
4. 根据情况自己选做（提高题）：用 MATLAB 或者 JAVA 完成状态轨迹的模拟。

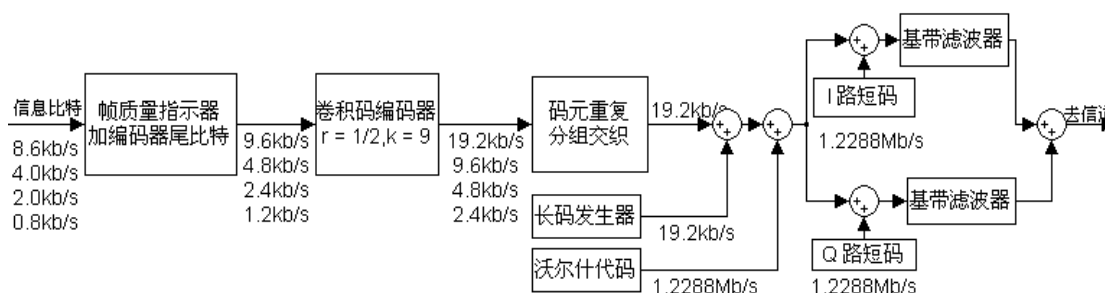
实验四 CDMA 前向数据链路仿真

1. 实验目的

在理论学习的基础上,通过本实验熟悉 IS-95A 前向业务信道的编解码模型,了解数据的传输过程,加深对 IS-95A 理解。

2. 实验原理

窄带 CDMA 的逻辑信道分控制信道和业务信道,其中控制信道包括导频信道、寻呼信道、同步信道和接入信道;业务信道包括前向业务信道和反向业务信道。其中前向业务信道最具有代表性,所以我们重点介绍前向业务信道。其模型可看下图:



基站在前向业务信道可以改变数据速率来传送信息。前向业务信道的帧长为 20ms,各个速率的信号经过增加 CRC 校验位,卷积编码,符号重复和块交织后,速率将固定在 19.2kbps。我们要对这个过程进行仿真。

3. 实验内容

(1) 首先,编写一个程序模块使其能产生 172bits, 80bits, 40bits 或 16bits 的数据,做为前向业务信道中四种不同速率的一帧数据。

(2) 然后,再编写一个程序模块,完成帧质量指示器以及加编码器尾比特。CRC 编码属于循环冗余编码,具有检纠错的能力,只有 9.6kbps 和 4.8kbps 有 CRC 编码器,2.4kbps 和 1.2kbps 没有。其中 9.6kbps 和 4.8kbps 的生成多项式分别为:

$$g(x) = 1 + x + x^4 + x^8 + x^9 + x^{10} + x^{11} + x^{12}$$

$$g(x) = 1 + x + x^3 + x^4 + x^7 + x^8$$

(3) 然后,再编写一个程序模块,完成卷积编码器的功能,卷积编码器的功能模型为:

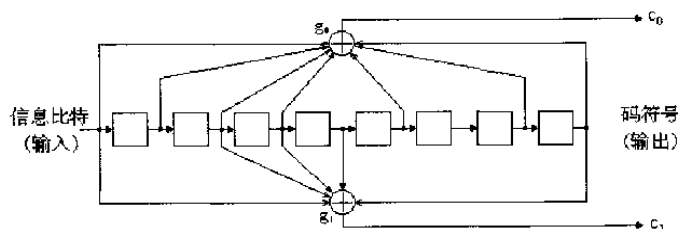


图 3-3 前向链路的卷积编码器: $r=1/2, K=9$

(4) 然后,再编写一个程序模块,使其完成码元重复的功能。并使其输出的码元速率固定在 19.2kbps。

(5) 大家可以找一下移动通信中有关章节,并再好好看一下,争取能够按照书

上的流程继续将这个仿真系统做的更完善一些。比如再加一个 PN 码发生器，Walsh 码发生器，I 路及 Q 路 m 序列发生器，基带滤波器等等。这都是完全有可能的，我们将作为补充练习，不做要求，但希望大家能够想一下，试着做一下。

4. 实验报告要求

简述实验目的及原理，按实验步骤附上相应的信号波形曲线，总结实验得出的主要结论。

实验五 Wav 信号的波形分析

一 实验目的

借助本实验帮助同学们巩固傅里叶变换及其反变换的知识，学习从时域和频域两个角度来观察信号，并尝试利用短时傅里叶变换分析非平稳信号的频谱变化。

二 实验原理

借助傅里叶变换，信号可以时间函数或频率函数两种形式描述，特别是周期信号和准周期信号（前者由一个基频成分和若干谐波成分，后者虽可分解为几个正弦分量，但它们的周期没有公倍数），从频率域可以很清楚地了解它们由哪些正弦分量组成。而对于非平稳信号，最典型的例子就是语音信号，它是非周期的，频谱随时间连续变化，因此由傅里叶变换得到的频谱无法获知其在各个时刻的频谱特性。最直观的想法就是用中心在某一时刻的时间窗截取一段信号，对其做傅里叶变换，得到这一时刻的频谱；然后将窗在时间轴上移动，从而得到不同时刻的频谱，这就是短时频谱的原理。最简单的窗就是矩形窗，即直接从原信号中截取一段。

三 实验内容

本实验利用 matlab 提供的工具来采集和分析声音信号的频谱，由以下几个部分组成：

(1). 声音的采集

Matlab 提供了读入、录制和播放声音以及快速傅里叶变换的函数，分别是 wavread、wavrecord、wavplay 和 fft。阅读这几个函数的帮助文档，熟练使用。

(2). 持续音的频谱分析

将 Windows 的系统目录下的 ding.wav 文件读入，这是一个双声道的声音，选择任一声道的信号，使用 fft 求取其频谱，并用 plot 显示它的幅度谱，观察主要的正弦分量；分别求取 2048、1024 点 FFT，观察产生的不同频谱；用 ifft 函数求取频谱的反傅里叶反变换，比较反变换后的信号波形与原始信号的波形；从频谱中找到幅度最大的正弦分量，构造一个同样幅度的正弦信号，将其波形与原始信号比较，并且试听一下。

