

信号与系统实验教程

(2010 年修订版)

《信号与系统》课程组 编

武汉大学电子信息学院

2010 年 4 月

目录

实验一	信号的表示与实现	1
实验二	信号的时域基本运算	7
实验三	信号的卷积运算	15
实验四	周期信号的合成与分解	21
实验五	二阶状态轨迹的显示	28
实验六	信号的抽样与内插	33
实验七	滤波器的设计	38
实验八	Wav信号的波形分析与合成	49
实验九	电话拨号音的合成与识别	58
实验十	CDMA前向数据链路仿真	66
附录一	MATLAB基础	68
附录二	学生实验报告模板	87

实验一 信号的表示与实现

一、实验目的

1. 理解信号的定义和描述方法
2. 学习并掌握 MATLAB 环境下的信号表示方法

二、实验原理

1. 信号的定义

信号含有关于某种物理现象的变化过程和特征信息。在数学上，信号是单个或多个独立变量的函数。

2. 信号的分类

- (1) 确定性信号和随机信号；
- (2) 周期信号和非周期信号；
- (3) 连续时间信号和离散时间信号；
- (4) 能量受限信号和功率受限信号。

三、实验内容

1. 单位阶跃信号

(连续) 单位阶跃信号:
$$u(t) = \begin{cases} 0 & t < 0 \\ 1 & t > 0 \end{cases}$$

(离散) 单位阶跃信号:
$$u(n) = \begin{cases} 0 & n < 0 \\ 1 & n \geq 0 \end{cases}$$

程序 1-1:

```
% Generate continuous time unit step signal
```

```
t1=-1:0.1:0;
```

```
l1=length(t1);
```

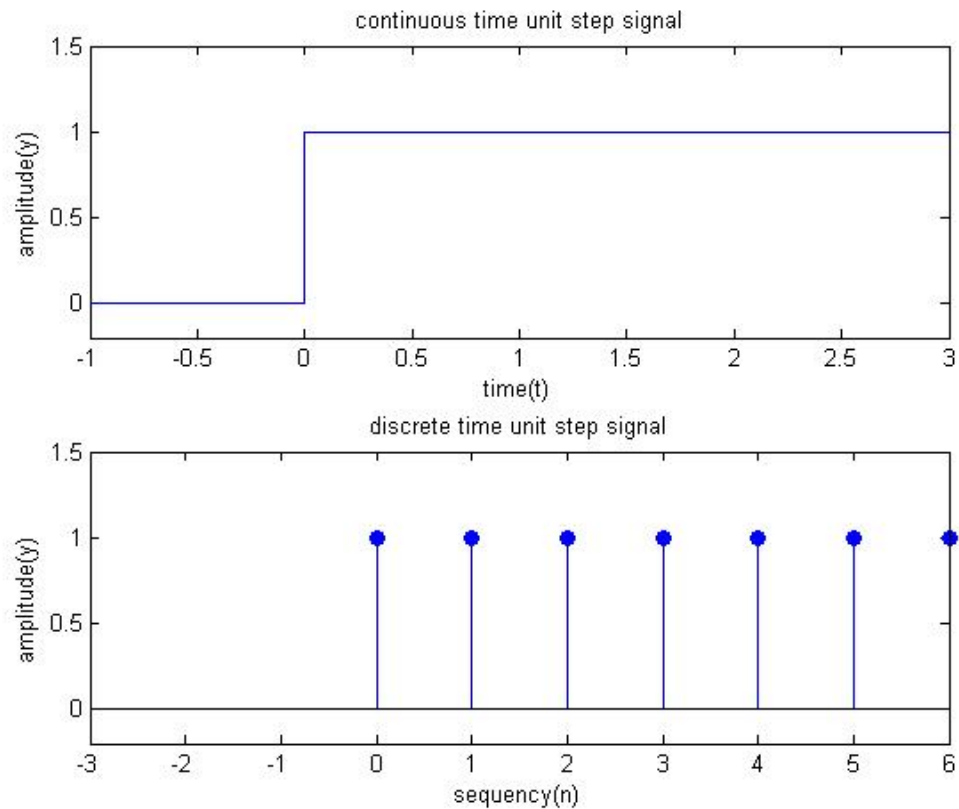
```
t2=0:0.1:3
```

```
l2=length(t2);
```

```

y1=zeros(1,11);
y2=ones(1,12);
subplot(2,1,1);
plot(t1,y1);
hold on;
plot(t2,y2);
plot([0,0],[0,1])
axis([-1,3,-0.2,1.5]);
xlabel('time(t)');
ylabel('amplitude(y)');
title('continuous time unit step signal')
% Generate discrete time unit step signal
n1=-3:-1;
l3=length(n1);
u1=zeros(1,l3);
n2=0:6
l4=length(n2);
u2=ones(1,l4);
subplot(2,1,2);
stem(n1,u1,'fill');
stem(n2,u2,'fill');
axis([-3,6,-0.2,1.5]);
xlabel('sequency(n)');
ylabel('amplitude(y)');
title('discrete time unit step signal')
hold off

```



2. 余弦信号

(连续) 余弦信号: $y(t) = A \cos(\Omega t + \vartheta) = 2 \cos\left(4\pi t + \frac{2\pi}{3}\right)$

(离散) 余弦信号: $y(n) = A \cos(\omega n + \phi) = 2 \cos\left(\frac{\pi}{6} n\right)$

程序1-2:

```
% The continuous cosine signal
```

```
t=0:0.01:1;
```

```
y=2.*cos(4*pi.*t-2*pi/3);
```

```
subplot(2, 1, 1)
```

```
plot(t, y)
```

```
hold on
```

```
plot([-0.2, 1.2], [0, 0], '--r')
```

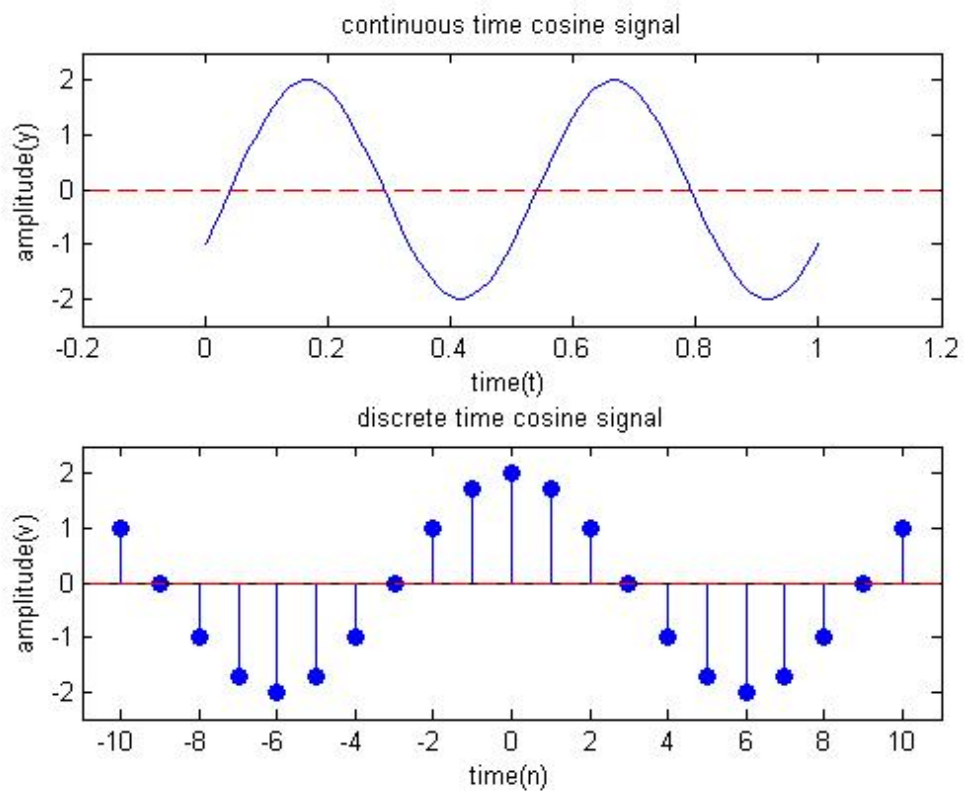
```
axis([-0.2, 1.2, -2.5, 2.5])
```

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

```

ylabel('amplitude(y)');
title('continuous time cosine signal')
% The discrete cosine signal
n=-10:10;
y2=2.*cos(pi/6.*n);
subplot(2,1,2)
stem(n,y2,'fill')
hold on
plot([-11,11],[0,0],'--r')
axis([-11,11,-2.5,2.5])
xlabel('time(n)');
ylabel('amplitude(v)');
title('discrete time cosine signal')

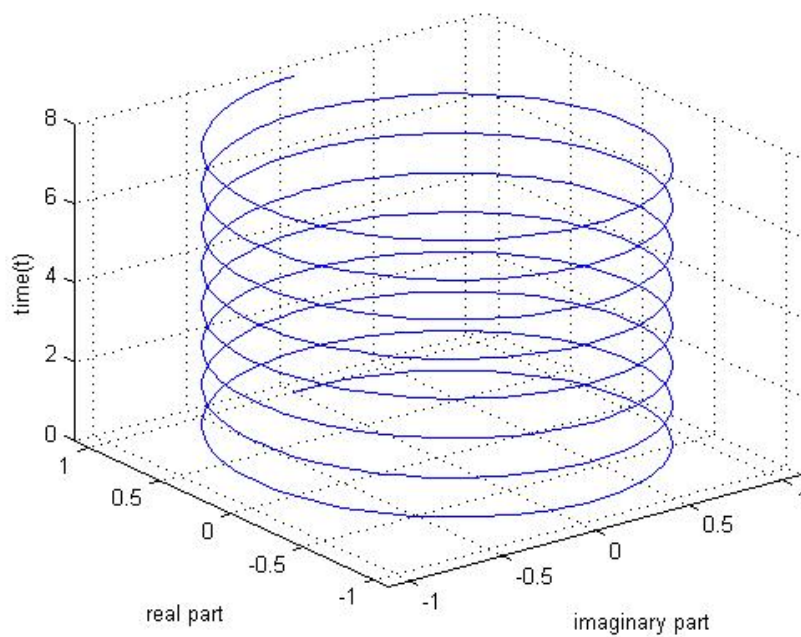
```



3. 复指数信号 $e^{j\omega_0 t}$, 或 $e^{j\omega_0 n}$

程序1-3: $e^{j2\pi t}$ 的三维立体图

```
% The 3-D line plot
t = 0:0.01:8;
plot3(sin(2*pi.*t), cos(2*pi.*t), t)
grid on
axis([-1.1 1.1 -1.1 1.1 0 8])
xlabel('imaginary part');
ylabel('real part');
zlabel('time(t)');
```



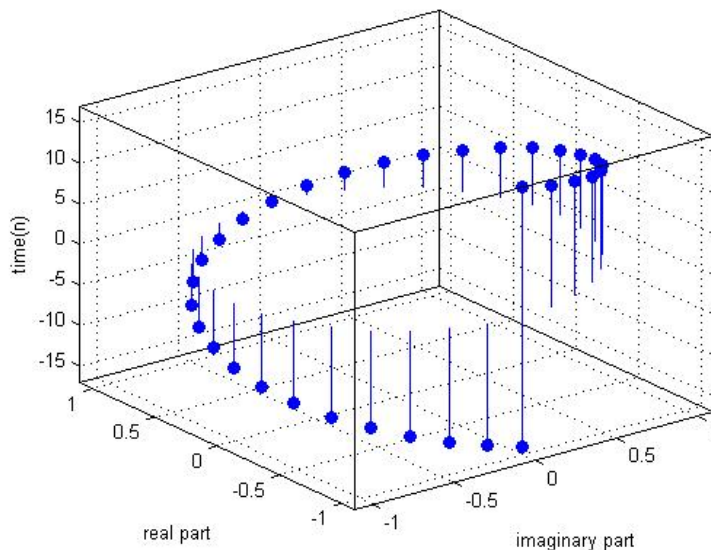
程序1-3: $e^{j\frac{\pi}{16}n}$ 的三维立体图

```
% The 3-D line plot for discrete time signal
% The 3-D line plot for discrete time signal
n = -16:16;
stem3(sin(2*pi./32.*n), cos(2*pi./32.*n), n, 'fill')
grid on
axis([-1.1 1.1 -1.1 1.1 -17 17])
```

```

xlabel('imaginary part');
ylabel('real part');
zlabel('time(n)');

```



4. 编制程序，实现如下时域信号的波形显示

(1) 连续时间信号

$y(t) = tu(t)$, $y(t) = \frac{\sin t}{t}$, $y(t) = e^{-t}u(t)$, $2e^{j6\pi t}$, 周期性方波信号, 周期性锯齿波信号。

(2) 离散时间信号

$y(n) = nu(n)$, $y(n) = a^n u(n)$, $0.5e^{j\frac{\pi}{32}n}$

三、实验要求

1. 输入本实验的示范程序，验证结果。
2. 按实验内容4的要求编制程序，完成时域信号的模拟显示。
3. 总结连续时间信号和离散时间信号的表示方法。

四、思考题

1. 在实现连续时间信号计算机表示时，时间间隔步长值的选取方法。
2. 任意函数信号的计算模拟表示方法。

实验二 信号的时域基本运算

一、实验目的

1. 掌握时域信号的基本运算。
2. 学习并实现时域信号基本运算的计算机仿真。

二、实验原理

1. 两信号相加、相乘

连续: $y(t) = y_1(t) + y_2(t)$, 或 $y(t) = y_1(t) \bullet y_2(t)$

离散: $y(n) = y_1(n) + y_2(n)$, 或 $y(n) = y_1(n) \bullet y_2(n)$

2. 延时

连续: $y(t) \rightarrow y(t - t_0)$

离散: $y(n) \rightarrow y(n - n_0)$

3. 反褶

连续: $y(t) \rightarrow y(-t)$

离散: $y(n) \rightarrow y(-n)$

4. 尺度变换

连续: $y(t) \rightarrow y(at)$, $a \neq 0$

离散: $y(n) \rightarrow y(an)$, $a \neq 0$

5. 微分/差分

连续: $y(t) \rightarrow \frac{dy(t)}{dt}$,

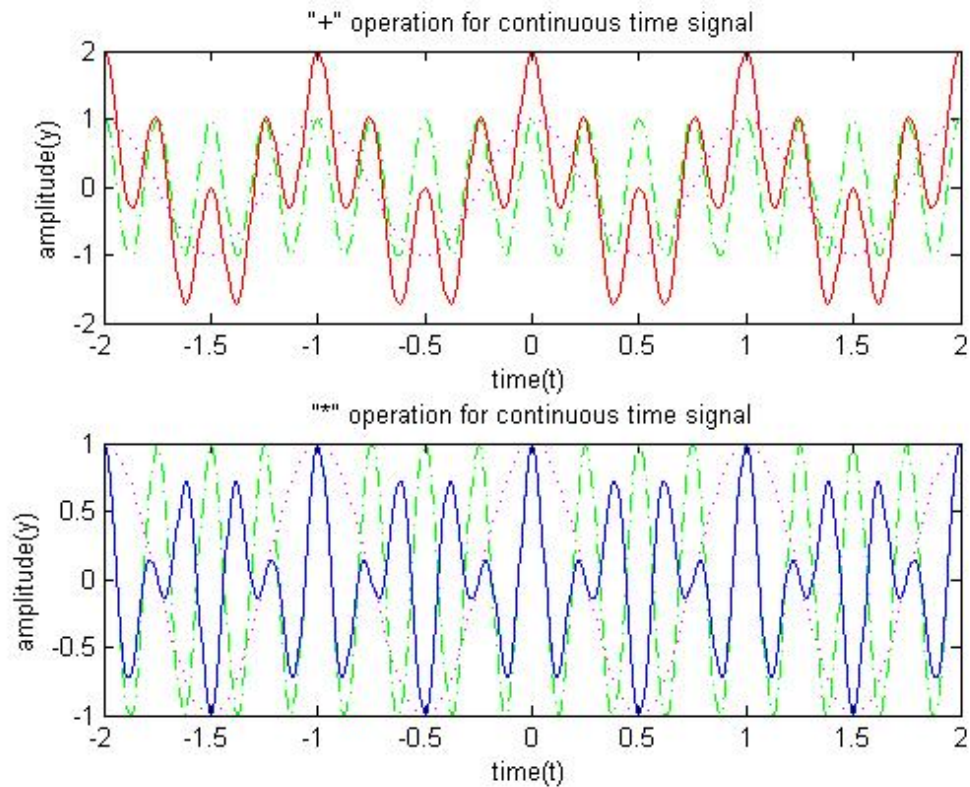
离散: $\nabla y(n) = y(n) - y(n-1)$, 或 $\Delta y(n) = y(n+1) - y(n)$

