实验一　连续时间信号波形的产生

一、实验目的

熟悉 MATLAB 软件的使用，了解连续时间信号的特点，掌握连续时间信号的表示方法，并熟悉使用 MATLAB 产生信号和绘制信号波形。

二、实验原理

（一）信　号

信号一般表现为随时间变化的某种物理量。信号按照特性的不同，可以分为确定性信号和随机信号、连续信号和离散信号、周期信号和非周期信号等。

信号按照自变量的取值是否连续可分为连续时间信号和离散时间信号。连续时间信号指在信号讨论的时间范围内，任意时刻都可以给出确定的函数值，可以有有限个间断点。离散时间信号是指其时间自变量是离散的，只在某些不连续的规定时刻给出函数值，其他时刻没有定义。

常见的连续时间信号有：指数信号、正弦信号、抽样函数信号、单位阶跃信号、符号函数、单位冲激信号、矩形脉冲信号等。

1. 指数信号

$$f(t) = A\mathrm{e}^{at}$$

式中，A 和 a 均为实数，A 是指信号在 $t=0$ 时刻的幅度，a 可以取正值也可以取负值。若 $a>0$，则指数信号随时间增长而增长；若 $a<0$，则指数信号随时间增长而衰减；若 $a=0$，则指数信号成为直流信号。

2. 正弦信号

连续时间正弦信号表示为

$$f(t) = A\cos(\omega t + \varphi)$$

式中，A 是振幅，ω 是角频率，φ 是初相位。

3. 复指数信号

连续时间复指数信号表示为

$$f(t)=A\mathrm{e}^{st}$$

式中，$s=\sigma+\mathrm{j}\omega$，称为复频率，因此

$$\mathrm{e}^{st}=\mathrm{e}^{(\sigma+\mathrm{j}\omega)t}=\mathrm{e}^{\sigma t}\mathrm{e}^{\mathrm{j}\omega t}=\mathrm{e}^{\sigma t}(\cos\omega t+\mathrm{j}\sin\omega t)$$

上式表明，一个复指数信号可分解为实部、虚部两部分。实部、虚部分别为振幅按指数规律变化的正弦信号。若$\sigma<0$，复指数信号的实部、虚部为衰减正弦信号；若$\sigma>0$，复指数信号的实部、虚部为增幅正弦信号；若$\sigma=0$，则为虚指数信号$\mathrm{e}^{\mathrm{j}\omega t}$；若$\omega=0$，则复指数信号成为一般的实指数信号；若$\sigma=0$，$\omega=0$，复指数信号的实部、虚部均与时间无关，成为直流信号。

4. 抽样函数信号

$$\mathrm{Sa}(t)=\frac{\sin t}{t}$$

5. 单位阶跃信号

单位阶跃信号$u(t)$表达式为

$$u(t)=\begin{cases}1 & t>0\\ 0 & t<0\end{cases}$$

6. 单位冲激信号$\delta(t)$

$$\begin{cases}\delta(t)=0 & t\neq 0\\ \int_{-\infty}^{\infty}\delta(t)\mathrm{d}t=1\end{cases}$$

对信号进行时域分析，首先需要将信号随时间变化的规律用二维曲线表示出来。对于简单信号可以通过手工绘制其波形，但对于复杂的信号，手工绘制信号波形显得十分困难，且难以绘制精确的曲线。MATLAB 提供了大量函数用于生成基本信号。

（二）MATLAB 函数介绍

1. plot 函数

plot 函数用来绘制二维线性坐标曲线。plot 函数是最基本的绘图命令函数。

调用格式如下：

① plot(y)：当 y 为一向量时，以 y 的序号为 x 轴，按向量 y 的值绘制曲线。

② plot(x,y)：x、y 都为向量，以向量 x 作为 x 轴，向量 y 作为 y 轴，绘制出 x 对 y 的二维曲线。

③ plot(x,y1,'cs',x,y2,'cs',…)：以公共的向量 x 作为 x 轴，分别以向量 y1、y2、…的数据绘制多条曲线，每条曲线的外形可由相应的字符'cs'来指定。其中，c 表示曲线颜色的字符，s 表示线型格式，参见附录表 1 所示。

2. sym 函数

sym 函数是最常用的创建符号变量的函数。

调用格式为：

sym('变量名')

例如，x＝sym('x')，将建立一个符号变量 x，它代表字符 x。

当需要给多个符号变量赋值时，可以合并为一个命令：

syms 变量名列表

其中各个变量名之间用空格分隔。例如，syms x y z。

3. ezplot 函数

ezplot 函数可以在图形窗口绘制出符号函数的图形。

调用格式如下：

① ezplot(f)：对于符号函数 f＝f(x)，按照 x 的默认范围-2*pi<x<2*pi，在图形窗口中绘制出 f＝f(x)的图形。

② ezplot(f,[a,b])：在图形窗口中绘制出 f＝f(x)的图形，x 的范围为[a,b]。

4. subs 函数

subs 函数将符号变量替换成数字或其他符号。

调用格式如下：

① subs(s,new)：用新变量 new 替换 s 中的默认变量。其中，s 可以是符号表达式、符号代数方程或微分方程。

② subs(s,new,old)：用新变量 new 替换 s 中的指定变量 old。

5. subplot 函数

subplot 函数可以将图形窗口分割，则可以在同一个图形窗口中显示多个图形。

调用格式如下：

subplot(m,n,p)或 subplot(mnp)：将当前图形窗口中的坐标轴分割为 m 行、n 列的子坐标轴，选定第 p 个子坐标轴为当前坐标轴。

例如，subplot(232)表示 2 行 3 列共有 6 个子坐标轴，选择第 2 个坐标作为当前绘图坐标。

6. linspace 函数

利用 linspace 函数生成向量。

① a＝linspace(i,j)：表示生成有 100 个元素的线性分布行向量，其中：a（1）＝i，a（2）＝j。

② a＝linspace(i,j,n)：表示生成有 n 个元素的线性分布行向量，其中：a（1）＝i，a（2）＝j。

例如，linspace 生成向量：

a1＝linspace(1,11,6)

a1＝

1　　3　　5　　7　　9　　11

（三）连续信号的 MATLAB 实现

MATLAB 用两种方法来表示连续信号：一种是用数值计算的方法将连续信号离散化后，用数值表示信号；另一种是用符号运算的方法来表示信号。MATLAB 通过符号数学工具箱(Symbolic Math Toolbox)实现符号运算。

1. 指数信号

指数信号 $Ae^{\alpha t}$ 在 MATLAB 中可以用 exp 函数表示，调用格式为：

y = A*exp(α*t)

（1）衰减指数信号的产生

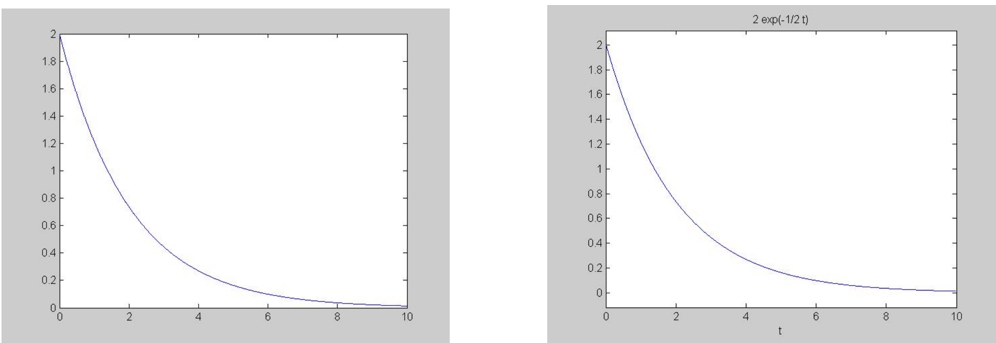
用两种方法实现的程序如下：

```
% 方法一用数值计算的方法产生衰减指数信号
    A＝2;
    a＝-0.5;
    t＝0:0.01:10;              % 信号时间向量
    y＝A*exp(a*t);             % 产生衰减指数信号
    plot(t,y);                 % 绘制信号图形
```

波形如图 1-1(a)所示。

```
    % 方法二用符号运算的方法来表示衰减指数信号
    syms t;                    % 定义 t 为符号变量
    y＝2*exp(-0.5*t);
    ezplot(y,[0,10]);          % 绘制符号函数图形
```

波形如图 1-1(b)所示。



(a)　　　　　　(b)

图 1-1　衰减指数信号

（2）a 值不同情况下的指数信号波形

用三种方法实现的程序如下：

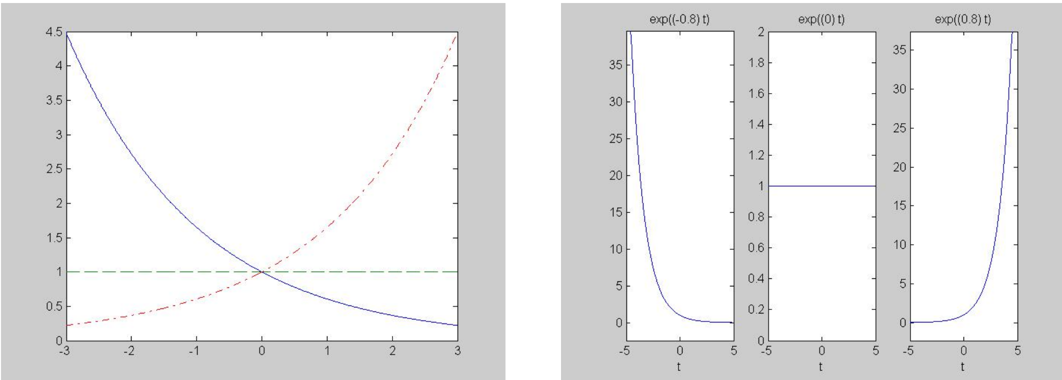
```
    % 方法一用数值计算的方法产生指数信号
```

```
t=-3:0.01:3;
f1=exp(-0.5*t);
f2=exp(0*t);
f3=exp(0.5*t);
plot(t,f1,t,f2, '--',t,f3,'-.');
```

波形如图 1-2(a)所示。

```
% 方法二用符号运算的方法来表示指数信号
syms a t;
y=exp(a*t);
y1=subs(y,'a','-0.8');              % 将变量 a 替换成-0.8
subplot(131);
ezplot(y1,[-5,5]);
y2=subs(y,'a','0');
subplot(132);
ezplot(y2,[-5,5]);
y3=subs(y,'a','0.8');
subplot(133);
ezplot(y3,[-5,5]);
```

波形如图 1-2(b)所示。



(a) (b)

图 1-2 指数信号

也可以先创建函数 rexp 如下：

```
function rexp (a,t1,t2)
t=t1:0.01:t2;
f=exp(a*t);
plot(t,f);
```

调用函数 sexp 可绘出指数信号在不同情况下的时域波形，例如：

```
rexp (-0.5,-3,3)
rexp (0,-3,3)
```

```
rexp (0.5,-3,3)
```

波形如图 1-3 所示。

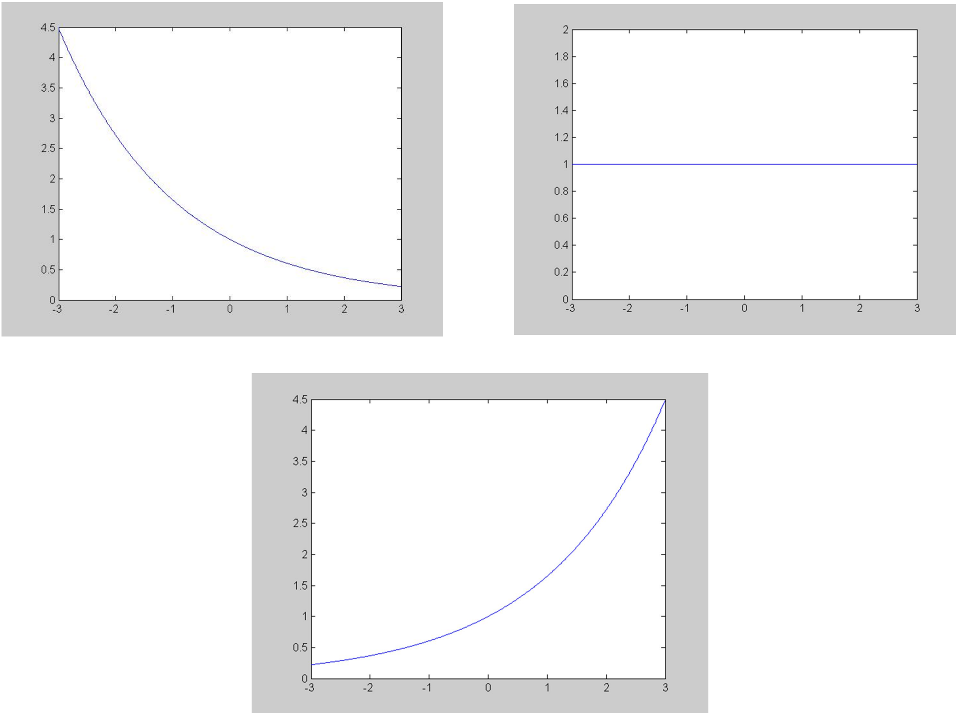


图 1-3　指数信号

2. 正弦信号

正弦信号 $A\cos(\omega_0 t+\varphi)$ 和 $A\sin(\omega_0 t+\varphi)$ 分别用 MATLAB 函数 cos 和 sin 表示，调用格式为：

```
A*cos(w0*t+phi)
A*sin(w0*t+phi)
```

（1）正弦信号的产生

```
A＝1;
w0＝2*pi*0.5;
phi＝pi/6;
t＝0:0.01:5;
y＝A*sin(w0*t+phi);
plot(t,y);
grid on;                      % 设置网格线
title('y(t)＝sin(πt+π/6)');   % 设置图形标题为 y(t)＝sin(πt+π/6)
xlabel('t');                  % 设置横坐标标题为 t
axis([0,5,-1.1,1.1]);         % 设置坐标轴范围，x 轴最小值 0，最大值 10；y 轴
                              % 最小值－0.1，最大值 2.1。通过 axis 命令可扩大水平、
```

% 垂直方向坐标边界。

波形如图 1-4 所示。

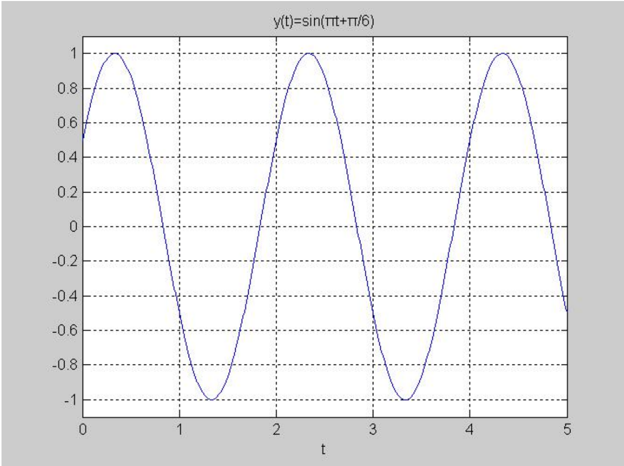


图 1-4　正弦信号

（2）指数衰减正弦信号的产生

前面已经看到，在所有的 MATLAB 信号产生命令中，用代表单位幅度信号的向量乘以标量 A 来得到所要求的幅度，例如，A*sin(w0*t+phi)，这种运算用星号“*”表示。下面考虑需要两个向量对应“元素-元素相乘”来产生信号的情况，这种运算用一个点后接星号“.*”表示。

指数衰减正弦信号 $Ae^{-at}\sin(\omega_0 t+\varphi)$ 的命令是：

A*exp(-a*t).*sin(w0*t+phi)

例 1-1　用 MATLAB 产生指数衰减正弦信号 $y=2e^{-t}\cos\left(2\pi t+\dfrac{\pi}{3}\right)$，$t$ 的范围是在－1～5 s，以 0.01 s 递增。

解　程序如下：

```
% 指数衰减正弦信号
A＝2;
a＝-1;
t＝-1:0.01:5;
y＝A*exp(a*t).*cos(2*pi*t+pi/3);
plot(t,y);
axis([-1 5 -4 4])
```