(3). 时变音的短时频谱分析

使用”load chirp”载入 matlab 自带的一个时变音；从信号中依次截取 1024 个点，利用上述方法求取其幅度谱，并显示出来，观察幅度谱随时间的变化情况。

四 实验报告要求

简述实验目的及原理，列出第 2、3 个内容的详细步骤和相应的代码，并附上各步的结果图。

实验六 电话拨号音的合成与识别

1. 实验目的

本实验基于对电话通信系统中拨号音合成与识别的仿真实现，主要涉及到电话拨号音合成的基本原理及识别的主要方法，利用 MATLAB 软件以及 FFT 算法实现对电话通信系统中拨号音的合成与识别。并进一步利用 MATLAB 中的图形用户界面 GUI 制作简单直观的模拟界面。使其对电话通信系统拨号音的合成与识别有个基本的了解。

能够利用矩阵不同的基频合成 0 — 9 不同按键的拨号音，并能够对不同的拨号音加以正确的识别，实现由拨号音解析出电话号码的过程。进一步利用 GUI 做出简单的图形操作界面。要求界面清楚，画面简洁，易于理解，操作简单。从而实现对电话拨号音系统的简单的实验仿真。

2. 实验原理


双音多频 DTMF (Dual Tone Multi-Frequency) 信号，是用两个特定的单音频率信号的组合来代表数字或功能。在 DTMF 电话机中有 16 个按键，其中 10 个数字键 0 — 9 ， 6 个功能键 * 、 # 、 A 、 B 、 C 、 D 。其中 12 个按键是我们比较熟悉的按键，另外由第 4 列确定的按键作为保留，作为功能键留为今后他用。根据 CCITT 建议，国际上采用 697Hz 、 770Hz 、 852Hz 、 941Hz 低频群及 1209Hz 、 1336Hz 、 1477Hz 、 1633Hz 高频群。从低频群和高频群任意各抽一种频率进行组合，共有 16 种组合，代表 16 种不同的数字键或功能，每个按键唯一地由一组行频和列频组成，如表 1 所示。

$f_H(\text{Hz})$ $f_L(\text{Hz})$	1209	1336	1477	1633
697	1	2	3	A
770	4	5	6	B
852	7	8	9	C
941	*	0	#	D


表 1 : DTMF 的组合功能



3. 实验内容

(1). 图形电话拨号面板的制作

利用 GUI 图形用户界面设计工具制作电话拨号面板，把 DTMF 信号和电话机的键盘矩阵对应起来。其中选用我们熟悉的 10 个数字键 0 — 9 ， 2 个功能键 “ * ” 、 “ # ” ，另四个键省略。按照图 1 电话机键盘矩阵的排列方式制作四行三列的按键控件。每个按键可用  (Push Button) 添加。

然后，为了更直观的反映对应的按键号码，可以设置一个编辑框，用于动态的

显示拨号号码，模拟实际电话的拨号显示窗口。编辑框可用  (Edit Text) 添加。

另外，为了图形电话拨号面板的简洁美观，可以添加空白区域作为背景，并用静态文本框制作文字信息。背景可用  (Frame) 添加，静态文本框可用  (Static Text) 添加。

最终利用 GUI 图形用户界面设计工具生成的图形电话拨号面板用于拨号音的合成产生部分，如下图所示。这里将其保存为 `tu1.fig` 文件。



(2). DTMF 信号的产生合成

现在将对上节制作的图形电话拨号面板上的各控件单位的动作和变化进行设置，即对 `tu1.m` 文件进行编辑。其主要的功能是使对应的按键，按照表 1 的对应关系产生相应的拨号音，完成对应行频及列频的叠加输出。此外，对于图形界面的需要，还要使按键的号码数字显示在拨号显示窗口中。

鉴于 CCITT 对 DTMF 信号规定的指标，这里每个数字信号取 1000 个采样点模拟按键信号，并且每两个数字之间用 100 个 0 来表示间隔来模拟静音。以便区别连续的两个按键信号。间隔的静音信号也是在按键时产生的。

以按键 0 为例，简单介绍拨号音产生的过程：

% 按键 0 的响应函数

```
function varargout = pushbutton0_Callback(h, eventdata, handles, varargin)
```

```
n=[1:1000]; % 每个数字 1000 个采样点表示
```

```
d0=sin(0.7217*n)+sin(1.0247*n); % 对应行频列频叠加
```

```
n0=strcat(get(handles.edit1,'string'),'0'); % 获取数字号码
```

```
set(handles.edit1,'string',n0); % 显示号码
```

```
space=zeros(1,100); %100 个 0 模拟静音信号
```

```
global NUM
```

```

phone=[NUM, d0];

NUM=[phone, space]; % 存储连续的拨号音信号

wavplay(d0, 8192); % 产生拨号音

```

程序解释：

NUM 为定义的全局变量，用于存储连续的拨号音（DTMF）信号，包括数字信号音以及静音信号。

$d0 = \sin(0.7217 * n) + \sin(1.0247 * n)$ 中的行频与列频是由表 1 中 0 键对应的 $f_L = 941\text{Hz}$ ， $f_H = 1336\text{Hz}$ 计算得出，已知声音取样频率 $f_s = 8192\text{Hz}$ ，则取样后 $\omega_L = 2\pi f_L / f_s = 0.7217$ ， $\omega_H = 2\pi f_H / f_s = 1.0247$ 。

对于保留的两个功能键“*”“#”，按照现行键盘式拨号电话的习惯，将“*”作为删除键，“#”作为确认键。“*”删除键的作用是将前面拨错的号码删除退回，表现为将显示窗口已经显示的错误号码退回一位数字，并且将连续拨号音信号的存储单元 NUM 中退回一位拨号音信号和静音信号。删除可以进行连续的操作。“#”确认键的作用是将前面拨过的号码进行确认保留，意味着此时连续拨号音信号的存储单元 NUM 中的信号即为最后用于识别的连续拨号音 DTMF 信号，并在显示窗口中显示“#”号作为标记。