三、实验内容

1. 相加、相乘

程序2-1：连续时间信号的加、乘运算

```
% 'add' and 'multiply' operation for continuous time signal
t=-2:0.01:2;
y1=cos(2*pi.*t);
y2=cos(8*pi.*t);
y3=cos(2*pi.*t)+cos(8*pi.*t);
y4=cos(2*pi.*t).*cos(8*pi.*t);
subplot(2,1,1)
plot(t,y1,':m');
hold on;
plot(t,y2,'-.g');
plot(t,y3,'r');
xlabel('time(t)');
ylabel('amplitude(y)');
title('"+" operation for continuous time signal')
subplot(2,1,2)
plot(t,y1,':m');
hold on;
plot(t,y2,'-.g');
plot(t,y4,'b');
xlabel('time(t)');
ylabel('amplitude(y)');
title('"*" operation for continuous time signal')
```



程序2-2：离散时间信号的加、乘运算

% 'add' and 'multiply' operation for discrete time signal

n=-20:20;

y1=cos(1*pi/8.*n);

y2=cos(4*pi/8.*n);

y3=cos(1*pi/8.*n)+cos(4*pi/8.*n);

y4=cos(1*pi/8.*n).*cos(4*pi/8.*n);

subplot(2,1,1)

plot(n,y1,':m');

hold on;

plot(n,y2,'-.g');

plot(n,y3,'r');

xlabel('time(t)');

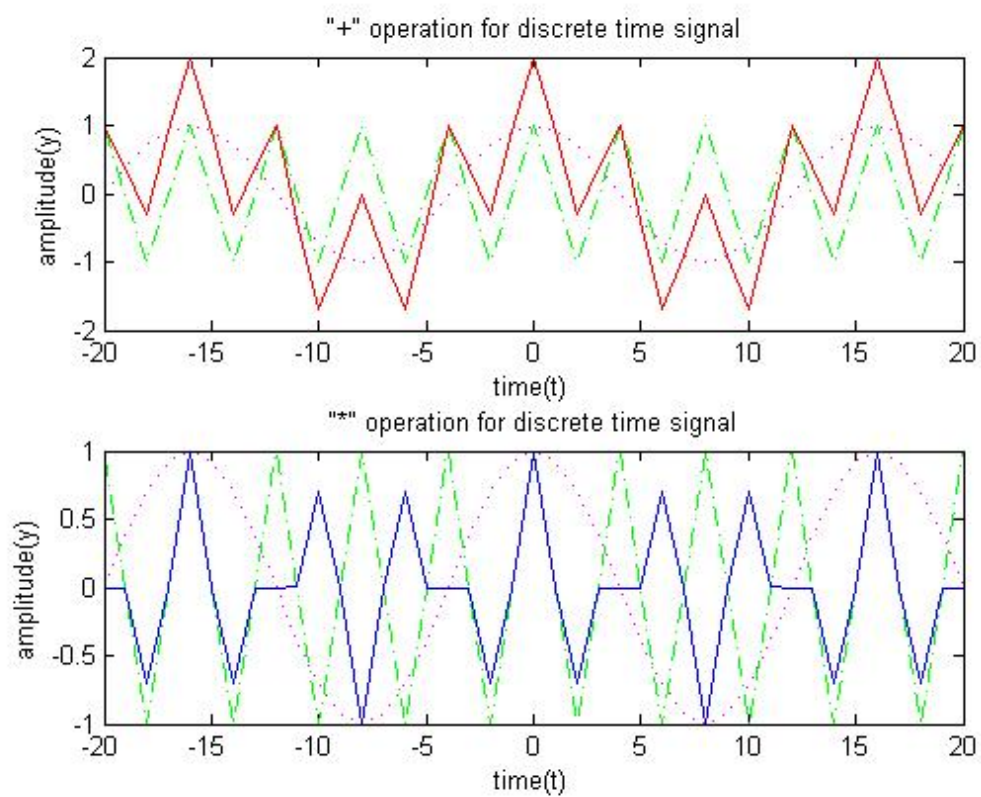
ylabel('amplitude(y)');

title('+ operation for discrete time signal')

```

subplot(2,1,2)
plot(n,y1,'m');
hold on;
plot(n,y2,'-.g');
plot(n,y4,'b');
xlabel('time(t)');
ylabel('amplitude(y)');
title('"*" operation for discrete time signal')

```



2. 尺度变换

程序2-3: 连续时间信号的尺度变换

```

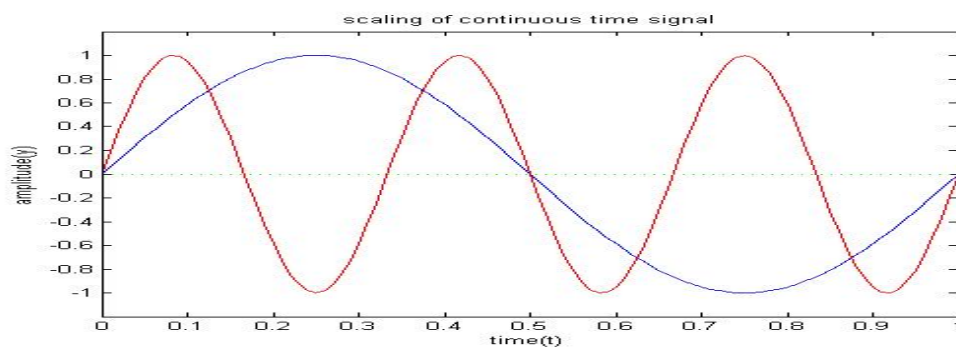
% scaling of continuous time signal
t=0:0.01:1;
m=3;
y1=sin(2*pi.*t);
y2=sin(2*m*pi.*t);

```

```

plot(t,y1,'b');
hold on
plot(t,y2,'r');
plot([0,1],[0,0],':g');
axis([0,1,-1.2,1.2]);
xlabel('time(t)');
ylabel('amplitude(y)');
title('scaling of continuous time signal')

```

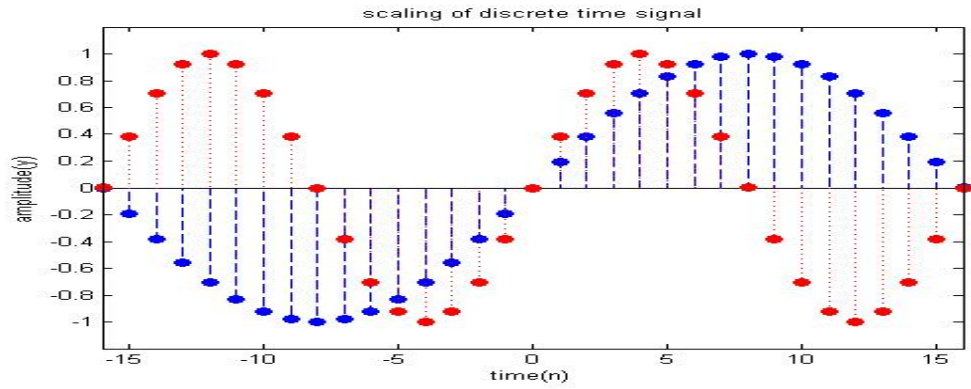


程序2-4：离散时间信号的尺度变换

```

% scaling of discrete time signal
n=-16:16;
m=2;
y1=sin(2*pi./32.*n);
y2=sin(2*m*pi./32.*n);
stem(n,y1,'fill','--b');
hold on
stem(n,y2,'fill',':r');
axis([-16,16,-1.2,1.2]);
xlabel('time(n)');
ylabel('amplitude(y)');
title('scaling of discrete time signal')

```



3. 微分/差分运算

程序2-5：微分运算

```
% approximate derivatives of continuous time signal
```

```
t=0:0.01:1;
```

```
syms t;
```

```
y1=sin(2*pi*t);
```

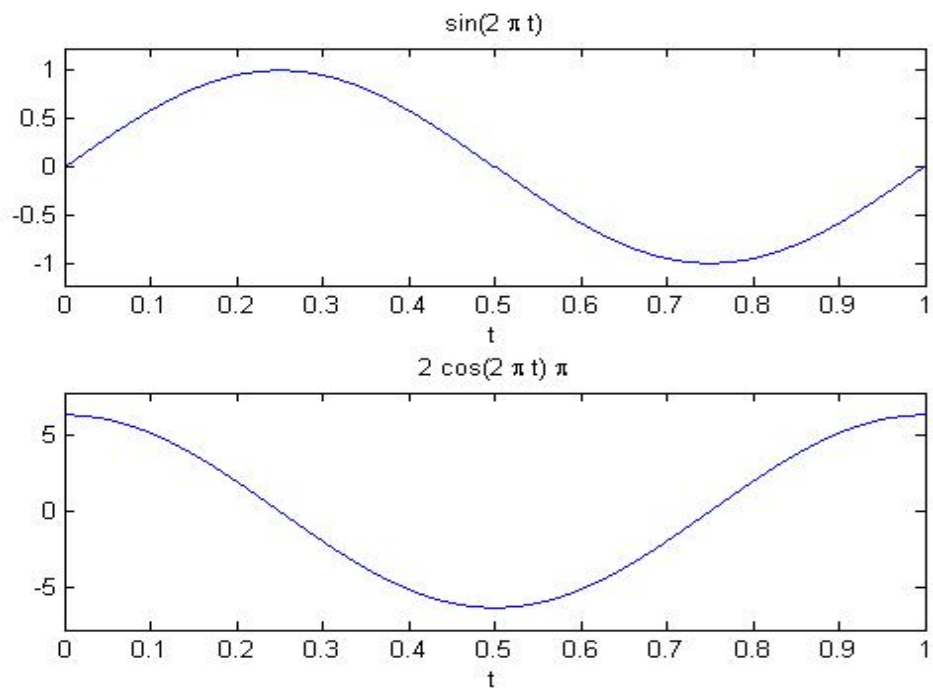
```
y2=diff(y1);
```

```
subplot(2,1,1)
```

```
ezplot(y1,[0 1]);
```

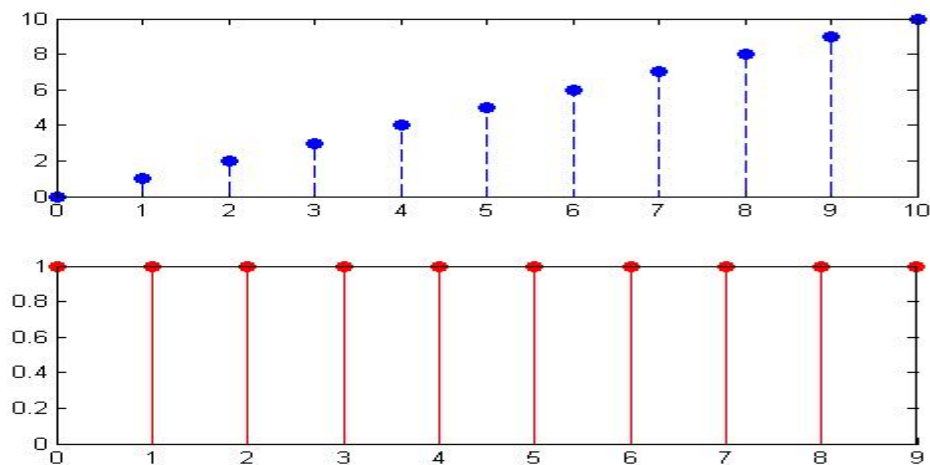
```
subplot(2,1,2)
```

```
ezplot(y2,[0,1]);
```



程序2-5：差分运算

```
clear all
% Differences of discrete time signal
n=0:10;
y1=[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10];
y2=diff(y1);
subplot(2, 1, 1)
stem(n, y1, 'fill', '--b')
subplot(2, 1, 2)
n2=0:9
stem(n2, y2, 'fill', 'r')
```



4. 设计程序，完成如下算式：

(1) 连续时间信号

(a) $y(t) = 2 \bullet \frac{\sin(t-1)}{t-1},$

(b) 设 $y(t) = e^{-2t}u(t)$ ，求 $y(-t)$ ， $y(3t-6)$ ， $y(t) \bullet \cos(2\pi t)$ ， $\frac{dy(t)}{dt}$ ， $\int_0^4 y(t)dt$ 。

(2) 离散时间信号

设 $y(n) = (0.5)^n u(n)$ ，求 $y(-n)$ ， $y(2n-4)$ ， $\nabla y(n)$ ， $y(n) \bullet \cos\left(\frac{1}{16}\pi n\right)$ 。

三、实验要求

1. 输入本实验的示范程序，验证结果；
2. 编制程序完成实验内容4；
3. 总结计算机实现连续时间信号和离散时间信号时域基本运算的方法

四、思考题

1. 分析和总结连续时间信号和离散时间信号时域基本运算的相同点和不同点。
2. 冲激函数及冲激偶函数的计算机模拟。

实验三 信号的卷积运算

一、实验目的

1. 熟悉连续时间信号和离散时间信号的卷积定义。
2. 理解连续时间信号和离散时间信号卷积运算的物理意义。
3. 掌握连续时间信号和离散时间信号卷积运算的计算方法。
4. 学会编程实现连续时间信号和离散时间信号的卷积运算并显示结果。

二、实验原理

卷积运算是指连续时间信号的卷积积分运算和离散时间信号的卷积和运算。卷积方法的原理就是将信号分解为冲激信号之和,借助系统的冲激响应 $h(t)$ (连续时间系统) 或者 $h(n)$ (离散时间系统), 求解线性时不变系统对任意激励信号的零状态响应 $y_{zs}(t)$ 或者 $y_{zs}(n)$:

$$y_{zs}(t) = e(t) * h(t)$$

$$y_{zs}(n) = e(n) * h(n)$$

1. 卷积积分和卷积和的定义

卷积积分的定义为:

$$f(t) = f_1(t) * f_2(t) = \int_{-\infty}^{+\infty} f_1(\tau) f_2(t - \tau) d\tau = \int_{-\infty}^{+\infty} f_2(\tau) f_1(t - \tau) d\tau \quad (1)$$

卷积和的定义为:

$$f(n) = f_1(n) * f_2(n) = \sum_{m=-\infty}^{+\infty} f_1(m) f_2(n - m) = \sum_{m=-\infty}^{+\infty} f_2(m) f_1(n - m) \quad (2)$$

2. 卷积运算的图形解释

卷积运算可以分解为四个步骤,前三个步骤为反褶、平移和相乘,最后一个步骤对于卷积积分(连续时间信号)为积分,而对于卷积和(离散时间信号)则为求和。

3. 卷积运算的计算机实现

计算机是一个离散时间系统,只能直接处理离散时间信号。因此,利用计算机来求取离散时间信号的卷积和是非常方便的。但若要求取两个连续时间信号的

卷积积分，就只能将这两个信号化为离散时间信号再计算了。卷积积分的计算机实现实际上只能采用信号的分段求和：

$$f(t) = f_1(t) * f_2(t) = \int_{-\infty}^{+\infty} f_1(\tau) f_2(t - \tau) d\tau = \lim_{\Delta t \rightarrow 0} \sum_{m=-\infty}^{m=\infty} f_1(m\Delta t) f_2(t - m\Delta t) \cdot \Delta t \quad (3)$$

显然，无论如何 Δt 总是一个有限值，因此利用计算机来计算卷积只能得到离散的近似值：

$$f(n\Delta t) = \sum_{m=-\infty}^{m=\infty} f_1(m\Delta t) f_2[(n - m)\Delta t] \cdot \Delta t \quad (4)$$