指数衰减正弦信号如图 1-5 所示。

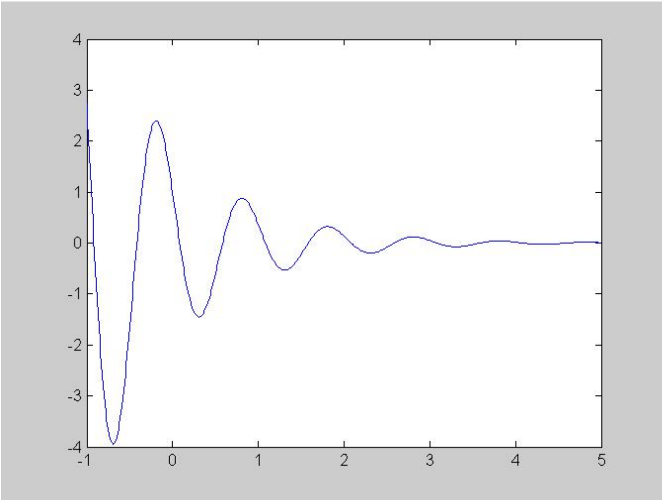


图 1-5　指数衰减正弦信号

3. 复指数信号

复指数信号 Ae^{st} 是复频域分析的基本信号，其中 $s=\sigma+\mathrm{j}\omega$ 为复数，即复频率。由欧拉公式得

$$Ae^{st}=Ae^{\sigma t}e^{\mathrm{j}\omega t}=Ae^{\sigma t}\cos(\omega t)+\mathrm{j}Ae^{\sigma t}\sin(\omega t)$$

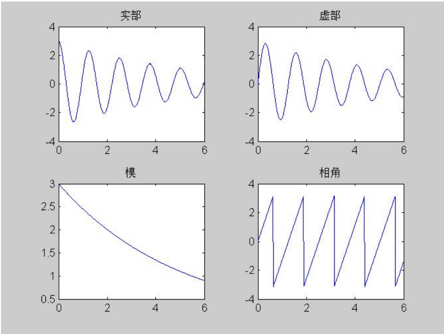
所以复指数信号是时间 t 的复函数，需要两个实信号来表示，即用模和相角或实部和虚部来表示复指数信号随时间变化的规律。

例 1-2　用 MATLAB 产生复指数信号 f(t)＝3*exp((-0.2+i*5)*t)的程序。

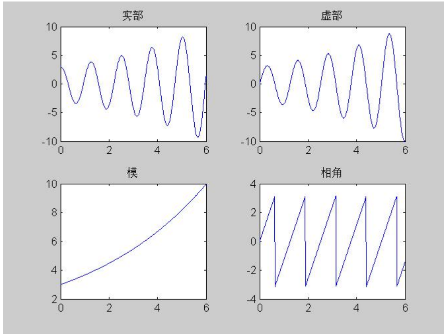
解　程序如下：

```
% 绘制复指数信号波形程序
t＝0:0.01:6;
y＝3*exp((-0.2+i*5)*t);            %产生复指数信号 y
yr＝real(y);                       %复指数信号 y 取实部
yi＝imag(y);                       %复指数信号 y 取虚部
ya＝abs(y);                        %求复指数信号 y 的模
yg＝angle(y);                      %求复指数信号 y 的相角
subplot(221);plot(t,yr);title('实部');
subplot(222);plot(t,yi);title('虚部');
subplot(223);plot(t,ya);title('模');
subplot(224);plot(t,yg);title('相角');
```

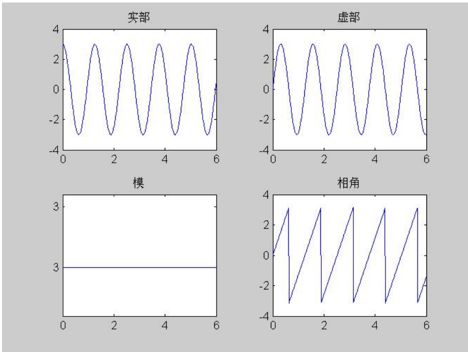
复指数信号 f(t)＝3*exp((-0.2+i*5)*t)的波形如图 1-6(a)所示。同理，可以绘出复指数信号 f(t)＝3*exp((0.2+i*5)*t)的波形如图 1-6(b)所示。



(a)



(b)



(c)

图 1-6　复指数信号

```
function fuexp (a,w,t1,t2,A)
% 绘制复指数信号波形程序
%  a 是复频率实部
%  w 是复频率虚部
%  t1,t2 分别是起始时间和终止时间
%  A 是复指数信号幅值
t＝t1:0.01:t2;
y＝A*exp((a+i*w)*t);
yr＝real(y);
yi＝imag(y);
ya＝abs(y);
yg＝angle(y);
subplot(221);
plot(t,yr);
title('实部');
subplot(222);
plot(t,yi);
title('虚部');
```

```
subplot(223);
plot(t,ya);
title('模');
subplot(224);
plot(t,yg);
title('相角');
```

调用函数 fuexp 可绘出复指数信号在不同情况下的时域波形，例如，绘制复指数信号 f(t)＝3*exp(i*5*t)的波形，对应的 MATLAB 程序为：

```
fuexp(0,5,0,6,3)
```

波形如图 1-6(c)所示。

4. 抽样函数

抽样函数信号 Sa(t)在 MATLAB 中用 sinc 函数表示，定义为

$$\operatorname{sin c}(t)=\frac{\sin(\pi t)}{\pi t}$$

调用格式为：

```
y＝sinc(t)
```

程序如下：

```
%Sample function
t＝-3*pi:pi/100:3*pi;
y＝sinc(t/pi);
plot(t,y);
grid on;
axis([-3*pi,3*pi,-0.4,1.1]);
```

波形如图 1-7 所示。

或用如下的程序实现：

```
x＝linspace(-5,5);                %利用 linspace 函数生成向量
y＝sinc(x);
plot(x,y);
```

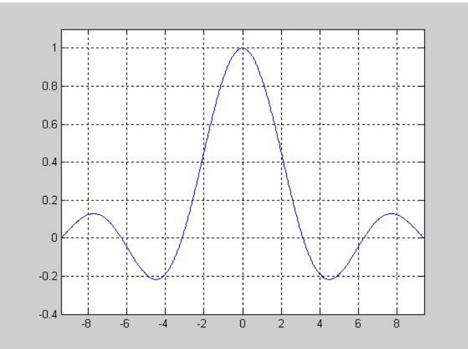


图 1-7　抽样函数信号

也可以通过 sinc(t)＝sin(πt)/(πt)得到。注意，一个点后接除号“./”表示逐点相除运算。

5. 阶跃信号

单位阶跃信号定义为

$$\varepsilon(t)=\begin{cases}1 & t>0\\ 0 & t<0\end{cases}$$

表示单位阶跃信号的一种方法是借助关系运算符。在 MATLAB 中，一个关系运算符对两项进行比较，若结果为真，则返回逻辑真“1”；若结果为假，则返回逻辑假“0”。

MATLAB 有六种关系运算符：

< 小于	> 大于
<= 小于等于	>= 大于等于
== 等于	~= 不等于

产生阶跃信号 $\varepsilon(t)$ 的程序为：

```
% 用关系运算符产生阶跃信号
t＝-2:0.02:6;
y＝(t>0);
plot(t,y);
axis([-2,6,0,1.2]);
title('单位阶跃函数');
```

波形如图 1-8 所示。

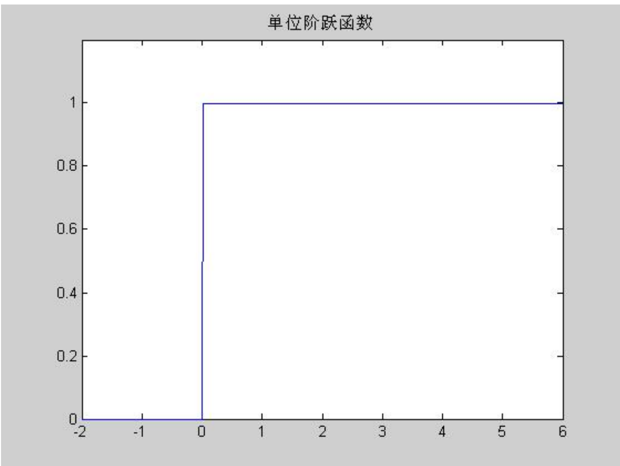


图 1-8　单位阶跃函数

另一种有效方法是根据单位阶跃信号的定义，调用 MATLAB 的 Symbolic Math Toolbox 中的单位阶跃函数 Heaviside(t)。因为函数 ezplot 只能画出既存在于 Symbolic Math 工具箱中，又存在于总 MATLAB 工具箱中的那些函数，而 Heaviside 函数仅存在于 Symbolic Math 工具箱中，所以需要在 MATLAB 的当前工作目录下要创建 Heaviside.m 的 M 文件。

创建 Heaviside(t)函数方法如下：

```
function f=Heaviside(t)
f=(t>0);
```

且以 Heaviside.m 文件名保存。MATLAB 表达式 f=(t>0)的含义是 t>0 时 f=1，而 t≤0 时 f=0。

在当前工作目录下创建 Heaviside 函数并保存后，就可以调用该函数。运行下列程序：

```
%利用符号运算的方法来表示单位阶跃函数
syms t;
f=sym('Heaviside(t)');
ezplot(f,[-2,6]);
title('单位阶跃函数');
```

绘制的波形如图 1-9 所示。

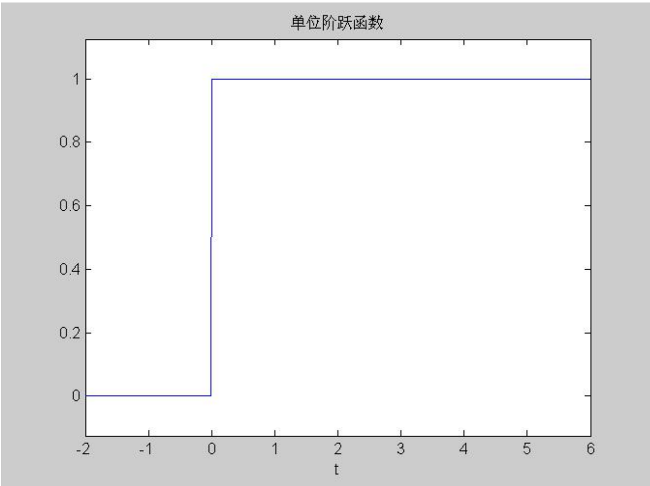


图 1-9　单位阶跃函数

6. 符号函数

符号函数 Sgn(t)的定义为

$$\mathrm{Sgn}(t)=\begin{cases}1 & t>0\\ -1 & t<0\end{cases}$$

符号函数在 MATLAB 中用 sign 函数表示。调用格式为：

```
y=sign(t)
```

程序如下：

```
% 符号函数
t=-3:0.02:3;
y=sign(t);
```

```
plot(t,y);
axis([-3,3,-1.2,1.2]);
title('符号函数');
```

绘制的波形如图 1-10 所示。

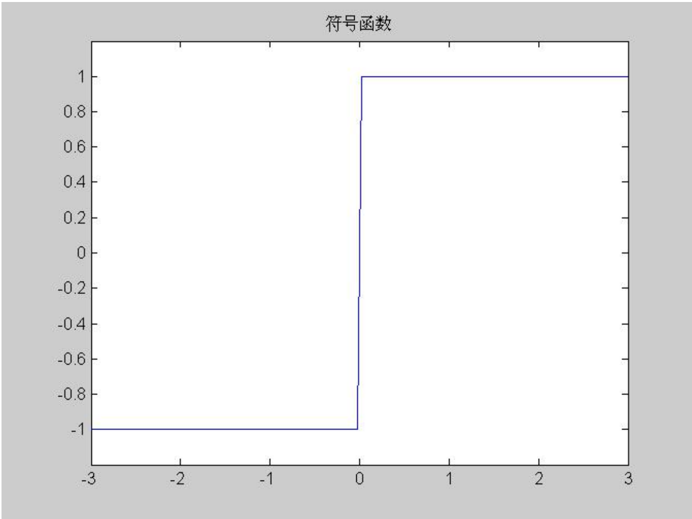


图 1-10 符号函数

7. 矩形脉冲信号

矩形脉冲信号在 MATLAB 中用 rectpuls 函数表示，调用格式如下：

① y＝rectpuls(t)

② y＝rectpuls(t,width)

格式①产生指定宽度为 1、单位高度（高度为 1）的非周期矩形脉冲信号。

格式②产生指定宽度为 width、单位高度的非周期矩形脉冲信号。

例如，产生宽度为 1、高度为 1 的非周期矩形脉冲信号和宽度为 3、高度为 1.5 的非周期矩形脉冲信号，程序如下：

```
% 矩形脉冲信号
    t＝-3:0.02:3;
    y1＝rectpuls(t);
    y2＝1.5*rectpuls(t,3);
    plot(t,y1,t,y2,'--');
    title('矩形脉冲信号');
    axis([-3,3,-0.2,1.6]);
```

绘制的波形如图 1-11 所示。

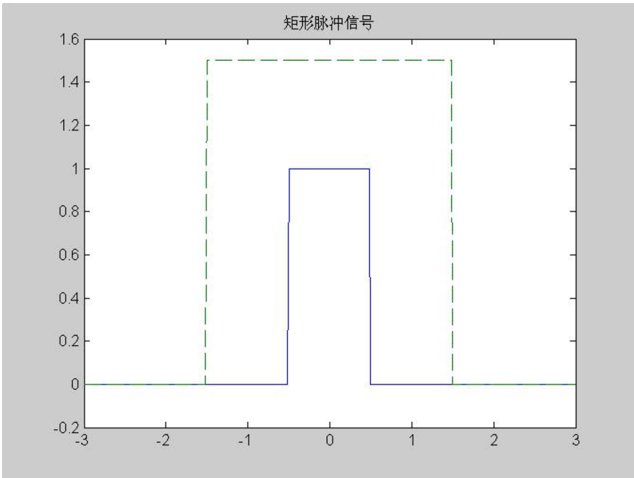


图 1-11　矩形脉冲信号

8. 三角形脉冲信号

矩形脉冲信号在 MATLAB 中用 tripuls 函数表示，调用格式如下：

① y＝tripuls(t)

② y＝tripuls(t,width)

③ y＝tripuls(t,width,skew)

格式①产生指定宽度为 1、单位高度（高度为 1）的非周期对称三角形脉冲信号，三角波的中心位置位于 t＝0 处。

格式②产生指定宽度为 width、单位高度的非周期对称三角波。

格式③产生指定宽度为 width、单位高度的非对称三角波，由 skew 指定三角形的斜率，其中 skew 的取值范围为-1 到+1 之间，当 skew 等于 0 时产生对称三角波。

例如，产生宽度为 1、高度为 1 的非周期对称三角波和宽度为 4、高度为 1.5 的非周期对称三角波和宽度为 4、高度为 1、skew 为-0.5 的非周期三角波，程序如下：

```
% 三角形脉冲信号
t＝-3:0.02:3;
y1＝tripuls(t);                % 产生宽度为 1、高度为 1 的非周期对称三角波
subplot(311);plot(t,y1);
title('三角形脉冲信号');
y2＝1.5*tripuls(t,4);          % 产生宽度为 4、高度为 1.5 的非周期对称三角波
subplot(312); plot(t,y2);
y3＝tripuls(t,4, -0.5);        % 产生宽度为 4、高度为 1、skew 为-0.5 的非周期三角波
subplot(313); plot(t,y3);
axis([-2.2,2.2,-0.2,1.2]);
```

绘制的波形如图 1-12 所示。

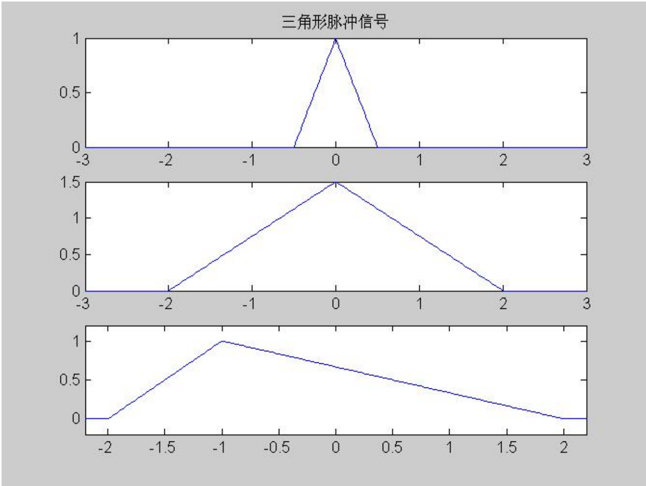


图 1-12　三角形脉冲信号

9. 周期矩形脉冲信号

square 函数用于产生周期矩形脉冲信号，调用格式如下：

① square(t)

② square(w0*t,duty)

格式①产生周期为 2π、峰值为±1 的周期方波。

格式②产生基本频率为 w0(周期 T＝2π/w0)、占空比为 duty 的周期矩形脉冲。占空比指一个周期内信号为正的部分所占的比例。

例如，产生峰值为±1、周期为 2π的方波和峰值为±1、基本频率为 0.5 Hz 的方波和峰值为±1、基本频率为 0.5 Hz、占空比为 80% 的周期矩形脉冲，程序如下:

```
% 周期矩形脉冲信号
t＝0:0.01:10;
y1＝square(t);                % 产生周期为 2π的方波
subplot(311);plot(t,y1);
axis([0,10,-1.5,1.5]);
title('周期方波信号');
w0＝2*pi*0.5;
y2＝square(w0*t);             % 产生周期为 2 的方波
subplot(312);plot(t,y2);
axis([0,10,-1.5,1.5]);
y3＝square(w0*t,80);          % 产生周期为 2、占空比为 80%的周期矩形脉冲
subplot(313);plot(t,y3);
axis([0,10,-1.5,1.5]);
title('周期矩形脉冲信号');
```