% 删除键的响应函数

```

function varargout = pushbuttonback_Callback(h, eventdata, handles,
varargin)

n=[1:1000];

num=get(handles.edit1,'string');

l=length(num);

n11=strrep(num, num, num(1:l-1)); %去掉末尾号码在面板上的显示

d11=sin(0.7217*n)+sin(0.9273*n);

set(handles.edit1,'string',n11);

global NUM

L=length(NUM);

NUM=NUM(1:L-1100); %删除末尾号码在拨号音信号中的存储

```

```
wavplay(d11,8192);
```

(3). DTMF 信号的检测识别

要实现电话拨号音（DTMF）信号的检测识别，可以通过直接计算付里叶变换得到输入信号的组成频率。这里采用 FFT 算法对信号进行解码分析。首先对接收到的数字信号作 FFT 分析，计算出其幅频谱，进而得到功率谱，组成输入信号的频率必定对应功率谱的峰值。对于连续的双音多频（DTMF）信号，需要把有效的数字拨号信号从静音间隔信号中分割提取出来，然后再用 FFT 算法对信号进行解码分析。

MATLAB 实现信号音的识别如下：

% 信号音识别

```
function varargout = pushbuttonNUM_Callback(h, eventdata, handles,  
varargin)
```

```
global NUM
```

```
wavplay(NUM,8192);
```

```
L=length(NUM);
```

```
n=L/1100;
```

```
number='';
```

```
for i=1:n
```

```
    j=(i-1)*1100+1;
```

```
    d=NUM(j:j+999); % 截取出每个数字
```

```
    f=fft(d,2048); % 以 N=2048 作 FFT 变换
```

```
    a=abs(f);
```

```
    p=a.*a/10000; % 计算功率谱
```

```
    num(1)=find(p(1:250)==max(p(1:250))); % 找行频
```

```
    num(2)=300+find(p(300:380)==max(p(300:380))); % 找列频
```

```
    if (num(1) < 180) row=1; % 确定行数
```

```
    elseif (num(1) < 200) row=2;
```

```

elseif (num(1) < 220) row=3;

else row=4;

end

if (num(2) < 320) column=1; % 确定列数

elseif (num(2) < 340) column=2;

else column=3;

end

z=[row,column]; % 确定数字

if z==[4,2] tel=0;

elseif z==[1,1] tel=1;

elseif z==[1,2] tel=2;

elseif z==[1,3] tel=3;

elseif z==[2,1] tel=4;

elseif z==[2,2] tel=5;

elseif z==[2,3] tel=6;

elseif z==[3,1] tel=7;

elseif z==[3,2] tel=8;

elseif z==[3,3] tel=9;

end

t(i)=tel;

c=strcat(number,int2str(tel));

number=c;

i=i+1;

end

```



```
set(handles.edit3,'string',number);
```

程序解释:

确定行频和列频的数值范围是通过计算得出的: 已知输入信号的取样频率

$f_s = 8192\text{Hz}$, 而做 FFT 的 $N=2048$, 则频谱分辨率为 $F = f_s / N = 8192 / 2048 = 4\text{Hz}$,

由此可算出频谱图上任意点对应的频率 $K = f / F$ 。例如, 数字 8 的高、低端频率

为 $f_L = 852\text{Hz}, f_H = 1336\text{Hz}$, 则在谱图上对应的点

$K_L = f_L / F = 213, K_H = f_H / F = 334$ 。

4. 实验内容实验报告要求:

- (1) 简述实验目的及原理。
- (2) 打印出一个数字拨号音的频谱图, 加以分析说明, 并解释 DTMF 信号的检测识别的原理。
- (3) 总结实验得出主要结论。

附录 1 MATLAB 基础

§1 MATLAB 简介

1.1 什么是 MATLAB?

MATLAB 是由美国 MathWorks 公司推出的用于数值计算和图形处理计算系统环境,除了具备卓越的数值计算能力外,它还提供了专业水平的符号计算,文字处理,可视化建模仿真和实时控制等功能。MATLAB 的基本数据单位是矩阵,它的指令表达式与数学,工程中常用的形式十分相似,故用 MATLAB 来解算问题要比用 C, FORTRAN 等语言简捷得多。MATLAB 是国际公认的优秀数学应用软件之一。

MATLAB 是英文 MATrix LABoratory(矩阵实验室)的缩写。20 世纪 80 年代初期, Cleve Moler 与 John Little 等利用 C 语言开发了新一代的 MATLAB 语言,此时的 MATLAB 语言已同时具备了数值计算功能和简单的图形处理功能。1984 年, Cleve Moler 与 John Little 等正式成立了 Mathworks 公司,把 MATLAB 语言推向市场,并开始了对 MATLAB 工具箱等的开发设计。1993 年, Mathworks 公司推出了基于个人计算机的 MATLAB 4.0 版本,到了 1997 年又推出了 MATLAB 5.X 版本 (Release 11),并在 2000 年推出 MATLAB 6 版本 (Release 12),2004 年 6 月份正式推出 MATLAB 7.0 版本 (Release 14)。

概括地讲,整个 MATLAB 系统由两部分组成,即 MATLAB 内核及辅助工具箱,两者的调用构成了 MATLAB 的强大功能。MATLAB 语言以数组为基本数据单位,包括控制流语句、函数、数据结构、输入输出及面向对象等特点的高级语言,它具有以下主要特点:

1) 运算符和库函数极其丰富,语言简洁,编程效率高, MATLAB 除了提供和 C 语言一样的运算符外,还提供广泛的矩阵和向量运算符。利用其运算符和库函数可使其程序相当简短,两三行语句就可实现几十行甚至几百行 C 或 FORTRAN 的程序功能。