比较式(4)与式(2)可知，在利用计算机求取卷积积分运算时，实际上是用卷积和来近似的，但要多乘一个 Δt 。

三、需要掌握的 MATLAB 函数

卷积和的计算机实现有两种方法：利用循环语句或者调用 conv 函数。显然直接调用函数会更加简单，但前提是必须掌握 conv 函数的用法。conv 函数的功能为计算卷积和。调用格式为： $f = \text{conv}(f_1, f_2)$ ，其中是 f_1 和 f_2 是两个有限长度的序列，长度分别为 L_1 和 L_2 ，则卷积和 f 的长度为 $L_1 + L_2 - 1$ ；如果 f_1 和 f_2 的起始时刻分别为 m_1 和 m_2 ，则卷积和 f 的起始时刻为 $m_1 + m_2$ 。

卷积运算结果的显示会用到 subplot 和 plot 函数，请参考 MATLAB 帮助。

四、实验内容

1. 连续时间信号卷积运算

已知 $f_1 = u(t) - u(t - 1)$ ， $f_2 = u(t) - u(t - 3)$ ，请编程计算 $f_1 * f_1$ 和 $f_1 * f_2$ 并显示。

MATLAB 程序如下：

```
%时间连续函数卷积运算
deltat=1/1000;
t=-5:deltat:5;
f1=stepfun(t,0)-stepfun(t,1);
f2=stepfun(t,0)-stepfun(t,3);
subplot(221);
plot(t,f1);
```

```

axis([-1, 5, 0, 1.5]);
xlabel('time');
ylabel('f1(t)');
subplot(222);
plot(t, f2);
axis([-1, 5, 0, 1.5]);
xlabel('time');
ylabel('f2(t)');
fa=conv(f1, f1)*deltat;
t=-10:deltat:10;
subplot(223);
plot(t, fa);
axis([-1, 5, 0, 1.5]);
xlabel('time');
ylabel('f1(t)*f2(t)');
fb=conv(f1, f2)*deltat;
subplot(224);
plot(t, fb);
axis([-1, 5, 0, 1.5]);
xlabel('time');
ylabel('f2(t)*f2(t)');

```

显示结果如图 3-1 所示。

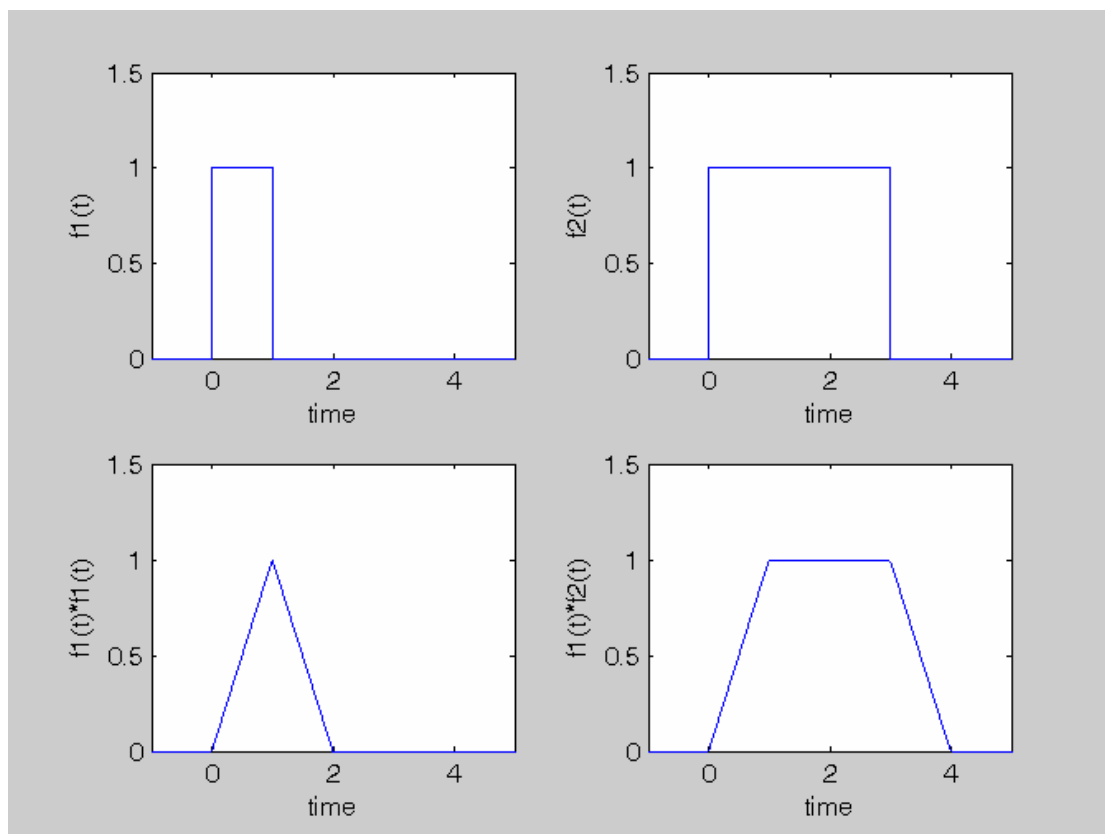


图 3-1 连续时间信号的卷积运算结果

2. 离散时间信号卷积运算

已知 $f_1 = u(n) - u(n-10)$, $f_2 = u(n) - u(n-30)$, 请编程计算 $f_1 * f_1$ 和 $f_1 * f_2$ 并显示。

MATLAB 程序如下：

%时间离散函数卷积运算

```
deltat=1;
```

```
t=-50:deltat:50;
```

```
f1=stepfun(t,0)-stepfun(t,10);
```

```
f2=stepfun(t,0)-stepfun(t,30);
```

```
subplot(221);
```

```
plot(t,f1);
```

```
axis([-50,50,0,1.5]);
```

```
xlabel('time');
```

```
ylabel('f1(t)');
```

```
subplot(222);
```

```

plot(t, f2);
axis([-50, 50, 0, 1.5]);
xlabel('time');
ylabel('f2(t)');
fa=conv(f1, f1);
t=-100:deltat:100;
subplot(223);
plot(t, fa);
axis([-50, 50, 0, 15]);
xlabel('time');
ylabel('f1(t)*f2(t)');
fb=conv(f1, f2)*deltat;
subplot(224);
plot(t, fb);
axis([-50, 50, 0, 15]);
xlabel('time');
ylabel('f2(t)*f2(t)');

```

显示结果如图 3-2 所示。

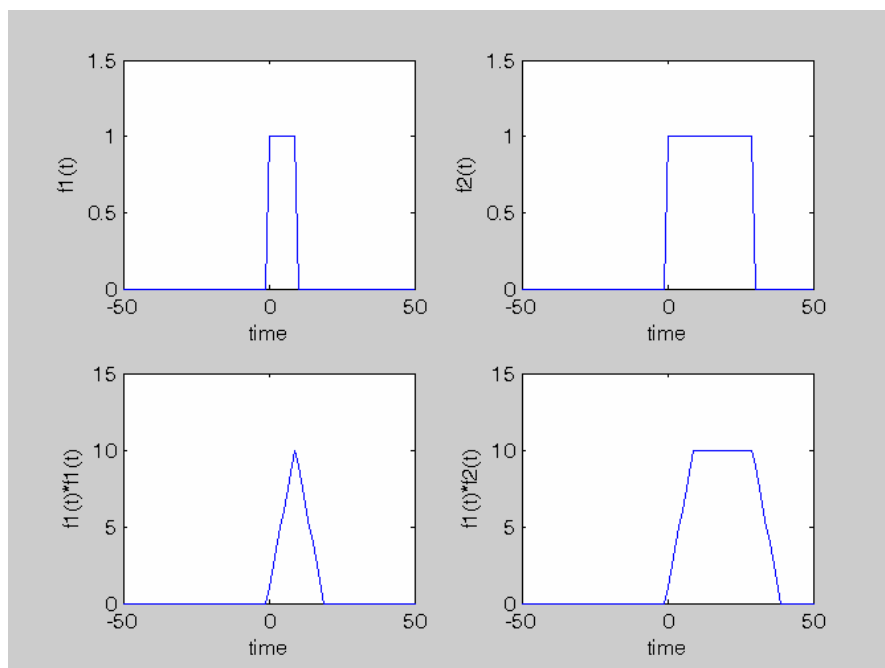


图 3-2 离散时间信号的卷积运算结果

3. 编程实现连续时间信号和离散时间信号的卷积运算

(1) 已知 $f_1 = u(t) - u(t-1)$, $f_2 = t[u(t) - u(t-3)]$, 请编程计算 $f_1 * f_2$ 并显示。

(2) 已知 $f_1 = 0.2^n[u(n) - u(n-10)]$, $f_2 = u(n) - u(n-30)$, 请编程计算 $f_1 * f_2$ 并显示。

五、实验要求

1. 输入实验内容 1 和 2 中提供的 MATLAB 程序, 生成 M 文件, 编译并运行, 观察实验结果。与理论计算结果相比较, 分析连续时间信号与离散时间信号卷积的差异。
2. 自行编制完整的 MATLAB 程序, 完成实验内容 3 中相应连续时间信号和离散时间信号的卷积运算。在实验报告中给出程序和显示结果。

六、思考题

1. 如果不采用 conv 函数, 如何利用循环实现卷积运算?
2. 函数 conv 给出的是卷积后得到的序列, 它所对应的时间序列是什么?

实验四 周期信号的合成与分解

一、实验目的

1. 在理论学习的基础上，通过实验深刻领会周期信号傅里叶级数分解的物理意义。
2. 理解实际应用中通常采用有限项级数来逼近无限项级数，此时方均误差随项数的增加而减小。
3. 观察并初步了解 Gibbs 现象。
4. 深入理解周期信号的频谱特点，比较不同周期信号频谱的差异。

二、实验原理

满足 Dirichlet 条件的周期信号 $f(t)$ 可以分解成三角函数形式的傅里叶级数，表达式为：

$$\begin{aligned} f(t) &= a_0 + a_1 \cos(\omega_1 t) + b_1 \sin(\omega_1 t) + \cdots + a_n \cos(n\omega_1 t) + b_n \sin(n\omega_1 t) + \cdots \\ &= a_0 + \sum_{n=1}^{\infty} [a_n \cos(n\omega_1 t) + b_n \sin(n\omega_1 t)] \end{aligned}$$

式中 n 为正整数；角频率 ω_1 由周期 T_1 决定： $\omega_1 = \frac{2\pi}{T_1}$ 。该式表明：任何满足

Dirichlet 条件的周期信号都可以分解成直流分量及许多正弦、余弦分量。这些正弦、余弦分量的频率必定是基频 $f_1 = \frac{1}{T_1}$ 的整数倍。通常把频率为 f_1 的分量称为基波，频率为 nf_1 的分量成为 n 次谐波。周期信号的频谱只会出现在 $0, \omega_1, 2\omega_1, \cdots, n\omega_1, \cdots$ 等离散的频率点上，这种频谱称为离散谱，是周期信号频谱的主要特点。 $f(t)$ 波形变化越剧烈，所包含的高频分量的比重就越大；变化越平缓，所包含的低频分量的比重就越大。

一般来说，将周期信号分解得到的三角函数形式的傅里叶级数的项数是无限的。也就是说，通常只有无穷项的傅里叶级数才能与原函数精确相等。但在实际应用中，显然无法取至无穷多项，而只能采用有限项级数来逼近无穷项级数。而且，所取项数越多，有限项级数就越逼近原函数，原函数与有限项级数间的方均误差就越小，而且低次谐波分量的系数不会因为所取项数的增加而变化。当选取

的傅里叶有限级数的项数越多，所合成的波形的峰起就越靠近 $f(t)$ 的不连续点。当所取得项数 N 很大时，该峰起值趋于一个常数，约等于总跳变值的 9%，这种现象称为 Gibbs 现象。

三、需要掌握的 MATLAB 函数

结果的显示会用到 plot 和 pause 函数，请参考 MATLAB 帮助。

四、实验内容

1. 周期对称方波信号的合成

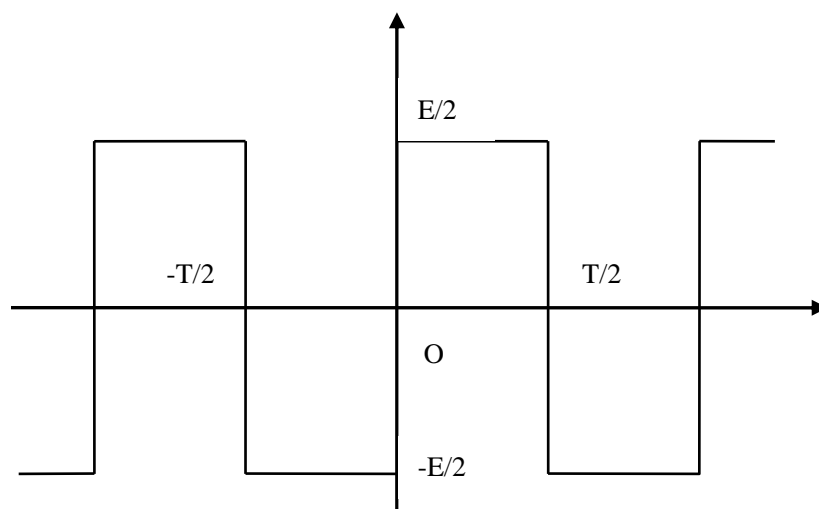


图 4-1 奇对称的周期方波信号

图示方波既是一个奇对称信号，又是一个奇谐信号。根据函数的对称性与傅里叶系数的关系可知，它可以用无穷个奇次谐波分量的傅里叶级数来表示：

$$f(t) = \frac{2E}{\pi} \sum_{k=0}^{\infty} \sin[2\pi(2k+1)f_0t] \cdot \frac{1}{2k+1}$$

选取奇对称周期方波的周期 $T = 0.02s$ ，幅度 $E = 6$ ，请采用有限项级数替代无限项级数来逼近该函数。分别取前 1、2、5 和 100 项有限级数来近似，编写程序并把结果显示在一幅图中，观察它们逼近方波的过程。

MATLAB 程序如下：

%奇对称方波合成

t=0:0.001:0.1;

sishu=12/pi;


```

y=sishu*sin(100*pi*t);
subplot(221)
plot(t,y);
axis([0,0.1,-4,4]);
xlabel('time');
ylabel('前1项有限级数');
y=sishu*(sin(100*pi*t)+sin(3*100*pi*t)/3);
subplot(222);
plot(t,y);
axis([0,0.1,-4,4]);
xlabel('time');
ylabel('前2项有限级数');
y=sishu*(sin(100*pi*t)+sin(3*100*pi*t)/3+sin(5*100*pi*t)/5+sin(7*
100*pi*t)/7+sin(9*100*pi*t)/9);
subplot(223)
plot(t,y);
axis([0,0.1,-4,4]);
xlabel('time');
ylabel('前5项有限级数');
t=0:0.001:0.1;
y=0;
for i=1:100
    y=y+sishu*(sin((2*i-1)*100*pi*t)/(2*i-1));
end
subplot(224);
plot(t,y);
axis([0,0.1,-4,4]);
xlabel('time');
ylabel('前100项有限级数');