绘制的波形如图 1-13 所示。

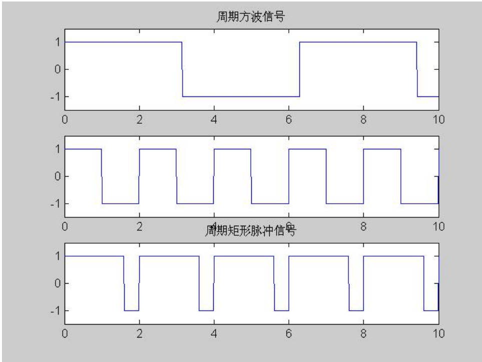


图 1-13　周期矩形脉冲信号

10. 周期锯齿波信号

sawtooth 函数用于产生周期齿波信号或三角波，调用格式如下：

① sawtooth(t)

② sawtooth(t,width)

③ sawtooth(w0*t,width)

格式①产生周期为 2π、峰值为±1 的周期锯齿波。

格式②产生周期为 2π、峰值为±1 的周期锯齿波，width 是值为 0 到 1 之间的常数，用于指定在一个周期（0 到 2π）内，三角波最大值出现的位置。当 width 等于 0.5 时产生对称三角波；当 width 等于 1 时，sawtooth(t,1)与 sawtooth(t)相同。

格式③产生指定周期(周期 T＝2π/w0)、峰值为±1 的周期锯齿波，width 是值为 0 到 1 之间的常数，用于指定在一个周期内，三角波最大值出现的位置。

例如，产生峰值为±1、周期为 2π的锯齿波和峰值为±1、基本频率为 0.5Hz 的锯齿波和峰值为±1、基本频率为 0.5Hz、width 为 0.5 的周期矩形脉冲，程序如下:

```
% 周期锯齿波信号
t＝0:0.001:10;
w0＝2*pi*0.5;
y1＝sawtooth(t);                    % 产生周期为 2π的锯齿波
subplot(311);plot(t,y1);
axis([0,10,-1.5,1.5]);
title('周期锯齿波信号');
y2＝sawtooth(w0*t,1);               % 产生周期为 2 的锯齿波
subplot(312);plot(t,y2);
axis([0,10,-1.5,1.5]);
y3＝sawtooth(w0*t,0.5);             % 产生周期为 2 的对称周期锯齿波（width 为 0.5）
subplot(313);plot(t,y3);
axis([0,10,-1.5,1.5]);
```

绘制的波形如图 1-14 所示。

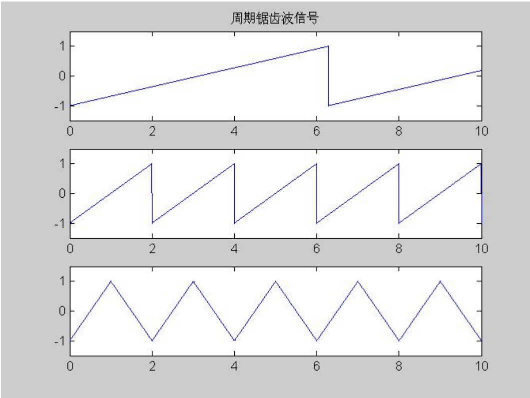


图 1-14　周期锯齿波信号

三、实验内容

1.1　试用 MATLAB 产生以下信号并绘出波形。

① $3\sin(0.2\pi t)$　　　② $5e^{-0.3t}\cos(\frac{3t}{4})$

③ $\frac{\sin(t)}{t}$　　　④ $1-2\frac{|t|}{a}$

⑤ $\sqrt{2t}$

1.2　试用 MATLAB 绘制如下矩形信号(门信号)的时域波形。

$$g(t)=\begin{cases}1 & -1<t<1\\ 0 & \text{其他}\end{cases}$$

1.3　试用 MATLAB 绘制单位冲激信号 $\delta(t)$ 的时域近似波形。$\delta(t)$ 用宽度为 τ、幅度为 $\frac{1}{\tau}$ 当 $\tau\to\infty$ 时的矩形信号来近似(取 $\tau=0.2$ 和 $\tau=0.1$)。

1.4　绘制出（1）$f_1(t)=5\cos(12\pi t)+6\cos(18\pi t)$，（2）$f_2(t)=\cos^2(2\pi t)$ 的波形，并求出信号的周期。

1.5　产生一个峰值为 ±1、基本频率为 1Hz 周期锯齿波及 width＝0.5 时的锯齿波。

四、实验要求

1. 熟悉 MATLAB 软件的常用命令的使用。

2. 在 MATLAB 中输入程序，并将实验结果存入指定存储区域。

3. 规范化地书写实验报告。在实验报告中写出自编程序，并给出实验结果，附上相应的信号波形曲线。

实验四　连续时间系统的时域分析

一、实验目的

通过使用 MATLAB 软件对连续时间线性非时变系统的时域特性进行仿真分析，熟悉 LTI 系统在典型激励下的响应及特征，熟悉相应 MATLAB 函数的调用格式和作用，熟悉并掌握用 MATLAB 函数求解冲激响应、阶跃响应、零输入响应、零状态响应及全响应的方法。

二、实验原理

（一）连续时间系统的时域分析方法

连续时间线性非时变系统(LTI)的输入 $f(t)$与输出 $y(t)$可以用线性常系数微分方程来描述

$$\begin{aligned}&a_n y^{(n)}(t)+a_{n-1}y^{(n-1)}(t)+\cdots+a_1 y'(t)+a_0 y(t)\\ =\ &b_m f^{(m)}(t)+\cdots+b_1 f'(t)+b_0 f(t)\end{aligned}$$

如果已知系统的输入信号 $f(\mathrm{t})$及系统的初始条件为 $y(0_-)$，$y'(0_-)$，$y''(0_-)$，…，$y^{(n-1)}(0_-)$，就可以利用解析方法求出系统的响应。

线性系统的全响应由零输入响应分量和零状态响应分量组成。零输入响应是指当输入为零时仅由 $t=0$ 的初始条件产生的系统响应，零状态响应是当初始条件（在 $t=0$ ）假定为零时仅由 $t\geqslant 0$ 时的输入产生的系统响应分量。

零输入响应(单极点时)为

$$y_{\mathrm{x}}(t)=c_1\mathrm{e}^{\lambda_1 t}+c_2\mathrm{e}^{\lambda_2 t}+\cdots+c_n\mathrm{e}^{\lambda_n t}=\sum_{k=1}^{n}c_k\mathrm{e}^{\lambda_k t}$$

式中，c_1、c_2、…、c_n 为任意待定常数，由初始条件确定。

零状态响应为

$$y_{\mathrm{f}}(t)=\int_{-\infty}^{\infty}f(\tau)h(t-\tau)\mathrm{d}\tau$$

此式是对任意输入 $f(t)$，用单位冲激响应 $h(t)$ 形式表示的零状态响应 $y_{\mathrm{f}}(t)$ 的公式。已知 $h(t)$ 就可确定任意输入 $f(t)$ 的零状态响应 $y_{\mathrm{f}}(t)$，即系统对任意输入的响应都可以用单位冲激响应确定。

系统总响应为：

$$
\begin{aligned}
y(t) &= y_{\mathrm{x}}(t) + y_{\mathrm{f}}(t) \\
&= \sum_{j=1}^{n} c_j \mathrm{e}^{\lambda_j t} + \int_{-\infty}^{\infty} f(\tau) h(t-\tau) \mathrm{d}\tau
\end{aligned}
$$

对于高阶系统，手工计算非常繁琐。MATLAB 的计算功能能比较容易地确定系统的各种响应，如冲激响应、阶跃响应、零输入响应、零状态响应和全响应等。

（二）MATLAB 函数介绍

1. impulse 函数

impulse 函数用来求线性时不变(LTI)系统模型的冲激响应。

在调用 impulse 函数时，需要用向量来对连续系统进行分析。

设描述连续系统的微分方程为：

$$a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \cdots + a_1 y'(t) + a_0 y(t) = b_m f^{(m)}(t) + \cdots + b_1 f'(t) + b_0 f(t)$$

则可用向量 a 和 b 来表示该系统，即：

a＝[a_n，a_{n-1}，……a_1，a_0]

b＝[b_m，b_{m-1}，……b_1，b_0]

注意，向量 a 和 b 的元素一定要以微分方程中时间求导的降幂次序来排列，且缺项要用 0 来补齐。例如，对微分方程 $y''(t)+3y'(t)+2y(t)=f''(t)+f(t)$，则表示该系统的对应向量应为 a＝[1 3 2]，b＝[1 0 1]。

函数 impulse 将绘出由向量 a 和 b 表示的连续系统在指定时间范围内的冲激响应 h(t)的时域波形图，并能求出指定时间范围内冲激响应的数值解。

impulse 函数的调用格式为：

① impulse(b,a)：该调用格式以默认方式绘出向量 a 和 b 定义的连续时间系统的冲激响应的时域波形。

② impulse(b,a,t)：绘出系统在 0～t 时间范围内冲激响应的时域波形。

③ impulse(b,a,t1:p:t2)：绘出在 t1～t2 时间范围内，且以时间间隔 p 均匀取样的冲激响应波形。

④ y＝impulse(b,a,t1:p:t2)：不绘出波形，而是求出系统冲激响应的数值解。

求解系统的冲激响应 h(t)也可采用如下调用格式：

① impulse(sys)：绘制 LTI 系统模型 sys 的冲激响应(由 tf、zpk 或 ss 创建)。

② impulse(sys,t)：绘出系统在 0～t 时间范围内冲激响应的时域波形。

③ h＝impulse(sys,t)：求出系统冲激响应的数值解，不绘出波形。

sys 是系统函数对象，可以由 MATLAB 的 tf 函数根据系统微分方程的系数获得。

tf 函数的调用格式为：

sys＝tf (b,a) 其中输入参量 b 为上面描述系统的微分方程右边多项式系数构成的行向量，a 为微分方程左边多项式系数构成的行向量，输出参量 sys 为返回 MATLAB 定义的系统函数对象。

例如，系统 $y''(t)+3y'(t)+2y(t)=-f'(t)+2f(t)$，由 tf 函数生成其系统函数对象 sys 的命令为：

a＝[1 3 2]; b＝[-1 2];sys＝tf (b,a)

结果为：

Transfer function:

```
      -s+2
   -----------
   s^2+3s+2
```

2. step 函数

step 函数用来求 LTI 系统模型的阶跃响应。

调用格式为：

① step(b,a)：以默认方式绘出向量 a 和 b 定义的连续时间系统的阶跃响应的时域波形。

② step(b,a,t)：绘出系统在 0～t 时间范围内阶跃响应的时域波形。

③ step(b,a,t1:p:t2)：绘出在 t1~t2 时间范围内，且以时间间隔 p 均匀取样的阶跃响应波形。

④ y＝step(b,a,t1:p:t2)：返回系统阶跃响应的数值解，不绘出波形。

也可用如下调用格式：

① step(sys)

② step(sys,t)

③ g＝step(sys,t)

3. lsim 函数

lsim 函数可以仿真线性时不变(LTI)系统在任意输入下的时域响应。

调用格式为：

① lsim(sys,u,t)：绘制 LTI 系统在输入信号 u 作用下、在指定的时间 t 范围内系统时域响应的波形及输入信号 u 的时域波形。

其中 sys 是状态空间形式的系统函数。t 为定义时间范围的向量，u 是输入信号在向量 t 定义的时间点上的采样值。例如，t＝0:0.01:5; u＝sin(t); lsim(sys,u,t)，仿真一个单输入系统 sys 在输入 u＝sin(t)时 5 秒内的响应。

② lsim(sys,u,t,x0)：绘制系统响应的时域波形。x0 是 t（1）时刻状态向量的的初始值向量，由系统的初始状态经计算而得到。

③ lsim(sys1, sys2,...,u,t,x0)：在一个图中绘出多个 LTI 系统 sys1, sys2,…的响应波形图，初始条件 x0 是可选择的，也可以指定颜色、线型、 标记，例如 lsim(sys1,'r',sys2,'y--', sys3,'gx',u,t)。

④ y＝lsim(sys,u,t)：返回输出响应 y 的值，不绘制波形。

4. dsolve 函数

dsolve 函数是符号数学工具箱(Symbolic Math toolbox)中的专门用来求解常微分方程的函数。

调用格式为：

dsolve (' eqn1', ' eqn2',…)：接受多个表示常微分方程的符号方程和初始条件，方程或初始

条件可以集中在一起，用逗号分开。例如，dsolve('D2y＝-a^2*y', 'y(0)＝1, Dy(pi/a)＝0')。

dsolve 函数在应用时需要注意：

① t 为缺省的自变量。

② 符号 D 表示对变量进行微分运算 d/dt，D2 表示 d^2/dt^2，对应第二阶导数，D3 对应第三阶导数。例如，D3y 表示对函数 y 求三阶导数。注意，函数的名称应位于符号 D 和表示结束的数字之后。

③ 初始条件由方程给定，如'y(a)＝b'或'Dy(a)＝b'，其中 y 是因变量，a 和 b 是常数。如果无必要的初始条件，将自动定出初始条件。

例如，y＝dsolve('Dx＝-a*x')，结果为 y＝C1*exp(-a*t)，相当于初始条件为 x(0)＝C1，等价于命令 y＝dsolve('Dx＝-a*x', 'x(0)＝C1')。

（三）连续系统的冲激响应

对于 LTI 连续系统，求解系统的冲激响应 $h(t)$对连续系统的分析具有非常重要的意义。MATLAB 为用户提供了专门用于求连续系统冲激响应并绘制其时域波形的函数 impulse。

例 4-1 描述连续系统的微分方程为 $y''(t)+5y'(t)+6y(t)=3f'(t)+2f(t)$，计算系统的冲激响应 $h(t)$。

解 运行如下 MATLAB 命令：

```
a=[1 5 6];
b=[3 2];
impulse(b,a);
```

则绘出系统的冲激响应波形，如图 4-1 所示。

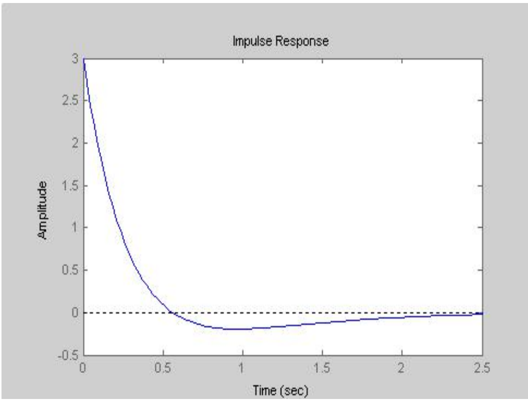


图 4-1 例 4-1 图（1）

对上例，若运行命令 impulse(b,a,10)，则绘出系统在 0～10 秒范围内冲激响应的时域波形。若运行命令 impulse(b,a,1:0.1:2)，则绘出 1～2 秒内每隔 0.1 秒取样的冲激响应的时域波形，如图 4-2 所示。运行程序如下：

```
a=[1 5 6];b=[3 2];
subplot(1,3,1);impulse(b,a);
```

```
subplot(1,3,2);impulse(b,a,10);
subplot(1,3,3);impulse(b,a,1:0.1:2);
```

对上例，若运行命令 y＝impulse(b,a,1:0.1:2)，则求出 1～2 秒内，每隔 0.1 秒取样的系统冲激响应的数值解。

```
a＝[1 5 6];b＝[3 2];y＝impulse(b,a,1:0.1:2)'
```

运行结果为：

```
y＝
-0.1928   -0.1850   -0.1716   -0.1554   -0.1383   -0.1214   -0.1054   -0.0908
-0.0777   -0.0661   -0.0559
```

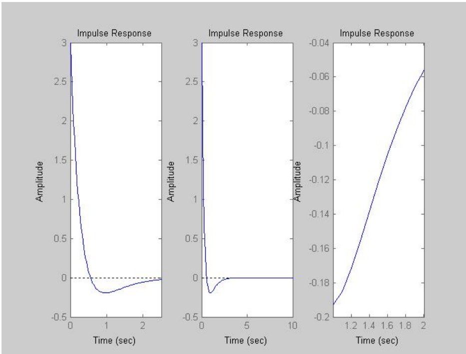


图 4-2　例 4-1 图（2）

例 4-2　描述连续系统的微分方程为 $y''(t)+3y'(t)+2y(t)=-f'(t)+2f(t)$，计算系统的冲激响应 $h(t)$。

解　运行如下 MATLAB 命令：

```
a＝[1 3 2];b＝[-1 2];sys＝tf(b,a);
t＝0:0.01:5;h＝impulse(sys,t);
plot(h);xlabel('t');title('h(t)');
```