2) 既具有结构化的控制语句(如 for 循环、while 循环、break 语句、if 语句和 switch 语句),又有面向对象的编程特性。

3) 图形功能强大。它既包括对二维和三维数据可视化、图像处理、动画制作等高层次的绘图命令,也包括可以修改图形及编制完整图形界面的、低层次的绘图命令。

4) 功能强大的工具箱。工具箱可分为两类:功能性工具箱和学科性工具箱。功能性工具箱主要用来扩充其符号计算功能、图示建模仿真功能、文字处理功能以及与硬件实时交互的功能。而学科性工具箱是专业性比较强的,如优化工具箱、统计工具箱、控制工具箱、小波工具箱、图象处理工具箱、通信工具箱等。

5) 易于扩充。除内部函数外,所有 MATLAB 的核心文件和工具箱文件都是可读可改的源文件,用户可修改源文件和加入自己的文件,它们可以与库函数一样被调用。

§2 MATLAB 的开发环境

桌面平台是各桌面组件的展示平台,其中重要的窗口具体如下:

2.2.1 MATLAB 主窗口

MATLAB6 比早期版本增加了一个主窗口。该窗口不能进行任何计算任务的操作,只用来进行一些整体的环境参数的设置。

2.2.2 命令窗口 (Command Window)

命令窗口是对 MATLAB 进行操作的主要载体,默认的情况下,启动 MATLAB 时就会打开命令窗口,显示形式如图 1-1 所示。一般来说, MATLAB 的所有函数和命令都可以在命令窗口

中执行。在 MATLAB 命令窗口中，命令的实现不仅可以由菜单操作来实现，也可以由命令行操作来执行，下面就详细介绍 MATLAB 命令行操作。

实际上，掌握 MATLAB 命令行操作是走入 MATLAB 世界的第一步，命令行操作实现了对程序设计而言简单而又重要的人机交互，通过对命令行操作，避免了编程序的麻烦，体现了 MATLAB 所特有的灵活性。

例如：

```
% 在命令窗口中输入 sin(pi/5)，然后单击回车键，则会得到该表达式的值
sin (pi/5)
ans=
    0.5878
```

由例可以看出，为求得表达式的值，只需按照 MATLAB 语言规则将表达式输入即可，结果会自动返回，而不必像其他的程序设计语言那样，编制冗长的程序来执行。当需要处理相当繁琐的计算时，可能在一行之内无法写完表达式，可以换行表示，此时需要使用续行符“.....”否则 MATLAB 将只计算一行的值，而不理会该行是否已输入完毕。

例如：

```
sin(1/9*pi)+sin(2/9*pi)+sin(3/9*pi)+.....
sin(4/9*pi)+sin(5/9*pi)+sin(6/9*pi)+.....
sin(7/9*pi)+sin(8/9*pi)+sin(9/9*pi)+.....
ans=
    5.6713
```

使用续行符之后 MATLAB 会自动将前一行保留而不加以计算，并与下一行衔接，等待完整输入后再计算整个输入的结果。

在 MATLAB 命令行操作中，有一些键盘按键可以提供特殊而方便的编辑操作。比如：“↑”可用于调出前一个命令行，“↓”可调出后一个命令行，避免了重新输入的麻烦。当然下面即将讲到的历史窗口也具有此功能。

2.2.3 历史窗口 (Command History)

历史命令窗口是 MATLAB6 新增添的一个用户界面窗口，默认设置下历史命令窗口会保留自安装时起所有命令的历史记录，并标明使用时间，以方便使用者的查询。而且双击某一行命令，即在命令窗口中执行该命令。

2.2.4 当前目录窗口 (Current Directory)

在当前目录窗口中可显示或改变当前目录，还可以显示当前目录下的文件，包括文件名、文件类型、最后修改时间以及该文件的说明信息等并提供搜索功能。

2.2.5 工作空间管理窗口 (Workspace)

工作空间管理窗口是 MATLAB 的重要组成部分。在工作空间管理窗口中将显示所有目前保存在内存中的 MATLAB 变量的变量名、数据结构、字节数以及类型，而不同的变量类型分别对应不同的变量名图标。

§3 MATLAB 帮助系统

完善的帮助系统是为任何应用软件必要的组成部分。MATLAB 提供了相当丰富的帮助信息，同时也提供了获得帮助的方法。首先，可以通过桌面平台的【Help】菜单来获得帮助，也可以通过工具栏的帮助选项获得帮助。此外，MATLAB 也提供了在命令窗口中的获得帮助的多种方法，在命令窗口中获得 MATLAB 帮助的命令及说明列于表 3-1 中。其调用格式为：

命令+指定参数

表 3-1

命 令	说 明
doc	在帮助浏览器中显示指定函数的参考信息
help	在命令窗口中显示 M 文件帮助
helpbrowser	打开帮助浏览器，无参数
helpwin	打开帮助浏览器，并且见初始界面置于 MATLAB 函数的 M 文件帮助信息
lookfor	在命令窗口中显示具有指定参数特征函数的 M 文件帮助
web	显示指定的网络页面，默认为 MATLAB 帮助浏览器

例如：

```
>>help sin
SIN    Sine
      SIN(X) is the sine of the elements of  X
Overloaded methods
      Help sym/sin.m
```

另外也可以通过在组件平台中调用演示模型（demo）来获得特殊帮助。

§4 数据交换系统

MATLAB 提供了多种方法将数据从磁盘或剪贴板中读入 MATLAB 工作空间。具体的读写方法可依据用户的喜好以及数据的类型来选择。这里主要介绍文本数据的读入。

对于文本数据(ASCII)而言，最简单的读入方法就是通过 MATLAB 的数据输入向导(Import Wizard)，也可以通过 MATLAB 函数实现数据读入。

例如，对于文本文件 test.txt：

```
students' scores
      English      Chinese      Mathmatics
Wang      99        98         100
Li         98        89         70
Zhang      80        90         97
Zhao       77        65         87
```

下面通过上述两种方法将该文件数据读入 MATLAB 工作空间，先介绍 MATLAB 数据交换系统对文本数据的识别。此时文件的前几行（此处为“students' scores”）将被识别为文件头，文件头可以为一行或几行，也可以识别出数据的列头（此处为：“English”、“Chinese”、和“Mathmatics”）和行头（此处为“wang”、“li”、“zhang”和“zhao”），其余的为可分断数据（此处为“99”、“98”、和“100”等）。