```

显示结果如图 4-2 所示。

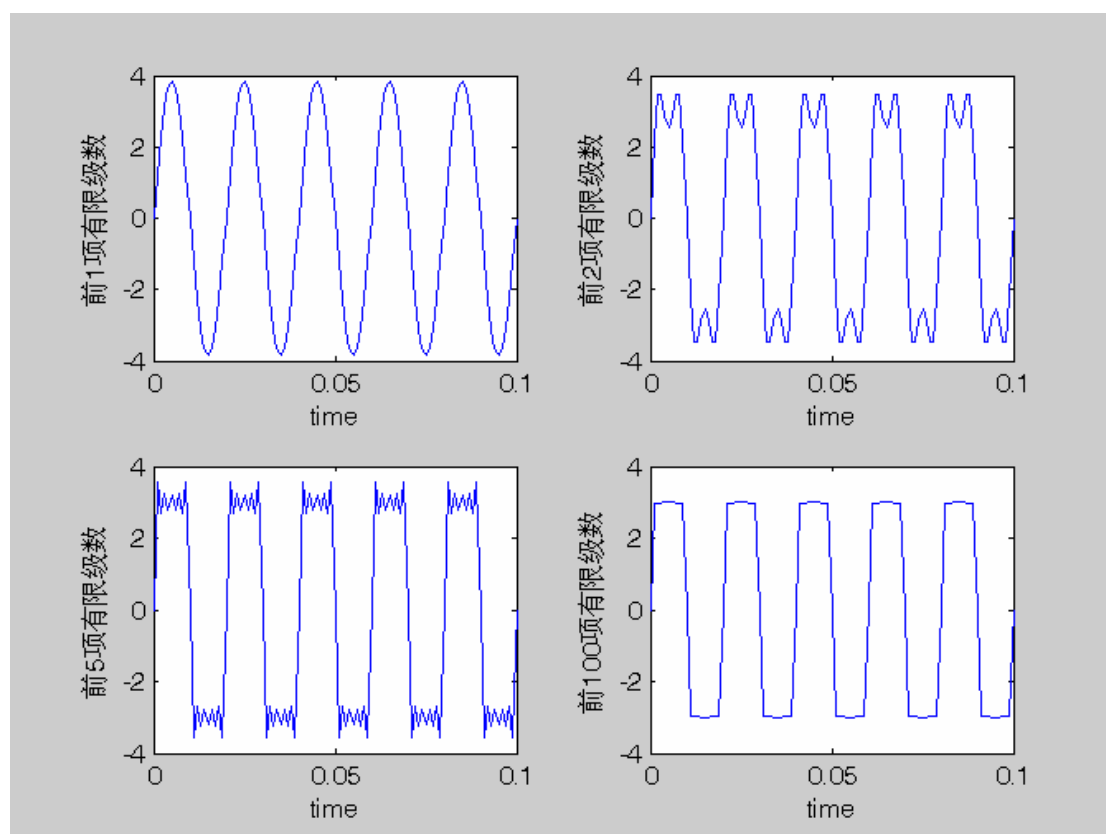


图 4-2 奇对称方波信号的合成

2. 观察 Gibbs 现象

分别取前 10、20、30 和 40 项有限级数来逼近奇对称方波，观察 Gibbs 现象。

MATLAB 程序如下：

%观察 Gibbs 现象

t=0:0.001:0.04;

sishu=12/pi;

y=0;

for i=1:5

y=y+sishu*(sin((2*i-1)*100*pi*t)/(2*i-1));

end

subplot(221)

plot(t,y);

```

axis([0, 0.04, -4, 4]);
xlabel(' time');
ylabel(' 前 5 项有限级数');
y=0;
for i=1:6
    y=y+sishu*(sin((2*i-1)*100*pi*t)/(2*i-1));
end
subplot(222);
plot(t,y);
axis([0, 0.04, -4, 4]);
xlabel(' time');
ylabel(' 前 6 项有限级数');
y=0;
for i=1:7
    y=y+sishu*(sin((2*i-1)*100*pi*t)/(2*i-1));
end
subplot(223)
plot(t,y);
axis([0, 0.04, -4, 4]);
xlabel(' time');
ylabel(' 前 7 项有限级数');
y=0;
for i=1:8
    y=y+sishu*(sin((2*i-1)*100*pi*t)/(2*i-1));
end
subplot(224);
plot(t,y);
axis([0, 0.04, -4, 4]);
xlabel(' time');
ylabel(' 前 8 项有限级数');

```

显示结果如图 4-3 所示。

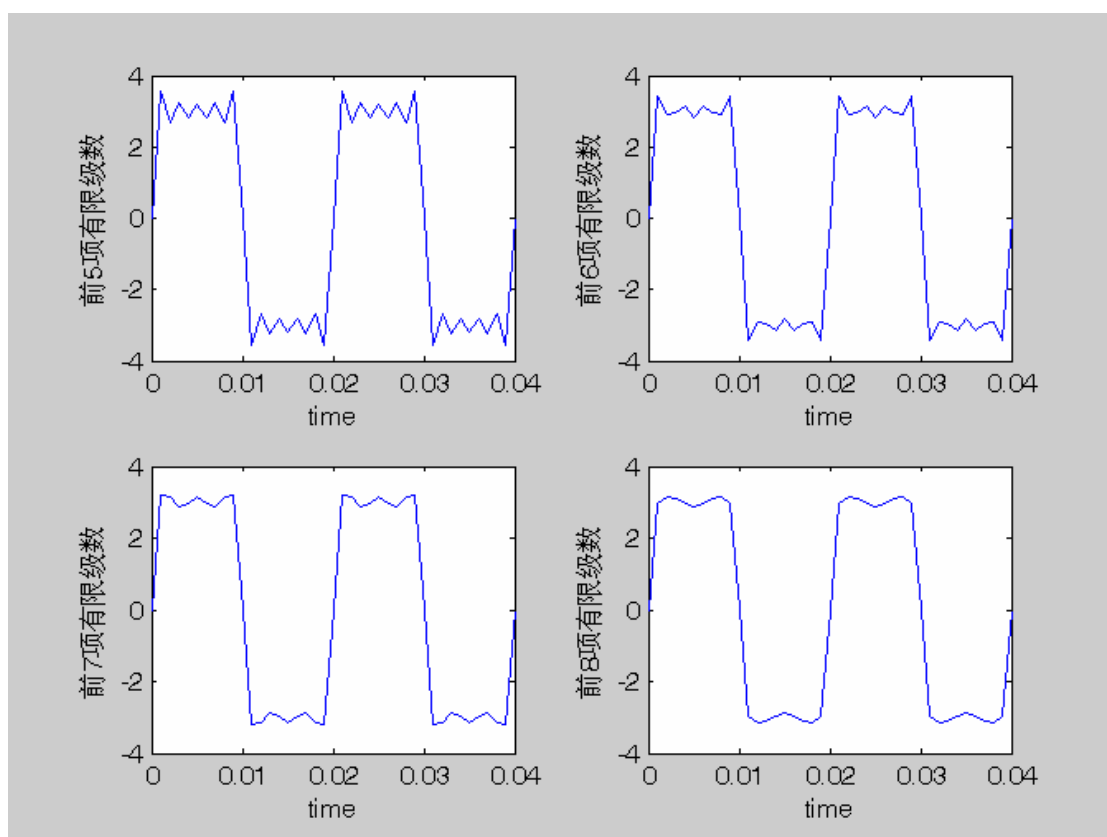


图 4-3 Gibbs 现象

3. 周期对称三角信号的合成

设计采用有限项级数逼近偶对称周期三角信号的实验，编制程序并显示结果。

4. 周期信号的频谱

分析奇对称方波信号与偶对称三角信号的频谱，编制程序并显示结果，深入讨论周期信号的频谱特点和两信号频谱的差异。

五、实验要求

1. 输入实验内容 1 中提供的奇对称方波信号合成的 MATLAB 程序，生成 M 文件，编译并运行，观察合成结果。
2. 输入实验内容 2 中提供的有限项级数逼近方波信号的 MATLAB 程序，生成 M 文件，编译并运行，观察 Gibbs 现象。
3. 自行编制完整的 MATLAB 程序，完成实验内容 3 中偶对称三角信号的合成。在实验报告中给出程序和显示结果。

4. 自行编制完整的 MATLAB 程序, 完成实验内容 4 中奇对称方波信号和偶对称三角信号的频谱分析。在实验报告中给出程序和显示结果, 讨论周期信号的频谱特点和两信号频谱的差异。

六、思考题

1. 利用有限项的指数形式的傅里叶级数重复奇对称方波信号的合成。
2. 分析时域信号的间断性与其频谱谐波收敛速率的对应关系。

实验五 二阶状态轨迹的显示

一、实验目的

1. 熟悉二阶连续时间系统状态轨迹的概念。
2. 掌握连续时间系统冲激响应、阶跃响应的求解方法。
3. 观察过阻尼，欠阻尼，临界阻尼情况下，RLC 电路的状态轨迹。

二、实验原理

系统数学模型的描述方法有输入输出描述法和状态变量分析法。在输入输出描述法中，主要建立系统的输入（激励）与系统的输出（响应）之间的关系，不关心系统内部的变化情况。在状态变量分析法中，需先确定状态变量后，建立描述系统状态变量与输入之间的关系（状态方程），以及建立系统输出变量与系统状态变量及系统输入之间的关系（输出方程），这种分析法不仅能反映输入与输出的关系，而且能了解系统内部的变化过程。

在状态变量分析法中，状态变量是建立状态方程和输出方程的关键变量，是能描述系统动态特性的一组独立完备的变量。对于一个二阶系统，则可以用两个状态变量来描述系统的动态特性，这两个状态变量构成的列矢量称为状态矢量，以这两个状态变量为坐标轴而形成的空间称为二维状态空间。在状态空间中状态矢量端点随时间变化而描述出的路径为状态轨迹。因此状态轨迹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。

三、涉及的 MATLAB 相关内容

1. MATLAB 图形用户界面（GUI）设计

利用 MATLAB 图形用户界面工具设计修改 RLC 电路参数及显示 RLC 电路二阶状态轨迹的界面；并用程序控制界面中的控件，显示 RLC 电路的二阶状态轨迹。

2. ss 函数

功能：建立系统状态空间模型。

调用格式：sys = ss(a, b, c, d)

其中，a, b, c, d 为状态方程和输出方程的矩阵，sys 为建立的状态空间模型。

3. step 函数

功能：求线性时不变系统的阶跃响应。

调用格式：y = step(sys, t)

其中，y 为系统的阶跃响应，sys 为系统的状态空间模型。

4. axes 函数

功能：设置当前轴。

调用格式：axes(h)

其中，h 为已存在轴的句柄

四、 实验内容与方法

1. 验证性实验

图 1 所示为 RLC 电路，可看作一个二阶连续时间系统。对于该二阶系统，若要用状态变量分析来描述该系统的数学模型，可选用 $i_L(t)$ 和 $v_c(t)$ 作为状态变量，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，状态矢量 $\begin{bmatrix} i_L(t) \\ v_c(t) \end{bmatrix}$ 随时间变化而描出的路径叫状态轨迹。

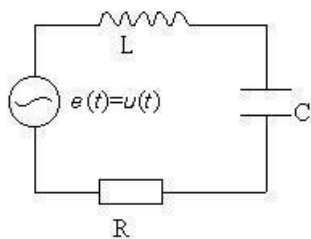


图 1 RLC 电路

本实验将利用计算机模拟该系统的状态轨迹，实验步骤如下：

- (a) 在 MATLAB 命令窗口重输入 “guide”，启动 GUI
- (b) 利用 GUI 编辑图 2 所示界面，并将其保存为 trace.fig 文件。

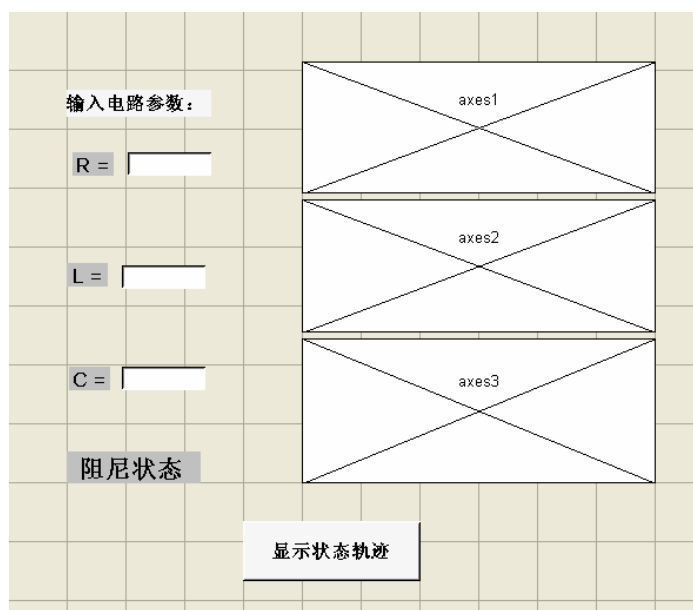


图 2 修改 RLC 电路参数及显示 RLC 电路二阶状态轨迹的界面

- (c) 运行 GUI，并生成 trace.m 文件。
- (d) 选中图 2 所示界面中“显示状态轨迹”按钮，点击右键选择菜单上的 View Callbacks，选择 Callback，MATLAB Editor 会自动调到该按钮对用的 Callback Function 上，可以直接在那里填写代码，编程控制 GUI。其中“显示状态轨迹”按钮 Callback Function 的参考程序代码如下：

```
function pushbutton1_Callback(hObject, eventdata, handles)
    t = 0:0.1:100;
    %从界面上获取电路参数
    R = str2num(get(handles.edit1,'string'));
    L = str2num(get(handles.edit2,'string'));
    C = str2num(get(handles.edit3,'string'));
    %若系统以  $i_L(t)$ ， $v_c(t)$  为响应，以  $e(t)$  为激励，
    %确定系统状态方程和输出方程中的 a, b, c, d 矩阵
    a = [-R/L -1/L; 1/C 0];
    b = [1/L; 0];
    c = [1 0; 0 1];
    d = [0];
```



```

sys = ss(a, b, c, d);          %建立系统状态空间模型
Response = step(sys, t); %求系统的阶跃响应
axes(handles.axes1);
plot(t, Response(:, 1), 'b-', 'linewidth', 3); %显示  $i_L(t)$ 
ylabel('il(t)', 'fontsize', 14)
axes(handles.axes2);
plot(t, Response(:, 2), 'r-', 'linewidth', 3); %显示  $v_c(t)$ 
ylabel('vc(t)', 'fontsize', 14)
axes(handles.axes3);
plot(Response(:, 2), Response(:, 1), 'linewidth', 3); %显示状态轨迹
xlabel('vc(t)', 'fontsize', 14)
ylabel('il(t)', 'fontsize', 14)
%判断系统的阻尼状态
alph = R/(2*L);
omega = 1/sqrt(L*C);
if (R==0)
    str = '无阻尼';
else
    if(alph>omega)
        str = '过阻尼';
    end
    if(alph==omega)
        str = '临界阻尼';
    end
    if(alph<omega)
        str = '欠阻尼';
    end
end
set(handles.text1, 'string', str);

```

2. 程序设计实验

已知某系统的系统函数为 $H(s) = \frac{s+3}{s^2+3s+2}$ ，若系统起始状态为零，在激励信号为 $e(t) = \delta(t)$ 情况下，画出该系统的状态轨迹。

五、 实验要求

1. 当 RLC 电路参数变化时，分别给出过阻尼，欠阻尼，临界阻尼情况下 RLC 电路的状态轨迹。
2. 编写程序完成程序设计实验，并给出实验结果。

六、 思考题

连续时间系统状态方程的求解方法。

实验六 信号的抽样与内插

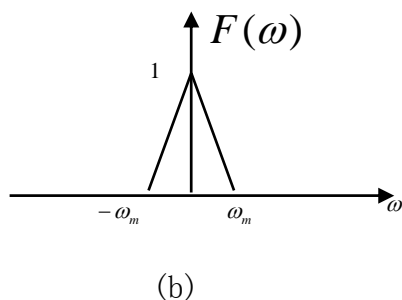
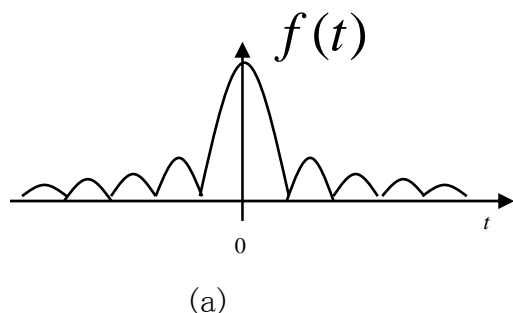
一、实验目的

1. 熟悉信号的抽样与恢复过程；
2. 观察欠采样与过采样时信号频谱的变化；
3. 掌握采样频率的确定方法。

二、实验原理

由时域抽样定理可知，若有限带宽的连续时间信号 $f(t)$ 的最高角频率为 ω_m ，则信号 $f(t)$ 可以用等间隔的抽样值唯一表示，且抽样间隔 T_s 必须不大于 $\frac{1}{2f_m}$ ，或者说抽样频率 $\omega_s \geq 2\omega_m$ 。