波形如图 4-3 所示。

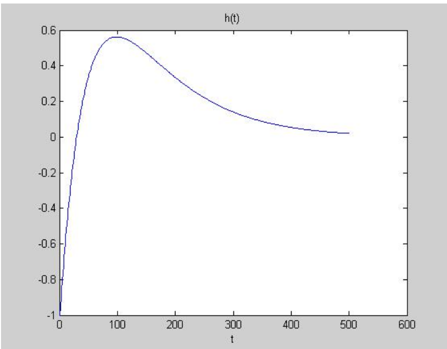


图 4-3　例 4-2 图

例 4-3　使用 impulse 函数，求下列系统的单位冲激响应 h(t)。

① $H(p)=\dfrac{p+5}{p^2+5p+6}$；

② $H(p)=\dfrac{p+5}{p^2+2p+5}$；

③ $H(p)=\dfrac{p+5}{p^2+2p+1}$。

比较这三种情况，在一张图上画出这三个信号。

解　程序如下：

```
% 系统的单位冲激响应
t＝0:0.1:10;
a＝[1 5 6];b＝[1 5];h1＝impulse(b,a,t);
a＝[1 2 5];b＝[1 5];h2＝impulse(b,a,t);
a＝[1 2 1];b＝[1 5];h3＝impulse(b,a,t);
plot(t,h1,t,h2,'--',t,h3,'+');
xlabel('t');title('单位冲激响应 h(t)');
legend('h_1(t)', 'h_2(t)', ' h_3(t)');          % legend 函数用于图表说明
```

波形如图 4-4 所示。

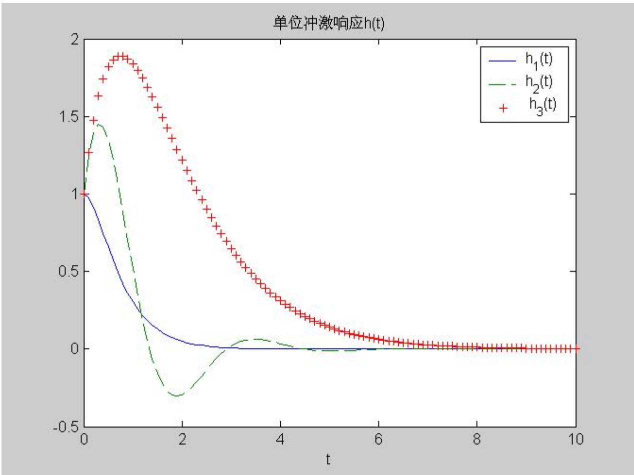


图 4-4　例 4-3 图

（四）连续系统的阶跃响应

对于 LTI 连续系统，求解系统的阶跃响应 $g(t)$对连续系统的分析同样具有非常重要的意义。MATLAB 为用户提供了专门用于求连续系统阶跃响应并绘制其时域波形的函数 step。

例 4-4　描述连续系统的微分方程为 $y''(t)+3y'(t)+2y(t)=-f'(t)+2f(t)$，计算系统的阶跃响应 $g(t)$。

解　运行如下 MATLAB 命令：

```
a＝[1 3 2]; b＝[-1 2];
subplot(1,3,1);step(b,a);
subplot(1,3,2);step(b,a,10);
subplot(1,3,3);step (b,a,0:0.1:1);
```

波形如图 4-5 所示。

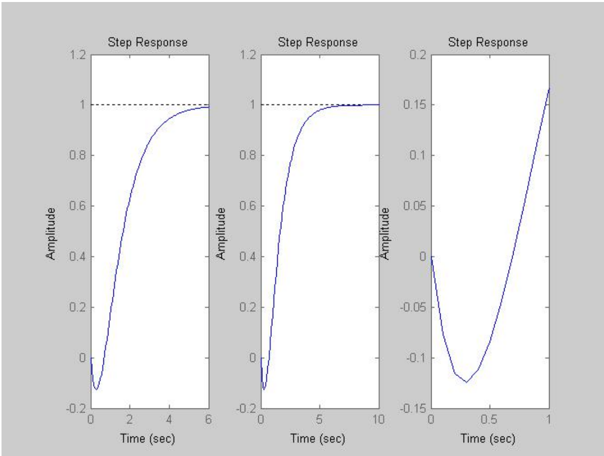


图 4-5　例 4-4 的阶跃响应

（五）连续系统响应 $y(t)$

1. 用 lsim 函数实现连续时间系统响应分析

lsim 函数不但能绘制连续系统在指定的任意时间范围内系统响应的时域波形及任意输入信号下的时域波形，还能求出连续系统在指定的任意时间范围内系统响应的数值解。lsim 函数的几种调用格式见前面的 MATLAB 函数介绍。

例 4-5　已知系统 $y''(t)+3y'(t)+2y(t)=-f'(t)+2f(t)$，计算在输入信号为 $f(t)=\mathrm{e}^{-2t}u(t)$ 时的系统零状态响应 $y(t)$。

解　运行如下 MATLAB 命令：

```
a＝[1 3 2];b＝[-1 2];sys＝tf(b,a);
t＝0:0.01:5;f＝exp(-2*t);
subplot(1,2,1);
lsim(sys,f,t);                % 绘制系统零状态响应波形和输入信号波形
subplot(1,2,2);
y＝lsim(sys,f,t);              % 绘制系统零状态响应波形
plot(t,y);xlabel('t');title('y(t)')
```

系统响应时域波形如图 4-6 所示。

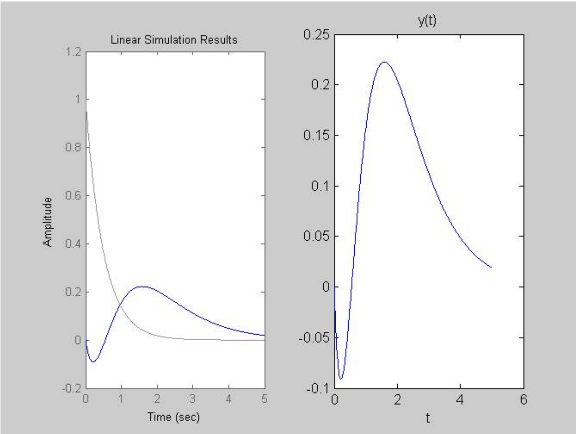


图 4-6　例 4-5 图

当系统具有初始状态时，使用 lsim(sys,u,t,x0)命令格式，sys 必须以状态方程形式求解，而且系统的初始状态必须转化为状态向量在 0 时刻的初始值。

例如，求系统 $y''(t)+20y'(t)+500y(t)=f(t)$ 在初始值为 $y(0\_)=30$，$y'(0\_)=0$ 时的零输入响应。使用下列命令可得到状态方程形式的系统函数：

```
a＝[1 20 500];
b＝[0 1];
[A,B,C,D]＝tf2ss(b,a);          % tf2ss 函数将传递函数模型转换为状态空间模型
sys＝ss(A,B,C,D)
```

运行结果如下：

```
a＝     x1      x2
      x1   -20     -500
      x2    1       0
b＝     u1
      x1    1
      x2    0
c＝     x1   x2
      y1    0     1
d＝     u1
      y1    0
```

因此系统的状态方程形式为：

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}-20 & -500\\ 1 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}1\\ 0\end{bmatrix}\begin{bmatrix}f\end{bmatrix}$$

$$\begin{bmatrix}y\end{bmatrix}=\begin{bmatrix}0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_1\end{bmatrix}+\begin{bmatrix}0\end{bmatrix}\begin{bmatrix}f\end{bmatrix}$$

将 $f=0$ 和初始状态 $y(0\_)=30$，$y'(0\_)=0$ 代入上式,可解出状态向量 x 在 0 时刻的初始值为：

$$\begin{bmatrix} x_1(0_-) \\ x_2(0_-) \end{bmatrix} = \begin{bmatrix} 0 \\ 30 \end{bmatrix}$$

例 4-6 已知系统 $y''(t)+20y'(t)+500y(t)=f(t)$，初始值为 $y(0_-)=30$， $y'(0_-)=0$，计算在输入信号为 $f(t)=\mathrm{e}^{-t}u(t)$ 时的系统零输入响应、零状态响应和完全响应。

解 计算零输入响应的程序如下：

```
a＝[1 20 500];
b＝[0 1];
[A,B,C,D]＝tf2ss(b,a);
sys＝ss(A,B,C,D);
t＝0:1/50:1;
f＝zeros(1,length(t));
zi＝[0,30];
y＝lsim(sys,f,t,zi);
plot(t,y);
xlabel('t');
title('零输入响应 y(t)');
```

计算零状态响应的程序如下：

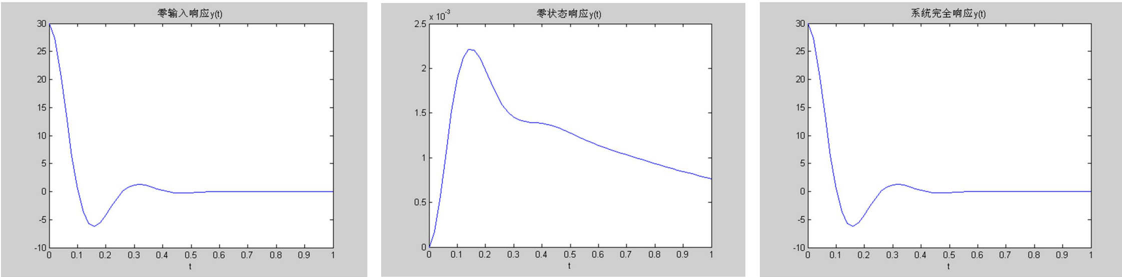
```
a＝[1 20 500];
b＝[0 1];
sys＝tf(b,a);
t＝0:0.02:1;
f＝exp(-t);
y＝lsim(sys,f,t);
plot(t,y);
xlabel('t');
title('零状态响应 y(t)');
```

计算系统完全响应的程序如下：

```
a＝[1 20 500];
b＝[0 1];
[A,B,C,D]＝tf2ss(b,a);
sys＝ss(A,B,C,D);
t＝0:1/50:1;
f＝exp(-t);
zi＝[0,30];
y＝lsim(sys,f,t,zi);
plot(t,y);
```

```
xlabel('t');
title('系统完全响应 y(t)')
```

波形如图 4-7 所示。



（a）　（b）　（c）

图 4-7　例 4-6 图

例 4-7　已知系统 $H(p)=\dfrac{p+5}{p^2+5p+6}$，输入信号 f(t)＝sint，t＝0:0.1:200，x0＝[2,5]。使用 lsim 函数，求系统的零输入响应、零状态响应和全响应 y(t)。比较这三种情况，在一张图上画出这三个信号。

解　程序如下：

```
% 系统的响应
a=[1 5 6];
b=[1 5];
[A,B,C,D]=tf2ss(b,a)
sys=ss(A,B,C,D);
t=0:0.1:10;
f=zeros(1,length(t));
x0=[2,5];
yx=lsim(sys,f,t, x0);
f=5*sin(t);
yf=lsim(sys,f,t);
y=lsim(sys,f,t, x0);
plot(t,yf,t,yx,'--',t,y,'+');
```

波形如图 4-8 所示。

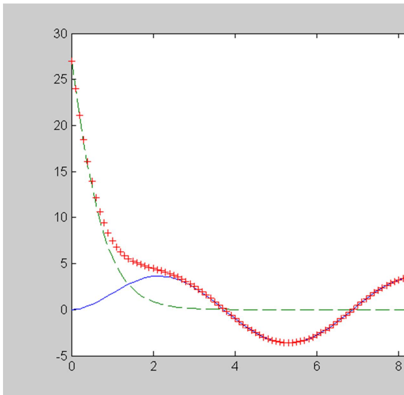


图 4-8　例 4-7 图

2.　用 dsolve 函数实现连续时间系统响应分析

系统微分方程的求解可以利用符号数学工具箱(Symbolic Math toolbox)中的 dsolve 函数。dsolve 函数专门用来求解常微分方程。

例 4-8　已知 $H(p)=\dfrac{2p^2+8p+3}{(p+1)(p+3)^2}$，初始条件为 $y(0^-)=2$，$y'(0^-)=1$，$y''(0^-)=0$，求零输入响应 $y_x(t)$。

解 计算零输入响应的程序如下：

```
y_0＝dsolve('D3y+7*D2y+15*Dy+9*y＝0','y(0)＝2','Dy(0)＝1','D2y(0)＝0','t')
y_0＝6*exp(-t)-4*exp(-3*t)-5*exp(-3*t)*t
```

即 $y(t)=6\mathrm{e}^{-t}-4\mathrm{e}^{-3t}-5t\mathrm{e}^{-3t}\quad(t\geqslant 0)$

例 4-9 已知 $H(p)=\dfrac{p+2}{p^2+3p+2}$，$f(t)=t^2$，初始条件为 $y(0^+)=1$，$y'(0^+)=1$，求全响应 $y(t)$。

解 计算全响应的程序如下：

```
y_0＝dsolve('D2y+3*Dy+2*y＝2*t+2*t^2','y(0)＝1','Dy(0)＝1','t')
y_0＝2-2*t+t^2-2*exp(-2*t)+exp(-t)
```

即 $y(t)=\mathrm{e}^{-t}-2\mathrm{e}^{-2t}+(t^2-2t+2)\quad(t\geqslant 0)$

三、实验内容

4.1 已知连续系统的微分方程为：

$$2y''(t)+y'(t)+8y(t)=f(t)$$

试用 MATLAB：

① 绘出该系统在 0～30 秒范围内，并以时间间隔 0.01 秒取样的冲激响应和阶跃响应的时域波形。

② 求出系统在 0～30 秒范围内，并以时间间隔 0.01 秒取样的冲激响应和阶跃响应的数值解。

4.2 使用 step 函数，求下列系统的单位阶跃响应。

① $H(p)=\dfrac{p+5}{p^2+5p+6}$；

② $H(p)=\dfrac{p+5}{p^2+2p+5}$；

③ $H(p)=\dfrac{p+5}{p^2+2p+1}$。

比较这三种情况，在一张图上画出这三个信号。

4.3 已知系统 $y''(t)+3y'(t)+2y(t)=-f'(t)+2f(t)$，计算在输入信号为 $f(t)=\cos t u(t)$ 时的系统零状态响应 $y(t)$。

4.4 已知系统 $H(p)=\dfrac{p+2}{p^2+3p+2}$，输入信号为 $f(t)=t^2$，初始条件为 $y(0^+)=1$，$y'(0^+)=1$，求零输入响应、零状态响应和全响应。

4.5 已知系统 $y''(t)+5y'(t)+6y(t)=2f'(t)+8f(t)$，初始值为 $y(0_-)=-3$，$y'(0_-)=0$，计算在输入信号为 $f(t)=\mathrm{e}^{-t}u(t)$ 时的系统零输入响应、零状态响应和全响应。

四、实验要求

1. 在 MATLAB 中输入程序，并将实验结果存入指定存储区域。

2. 在实验报告中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。写出自编程序，并给出实验结果。

实验五　信号的卷积计算

一、实验目的

熟悉离散时间信号的卷积和、连续时间信号的卷积积分的定义，掌握使用 MATLAB 软件来分析离散时间信号的卷积和、连续时间信号的卷积积分运算的原理和方法，熟悉 MATLAB 的 conv 函数应用，并用图形可视化相关结果。

二、实验原理

（一）卷积计算的原理

1. 卷积和

卷积和是离散信号与系统时域分析的基本方法之一，是用来计算离散系统零状态响应的有力工具。将离散时间序列 $f_1(k)$ 和 $f_2(k)$ 的卷积和 $f(k)$ 定义为：

$$f(k) = f_1(k) * f_2(k) = \sum_{j=-\infty}^{\infty} f_1(j) f_2(k-j)$$

对任意离散序列 $f(k)$ 可进行时域分解，即表示为一系列幅度由 $f(k)$ 决定的单位脉冲序列 $\delta(k)$ 及其时间平移序列的线性和，即

$$f(k) = f(k) * \delta(k) = \sum_{j=-\infty}^{\infty} f(j)\delta(k-j)$$

现将 $f(k)$ 作为激励信号接入 LTI 离散系统，系统单位脉冲响应为 h(k)，则由 LTI 系统的性质可得系统的零状态响应为

$$y(k) = f(k) * h\left(k\right) = \sum_{j=-\infty}^{\infty} f(j) h(k-j)$$

即系统的零状态响应也表示为一系列幅度由输入序列 $f(k)$ 决定的单位脉冲响应 $h(k)$ 及其时间平移序列的线性和。

2. 卷积积分

卷积积分是连续信号与系统时域分析的有效工具和基本方法之一，在信号与线性系统分析中具有非常重要的意义。

连续时间信号 $f_1(t)$和 $f_2(t)$的卷积积分（简称为卷积）$f(t)$定义为：

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

由此可得到两个与卷积相关的重要结论，即：

① $f(t)=f(t)*\delta(t)$，即连续信号可分解为一系列幅度由 $f(t)$ 决定的冲激信号 $\delta(t)$ 及其平移信号之和。

② 线性时不变连续系统，设其输入信号为 $f(t)$，单位响应为 $h(t)$，其零状态响应为 $y(t)$，则有：$y(t)=f(t)*h(t)$。

（二）MATLAB 函数介绍

1. conv 函数

conv 函数是 MATLAB 信号处理工具箱提供的一个计算两个离散序列卷积和的函数。

调用格式为：

f＝conv(f1,f2)：输入参量 f1 为包含序列 $f_1(k)$ 的所有非零样值点的行向量，f2 为包含序列 $f_2(k)$ 的所有非零样值点的行向量，输出参量 f 则为返回序列 $f(k)=f_1(k)*f_2(k)$ 的所有非零样值点的行向量。