首先是通过数据输入向导编辑器读入数据，通过桌面平台上的【File】菜单中的【Import Data】选项打开输入向导编辑器，按向导提示进行操作完成整个文本数据的输入，则用户可以在

MATLAB 开发环境中使用该文本数据。

例如：

```
>> whos

      Name      Size      Bytes      Class
      Data      4×3        96      double array
```

Grand total is 12 elements using 96 bytes

```
>> Data
Data =
    99    98   100
    98    89    70
    80    90    97
    77    65    87
```

“whos”用于显示当前 MATLAB 工作空间的变量，而在命令窗口中输入 data 后，将显示该数据。在命令窗口或 M 文件中调用相应的函数也可以实现数据的读入。

例如：

```
>> [a,b,c,d]=textread('text.txt','%s %s %s %s','headlines',2)
```

§5 MATLAB 数值计算功能

MATLAB 强大的数值计算功能使其在诸多数学计算软件中傲视群雄，是 MATLAB 软件的基础。本节将简要介绍 MATLAB 的数据类型、矩阵的建立及运算。

5.1 MATLAB 数据类型

MATLAB 的数据类型主要包括：数字、字符串、矩阵、单元型数据及结构型数据等，限于篇幅我们将重点介绍其中几个常用类型。

5.1.1 变量与常量

变量是任何程序设计语言的基本要素之一，MATLAB 语言当然也不例外。与常规的程序设计语言不同的 MATLAB 并不要求事先对所使用的变量进行声明，也不需要指定变量类型，MATLAB 语言会自动依据所赋予变量的值或对变量所进行的操作来识别变量的类型。在赋值过程中如果赋值变量已存在时，MATLAB 语言将使用新值代替旧值，并以新值类型代替旧值类型。在 MATLAB 语言中变量的命名应遵循如下规则：

- (1) 变量名区分大小写。
- (2) 变量名长度不超 31 位，第 31 个字符之后的字符将被 MATLAB 语言所忽略。
- (3) 变量名以字母开头，可以是字母、数字、下划线组成，但不能使用标点。

与其他的程序设计语言相同，在 MATLAB 语言中也存在变量作用域的问题。在未加特殊说明的情况下，MATLAB 语言将所识别的一切变量视为局部变量，即仅在其使用的 M 文件内有效。若要将变量定义为全局变量，则应当对变量进行说明，即在该变量前加关键字 global。一般来说全局变量均用大写的英文字符表示。

MATLAB 语言本身也具有一些预定义的变量，这些特殊的变量称为常量。表 4-1 给出了 MATLAB 语言中经常使用的一些常量值。

表 5-1

常 量	表 示 数 值
pi	圆周率
eps	浮点运算的相对精度
inf	正无穷大
NaN	表示不定值
realmax	最大的浮点数
i, j	虚数单位

在 MATLAB 语言中，定义变量时应避免与常量名重复，以防改变这些常量的值，如果已改变了某外常量的值，可以通过“clear+常量名”命令恢复该常量的初始设定值（当然，也可通过重新启动 MATLAB 系统来恢复这些常量值）。

5.1.2 数字变量的运算及显示格式

MALAB 是以矩阵为基本运算单元的，而构成数值矩阵的基本单元是数字。为了更好地学习和掌握矩阵的运算，首先对数字的基本知识作简单的介绍。

对于简单的数字运算，可以直接在命令窗口中以平常惯用的形式输入，如计算 2 和 3 的乘积再加 1 时，可以直接输入：

```
>> 1+2*3
ans=
    7
```

这里“ans”是指当前的计算结果，若计算时用户没有对表达式设定变量，系统就自动赋当前结果给“ans”变量。用户也可以输入：

```
>> a=1+2*3
a=
    7
```

此时系统就把计算结果赋给指定的变量 a 了。

MATLAB 语言中数值有多种显示形式，在缺省情况下，若数据为整数，则就以整数表示；若数据为实数，则以保留小数点后 4 位的精度近似表示。MATLAB 语言提供了 10 种数据显示格式，常用的有下述几种格式：

short	小数点后 4 位(系统默认值)
long	小数点后 14 位
short e	5 位指数形式
long e	15 位指数形式

MATLAB 语言还提供了复数的表达和运算功能。在 MATLAB 语言中，复数的基本单位表示为 i 或 j。在表达简单数数值时虚部的数值与 i、j 之间可以不使用乘号，但是如果是表达式，则必须使用乘号以识别虚部符号。

5.1.3 字符串

字符和字符串运算是各种高级语言必不可少的部分，MATLAB 中的字符串是其进行符号运算表达式的基本构成单元。

在 MATLAB 中，字符串和字符数组基本上是等价的；所有的字符串都用单引号进行输入或赋值（当然也可以用函数 char 来生成）。字符串的每个字符（包括空格）都是字符数组的一个元素。例如：

```
>>s='matrix laboratory';
s=
```

```

matrix    laboratory
>> size(s)                                % size 查看数组的维数
ans=
     1     17

```

另外，由于 MATLAB 对字符串的操作与 C 语言几乎完全相同这里不在赘述。

5.2 矩阵及其运算

矩阵是 MATLAB 数据存储的基本单元，而矩阵的运算是 MATLAB 语言的核心，在 MATLAB 语言系统中几乎一切运算均是以对矩阵的操作为基础的。下面重点介绍矩阵的生成、矩阵的基本运算和矩阵的数组运算。

5.2.1 矩阵的生成

1. 直接输入法

从键盘上直接输入矩阵是最方便、最常用的创建数值矩阵的方法，尤其适合较小的简单矩阵。在用此方法创建矩阵时，应当注意以下几点：

- 输入矩阵时要以 “[]” 为其标识符号，矩阵的所有元素必须都在括号内。
- 矩阵同行元素之间由空格或逗号分隔，行与行之间用分号或回车键分隔。
- 矩阵大小不需要预先定义。
- 矩阵元素可以是运算表达式。
- 若 “[]” 中无元素表示空矩阵。