图 6-1 所示为信号抽样与恢复示意图，其中图 6-1 (a) 中为抽样前带限信号 $f(t)$ ，其频谱 $F(\omega)$ 为图 6-1 (b) 所示，最高频率为 ω_m 。当该信号被抽样间隔为 T_s 的冲激序列抽样时，若 T_s 大于 $\frac{1}{2f_m}$ （过采样），则抽样后信号 $f_s(t)$ 的频谱为图 1(f) 所示，频谱没有产生混迭现象。将抽样后信号 $f_s(t)$ 通过一个低通滤波器，能恢复原信号 $f(t)$ 。若 T_s 小于 $\frac{1}{2f_m}$ （欠采样），则抽样后信号 $f_s(t)$ 的频谱将产生混迭现象，不能从抽样后信号 $f_s(t)$ 中恢复原信号 $f(t)$ 。



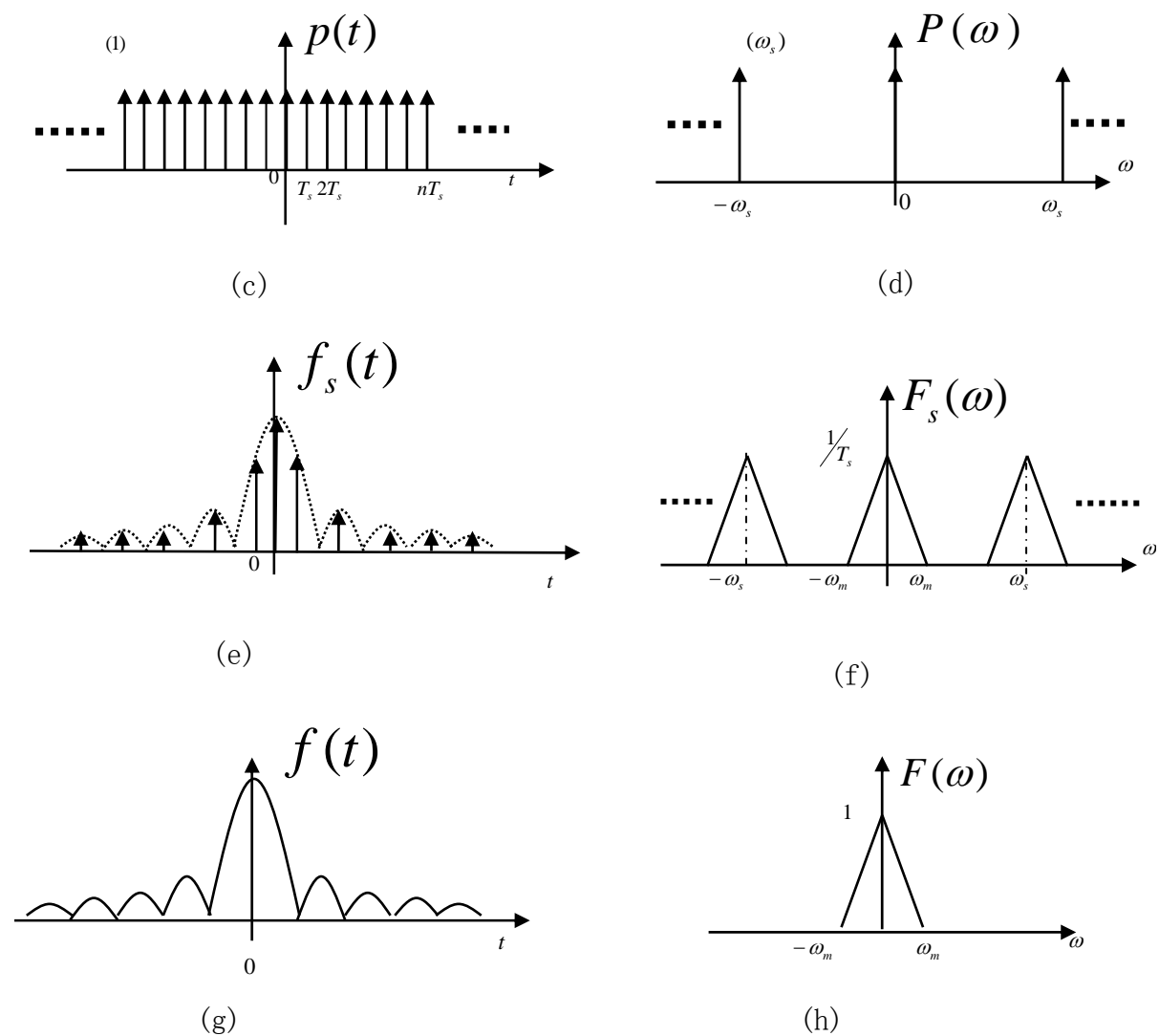


图 6-1 信号抽样与恢复示意图

三、涉及的 MATLAB 相关内容

1. Simulink 仿真

利用 Simulink 完成信号的抽样与内插实验仿真设计。

2. fft 函数

功能：离散傅里叶变换。

调用格式：y = fft(x, n)

3. abs 函数

功能：求绝对值和复数的模。

调用格式：y = abs(x)

四、实验内容与方法

设计信号 $x(t) = \sin(2\pi ft)$ ， $f = 1\text{Hz}$ 的抽样与恢复的实验，实验步骤如下：

1. 在 MATLAB 命令窗口中输入“simulink”，启动 Simulink Library Browser；
2. 在 Simulink Library Browser 中，新建一个模型文件，编辑模型文件，建立如图 2 所示的抽样与内插的仿真模型，并保存为 sample.mdl；

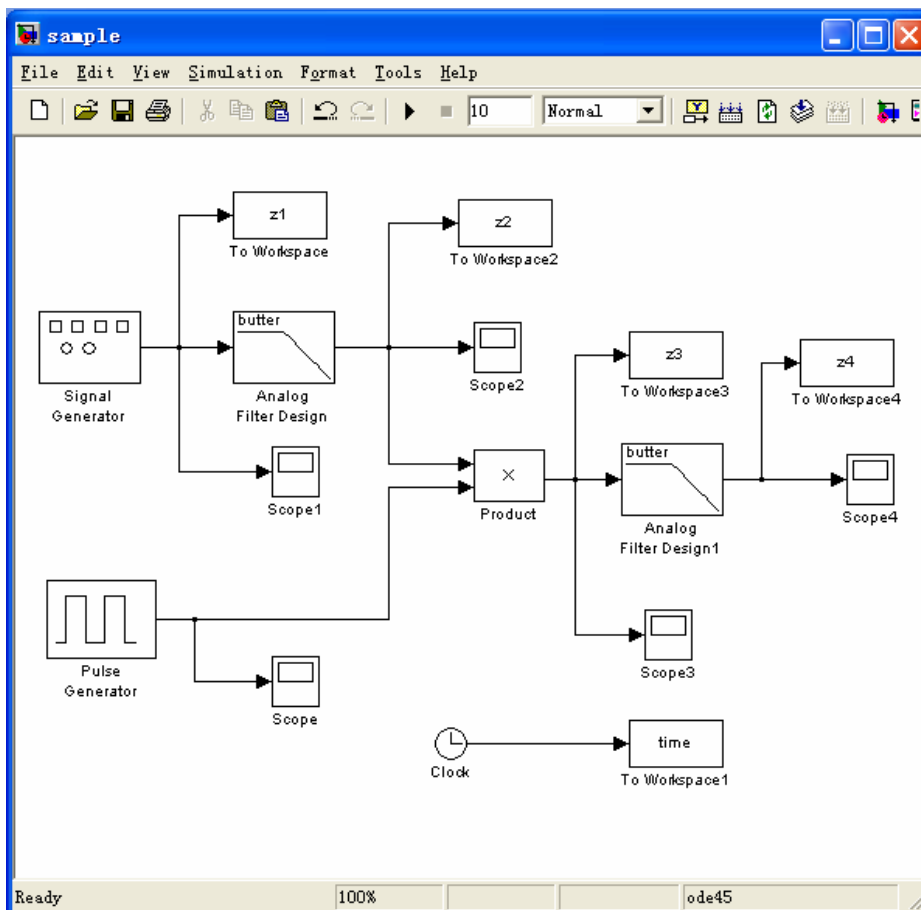
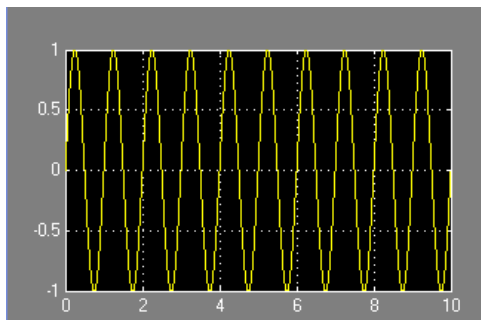


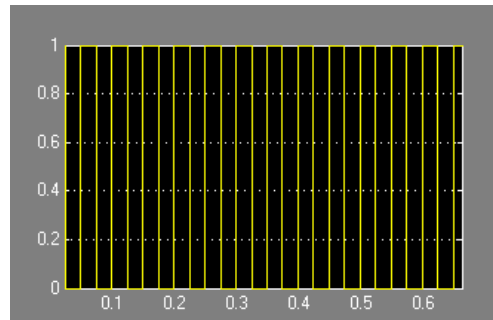
图 2 抽样与内插的仿真模型

3. 分别在欠采样与过采样条件下，配置各模块的参数（如信号源的频率，抽样脉冲的间隔，低通滤波器的截止频率等）。
4. 在模型文件的菜单中选择 Simulation→Start，运行在欠采样、与过采样条件下的仿真模型；
5. 仿真结束后，打开示波器，观察在欠采样与过采样条件下的仿真结果。

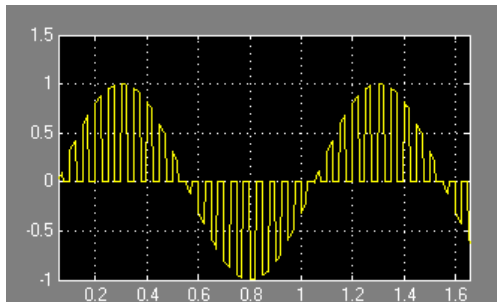
图 3 所示为过采样条件下的仿真结果：



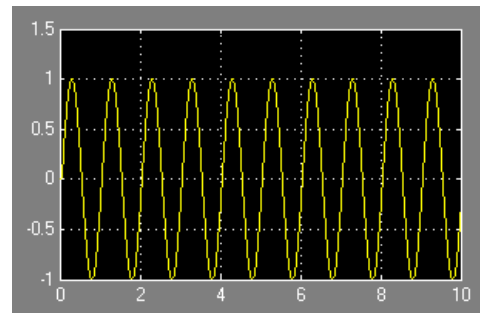
信号源的波形



抽样脉冲的波形



抽样后信号的波形



恢复以后信号的波形

图 6-3 过采样条件下的仿真结果

6. 画出各信号的频谱图，参考程序代码如下：

```

N=length(time);    Ts=(time(N) - time(1))/N;
m=floor(N/2);      Ws=2*pi/Ts;      W=Ws*(0:m)/N;
F=fft(z1,N);  FF=F(1:m+1);  F11=abs(FF);
F=fft(z2,N);  FF=F(1:m+1);  F12=abs(FF);
F=fft(z3,N);  FF=F(1:m+1);  F13=abs(FF);
F=fft(z4,N);  FF=F(1:m+1);  F14=abs(FF);
figure(1)
plot(W,F11,'b',-W,F11,'b');
title('输入信号的幅频特性');
xlabel('频率 (Rad/s)');
figure(2)
plot(W,F12,'b',-W,F12,'b');
title('滤波后信号的幅频特性');
xlabel('频率 (Rad/s)');
figure(3)

```

```

plot(W,F13,'b',-W,F13,'b');
title('抽样后信号的幅频特性');
xlabel('频率 (Rad/s)');
figure(4)
plot(W,F14,'b',-W,F14,'b');
title('恢复后信号的幅频特性');
xlabel('频率 (Rad/s)');

```

7. 改变信号源的波形、如将信号源的波形换成方波、三角波后重复上述实验，观察信号波形与频谱的变化。

五、实验要求

1. 当信号源波形改变为方波和三角波后，给出信号源、抽样后与恢复后信号的波形。
2. 自行编写程序画出各信号的频谱图。
3. 在实验报告中给出各信号的频谱图。

六、思考题

1. 说明采样频率变化对信号时域和频域特性的影响。
2. 分析采样与内插仿真模型中两个低通滤波器的作用。

实验七 滤波器的设计

一、实验目的

1. 掌握模拟滤波器的设计方法。

(1) 模拟原型滤波器的设计, 包括Butterworth、chebyshev I、chebyshev II、Elliptic滤波器的设计

(2) 由模拟低通滤波器到其它类型滤波器的变换, 包括低通→高通, 低通→带通, 低通→带阻。

2. 掌握无限冲激响应(IIR)数字滤波器设计方法。

3. 学习并掌握运用MATLAB设计滤波器的方法。

4. 学习运用MATLAB进行信号时域和频域分析的方法。

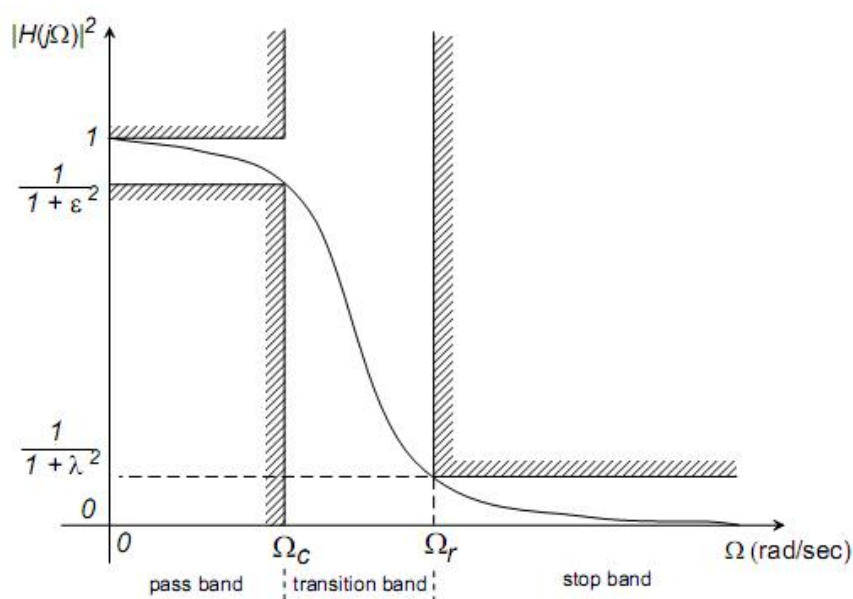
二、实验原理

低通原型滤波器设计是基于滤波器系统函数 $H(j\Omega)$ 的幅度平方函数

$|H(j\Omega)|^2$, 而非相位特性, 根据通带截止角频率 Ω_c 、阻带角频率 Ω_r 、通带起伏参数 ϵ 、和阻带衰落参数 λ , 计算原型滤波器的阶次、截至频率 Ω_c 和滤波器系统转移函数。原型滤波器的 $|H(j\Omega)|^2$ 如图7-1所示。

通带内, 滤波器应满足: $\frac{1}{1+\epsilon^2} < |H(j\Omega)|^2 \leq 1, |\Omega| \leq \Omega_c$ 。

阻带内, 滤波器应满足: $|H(j\Omega)|^2 < \frac{1}{1+\lambda^2}, |\Omega| > \Omega_c$ 。



1、Butterworth滤波器的设计

Butterworth原型滤波器的定义是：

$$|H(j\Omega)|^2 = \frac{1}{1 + \left(\frac{\Omega}{\Omega_c}\right)^{2N}} \quad (7-1)$$