2. length 函数

length 函数用于计算矩阵的长度（列数），调用格式为

Y＝length(X)

此命令将矩阵 X 的列数赋值给变量 Y。

例如，运行如下命令：

```
X1＝[1 2 3
     4 5 6];
X2＝[1 2 3 4 5 6];
Y1＝length(X1)
Y2＝length(X2)
```

运行结果为：

```
Y1＝3
Y2＝6
```

（三）卷积和的 MATLAB 实现

使用 conv 函数可以实现两个离散序列的卷积和。

例 5-1 已知序列 f1(k)＝{1,1,1,1;k＝0,1,2,3}， f2(k)＝{1,2,3,4,5;k＝0,1,2,3,4}， 计算 f1(k)∗f2(k)，并画出卷积结果。

解 程序如下：

```
% 序列的卷积和
f1＝[1,1,1,1];
f2＝[1,2,3,4,5];
y＝conv(f1,f2)                % 计算 f1(k)*f2(k)
L1＝length(f1)                % 计算序列 f1 的长度(列数)L1
L2＝length(f2)                % 计算序列 f2 的长度(列数)L2
L＝length(y)                  % 计算卷积和 y 的长度(列数)L
```

运行的结果为：

```
y＝1    3    6    10    14    12    9    5
L1＝4
L2＝5
L＝8
```

注意，当 f1(k)改变为 f1(k)＝{1,1,1,1;k＝-3,-2,-1,0}，f2(k)不变时，f1(k)∗f2(k)也可以由命令 conv([1,1,1,1],[1,2,3,4,5])计算并得到相同的结果。两种情况的区别在于卷积和的存在区间不同，前者为 0~7，后者为-3~4。虽然 conv 函数没有计算存在区间，但是这比较容易确定。若向量 w 开始于 n_w，向量 v 开始于 n_v，则 conv(w,v)开始于 n_w+n_v。

绘制卷积结果的语句为：

```
stem(0:L-1,y,'filled');
```

波形如图 5-1 所示。

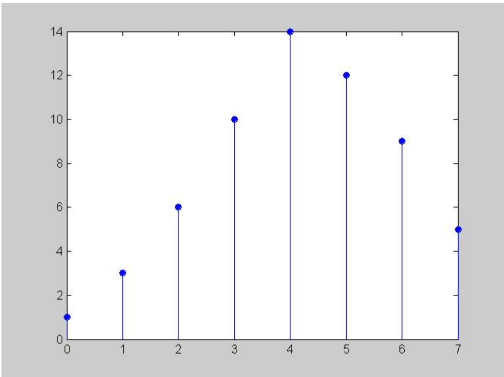


图 5-1 例 5-1 图

根据例 5-1 的程序，可得到如下结论：

① f1(k)在区间 0~3 非零，f2(k)在区间 0~4 非零，卷积和 y 在区间 0~7 非零。

② f1(k)的时域宽度为 L1＝4，f1(k)的时域宽度为 L2＝5，卷积和 y 的时域宽度为 L＝8。

③ L＝L1+L2-1，卷积和 y 在区间 0~ L-1 非零，或 0~ L1+L2-2 非零。

一般而言，$f_1(k)$ 和 $f_2(k)$ 为两个在有限区间非零的离散时间序列，$f_1(k)$ 在区间 $n_1 \sim n_2$ 非零，$f_2(k)$ 在区间 $m_1 \sim m_2$ 非零，则可求得序列 $f_1(k)$ 的时域宽度为 $L_1 = n_2 - n_1 + 1$，序列 $f_2(k)$ 的时域宽度为 $L_2 = m_2 - m_1 + 1$。由卷积和的定义可得，卷积和序列 $f(k) = f_1(k) * f_2(k)$ 的时域宽度为 $L = L_1 + L_2 - 1$，且只在区间 $(n_1 + m_1) \sim (n_1 + m_1) + (L_1 + L_2) - 2$ 非零。

conv 函数也可以用来计算两个多项式的积。例如，多项式$(s^2+2s+3)(s^4+5s^2+6s+2)$乘积可以通过下面的程序给出：

```
% 多项式的乘积计算
f1＝[1,2,3]; f2＝[1,0,5,6,2];
y＝conv(f1,f2)
```

运行的结果为：

```
y＝1    2    8    16    29    22    6
```

即

$$(s^2+2s+3)(s^4+5s^2+6s+2)=s^6+2s^5+8s^4+16s^3+29s^2+22s+6$$

f1＝[1,2,3]和 f2＝[1,0,5,6,2]分别是多项式(s^2+2s+3)和(s^4+5s^2+6s+2)的向量表示。注意，在用向量表示多项式时，应将多项式各项包括零系数项的系数均写入向量的对应元素中。如(s^4+5s^2+6s+2)中的 s^3 的系数为零，所以第二个元素为零。

由于 conv 函数只返回卷积和结果序列 $f(k)$ 的采样样值，并不返回其对应的时间序号，因此要完整地表示出卷积和的计算结果序列，还需产生参与卷积运算的两序列 $f_1(k)$ 和 $f_2(k)$ 的时间序号向量 k_1 和 k_2，以及卷积和结果序列的对应时间序号向量 k。

例 5-2　已知序列 $f_1(k) = 5(0.5)^k + 2(0.2)^k \ (k \geqslant 0)$，$f_2(k) = u(k) - u(k-11)$，计算 $f_1(k) * f_2(k)$。

解　计算 $f_1(k) * f_2(k)$ 的程序如下：

```
k1＝0:10; f1＝5*0.5.^k1+2*0.2.^k1;
k2＝0:10; f2＝ones(1,11);
f＝conv(f1,f2);                              %计算序列 f1 与 f2 的卷积和 f
k0＝k1（1）+k2（1）;                          %计算序列 f 非零样值的起点位置
L＝length(f);                                %计算卷积和 f 的非零样值的宽度
k＝k0:(k0+L-1);                              %确定卷积和 f 非零样值的时间向量
subplot(2,2,1);stem(k1,f1,'filled');         %在子图 1 绘 f1(t)时域波形图
title('f1(k)');xlabel('k');ylabel('f1(k)')
subplot(2,2,2);stem (k2,f2,'filled');        %在子图 2 绘 f2(t)时波形图
title('f2(k)');xlabel('k');ylabel('f2(k)')
subplot(2,2,3);stem (k,f,'filled');
title('f(k)＝f1(k)*f2(k)');xlabel('k');ylabel('f(k)')
h＝get(gca,'position');                      %获取当前坐标轴对象位置
h（3）＝2*h（3）;
set(gca,'position',h)                        %将第三个子图的横坐标范围扩为原来的 2 倍
```

波形如图 5-2 所示。

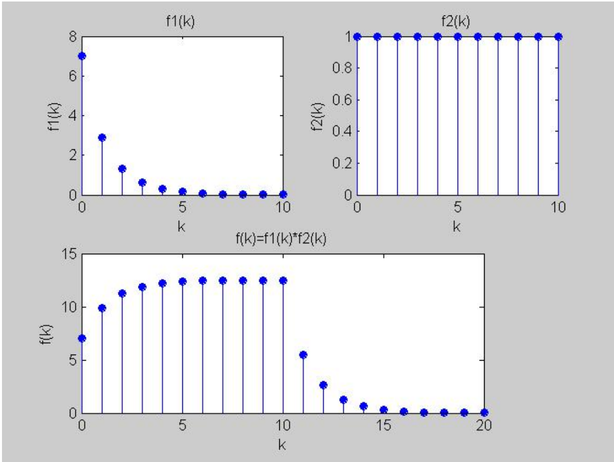


图 5-2　例 5-2 图

下面给出的函数 hconv 不仅能求出两离散序列的卷积和，还可求出其对应时间序号向量，并绘出各序列的时域波形。

```
function [f,k]＝hconv(f1,f2,k1,k2)
%计算离散序列卷积和 f(k)
f＝conv(f1,f2);                    %计算序列 f1 与 f2 的卷积和 f
k0＝k1（1）+k2（1）;               %计算序列 f 非零样值的起点位置
L＝length(f);                      %计算卷积和 f 的非零样值的宽度
k＝k0:(k0+L-1);                    %确定卷积和 f 的非零样值的时间向量
subplot(2,2,1)
stem(k1,f1,'filled')               %在子图 1 绘 f1(t)时域波形图
title('f1(k)')
xlabel('k')
ylabel('f1(k)')
subplot(2,2,2)
stem (k2,f2,'filled')              %在子图 2 绘 f2(t)时波形图
title('f2(k)')
xlabel('k')
ylabel('f2(k)')
subplot(2,2,3)
stem (k,f,'filled');
title('f(k)＝f1(k)*f2(k)')
xlabel('k')
ylabel('f(k)')
h＝get(gca,'position');            %获取当前坐标轴对象位置
h（3）＝2.5*h（3）;
```

```
set(gca,'position',h)              %将第三个子图的横坐标范围扩为原来的 2.5 倍
```

注意，conv 函数在计算两离散序列卷积和时，将参与卷积运算的两序列向量 $f_1(k)$ 和 $f_2(k)$ 视为有限时间区间非零的离散时间序列，即将向量 $f_1(k)$ 和 $f_2(k)$ 时域宽度以外的序列样值视为零。因此，当参与卷积运算的两序列 $f_1(k)$ 和 $f_2(k)$ 为时间无限长序列时，调用 conv 函数就可能产生误差，即只有部分计算结果是正确的。

例 5-3　已知系统传递算子为 $H(E)=\dfrac{E(7E-2)}{(E-0.5)(E-0.2)}$，利用 MATLAB 计算在输入信号为① $f(k)=u(k)-u(k-11)$，② $f(k)=u(k)$ 时的系统零状态响应。

解　由　$H(E)=\dfrac{E(7E-2)}{(E-0.5)(E-0.2)}=\dfrac{7E^2-2E}{E^2-0.7E+0.1}$

求得　$h(k)=5(0.5)^k+2(0.2)^k\quad(k\geqslant 0)$

则　$y_f(k)=f(k)*h(k)$

① 用 conv 函数计算 $f(k)=u(k)-u(k-11)$ 时的系统零状态响应的程序：

```
k1＝0:10; f1＝5*0.5.^k1+2*0.2.^k1;
k2＝0:10; f2＝ones(1,21);
[f,k]＝hconv(f1,f2,k1,k2);
```

波形如图 5-3(a)所示。

② 下面计算 $f(k)=u(k)$ 时的系统零状态响应：

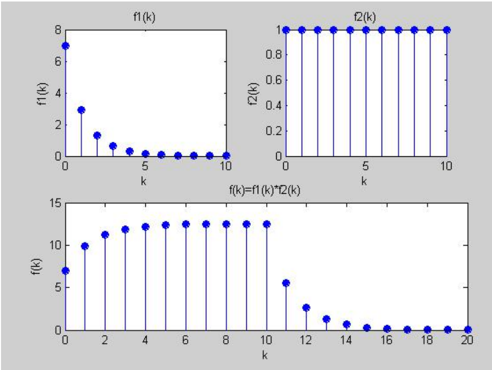
```
%用 z 变换法求解系统的零状态响应
syms n z;
f＝sym('Heaviside(n)');
F＝ztrans(f);
H＝z*(7*z-2)/((z-0.2)*(z-0.5));
y＝iztrans(H*F)
```

运行结果得：

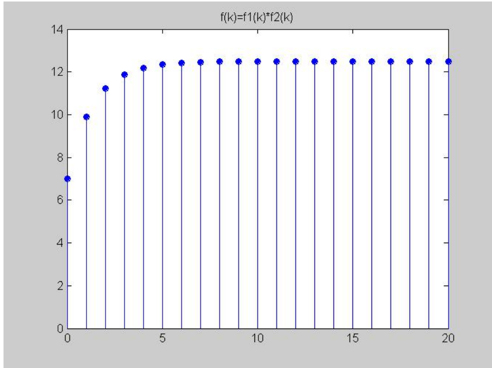
```
y＝-1/2*(1/5)^n-5*(1/2)^n+25/2
%绘制系统的零状态响应
k＝0:20;
y＝-1/2*(1/5).^k-5*(1/2).^k+25/2;
stem (k,y,'filled');
title('f(k)＝f1(k)*f2(k)')
```

波形如图 5-3(b)所示。

$f(k)$ 和 h(k) 是无限长因果序列，所以 conv 只能获得卷积的部分值。图 5-3(a)给出了 conv 的输出，图 5-3(b)给出的是卷积的理论结果输出，可以看出在 $0<k<10$ 的范围 conv 的结果是正确的，在其余($k\geqslant 10$)的范围 conv 的结果是不正确的，输出包络应该继续不变而不应该衰减。



(a)



(b)

图 5-3　例 5-3 图

（四）卷积积分的 MATLAB 实现

下面主要用数值方法来近似求两个信号之间的卷积积分。

由连续时间信号 $f_1(t)$ 和 $f_2(t)$ 的卷积积分 $f(t)$ 定义有：

$$f_1(t) * f_2(t) = \int_{-\infty}^{\infty} f_1(\tau) f_2(t-\tau) \mathrm{d}\tau$$

则有

$$f(kT) \approx \sum_{m=-\infty}^{\infty} f_1(mT) f_2(kT - mT)T \ = Tf_1(kT) * f_2(kT)$$

上式说明，连续时间信号的卷积积分可以用各自取样后的离散时间信号的卷积和再乘以取样间隔 T。取样间隔 T 越小，得到的误差就越小。

用 MATLAB 实现连续信号 $f_1(t)$ 与 $f_2(t)$ 卷积的过程如下：

① 将连续信号 $f_1(t)$ 与 $f_2(t)$ 按时间间隔 T 进行取样，得到离散序列 $f_1(kT)$ 和 $f_2(kT)$。

② 构造与 $f_1(kT)$ 和 $f_2(kT)$ 相对应的时间向量 k_1 和 k_2。

③ 调用 conv 函数计算卷积积分 $f(t)$ 的近似向量 $f(kT)$。

④ 构造 $f(kT)$ 对应的时间向量 k。

例 5-4　实现连续信号 $f_1(t) = u(t-2) - u(t-3)$，　$f_2(t) = \mathrm{e}^{(-t-1)} u(t+1)$ 的卷积积分。

解　程序如下：

```
%连续信号卷积积分
Ts＝0.001;
t＝-1.5:Ts:6;                    %f1(t)和 f2(t)都从 t＝-1.5 开始，到 t＝6 结束
f1＝(t>2)-(t>3);                 %产生信号 f1(t)
f2＝exp(-t-1).*(t>＝-1);         %产生信号 f2(t)
y＝conv(f1,f2)*Ts;               %计算 f1(t)和 f2(t)卷积积分的近似值 y
L＝length(y);                    %计算卷积和 y 的宽度
k＝(0:L-1)*Ts-3;                 %卷积和 y 的起点 t＝-3
```

```
subplot(221);
plot(t,f1);grid on;title('f1');
subplot(222);
plot(t,f2);grid on;title('f2');
subplot(212);
plot(k,y);grid on;title('f1 与 f2 的卷积积分');
```

波形如图 5-4 所示。

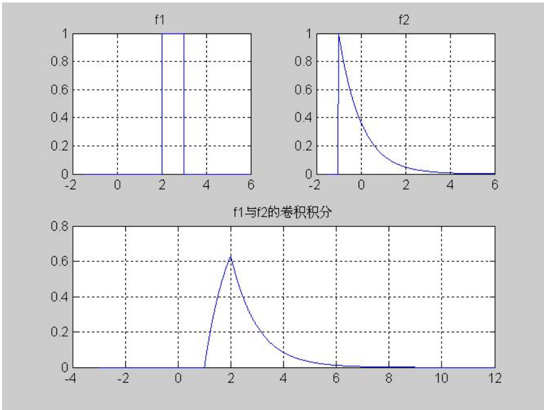


图 5-4　例 5-4 图

下面是利用 MATLAB 实现连续时间卷积的通用函数 pconv()，该程序在计算出卷积积分的近似数值的同时，还绘出 $f(t)$ 的时域波形图。需要注意的是，程序中如何构造 $f(t)$ 的对应时间向量 k。另外，程序在绘制 $f(t)$ 波形图时采用的是 plot 命令而不是 stem 命令。