另外，在 MATLAB 语言中冒号的作用是最为丰富的。首先，可以用冒号来定义行向量。

例如：

```

>> a=1:0.5:4
a=
Columns 1 through 7
     1     1.5     2     2.5     3     3.5     4

```

其次，通过使用冒号，可以截取指定矩阵中的部分。

例如：

```

>> A=[1 2 3; 4 5 6; 7 8 9]
A=
     1     2     3
     4     5     6
     7     8     9

>> B=A(1:2,:)
B=
     1     2     3
     4     5     6

```

通过上例可以看到 B 是由矩阵 A 的 1 到 2 行和相应的所有列的元素构成的一个新的矩阵。在这里，冒号代替了矩阵 A 的所有列。

2. 外部文件读入法

MATLAB 语言也允许用户调用在 MATLAB 环境之外定义的矩阵。可以利用任意的文本编辑器编辑所要使用的矩阵，矩阵元素之间以特定分断符分开，并按行列布置。读入矩阵的一种方法可参考 3.3 节数据交换系统。另外也可以利用 load 函数，其调用方法为：Load+文件名[参数]

Load 函数将会从文件名所指定的文件中读取数据，并将输入的数据赋给以文件名命名的变量，如果不给定文件名，则将自动认为 matlab.mat 文件为操作对象，如果该文件在 MATLAB 搜索路

径中不存在时，系统将会报错。

例如： 事先在记事本中建立文件：

1	1	1
1	2	3
1	3	6

(并以 data1.txt 保存)

在 MATLAB 命令窗口中输入：

```
>> load data1.txt
>> data1
data1=
     1     1     1
     1     2     3
     1     3     6
```

3. 特殊矩阵的生成

对于一些比较特殊的矩阵(单位阵、矩阵中含 1 或 0 较多)，由于其具有特殊的结构，MATLAB 提供了一些函数用于生成这些矩阵。常用的有下面几个：

zeros(m)	生成 m 阶全 0 矩阵
eye(m)	生成 m 阶单位矩阵
ones(m)	生成 m 阶全 1 矩阵
rand(m)	生成 m 阶均匀分布的随机阵
randn(m)	生成 m 阶正态分布的随机矩阵

5.2.2 矩阵的基本数学运算

矩阵的基本数学运算包括矩阵的四则运算、与常数的运算、逆运算、行列式运算、秩运算、特征值运算等基本函数运算，这里进行简单介绍。

1. 四则运算

矩阵的加、减、乘运算符分别为“+，-，*”，用法与数字运算几乎相同，但计算时要满足其数学要求（如：同型矩阵才可以加、减）。

在 MATLAB 中矩阵的除法有两种形式：左除“\”和右除“/”。在传统的 MATLAB 算法中，右除是先计算矩阵的逆再相乘，而左除则不需要计算逆矩阵直接进行除运算。通常右除要快一点，但左除可避免被除矩阵的奇异性所带来的麻烦。在 MATLAB6 中两者的区别不太大。

2. 与常数的运算

常数与矩阵的运算即是同该矩阵的每一元素进行运算。但需注意进行数除时，常数通常只能做除数。

3. 基本函数运算

矩阵的函数运算是矩阵运算中最实用的部分，常用的主要有以下几个：

det(a)	求矩阵 a 的行列式
eig(a)	求矩阵 a 的特征值
inv(a)或 a ^ (-1)	求矩阵 a 的逆矩阵
rank(a)	求矩阵 a 的秩
trace(a)	求矩阵 a 的迹（对角线元素之和）

例如：

```
>> a=[2 1 -3 -1; 3 1 0 7; -1 2 4 -2; 1 0 -1 5];
>> a1=det(a);
>> a2=det(inv(a));
>> a1*a2
ans=
     1
```


注意：命令行后加“;”表示该命令执行但不显示执行结果。

5.2.2 矩阵的数组运算

我们在进行工程计算时常常遇到矩阵对应元素之间的运算。这种运算不同于前面讲的数学运算，为有所区别，我们称之为数组运算。

1. 基本数学运算

数组的加、减与矩阵的加、减运算完全相同。而乘除法运算有相当大的区别，数组的乘除法是指两同维数组对应元素之间的乘除法，它们的运算符为“.*”和“./”或“.\”。前面讲过常数与矩阵的除法运算中常数只能做除数。在数组运算中有了“对应关系”的规定，数组与常数之间的除法运算没有任何限制。

另外，矩阵的数组运算中还有幂运算（运算符为.^）、指数运算（exp）、对数运算（log）、和开方运算（sqrt）等。有了“对应元素”的规定，数组的运算实质上就是针对数组内部的每个元素进行的。

例如：

```
>> a=[2 1 -3 -1; 3 1 0 7; -1 2 4 -2; 1 0 -1 5];
>> a^3
ans=
    32   -28  -101    34
    99   -12  -151   239
    -1    49    93     8
    51   -17   -98   139

>> a.^3
ans=
     8     1   -27    -1
    27     1     0   343
    -1     8    64    -8
     1     0    -1   125
```

由上例可见矩阵的幂运算与数组的幂运算有很大的区别。

2. 逻辑关系运算

逻辑运算是 MATLAB 中数组运算所特有的一种运算形式，也是几乎所有的高级语言普遍适用的一种运算。它们的具体符号、功能及用法见表 4-2。

表 4-2

符号运算符	功 能	函 数 名
==	等于	eq
~=	不等于	ne
<	小于	lt
>	大于	gt
<=	小于等于	le
>=	大于等于	ge
&	逻辑与	and
	逻辑或	or
~	逻辑非	not

说明：

- 在关系比较中，若比较的双方为同维数组，则比较的结果也是同维数组。它的元素值由 0 和 1 组成。当比较双方对应位置上的元素值满足比较关系时，它的对

应值为 1，否则为 0。

- 当比较的双方中一方为常数，另一方为一数组，则比较的结果与数组同维。
- 在算术运算、比较运算和逻辑与、或、非运算中，它们的优先级关系先后为：
比较运算、算术运算、逻辑与或非运算。