其中，N为正整数，是Butterworth滤波器的阶次， Ω_c 是滤波器的截止角频率。

当 $\Omega = \Omega_c$ 时， $|H(j\Omega)|^2 = \frac{1}{2}$ ，故 Ω_c 是滤波器的-3dB点，即

$$10\log_{10}\left(|H(j\Omega)|^2\right)\Big|_{\Omega=\Omega_c} = -3dB。$$

设Butterworth滤波器的系统函数为 $H(s)$ ，令 $\Omega = \frac{s}{j}$ 代入式 (7-1)，得

$$|H(j\Omega)|^2\Big|_{\Omega=\frac{s}{j}} = H(s)H(-s) = \frac{1}{1 + \left(\frac{s}{j\Omega_c}\right)^{2N}} = \frac{(j\Omega_c)^{2N}}{s^{2N} + (j\Omega_c)^{2N}}$$

$H(s)H(-s)$ 共有2N个极点： $s_k = \Omega_c e^{j\pi(2k+N-1)/(2N)}$

为了得到稳定的系统函数 $H(s)$ ，取左半平面的极点给 $H(s)$ 。

2、MATLAB低通滤波器设计函数

(1) 低通滤波器阶次的计算

模拟低通滤波器阶次的计算：

$[n, Wn] = \text{buttord}(Wp, Ws, Rp, Rs, 's')$ ——Butterworth 滤波器；

$[n, Wp] = \text{cheblord}(Wp, Ws, Rp, Rs, 's')$ ——Chebyshev I 型滤波器

$[n, Ws] = \text{cheb2ord}(Wp, Ws, Rp, Rs, 's')$ ——Chebyshev II 型滤波器

$[n, Wp] = \text{ellipord}(Wp, Ws, Rp, Rs, 's')$ ——椭圆滤波器

IIR 数字滤波器阶次计算：

$[n, Wn] = \text{buttord}(Wp, Ws, Rp, Rs)$ ——Butterworth 滤波器；

$[n, Wp] = \text{cheblord}(Wp, Ws, Rp, Rs)$ ——Chebyshev I 型滤波器

$[n, Ws] = \text{cheb2ord}(Wp, Ws, Rp, Rs)$ ——Chebyshev II 型滤波器

$[n, Wp] = \text{ellipord}(Wp, Ws, Rp, Rs)$ ——椭圆滤波器

(2) 滤波器设计

模拟滤波器设计：

$[b, a] = \text{butter}(n, Wn, 's')$ ；

$[b, a] = \text{cheby1}(n, R, Wp, 's')$ ；

$[b, a] = \text{cheby2}(n, R, Wst, 's')$ ；

$[b, a] = \text{ellip}(n, Rp, Rs, Wp, 's')$ ；

IIR 数字滤波器设计：

$[z, p, k] = \text{butter}(n, Wn)$ ；

$[z, p, k] = \text{cheby1}(n, R, Wp)$ ；

$[z, p, k] = \text{cheby2}(n, R, Wst)$ ；

$[z, p, k] = \text{ellip}(n, Rp, Rs, Wp)$ ；

(3) 由模拟低通滤波器到其它类型滤波器的变换函数

$[bt, at] = \text{lp2bp}(b, a, Wo, Bw)$ ；——低通到带通

$[bt, at] = \text{lp2bs}(b, a, Wo, Bw)$ ，——低通到带阻

$[bt, at] = \text{lp2hp}(b, a, Wo)$ ；——低通到高通

(4) 模拟滤波器到数字滤波器的转换函数

$[bz, az] = \text{impinvar}(b, a, fs)$ ；——冲激不变法

$[zd, pd, kd] = \text{bilinear}(z, p, k, fs)$ ；——双线性变换法

三、实验内容

1. 给定模拟低通滤波器技术指标：

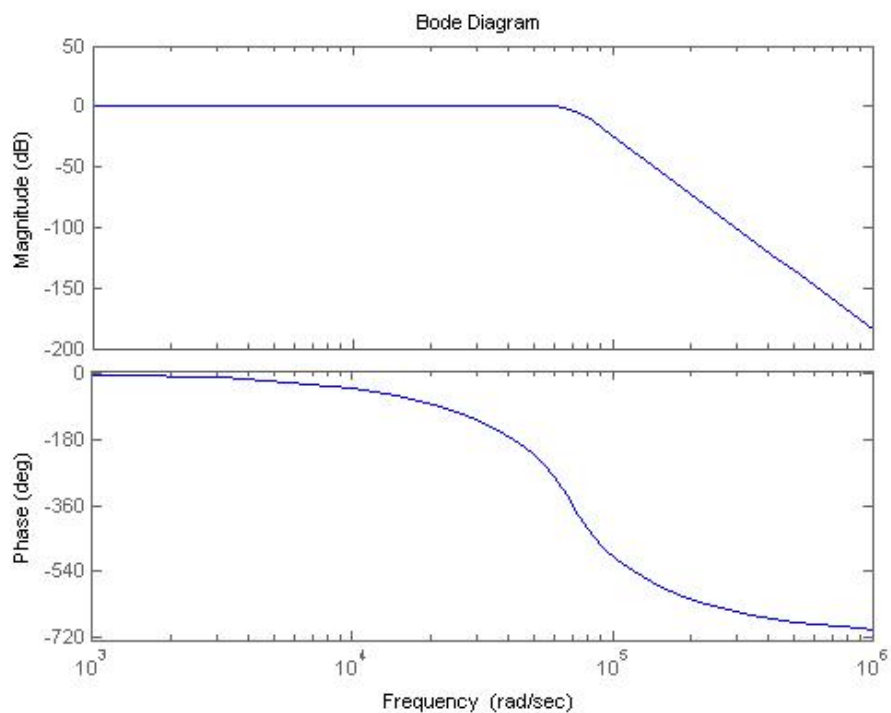
通带内允许起伏：-1dB， $0 \leq \Omega \leq 2\pi \times 10^4 \text{ rad/sec}$

阻带衰减： $\leq 40\text{dB}$ ， $\Omega \geq 2\pi \times 2 \times 10^4 \text{ rad/sec}$

设计 Butterworth 低通滤波器，实现上述指标。

程序 7-1：

```
clear all
close all
% design a low-pass Butterworth filter
% wp is passband corner frequency, wp=2*pi*10^4
% rp is the passband ripple, which is no more than -1dB (decibel)
% ws is the stopband corner frequency, 2*2*pi*10^4
% rs is the stopband ripple, which is less than -40dB
% Get the cutoff frequency and the order of the lowpass filter
wp=2*pi*10^4;
ws=2*2*pi*10^4;;
rp=1;
rs=40;
[n,wn]=buttord(wp,ws,rp,rs,'s');
% The cutoff frequency is wn.
% The filter order is n.
[b,a]=butter(n,wn,'s');
butter_lpf=tf(b,a);
% plot the bode curve.
bode(butter_lpf);
```



2. 设计一个 Butterworth 低通滤波器，拟从原始信号 $x(t) = \sin(2\pi \times 10t) + \sin(2\pi \times 50t)$ 中提出频率为 10 的信号。

滤波器技术指标：

通带内允许起伏：-3dB， $0 \leq \Omega \leq 2\pi \times 10 \text{ rad/sec}$

阻带衰减： $\leq 60\text{dB}$ ， $\Omega \geq 2\pi \times 50 \text{ rad/sec}$

程序 7-2：

```
clear all
close all
% create two continuous time signal
f1=10;
f2=50;
% set the sample rate fs=10*f2
fs=20*f2;
% design a low-pass Butterworth filter to extract y1 from y3
% wp is passband corner frequency, wp=2*f1/fs
```

```

% rp is the passband ripple, which is no more than -3dB (decibel)
% ws is the stopband corner frequency,  $ws=2*f2/fs$ 
% rs is the stopband ripple, which is less than -60dB
% Get the cutoff frequency and the order of the lowpass filter
wp=2*f1/fs;
ws=2*f2/fs;
rp=3;
rs=60;
[n, wn]=buttord(wp, ws, rp, rs)
% The cutoff frequency is wn.
% The filter order is n.
[b, a]=butter(n, wn);
figure (1)
freqz(b, a, 512, fs);
% plot the continuous time signal
w1=2*pi*f1/fs;
w2=2*pi*f2/fs;
N=1024;
n=0:N-1;
%
y1=sin(w1.*n);
y2=sin(w2.*n);
z=y1+y2;
% plot the signal on time domain
% plot the signal y1
t=n./fs;
figure(2);
subplot(2, 2, 1);
plot(t, y1, 'b');

```

```

xlim([0 0.3]);
title('Đ°Å1');
xlabel('time(t)');
ylabel('voltage(v)');
grid on
subplot(2,2,2);
plot(t,y2,':g');
xlim([0 0.3]);
title('Đ°Å2');
xlabel('time(t)');
ylabel('voltage(v)');
grid on
subplot(2,2,3);
plot(t,z,'r');
xlim([0 0.3]);
title('Ô-Ê¼Đ°Å');
xlabel('time(t)');
ylabel('voltage(v)');
grid on
%
% filtering y3 ;
fz=filter(b,a,z);
subplot(2,2,4);
plot(t,fz);
title('Â² ¨ ºóĐ°Å');
xlim([0 0.3]);
xlabel('time(s)')
ylabel('amplitude(v)')
grid on

```

```

% Spectrum analysis
fy1=fft(y1);
fy2=fft(y2);
fz1=fft(z);
fz2=fft(fz);
ffs=n./N.*fs;

%
figure (3)

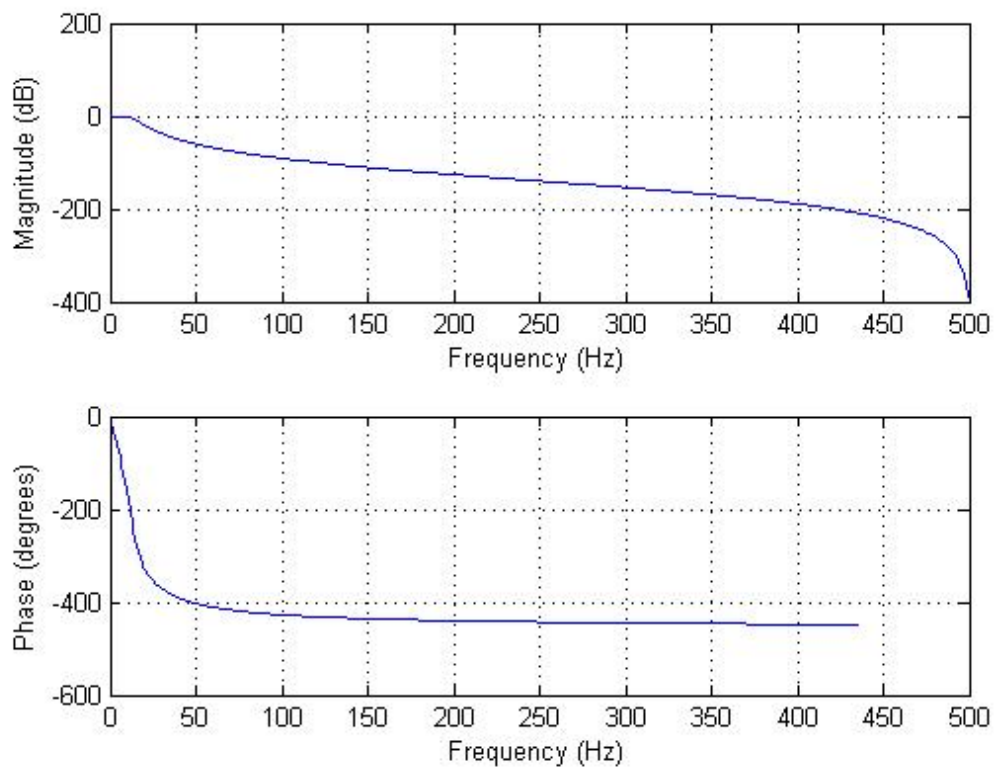
%
subplot(2,2,1);
plot(ffs,abs(fy1));
axis([0 0.2*fs 0 1024]);
title('100 Å 1 ÅμE×');
xlabel('frequency(/sec)');
ylabel('amplitude(v)');

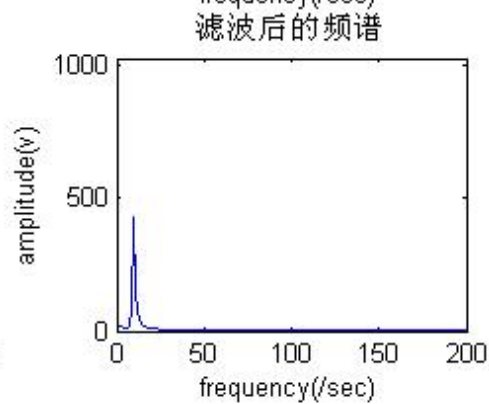
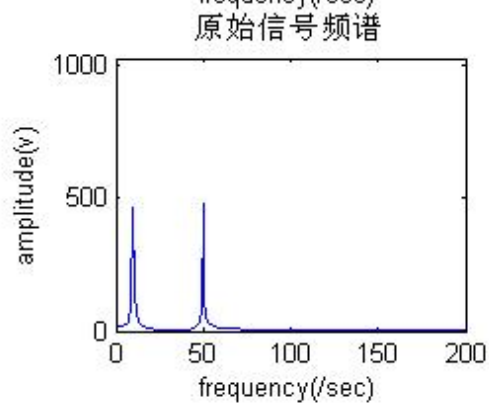
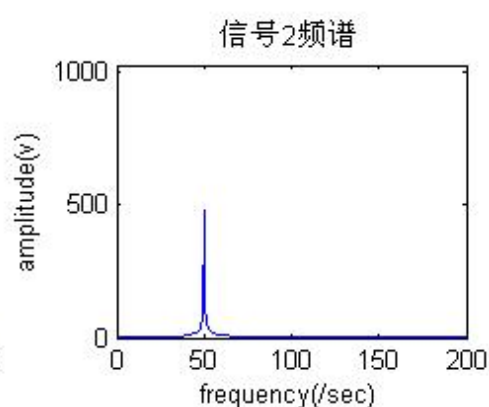
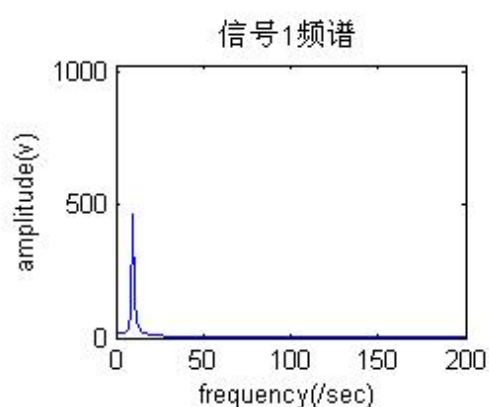
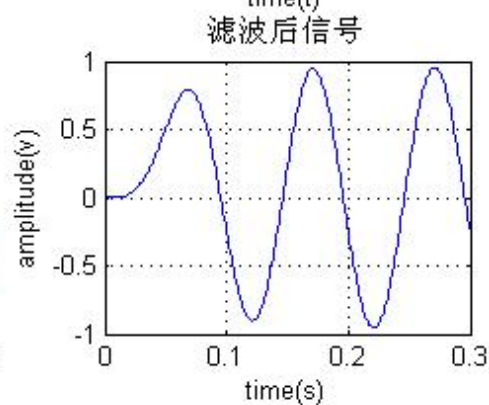
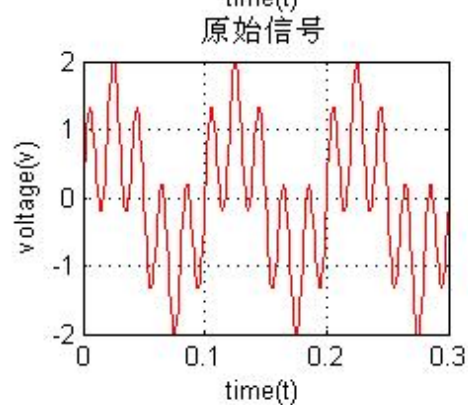
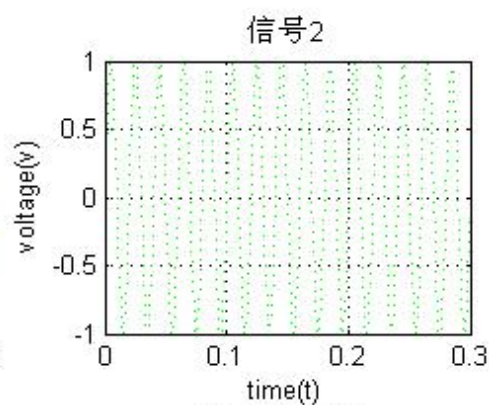
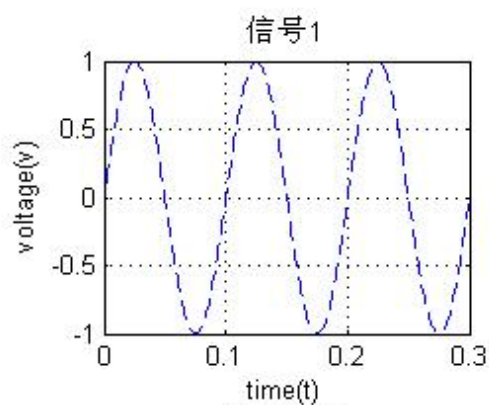
%
subplot(2,2,2);
plot(ffs,abs(fy2));
axis([0 0.2*fs 0 1024]);
title('100 Å 2 ÅμE×');
xlabel('frequency(/sec)');
ylabel('amplitude(v)');

%
subplot(2,2,3);
plot(ffs,abs(fz1));
axis([0 0.2*fs 0 1024]);
title('100 Å 1 ÅμE×');
xlabel('frequency(/sec)');
ylabel('amplitude(v)');