程序如下：

```
function [f,k]＝pconv(f1,f2,k1,k2,Ts)
%计算连续信号卷积积分 f(t))
% f: 卷积积分 f(t)对应的非零样值向量
% k：f(t)的对应时间向量
% f1: f1(t)非零样值向量
% f2: f2(t)的非零样值向量
% k1: f1(t)的对应时间向量
% k2: f2(t)的对应时间向量
% Ts：取样时间间隔
f＝conv(f1,f2);               %计算序列 f1 与 f2 的卷积和 f
f＝f*Ts;
k0＝k1（1）+k2（1）;             %计算序列 f 非零样值的起点位置
L＝length(f1)+length(f2)-2;
k＝k0:Ts:(k0+L*Ts);           %确定卷积和 f 非零样值的时间向量
subplot(2,2,1)
plot(k1,f1); grid on;         %在子图 1 绘 f1(t)时域波形图
```

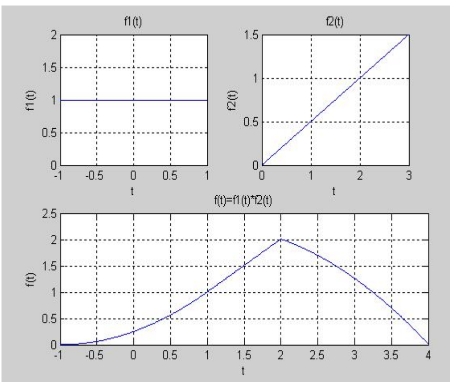
```
title('f1(t)')
xlabel('t')
ylabel('f1(t)')
subplot(2,2,2)
plot(k2,f2); grid on;                %在子图 2 绘 f2(t)时波形图
title('f2(t)')
xlabel('t')
ylabel('f2(t)')
subplot(2,2,3)
plot(k,f);grid on;                   %画卷积 f(t)的时域波形
h＝get(gca,'position');
h（3）＝2.5*h（3）;
set(gca,'position',h)                %将第三个子图的横坐标范围扩为原来的 2.5 倍
title('f(t)＝f1(t)*f2(t)')
xlabel('t')
ylabel('f(t)')
```

例 5-5　已知 $f_1(t)=\begin{cases}1 & |t|<1\\ 0 & |t|>1\end{cases}$，$f_2(t)=\dfrac{t}{2}$　$(0\leqslant t\leqslant 3)$，求 $f(t)=f_1(t)*f_2(t)$，并绘出 $f(t)$ 的时域波形图（设定取样时间间隔为 Ts）。

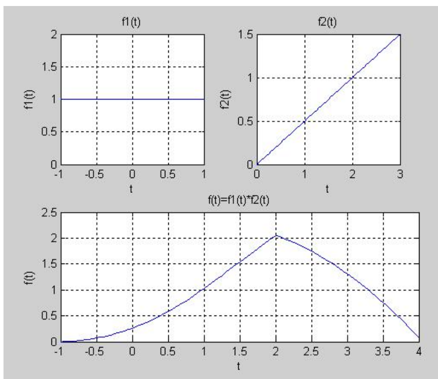
解　计算卷积积分的参考程序如下：

```
Ts＝0.05;
k1＝-1:Ts:1;
f1＝ones(1,length(k1)) ;
k2＝0:Ts:3;
f2＝0.5*k2;
[f,k]＝pconv(f1,f2,k1,k2,Ts)
```

运行结果如图 5-5 所示。



Ts＝0.01



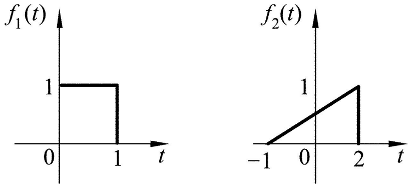
Ts＝0.05

图 5-5　例 5-5 图

三、实验内容

5.1　已知离散时间系统的单位脉冲响应 $h(k)=\mathrm{e}^{-\frac{1}{4}k}\left[u(k)-u(k-15)\right]$，试用 MATLAB 求当输入序列为 $f(k)=u(k)-u(k-5)$ 时，系统的零状态响应 $y(k)$，并绘出 $f(k)$、$h(k)$ 及 $y(k)$ 的时域波形。

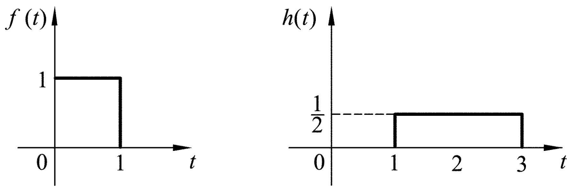
5.2　已知两连续时间信号如题 5.2 图所示，试用 MATLAB 求 $f(t)=f_1(t)*f_2(t)$，并绘出 $f(t)$ 的时域波形图。



题 5.2 图

5.3　已知两连续时间信号如题 5.3 图所示，试用 MATLAB 求 $y(t)=f(t)*h(t)$，并绘出 $y(t)$ 的时域波形图。

t



题 5.3 图

四、实验要求

1. 在 MATLAB 中输入程序，验证实验结果，并将实验结果存入指定存储区域。

2. 在实验报告中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，写出自编程序，并给出实验结果，总结实验得出的主要结论。

实验六　周期信号的傅里叶级数分析

一、实验目的

熟悉连续时间周期信号的傅里叶级数分解原理及方法，掌握周期信号的傅里叶频谱的概念及计算方法，熟悉相应 MATLAB 函数的调用格式和作用，掌握利用 MATLAB 计算傅里叶级数系数及绘制频谱图的方法。

二、实验原理

（一）周期信号的傅里叶级数分析原理

按傅里叶分析的原理，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。

1. 三角函数形式的傅里叶级数

$$\begin{aligned} f(t) &= a_0 + a_1\cos(\omega_0 t) + b_1\sin(\omega_0 t) + a_2\cos(2\omega_0 t) + b_2\sin(2\omega_0 t) + \cdots \\ &= a_0 + \sum_{n=1}^{\infty} a_n\cos(n\omega_0 t) + b_n\sin(n\omega_0 t) \end{aligned} \qquad (6\text{-}1)$$

式，中 a_0、a_n、b_n 称为傅里叶系数。

$$a_0 = \frac{1}{T}\int_{t_0}^{t_0+T} f(t)\,\mathrm{d}t$$

$$a_n = \frac{2}{T}\int_{t_0}^{t_0+T} f(t)\cos(n\omega)\,t\mathrm{d}t \quad n = 1,3,5,\cdots$$

$$b_n = \frac{2}{T}\int_{t_0}^{t_0+T} f(t)\cos(n\omega)\,t\mathrm{d}t \quad n = 1,3,5,\cdots$$

即可以用一组正弦波和余弦波来合成任意的周期信号。

式（6-1）的三角函数形式傅里叶级数可以写成余弦函数的形式

$$f(t)=c_0+\sum_{n=1}^{\infty}c_n\cos(n\omega_0 t+\varphi_n)\qquad(-\infty<t<\infty)\tag{6-2}$$

式中
$$\begin{cases}c_0=a_0\\c_n=\sqrt{a_n^2+b_n^2}\\\varphi_n=\arctan\left(-\dfrac{b_n}{a_n}\right)\end{cases}\qquad(n=1,\ 2,\ \cdots)\tag{6-3}$$

以及
$$\begin{cases}a_n=c_n\cos\varphi_n\\b_n=-c_n\sin\varphi_n\end{cases}$$

2. 指数函数形式的傅里叶级数

$$f(t)=\sum_{n=-\infty}^{\infty}F_n\mathrm{e}^{\mathrm{j}n\omega_0 t}$$

连续时间信号的傅里叶级数变换对：

$$f(t)=\sum_{n=-\infty}^{\infty}F_n\mathrm{e}^{\mathrm{j}n\omega_0 t}\ ,\quad F_n=\frac{1}{T}\int_T f(t)\mathrm{e}^{-\mathrm{j}n\omega_0 t}\mathrm{d}t$$

3. 周期信号的频谱

（1）三角函数形式频谱：

c_n-ω 关系曲线称为幅度频谱图。

φ_n-ω 关系曲线称为相位频谱图。

（2）指数函数形式频谱：

$|F_n|$-ω 关系曲线称为幅频图。

φ_n-ω 关系曲线称为相频图。

（二）周期信号的傅里叶级数的 MATLAB 实现

例 6-1 试用 MATLAB 求如图 6-1 所示的周期方波信号的傅里叶级数分解。

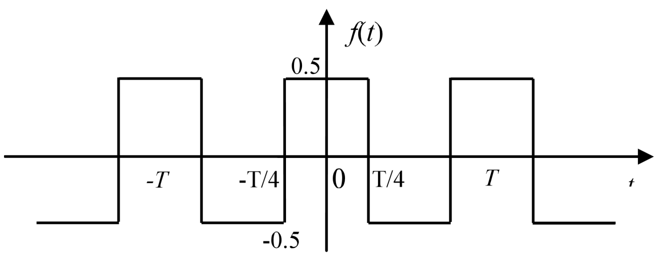


图 6-1 例 6-1 周期方波信号

解 周期方波信号是一个偶函数，又是一个奇谐波函数，因此其傅立叶级数只含有奇次谐波的余弦项，即周期方波信号可以分解为：

$$a_n = \frac{2}{T}\int_{t_0}^{t_0+T} f(t)\cos(n\omega)t\mathrm{d}t = \frac{4}{T}\int_{-\frac{T}{4}}^{\frac{T}{4}} 0.5\cos(n\omega)t\mathrm{d}t \quad (n=1,3,5,\cdots)$$

求傅里叶系数 a_n 的程序如下：

```
syms t n T;
y＝0.5*cos(n*(2*pi)/T*t);
an＝(4/T)*int(y,-T/4,T/4)
```

运行结果为：

```
an＝2*sin(1/2*pi*n)/pi/n
```

则此周期方波信号可以分解为：

$$b_n = 0\text{，}\quad a_n = \frac{2}{n\pi}\sin\left(\frac{n\pi}{2}\right) \quad (n=1,3,5,\cdots)$$

将其展开为三角函数形式的傅立叶级数：

$$f(t) = \frac{2}{\pi}\left[\cos(\omega t) - \frac{1}{3}\cos(3\omega t) + \frac{1}{5}\cos(5\omega t) - \cdots\right]$$

$$f(t) = \frac{2}{\pi}\sum_{n=2j-1}^{\infty}\frac{\sin\frac{n\pi}{2}}{n}\cos(n\omega t) \quad (j=1,2,3,\cdots)$$

例 6-2 根据例 6-1 的结果，试用正弦信号的叠加近似合成一频率为 50Hz、幅值为 3 的方波。

解 由例 6-1 知，周期方波信号可以分解为：

$$b_n = 0\text{，}\quad a_n = \frac{2}{n\pi}\sin\left(\frac{n\pi}{2}\right) \quad (n=1,3,5,\cdots)$$

则
$$f(t) = \frac{2}{\pi}\sum_{n=2j-1}^{\infty}\frac{\sin\frac{n\pi}{2}}{n}\cos(n\omega t) \quad (j=1,2,3,\cdots)$$

MATLAB 程序如下：

```
clear all;
fs＝10000;
t＝0:1/fs:0.1;
f0＝50;
sum＝0;
subplot(211)
for n＝1:2:9;
    plot(t,(2/pi)*(1/n)*sin(pi*n/2)*cos(n*2*pi*f0*t));
```

```
        title('谐波信号叠加前');
        hold on;
    end
    subplot(212)
    for n=1:2:9;
        sum=sum+2/pi*(1/n)*sin(pi*n/2)*cos(n*2*pi*f0*t);
    end
    plot(t,sum);
    title('谐波信号叠加后');
```

波形如图 6-2 所示。

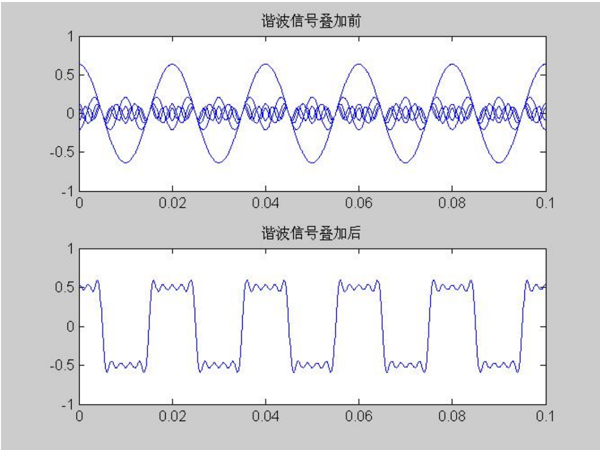


图 6-2　例 6-2 图

例 6-3　观察例 6-1 所示周期方波信号的吉布斯现象。

解　已知周期方波信号的傅立叶级数为：

$$b_n = 0\ ,\quad a_n = \frac{2}{n\pi}\sin\left(\frac{n\pi}{2}\right) \qquad (n = 1,3,5,\cdots)$$

执行下列程序，令 N 分别为 10，20，30，40，观察波形的特点，了解吉布斯现象的特点。MATLAB 程序如下：

```
    clear all;
    t=-1.5:0.01:1.5;
    w0=4;
    for M=1:4;
    N=10*M;
    xN=0;
    for n=1:N
        an=(2/(n*pi))*sin(n*pi/2);
        xN=xN+an.*cos(n*w0*t);
```

```
    end
    subplot(2,2,M);
    plot(t,xN);
    grid on;
    xlabel('time');
    axis([-2 2 -0.7 0.7]);
    end
```

波形如图 6-3 所示。

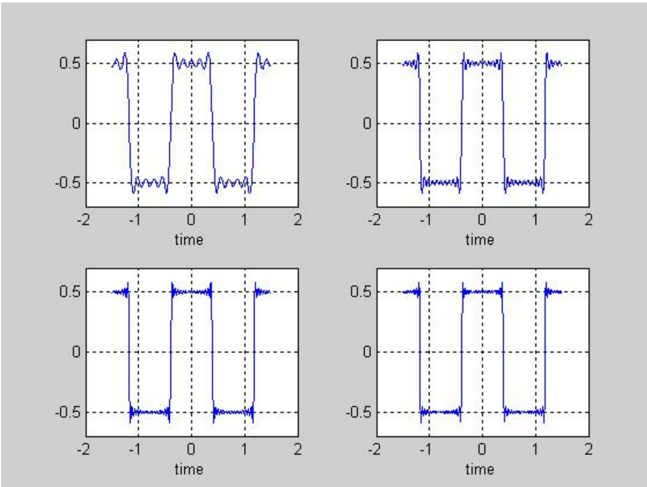


图 6-3　例 6-3 图

从上述波形可以看出，傅里叶级数所取项数越多，相加后波形越逼近原周期方波信号。但注意，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，部分和有起伏。随着 N 的增加，起伏的峰值大小保持不变而趋于一个常数，大约等于总跳变值的 9%，并从不连续点开始以起伏振荡的形式逐渐衰减下去，这种现象通常称为“吉布斯(Gibbs)现象”。

（三）周期信号的频谱分析的 MATLAB 实现

1. 周期信号的频谱分析的实现

例 6-4　已知周期方波信号如例 6-1 图所示，试用 MATLAB 计算并绘出周期方波信号的频谱图。

解　周期方波信号可以分解为：

$$b_n = 0\text{，}\quad a_n = \frac{2}{n\pi}\sin\left(\frac{n\pi}{2}\right) \qquad (n = 1,3,5,\cdots)$$

程序如下：

```
    %周期方波信号频谱图
```

```
display('Please input the value of m');
m＝input('m＝');
an＝zeros(m+1,1);
an（1）＝0;
for i＝1:m
an(i+1)＝2/(i*pi)*sin(i*pi/2);          %计算系数 an
cn(i+1)＝abs(2/(i*pi)*sin(i*pi/2));     %计算幅度谱 cn
end
t＝-5:0.01:5;
x＝0.5*square(pi*(t+0.5));
subplot(211);
plot(t,x);
axis([-5 5 -1 1]);
xlabel('t  单位:s','Fontsize',8);
subplot(212);
k＝0:m;
stem(k,cn);
hold on;
plot(k,cn);
title('幅度频谱','Fontsize',8);
xlabel('谐波次数','Fontsize',8);
%end
```