例如：

```
>>a=[1 2 3;4 5 6;7 8 9];
>> x=5;
>> y= ones(3)*5;
>> xa= x<=a
xa=
    0    0    0
    0    1    1
    1    1    1
>> b=[0 1 0;1 0 1;0 0 1];
>> ab=a&b
ab=
    0    1    0
    1    0    1
    0    0    1
```

§6 MATLAB 图形功能

MATLAB 有很强的图形功能，可以方便地实现数据的视觉化。强大的计算功能与图形功能相结合为 MATLAB 在科学技术和教学方面的应用提供了更加广阔的天地。下面着重介绍二维图形的画法，对三维图形只作简单叙述。

6.1 二维图形的绘制

6.1.1 基本形式

二维图形的绘制是 MATLAB 语言图形处理的基础，MATLAB 最常用的画二维图形的命令是 plot，看两个简单的例子：

```
>> y=[0 0.58 0.70 0.95 0.83 0.25];
>> plot(y)
```

生成的图形见图 6-1，是以序号 1,2,⋯,6 为横坐标、数组 y 的数值为纵坐标画出的折线。

```
>> x=linspace(0,2*pi,30);    % 生成一组线性等距的数值
>> y=sin(x);
>> plot(x,y)
```

生成的图形见图 6-2，是 $[0, 2\pi]$ 上 30 个点连成的光滑的正弦曲线。

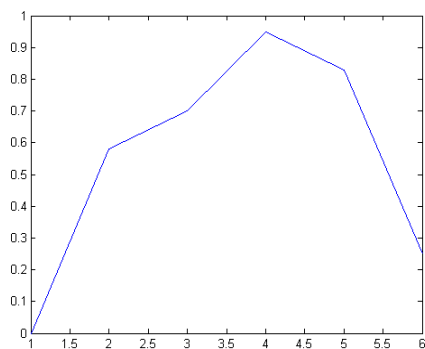


图 6-1

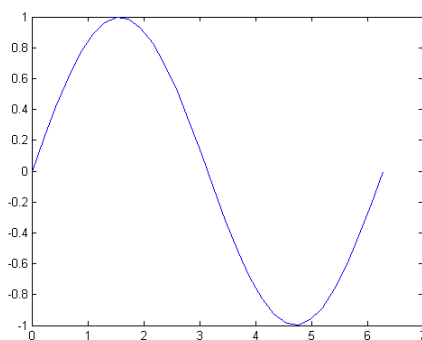


图 6-2

6.1.2 多重线

在同一个画面上可以画许多条曲线，只需多给出几个数组，例如

```
>> x=0:pi/15:2*pi;
>> y1=sin(x);
>> y2=cos(x);
>> plot(x,y1,x,y2)
```

则可以画出图 6-3。多重线的另一种画法是利用 hold 命令。在已经画好的图形上，若设置 hold on，MATLAB 将把新的 plot 命令产生的图形画在原来的图形上。而命令 hold off 将结束这个过程。例如：

```
>> x=linspace(0,2*pi,30); y=sin(x); plot(x,y)
```

先画好图 6-2，然后用下述命令增加 cos(x)的图形，也可得到图 6-3。

```
>> hold on
>> z=cos(x); plot(x,z)
>> hold off
```

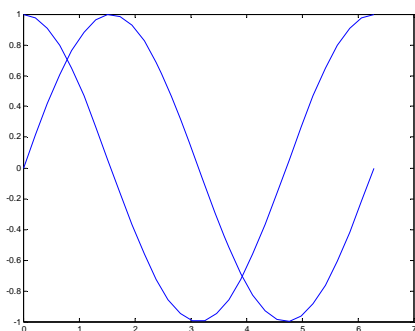


图 6-3

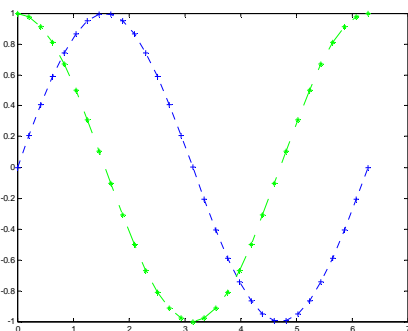


图 6-4

6.1.3 线型和颜色

MATLAB 对曲线的线型和颜色有许多选择，标注的方法是在每一对数组后加一个字符串参数，说明如下：

线型 线方式： - 实线 : 点线 -. 虚点线 -- 波折线。

线型 点方式： . 圆点 + 加号 * 星号 x x 形 o 小圆

颜色： y 黄； r 红； g 绿； b 蓝； w 白； k 黑； m 紫； c 青。

以下面的例子说明用法：

```
>> x=0:pi/15:2*pi;
```

```
>> y1=sin(x); y2=cos(x);
>> plot(x,y1,'b:+',x,y2,'g-.*')
```

可得图形 6-4。

6.1.4 网格和标记

在一个图形上可以加网格、标题、x 轴标记、y 轴标记，用下列命令完成这些工作。

```
>> x=linspace(0,2*pi,30); y=sin(x); z=cos(x);
>> plot(x,y,x,z)
>> grid
>> xlabel('Independent Variable X')
>> ylabel('Dependent Variables Y and Z')
>> title('Sine and Cosine Curves')
```

它们产生图 6-5：

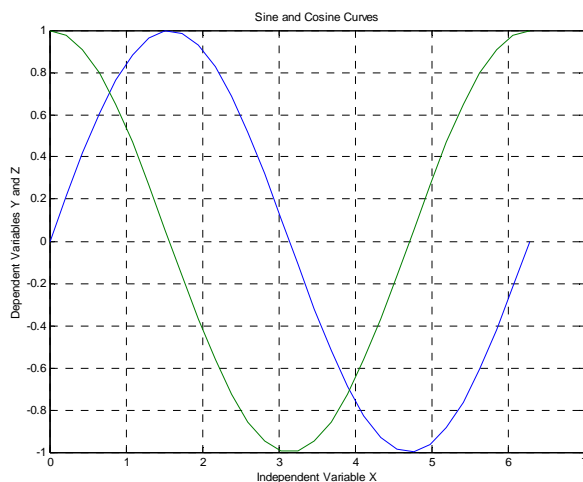


图 6-5

也可以在图形的任何位置加上一个字符串，如用：

```
>> text(2.5,0.7,'sinx')
```

表示在坐标 $x=2.5$, $y=0.7$ 处加上字符串 `sinx`。更方便的是用鼠标来确定字符串的位置，方法是输入命令：

```
>> gtext('sinx')
```