```

```
%
subplot(2, 2, 4);
plot(ffs, abs(fz2));
axis([0 0.2*fs 0 1024]);
title('Â² · °óμÄμ×');
xlabel('frequency(/sec)');
ylabel('amplitude(v)');
```





3. 设计程序，完成如下任务：

(1) 分别用Chebyshev I型、Chebyshev II型和椭圆滤波器设计低通滤波器，拟使该滤波器满足实验内容1的技术指标要求，显示滤波器的波特图。

(2) 分别用Butterworth、Chebyshev I型、Chebyshev II型和椭圆滤波器设计带通滤波器，拟从 $x(t)$ 中提取50Hz信号。要求显示滤波器的频率特性曲线，滤波器前后信号的波形、频谱。

四、实验要求

1. 输入本实验的示范程序，验证结果；
2. 自己编制程序完成实验内容3；
3. 总结计算机模拟滤波器的设计方法和IIR数字滤波器的设计方法。

五、思考题

1. 分析和总结IIR数字带通、带阻和高通滤波器的设计方法。
2. 分析模拟滤波器和IIR数字滤波器设计的相同点和不同点。
3. 思考滤波器在信号处理中的地位和作用。

实验八 Wav信号的波形分析与合成

一、实验目的

1. 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。
2. 尝试利用短时傅里叶变换分析非平稳信号的频谱变化。
3. 熟悉 MATLAB 环境中 wavread、wavrecord、wavplay、fft 和 ifft 等函数的应用。

二、实验原理

音频信号 (Audio) 是通过麦克风、A/D 等数据采集设备将声音转换而成的电信号,是声波频率、幅度变化的信息载体。声音的三个主观特性是音调、音强和音色。而声波的三个重要参数为频率 ω_0 、幅度 A_n 和相位 ψ_n , 这也就决定了音频信号的特征定义如下:

1. 基频与音调

基频指一个复杂声波中最低的一个频率 (其他频率叫谐波)。音调主要由声音的频率决定,同时也与声音强度有关。对一定强度的纯音,音调随频率的升降而升降;对一定频率的纯音、低频纯音的音调随声强增加而下降,高频纯音的音调却随强度增加而上升。

2. 谐波与音色

谐波是指周期函数或周期性的波形中不能用常数与原函数基频的正弦函数和余弦函数的线性组合表达的部分。 $n\omega_0$ 称为 ω_0 的 n 次谐波分量,也称为 $(n-1)$ 次泛音。音色是由混入基音的泛音所决定的,高次谐波越丰富,音色就越有明亮感和穿透力。不同的谐波具有不同的幅值 A_n 和相位偏移 ψ_n , 由此产生各种音色效果。

3. 幅度与音强

人耳对于声音细节的分辨只有在强度适中时才最灵敏。人的听觉响应与强度成对数关系。常用音量来描述音强,以分贝 ($\text{dB}=20\lg$) 为单位。

4. 音宽与频带

音频信号的频带宽度简称为音宽，它是描述组成复合信号的频率范围。

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、涉及的 MATLAB 函数

1. wavread

功能：读取 Microsoft 的 WAVE 文件。

基本调用格式：

`[y, Fs, nbits] = wavread(filename)` 载入以 filename 命名的 WAVE 文件。
y 为采样数据，Fs 为 WAVE 文件被采样时所用的采样频率，nbits 为用于描述每个采样点的编码数据位数（常用的有 8bits 和 16bits）。其中 filename 以单引号括起的字符串方式输入。

2. wavrecord

功能：应用 PC 机上音频输入设备录制声音。

基本调用格式：

`y = wavrecord(n, Fs)` 对某音频信号以 Fs Hz 采样率连续录入 n 个采样点的数据。其中 Fs 默认值为 11025Hz。

3. wavplay

功能：应用 PC 机上音频输出设备播放已录制的声音。

基本调用格式：

`Wavplay(y, Fs)` 以 Fs 播放存入矢量 y 中的音频信号。Fs 默认值仍为 11025Hz。

注：亦可参看 sound 函数。

4. fft

功能：离散傅立叶变换函数。

基本调用格式：

$Y = \text{fft}(X, n)$ 对矢量 X 进行 n 点离散傅立叶变换，结果存入矢量 Y 。计算时所用算法为快速傅立叶变换（FFT）。如果 X 为普通矩阵，那么返回为对此矩阵进行逐列的 FFT 的结果。

5. ifft

功能：离散傅立叶反变换。

基本调用格式：

$y = \text{ifft}(X, n)$ 返回对矢量 X 进行 n 点离散傅立叶反变换的结果。同样利用 FFT 算法进行具体计算，若 X 为矩阵，返回结果为逐列离散傅立叶反变换的 Y 矩阵。

四、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；

参考代码：

```
[y, fs]=wavread('ding.wav');  
%取其中的一个声道, 比如说, 右声道  
yr=y(:, 2);  
%截取前1024个点  
yr=yr(1:1024);
```

- 分别求取 2048、1024 点 FFT，观察产生的不同频谱；

参考代码：

%求取幅度谱并显示，首先是2048点的

```
YR2048=fft(yr,2048);
```

```
figure('numbertitle','off','name','2048点FFT');
```

```
plot(linspace(-pi,pi,2048),fftshift(abs(YR2048)));
```

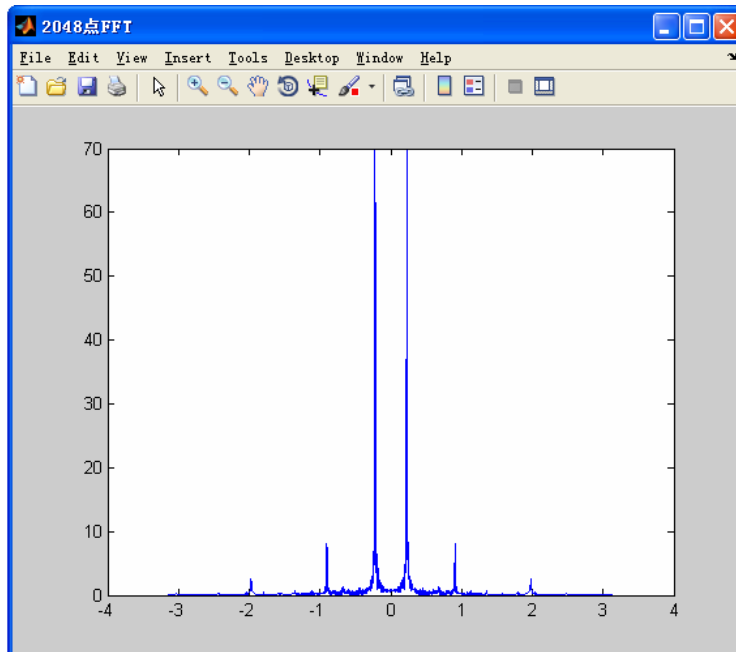
%求取1024点的

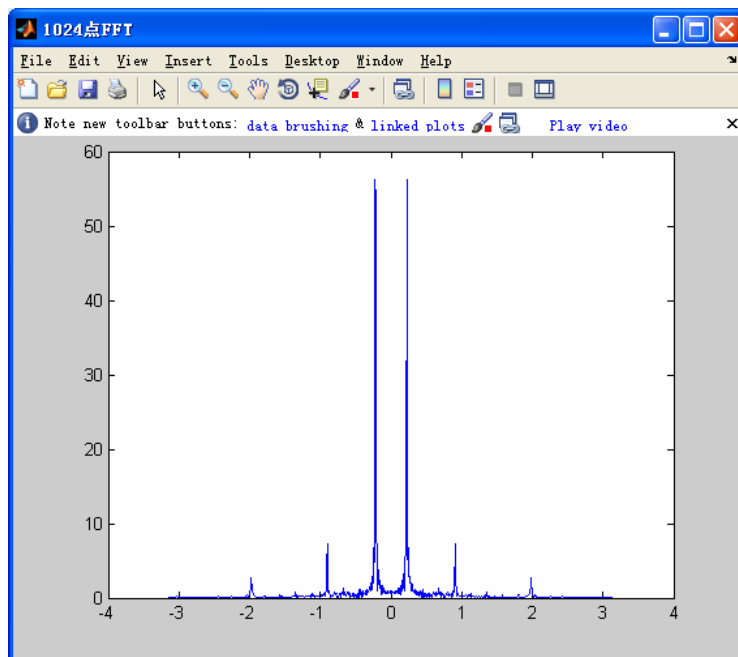
```
YR1024=fft(yr,1024);
```

```
figure('numbertitle','off','name','1024点FFT');
```

```
plot(linspace(-pi,pi,1024),fftshift(abs(YR1024)));
```

运行结果图：





- 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；

参考代码：

%反变换，结果应与原始信号相同

```
yr1024=real(ifft(YR1024));
```

```
figure('numbertitle','off','name',' 1024点fft的反变换');
```

```
subplot(2,1,1);
```

```
plot(yr1024)
```

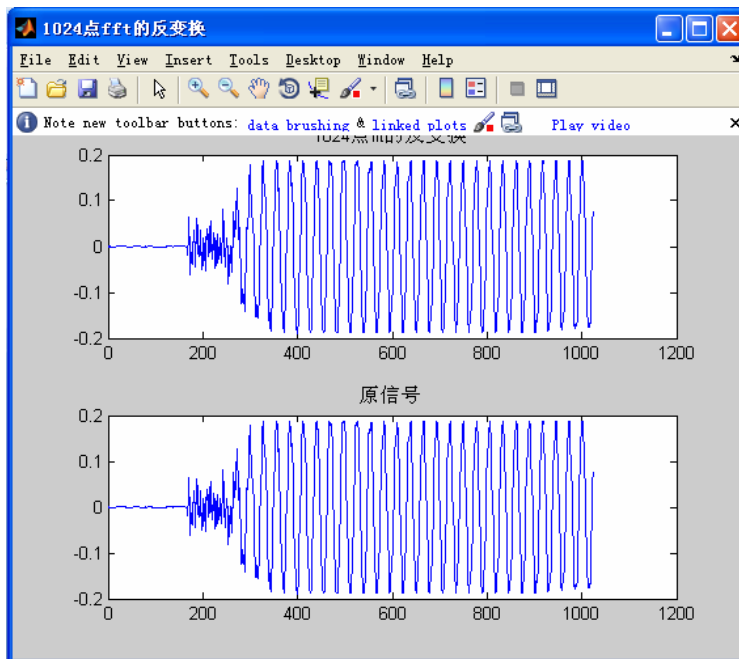
```
title(' 1024点fft的反变换')
```

```
subplot(2,1,2);
```

```
plot(yr);
```

```
title('原信号');
```

运行结果图：



- 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。

参考代码：

%重构

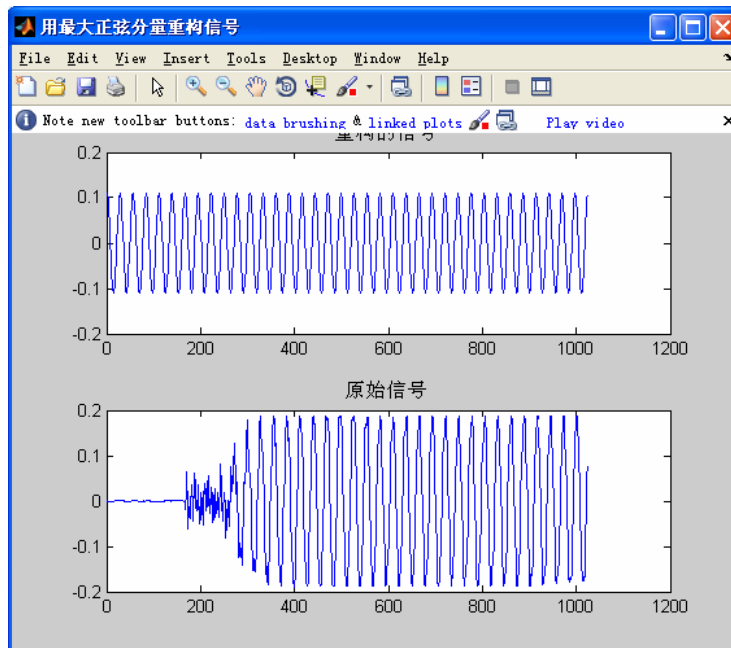
%寻找幅度最大的正弦分量

```
[maxpeak, peaki]=max(abs(YR1024(1:512)));
MAXSIN=zeros(1,1024);
MAXSIN(peaki)=maxpeak;
MAXSIN(1026-peaki)=maxpeak;
maxsin=ifft(MAXSIN);
figure('numbertitle','off','name','用最大正弦分量重构信号');
subplot(2,1,1)
plot(maxsin);
title('重构的信号');
subplot(2,1,2);
plot(yr1024);
title('原始信号');
%试听
wavplay(yr1024);
```



```
wavplay(maxsin);
```