波形如图 6-4 所示。

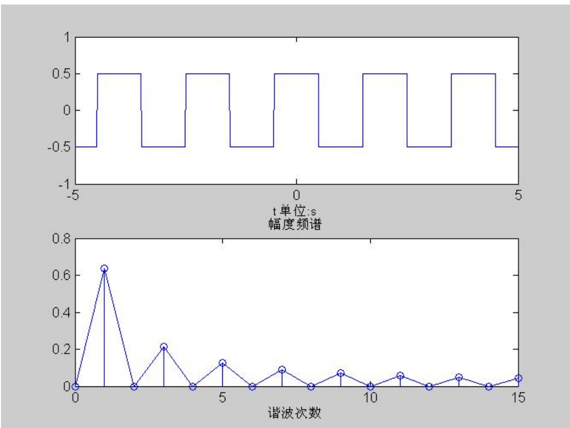


图 6-4　周期方波信号的频谱

例 6-5　试用 MATLAB 计算并绘出如图 6-5 所示的周期矩形脉冲的幅度频谱图和相位频谱图。

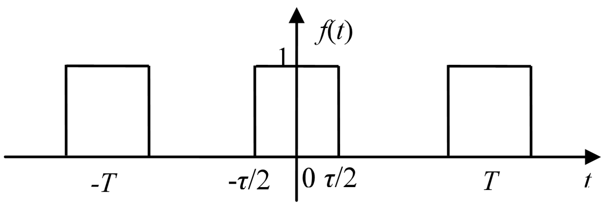


图 6-5　周期矩形脉冲信号

解　程序如下：

```
% 求矩形脉冲的傅立叶系数 an
syms t n T tao;
y＝1*cos(n*(2*pi)/T*t);
an＝(2/T)*int(y,-tao/2,tao/2)
y0＝1;
a0＝(1/T)*int(y0,-tao/2,tao/2)
```

运行的结果为：

```
an＝2*sin(tao*n*pi/T)/n/pi
a0＝1/T*tao
% 矩形脉冲的频谱图
display('Please input the value of T,tao and m');
T＝input('T＝');
tao＝input('tao＝');
m＝input('m＝');
an＝zeros(m+1,1); phase＝zeros(m+1,1);
an（1）＝1/T*tao;cn（1）＝1/T*tao;
for i＝1:m
an(i+1)＝2*sin(tao*i*pi/T)/i/pi;             %计算系数 an
cn(i+1)＝abs(2*sin(tao*i*pi/T)/i/pi);        %计算幅度谱 cn
if an(i)>＝0                                 %计算相位谱
      phase(i)＝0;
else
      phase(i)＝pi;
   end
end
t＝-5*T:0.01:5*T;
x＝0.5*(square(2*pi/T*(t+(tao/2)), tao*100/T)+1);
subplot(311); plot(t,x);
title('周期矩形脉冲  T＝10 tao＝2','Fontsize',8);
axis([-5*T 5*T -0.1 1.5]);
xlabel('t  单位:s','Fontsize',8);
```

```
subplot(312);
k=0:m;
stem(k,cn);
hold on;
plot(k,cn);
title('周期矩形脉冲的幅度频谱 cn','Fontsize',8);
subplot(313);
stem(k,phase);
title('周期矩形脉冲的相位频谱','Fontsize',8);
xlabel('谐波次数','Fontsize',8);
%end
```

Please input the value of T,tao and m

T＝10,tao＝2,m＝30

运行结果如图 6-6 所示。

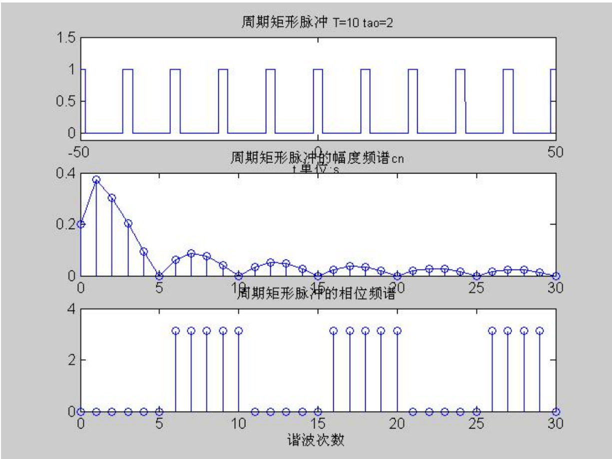


图 6-6　周期矩形脉冲的频谱

2. 周期信号的脉冲宽度与频谱的关系分析

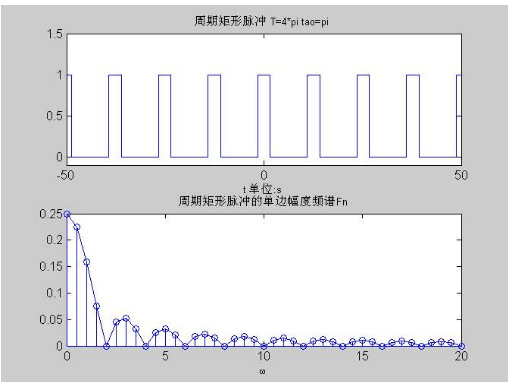
例 6-6　试用 MATLAB 分析当周期矩形脉冲的周期 T 保持不变时，脉冲宽度 τ 与频谱的关系。

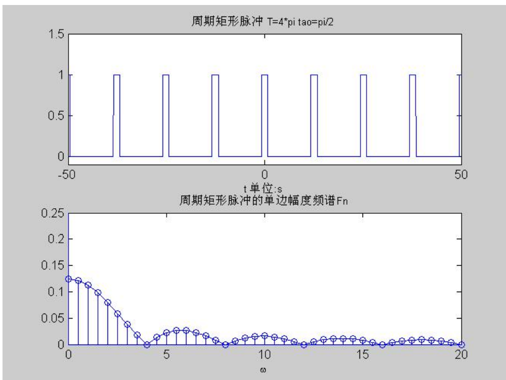
解　程序如下：

```
%矩形脉冲的脉冲宽度τ与频谱的关系
display('Please input the value of T,tao and m');
T＝input('T＝');tao＝input('tao＝');m＝input('m＝');
an＝zeros(m+1,1);
an（1）＝1/T*tao; cn（1）＝1/T*tao; Fn（1）＝1/T*tao;
for k＝1:m
```

```
        an(k+1)＝2*sin(tao*k*pi/T)/k/pi;                      %系数 an
        cn(k+1)＝abs(2*sin(tao*k*pi/T)/k/pi);                 %计算幅度谱 cn
         Fn(k+1)＝cn(k+1)/2;                                  %计算幅度谱 Fn
end
t＝-7*T:0.01:7*T;
x＝0.5*(square(2*pi/T*(t+(tao/2)), tao*100/T)+1);   %产生宽度为τ、周期为 T 的矩形脉冲
subplot(211);                                       %将显示窗口分为 2 个子窗口，
                                                      指向第一个子窗口
plot(t,x);                                                  %绘制周期矩形脉冲信号
title('周期矩形脉冲  T＝4*pi tao＝pi','Fontsize',8);         %标注标题
axis([-50 50 -0.1 1.5]);                                    %指定坐标轴范围
xlabel('t  单位:s','Fontsize',8);                            %x 轴标注
w0＝2*pi/T;                                                  %基波角频率ω0
N＝m*w0;
w＝0:w0:N;                                                   %形成 0:w0:N 的变量
subplot(212);
stem(w,Fn);                                                 %绘制幅度频谱 Fn
hold on;
plot(w,Fn);                                                  %绘制幅度频谱 Fn 的包络线
title('周期矩形脉冲的单边幅度频谱 Fn','Fontsize',8);
xlabel('\omega','Fontsize',8);
axis([0 20 0 0.25]);                                          %指定频谱图的坐标轴范围
line([0 20],[0 0]);
line([0 0],[0 0.25]);
%end
```

在周期矩形脉冲的周期 T 保持不变时，通过输入不同的脉冲宽度 τ，可以观察到频谱与脉冲宽度之间的关系。图 6-7 给出了三种输入情况下的频谱图：T＝4*pi,tao＝pi,m＝40，T＝4*pi,tao＝pi/2,m＝40，T＝4*pi,tao＝pi/4,m＝40。





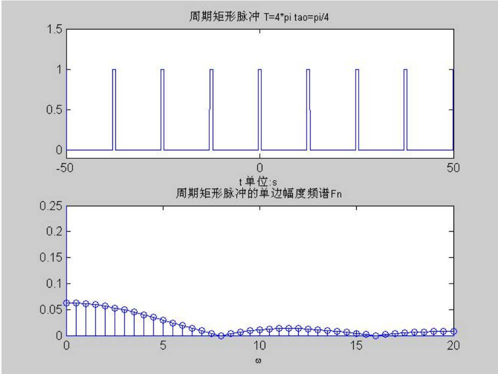


图 6-7　频谱与脉冲宽度的关系

由图 6-7 可见，周期 T 不变，谱线间隔也不变。改变脉冲宽度 τ 时，脉冲宽度 τ 越小，频谱包络线第一个过零点 $\left(\frac{2\pi}{\tau}\right)$ 的频率越高，即信号带宽越宽，频带内所含的分量越多。因此，信号的频带宽度与脉冲宽度 τ 成反比。另外，随着脉冲宽度 τ 的减小，各谐波分量的幅度也相应减小。

3. 周期信号的周期与频谱的关系分析

例 6-7　试用 MATLAB 分析当周期矩形脉冲的脉冲宽度 τ 保持不变时，周期 T 与频谱的关系。

解　程序如下：

```
%矩形脉冲的周期周期 T 与频谱的关系
display('Please input the value of T,tao and m');
T＝input('T＝');tao＝input('tao＝');m＝input('m＝');
an＝zeros(m+1,1); an（1）＝1/T*tao;cn（1）＝1/T*tao; Fn（1）＝1/T*tao;
for k＝1:m
    an(k+1)＝2*sin(tao*k*pi/T)/k/pi;              %系数 an
    cn(k+1)＝abs(2*sin(tao*k*pi/T)/k/pi);         %计算幅度谱 cn
    Fn(k+1)＝cn(k+1)/2;                           %计算幅度谱 Fn
end
t＝-7*T:0.01:7*T;
x＝0.5*(square(2*pi/T*(t+(tao/2)), tao*100/T)+1);
subplot(211);
plot(t,x);
title('周期矩形脉冲  T＝8*pi tao＝pi ','Fontsize',8);
axis([-50 50 -0.1 1.5]);
xlabel('t  单位:s','Fontsize',8);
w0＝2*pi/T;
N＝m*w0;
w＝0:w0:N;
```

```
subplot(212);
stem(w,Fn);
hold on;
plot(w,Fn);
title('周期矩形脉冲的单边幅度频谱 Fn ','Fontsize',8);
xlabel('\omega','Fontsize',8);
axis([0 20 0 tao/T]);
%end
```

在周期矩形脉冲的脉冲宽度 τ 保持不变时，通过输入不同的周期 T，可以观察到频谱与周期之间的关系。图 6-8 给出了四种输入情况下的频谱图：T＝4*pi,tao＝pi,m＝40；T＝8*pi,tao＝pi,m＝80；T＝16*pi,tao＝pi,m＝160；T＝100*pi,tao＝pi,m＝1000。

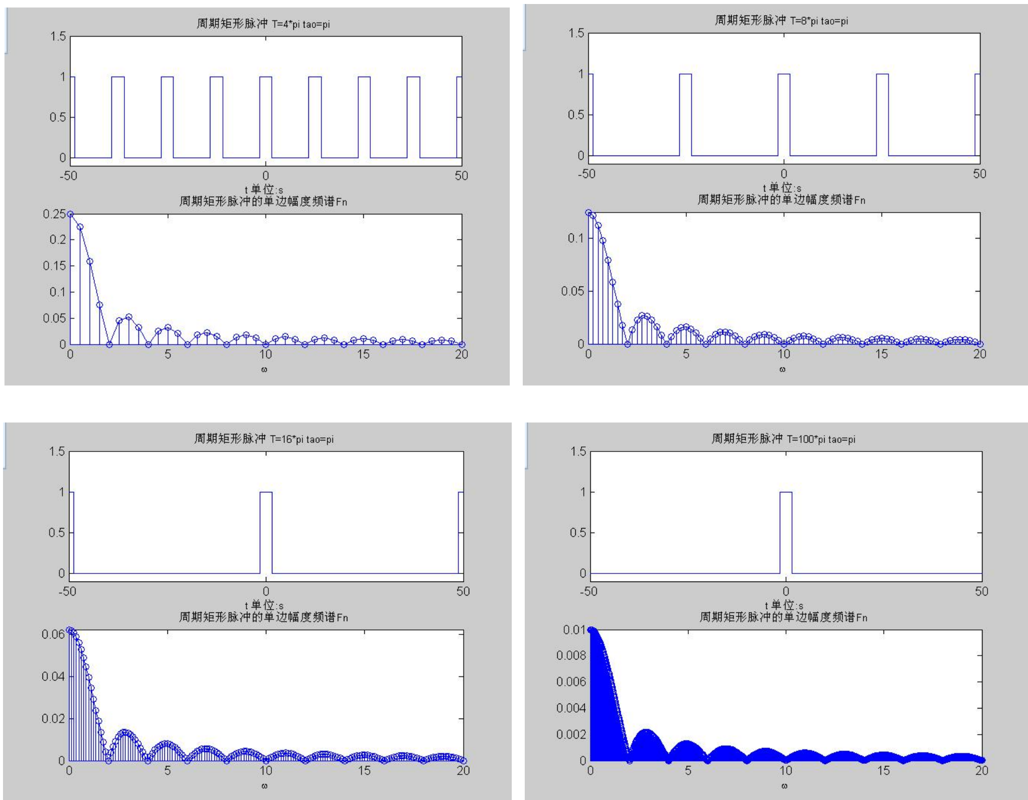


图 6-8　频谱与周期的关系

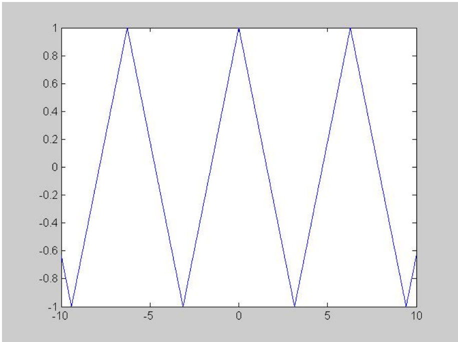
由图 6-8 可见，脉冲宽度 τ 不变，频谱包络线第一个过零点 $\left(\frac{2\pi}{\tau}\right)$ 的频率不变。周期 T 改变时，周期 T 越大，谱线间隔越小，频谱变密。当周期 T 趋于无穷时（周期信号变成非周期信号），谱线间隔趋于零，周期信号的离散频谱就变成非周期信号的连续频谱。另外，随着周期 T 的增大，各谐波分量的幅度也相应减小。

三、实验内容

6.1　分析如题 6.1 图所示的周期锯齿波信号，其中 $T=2\pi$，试用 MATLAB：

① 求信号的傅里叶级数展开式。

② 绘出信号的时域波形及频谱图。



题 6.1 图

6.2　试用 MATLAB 分析周期全波余弦信号，其中 $T=2\pi$，振幅为 1。

① 求信号的傅里叶级数展开式。

② 绘出信号的时域波形及频谱图。

6.3　设计一个三角波合成实验，写出实验步骤。

四、实验要求

1. 在 MATLAB 中输入程序，并将实验结果存入指定存储区域。

2. 在实验报告中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。

实验七　连续时间信号与系统的频域分析

一、实验目的

掌握连续时间信号的傅里叶变换及傅里叶逆变换的实现方法，掌握连续时间系统的频域分析方法，熟悉 MATLAB 相应函数的调用格式和作用，掌握使用 MATLAB 来分析连续时间信号与系统的频域特性及绘制信号频谱图的方法。

二、实验原理

（一）连续时间信号与系统的频域分析原理

1. 连续时间信号的频域分析

连续时间信号的傅里叶变换

$$F(\mathrm{j}\omega)=\int_{-\infty}^{\infty}f(t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t$$

傅里叶逆变换

$$f(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}F(\mathrm{j}\omega)\mathrm{e}^{\mathrm{j}\omega t}\mathrm{d}\omega$$

$F(\mathrm{j}\omega)$ 称为频谱密度函数，简称频谱。一般是复函数，可记为

$$F(\mathrm{j}\omega)=|F(\mathrm{j}\omega)|\mathrm{e}^{\mathrm{j}\varphi(\omega)}$$

$|F(\mathrm{j}\omega)|$ 反映信号各频率分量的幅度随频率 ω 的变化情况，称为信号幅度频谱。$\varphi(\omega)$ 反映信号各频率分量的相位随频率 ω 的变化情况，称为信号相位频谱。

2. 连续时间系统的频域分析

在 n 阶系统情况下，数学模型为

$$
\begin{aligned}
&a_n \frac{\mathrm{d}^n y(t)}{\mathrm{d}t^n}+a_{n-1}\frac{\mathrm{d}^{n-1}y(t)}{\mathrm{d}t^{n-1}}+\cdots+a_1\frac{\mathrm{d}y(t)}{\mathrm{d}t}+a_0 y(t)\\
&=b_m\frac{\mathrm{d}^m f(t)}{\mathrm{d}t^m}+b_{m-1}\frac{\mathrm{d}^{m-1}f(t)}{\mathrm{d}t^{m-1}}+\cdots+b_1\frac{\mathrm{d}f(t)}{\mathrm{d}t}+b_0 f(t)
\end{aligned}
$$