在图形窗口十字线的交点是字符串的位置，用鼠标点一下就可以将字符串放在那里。

6.1.5 坐标系的控制

在缺省情况下 MATLAB 自动选择图形的横、纵坐标的比例，如果你对这个比例不满意，可以用 `axis` 命令控制，常用的有：

```
axis([xmin xmax ymin ymax])    [ ]中分别给出 x 轴和 y 轴的最大值、最小值
axis equal 或 axis('equal')     x 轴和 y 轴的单位长度相同
axis square 或 axis('square')   图框呈方形
axis off 或 axis('off')         清除坐标刻度
```

还有 `axis auto` `axis image` `axis xy` `axis ij` `axis normal` `axis on` `axis(axis)`

用法可参考在线帮助系统。

6.1.6 多幅图形

可以在同一个画面上建立几个坐标系，用 `subplot(m,n,p)` 命令；把一个画面分成 $m \times n$ 个图形区域， p 代表当前的区域号，在每个区域中分别画一个图，如

```
>> x=linspace(0,2*pi,30); y=sin(x); z=cos(x);
>> u=2*sin(x).*cos(x); v=sin(x)./cos(x);
>> subplot(2,2,1),plot(x,y),axis([0 2*pi -1 1]),title('sin(x)')
>> subplot(2,2,2),plot(x,z),axis([0 2*pi -1 1]),title('cos(x)')
>> subplot(2,2,3),plot(x,u),axis([0 2*pi -1 1]),title('2sin(x)cos(x)')
>> subplot(2,2,4),plot(x,v),axis([0 2*pi -20 20]),title('sin(x)/cos(x)')
```

共得到 4 幅图形,见图 6-6.

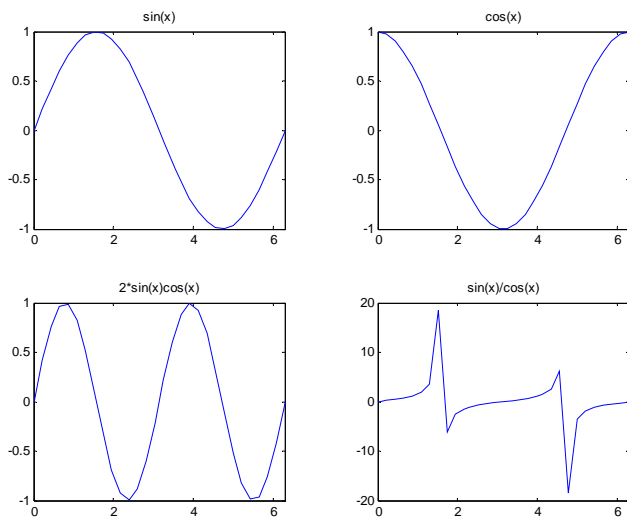


图 6-6

6.2 三维图形

限于篇幅这里只对几种常用的命令通过例子作简单介绍.

6.2.1 带网格的曲面

例 作曲面 $z=f(x,y)$ 的图形

$$z = \frac{\sin \sqrt{x^2 + y^2}}{\sqrt{x^2 + y^2}}, \quad -7.5 \leq x \leq 7.5, \quad -7.5 \leq y \leq 7.5$$

用以下程序实现:

```
>> x=-7.5:0.5:7.5;
>> y=x;
>> [X,Y]=meshgrid(x,y);      (3 维图形的 X,Y 数组)
>> R=sqrt(X.^2+Y.^2)+eps;    (加 eps 是防止出现 0/0)
>> Z=sin(R)./R;
>> mesh(X,Y,Z)                (3 维网格表面)
```

画出的图形如图 6-7. mesh 命令也可以改为 surf, 只是图形效果有所不同, 读者可以上机查看结果。

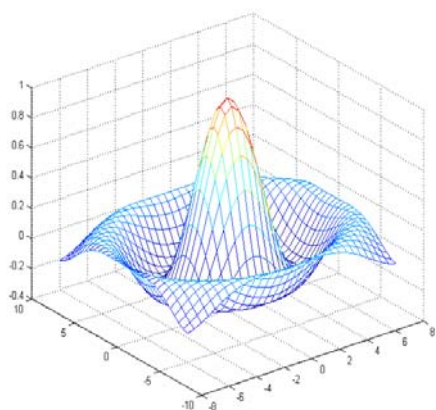


图 6-7

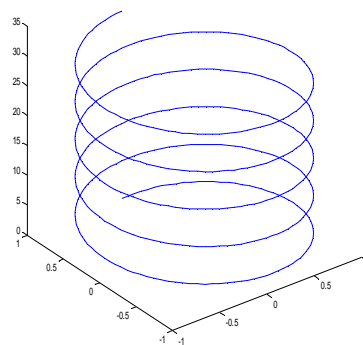


图 6-8

6.2.2 空间曲线

例 作螺旋线 $x=\sin t$, $y=\cos t$, $z=t$

用以下程序实现:

```
>> t=0:pi/50:10*pi;
```

```
>> plot3(sin(t),cos(t),t)    (空间曲线作图函数,用法类似于 plot)
```

画出的图形如图 5-8

6.2.3 等高线

用 `contour` 或 `contour3` 画曲面的等高线, 如对图 6-7 的曲面, 在上面的程序后接 `contour(X,Y,Z,10)` 即可得到 10 条等高线。

6.2.4 其它

较有用的是给三维图形指定观察点的命令 `view(azi,ele)`, `azi` 是方位角, `ele` 是仰角. 缺省时 `azi=-37.5°`, `ele=30°`.