运行结果图：



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；

参考代码：

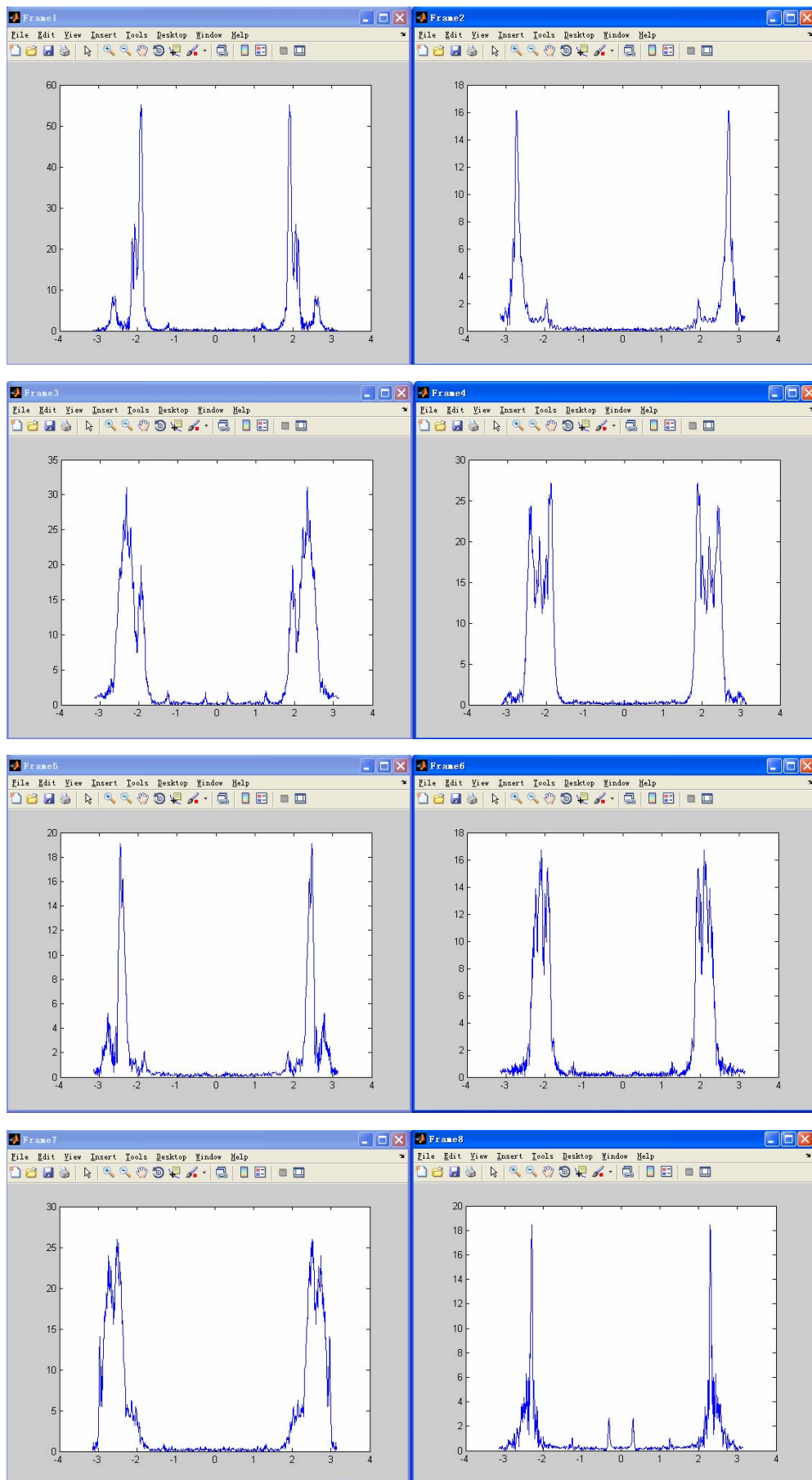
```
load chirp %y, Fs
```

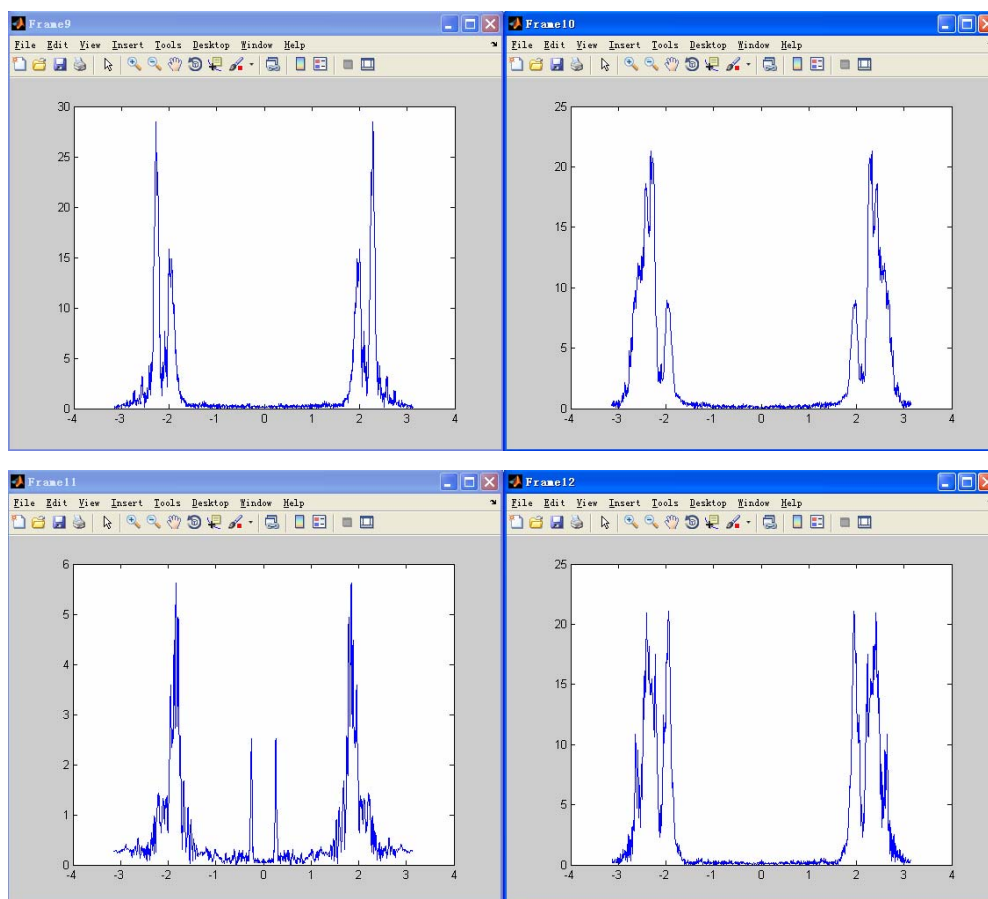
- 从信号中依次截取 1024 个点，利用上述方法求取其幅度谱，并显示出来，观察幅度谱随时间的变化情况。

参考代码：

```
len=length(y);  
for i=1:floor(len/1024)  
    seg=y((i-1)*1024+1:i*1024);  
    figure('numbertitle','off','name',['Frame',num2str(i)]);  
    plot(linspace(-pi,pi,1024),fftshift(abs(fft(seg))));  
end
```

运行结果图：





五、实验报告要求

简述实验目的及原理，列写完成思考题内容的详细步骤和相应的代码，并附上各步的结果图与分析。

六、思考题

分别录制男生和女生发元音“a”的声音，通过对音频文件的频谱分析对比两者的差异，并进行合理的解释。

实验九 电话拨号音的合成与识别

一、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。

二、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出一组频率进行组合，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。

表 1 ： DTMF 的组合功能

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

三、涉及的 MATLAB 函数

1. Set

功能：设置对象属性。

基本调用格式：set(H, 'PropertyName', PropertyValue, ...)

用属性值'PropertyValue'设置关于用参量H标志的对象（一个或多个）的属性名'PropertyName'（一个或多个）。H 可以为一句柄的向量。在这种情形下，命令 set 可以设置所有对象的属性值。

2. find

功能：找出矩阵X中非0项的坐标和取值。

基本调用格式：[row,col] = find(X, ...)

常与逻辑运算法一起使用，可进一步明确搜索数值的范围。

3. disp

功能：显示文本或数组。

基本调用格式：disp(X)

4. struct


功能：创建一个结构体数组。

基本调用格式：s = struct('field1', values1, 'field2', values2, ...)

给每一个域名赋值，赋值可以为矢量也可以为标量，但是矢量必须有相同的维数。


四、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。

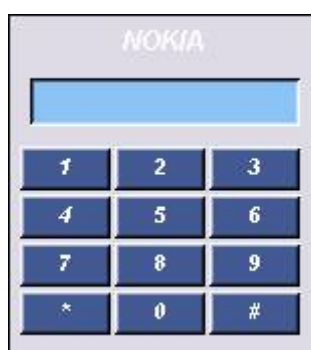
然后，为了更直观的反映对应的按键号码，可以设置一个编辑框，用于动态的显示拨号号码，模拟实际电话的拨号显示窗口。编辑框可用

 (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。

$d0 = \sin(0.7217 \cdot n) + \sin(1.0247 \cdot n)$ 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，计算出其幅度谱，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


```

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

五、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. 完成思考题要求内容；
4. 总结实验得出主要结论。

六、思考题

添加功能键，实现号码的预存储和来电识别。考虑不同位数号码的存储实现。

实验十 CDMA前向数据链路仿真

一、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。

二、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,所以重点介绍前向业务信道,其模型框图如图 9-1 所示:

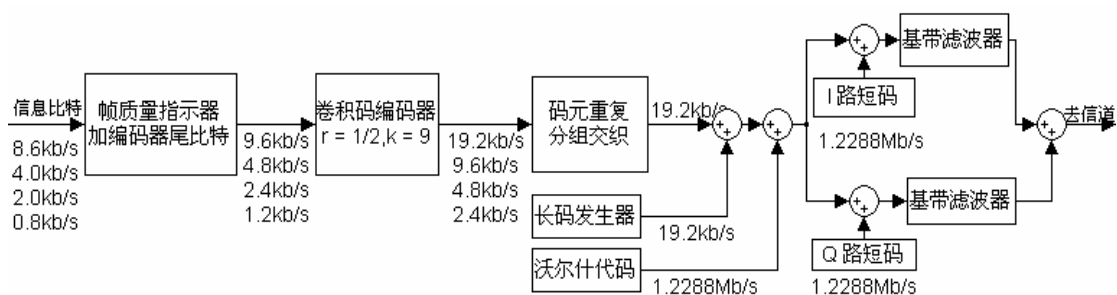


图 9-1 前向业务信道模型框图

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。

三、实验内容

1. 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. 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) 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：

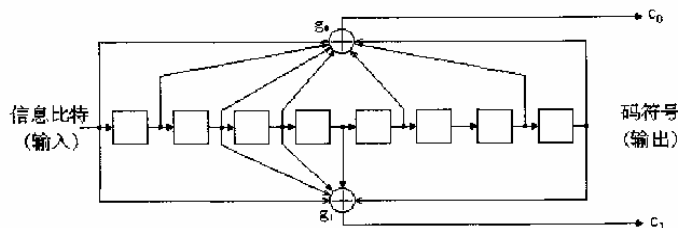


图 3-3 前向链路的卷积编码器: $r=1/2, K=9$

(4) 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) 可以找一下移动通信中有关章节，仔细研究，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。这些作为补充练习，不做要求。

五、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。

附录一 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 语言一样的运算符号外,还提供广泛的矩阵和向量运算符。利用其运算符号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，而不会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      4x3        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。

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

- 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：

<code>det(a)</code>	求矩阵 a 的行列式
<code>eig(a)</code>	求矩阵 a 的特征值
<code>inv(a)</code> 或 <code>a⁻¹</code>	求矩阵 a 的逆矩阵
<code>rank(a)</code>	求矩阵 a 的秩
<code>trace(a)</code>	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
<code>=</code>	等于	<code>eq</code>
<code>~ =</code>	不等于	<code>ne</code>
<code><</code>	小于	<code>lt</code>
<code>></code>	大于	<code>gt</code>
<code><=</code>	小于等于	<code>le</code>

>=	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，它的对应位置上的元素值应值为 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
```

§ 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。

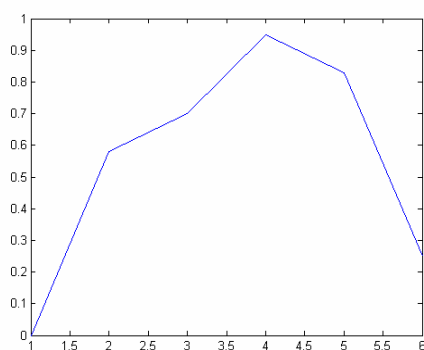


图 6-1

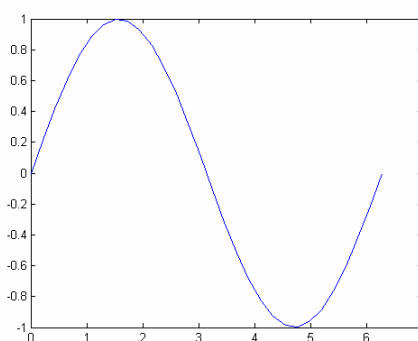


图 6-2

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
```

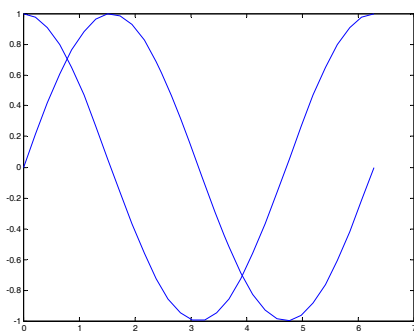


图 6-3

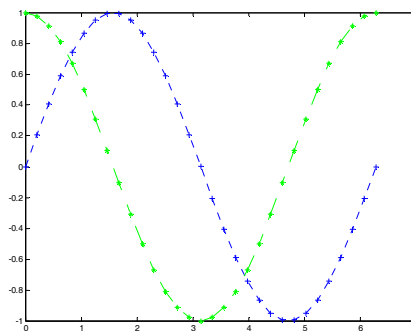


图 6-4

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:

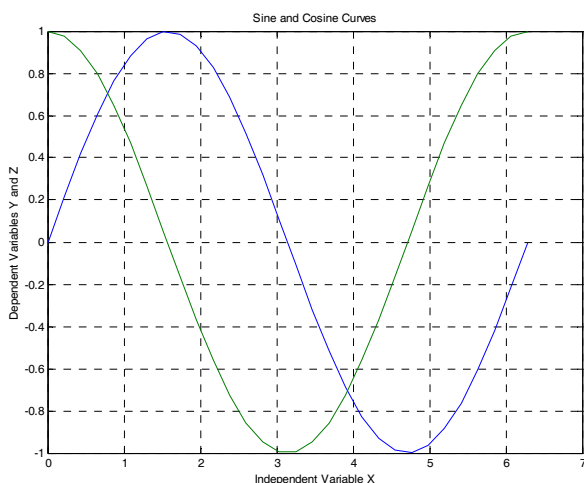


图 6-5

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 $\sin x$ 。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.

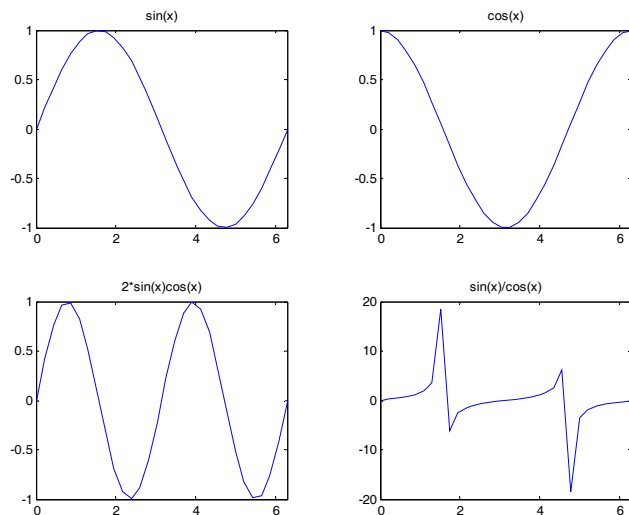


图 6-6

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。

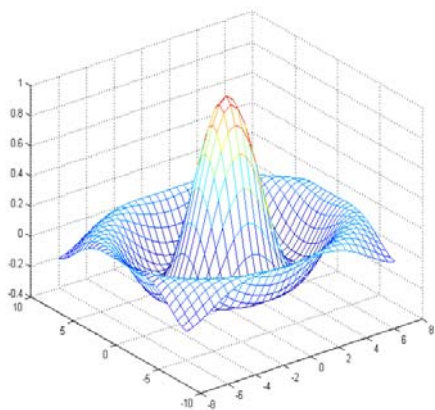


图 6-7

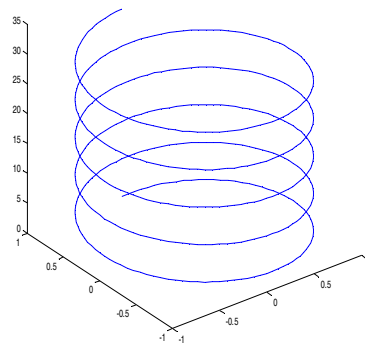


图 6-8

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°`。

附录二 学生实验报告模板

武汉大学教学实验报告

电子信息学院 _____ 专业 _____ 年 _____ 月 _____ 日

实验名称 _____ 指导教师 _____

姓名 _____ 年级 _____ 学号 _____ 成绩 _____

一、 预习部分

1. 实验目的
2. 实验基本原理
3. 主要仪器设备（含必要的元器件、工具）

二、 实验操作部分

1. 实验数据、表格及数据处理
2. 实验操作过程（可用图表示）
3. 实验结论

三、 实验效果分析（包括仪器设备等使用效果）

四、 教师评语

指导教师

年 月 日