令初始条件为零，两端取傅里叶变换，得

$$
\begin{aligned}
&\left[a_n(\mathrm{j}\omega)^n+a_{n-1}(\mathrm{j}\omega)^{n-1}+\cdots+a_1\mathrm{j}\omega+a_0\right]Y(\mathrm{j}\omega)\\
&=\left[b_m(\mathrm{j}\omega)^m+b_{m-1}(\mathrm{j}\omega)^{m-1}+\cdots+b_1\mathrm{j}\omega+b_0\right]F(\mathrm{j}\omega)
\end{aligned}
$$

表示为
$$\sum_{k=0}^{n}a_k(\mathrm{j}\omega)^k Y(\mathrm{j}\omega)=\sum_{k=0}^{m}b_k(\mathrm{j}\omega)^k F(\mathrm{j}\omega)$$

则
$$H(\mathrm{j}\omega)=\frac{Y(\mathrm{j}\omega)}{F(\mathrm{j}\omega)}=\frac{b_m(\mathrm{j}\omega)^m+b_{m-1}(\mathrm{j}\omega)^{m-1}+\cdots+b_1\mathrm{j}\omega+b_0}{a_n(\mathrm{j}\omega)^n+a_{n-1}(\mathrm{j}\omega)^{n-1}+\cdots+a_1\mathrm{j}\omega+a_0}=\frac{\sum_{k=0}^{m}b_k(\mathrm{j}\omega)^k}{\sum_{k=0}^{n}a_k(\mathrm{j}\omega)^k}$$

系统响应的频域分析指求任意输入 $f(t)$ 情况下的零状态响应 $y(t)$ 时，先计算 $y(t)$ 的傅里叶变换 $Y(\mathrm{j}\omega)$，然后再求傅里叶逆变换可得到 $y(t)$。

$$Y(\mathrm{j}\omega)=H(\mathrm{j}\omega)F(\mathrm{j}\omega)$$

式中，$F(\mathrm{j}\omega)$ 是输入信号 $f(t)$ 的傅里叶变换，$Y(\mathrm{j}\omega)$ 是输出信号 $y(t)$ 的傅里叶变换。即输出信号频谱 $Y(\mathrm{j}\omega)$ 等于 $H(\mathrm{j}\omega)$ 与输入信号的频谱 $F(\mathrm{j}\omega)$ 的乘积。由于此式是频率变量 ω 的函数，所以称之为系统的频域表示法。

3. 系统传递函数

系统传递函数定义为

$$H(\mathrm{j}\omega)=\frac{Y(\mathrm{j}\omega)}{F(\mathrm{j}\omega)}$$

系统传递函数反映了系统内在的固有的特性，它取决于系统自身的结构及参数，与外部激励无关，是描述系统特性的一个重要参数。

$$H(\mathrm{j}\omega)=|H(\mathrm{j}\omega)|\mathrm{e}^{\mathrm{j}\varphi(\omega)}$$

式中，$|H(\mathrm{j}\omega)|$ 随 ω 变化的规律称为系统的幅频特性，$\varphi(\omega)$ 随 ω 变化的规律称为系统的相频特性。频率特性不仅可以用函数表达式表示，还可用随频率 ω 变化的曲线来描述。

（二）MATLAB 函数介绍

MATLAB 的 Symbolic Math Toolbox 提供了能直接求解傅里叶变换及其逆变换的函数 fourier 和 ifourier。

1. fourier 函数

fourier 函数求解信号的傅里叶变换。

调用格式为：

① F＝fourier(f)：是符号函数 f 的傅里叶变换，输入参量 f 是关于符号变量 t 的符号表达式，输出参量 F 为返回符号表达式 f 的关于 ω 的傅里叶变换表达式。如果 f＝f(ω)，则 fourier 函数返回关于 t 的函数。

② F＝fourier(f,v)：返回函数 F 是关于符号变量 v 的符号表达式，而不是默认的ω，即

$$F(v)=\int_{-\infty}^{\infty} f(x)\mathrm{e}^{-\mathrm{j}vx}\mathrm{d}x$$

③ F＝fourier(f,u,v)：输入参量 f 是关于符号变量 u 的符号表达式，输出参量 F 是关于 v 的符号表达式，即

$$F(v)=\int_{-\infty}^{\infty} f(u)\mathrm{e}^{-\mathrm{j}vu}\mathrm{d}u$$

2. ifourier 函数

ifourier 函数求解信号的傅里叶逆变换。

调用格式为：

① f＝ifourier(F)：是函数 F 的傅里叶逆变换，输入参量 F 默认为符号变量ω的函数，输出参量 f 默认为符号变量 x 的函数。如果 f＝f(ω)，则 fourier 函数返回关于 t 的函数。

② f＝ifourier(F,u)：输入参量 F 默认为符号变量ω的函数，输出参量 f 为指定的符号变量 u 的函数。

③ f＝ifourier(F,v,u)：输入参量 F 为指定的符号变量 v 的函数，输出参量 f 为指定的符号变量 u 的函数。

注意，在调用函数 fourier 和 ifourier 之前，需用 syms 命令对所用到的变量如 t、u、v、ω 等进行说明，即要将这些变量说明成符号变量；采用 fourier 和 ifourier 得到的返回函数，仍然是符号表达式，如需对返回的函数作图，则应用 ezplot 绘图命令而不是用 plot 命令。如果返回函数中含有诸如狄拉克函数δ(t)等项，用 ezplot 也无法作图。

3. quadl 函数

quadl 函数是 MATLAB 中计算数值积分的函数。利用 quad8 和 quadl 可以计算非周期连续信号的频谱。

调用格式为：

q＝quadl(fun,a,b)

q＝quadl(fun,a,b,TOL,TRACE,p1,p2,...)

fun 是一个字符串，表示被积函数的 M 文件名；a、b 分别表示定积分的下限和上限；TOL 表示指定允许的相对或绝对积分误差，非零的 TRACE 表示以被积函数的点绘图方式来跟踪该 quadl 函数生成的返回值，如果 TOL 和 TRACE 都为空矩阵即“[]”，则两者均自动使用缺省值；“p1,p2,...”表示被积函数所需的多个额外输入参数。

例如，fun 可以指定为两种形式：

第一种形式为 inline 对象:

```
F＝inline('1./(x.^3-2*x-5)');
Q＝quadl(F,0,2);
```

运行结果为：

```
Q＝-0.4605
```

第二种形式为：

```
Q＝quadl(@myfun,0,2);
```

这里符号“@”表示取函数的句柄，myfun.m 是一个 M 文件，内容如下：

```
function y＝myfun(x)
y＝1./(x.^3-2*x-5);
```

运行结果为：

```
Q＝-0.4605
```

4. freqs 函数

freqs 函数用来实现连续时间系统函数 $H(\mathrm{j}\omega)$ 的频域特性分析。该函数可以求出 $H(\mathrm{j}\omega)$ 的数值解，并可绘出系统的幅频及相频响应曲线。

设连续系统的系统函数为

$$H(\mathrm{j}\omega)=\frac{B(\mathrm{j}\omega)}{A(\mathrm{j}\omega)}=\frac{b_m(\mathrm{j}\omega)^m+b_{m-1}(\mathrm{j}\omega)^{m-1}+\cdots+b_1(\mathrm{j}\omega)+b_0}{a_n(\mathrm{j}\omega)^n+a_{n-1}(\mathrm{j}\omega)^{n-1}+\cdots+a_1(\mathrm{j}\omega)+a_0}$$

freqs 函数有四种调用格式：

① H＝freqs(B,A,W)：其中输入参量 B 为系统函数 $H(\mathrm{j}\omega)$ 分子多项式系数，对应于[b_m, b_{m-1}, b_{m-2},…,b_0]，A 为系统函数 $H(\mathrm{j}\omega)$ 分母多项式系数，对应于[a_n, a_{n-1}, a_{n-2},…,a_0]， W 为需要计算的 $H(\mathrm{j}\omega)$ 的频率抽样点向量，形如 W1:P:W2 的冒号运算为定义的频率范围，W1 为频率起始值，W2 为频率终止值，P 为频率采样间隔。输出参量 H 为返回在 W 所定义的频率点上的系统函数的样值。

② [H,W]＝freqs(B,A)：该调用格式输出从计算出的频率响应中自动选择 200 个频率点的频率响应的样值。输出参量 H 保存 200 个频率点的系统频率响应的样值，W 保存 200 个频率点的位置。

③ [H,W]＝freqs(B,A,N)：该调用格式表示自动选择 N 个频率点的来计算频率响应的样值。N 的缺省值为 200。

④ freqs(B,A)：该调用格式不返回系统频率响应的样值，而是以波特图的方式绘出系统的幅频及相频响应曲线。

（三）连续时间信号频域分析的 MATLAB 实现

例 7-1　试求出下列连续时间信号的傅里叶变换。

① 单位阶跃信号 $f(t)=\varepsilon(t)$；

② $f(t)=\dfrac{1}{2}\mathrm{e}^{-2t}\varepsilon(t)$；

③ 矩形脉冲信号 $g_{\tau}(t)=\begin{cases}1 & |t|<5\\ 0 & |t|>5\end{cases}$

解 程序如下：

①
```
syms t;
f＝sym('Heaviside(t)');
F＝fourier(f)
```
结果为：
```
F＝pi*Dirac(w)-i/w
```
②
```
syms t;
f＝1/2*exp(-2*t)*sym('Heaviside(t)');
F＝fourier(f)
```
结果为：
```
F＝1/2/(2+i*w)
```
③
```
syms t;
g＝sym('Heaviside(t+5)-Heaviside(t-5)');
G＝fourier(g)
simplify(G)                    %简化结果
```
结果为：
```
G＝exp(5*i*w)*(pi*Dirac(w)-i/w)-exp(-5*i*w)*(pi*Dirac(w)-i/w)
ans＝2*sin(5*w)/w
```
例 7-2 试绘出连续时间信号 $f(t)=\dfrac{1}{2}\mathrm{e}^{-2t}\varepsilon(t)$ 的时域波形 $f(t)$ 及相应的幅度频谱和相位频谱图。

解 参考程序如下：
```
clear;
syms t;
f＝1/2*exp(-2*t)*sym('Heaviside(t)');
subplot(3,1,1);
ezplot(f); ylabel('f(t)');title('单边指数信号');
axis([-0.1 2.5 0 0.6]);
F＝fourier(f);
subplot(3,1,2);
ezplot(abs(F));xlabel('\omega');ylabel('|F(j\omega)|');title('幅度频谱');
im＝imag(F);
re＝real(F);
phase＝atan(im/re);
subplot(3,1,3);
```

```
ezplot(phase); xlabel('\omega');ylabel('\angle F(j\omega)');title('相位频谱');
%end
```

程序绘制的信号时域波形及幅度频谱和相位频谱如图 7-1 所示。

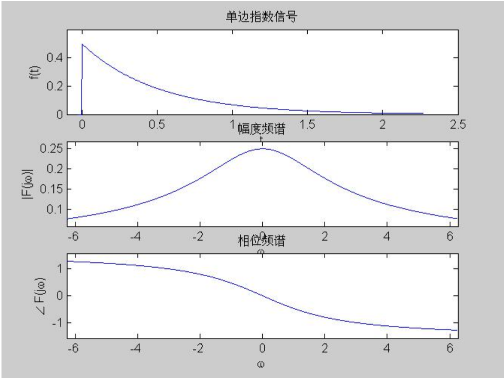


图 7-1　例 7-2 图

注意，Heaviside(t)函数即为单位阶跃函数 $\varepsilon(t)$。在使用 Heaviside(t)函数之前，要在当前工作目录下创建函数 Heaviside.m，且以 Heaviside.m 文件名保存。

例 7-3　试绘出矩形脉冲信号(门信号)的时域波形和相应的频谱图。

$$g_\tau(t)=\begin{cases}1 & |t|<\dfrac{\tau}{2}\\ 0 & |t|>\dfrac{\tau}{2}\end{cases}$$

解　绘制当 $\tau=1$ 时的矩形脉冲信号的时域波形和相应的频谱图。

方法一程序如下：

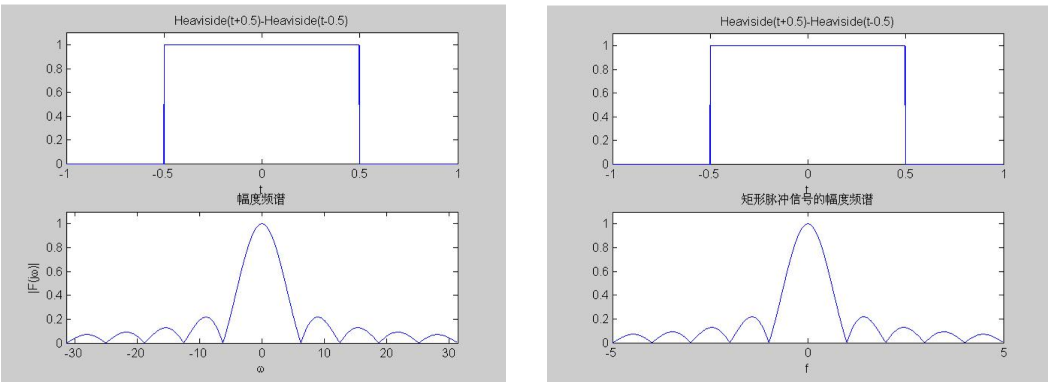
```
syms t w;
g＝sym('Heaviside(t+0.5)-Heaviside(t-0.5)');
subplot(2,1,1);
ezplot(g,[-1 1]);                    %
hold on;
axis([-1 1 0 1.1]);
plot([-0.5 -0.5],[ 0 1]);
plot([0.5 0.5],[ 0 1]);
gw＝fourier(g,t,w);
ggw＝maple('convert',gw,'piecewise');
subplot(2,1,2);
ezplot(abs(ggw),[-10*pi 10*pi ]);
xlabel('\omega');ylabel('|F(j\omega)|');title('幅度频谱');
axis([-10*pi 10*pi 0 1.1]);
%end
```

运行结果如图 7-2(a)所示。

方法二程序如下：

```
syms t w;
g＝sym('Heaviside(t+0.5)-Heaviside(t-0.5)');
subplot(2,1,1);
ezplot(g,[-1 1]);
hold on;
axis([-1 1 0 1.1]);
plot([-0.5 -0.5],[ 0 1]);
plot([0.5 0.5],[ 0 1]);
gw＝fourier(g,t,w);
Gw＝simplify(gw);
Gf＝subs(Gw,'2*pi*f','w');
Gfconj＝conj(Gf);
F＝sqrt(Gf*Gfconj);
subplot(2,1,2);
ezplot(F,[-5 5 ]);
axis([-5 5 0 1.1]);
title('矩形脉冲信号的幅度频谱');
%end
```

运行结果如图 7-2(b)所示。



(a) (b)

图 7-2 例 7-3 图

例 7-4 已知信号 $f(t)$ 的傅里叶变换为 $F(\mathrm{j}\omega)=\dfrac{4}{4+\omega^2}$，试绘出该信号的时域波形和相应的频谱图。

解 程序如下：

```
clear;
syms t w;
```

```
F=4/(4+w^2);
subplot(2,2,1);ezplot(F);
f=ifourier(F,t);
subplot(2,2,2);ezplot(f);title('f(t)');
subplot(2,2,3);ezplot(abs(F));
xlabel('\omega');ylabel('|F(j\omega)|');title('幅度频谱');
im=imag(F);
re=real(F);
phase=atan(im/re);
subplot(2,2,4);ezplot(phase);
xlabel('\omega');ylabel('\angle F(j\omega)');title('相位频谱');
%end
```

程序绘制的信号时域波形及相应的频谱图如图 7-4 所示。

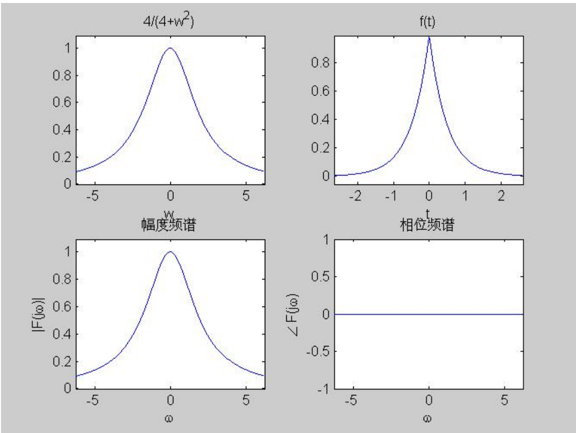


图 7-4　例 7-4 图

（四）连续时间信号的傅里叶变换的数值计算方法

由于有些连续信号不能表示成符号表达式，不能用 fourier 函数进行傅里叶变换，而且很多实际连续信号是现场测量的离散数值量。对这些情况，可以利用 MATLAB 的数值计算功能，用数值方法求信号的傅里叶变换。连续时间信号的傅里叶变换的数值计算方法的理论依据如下。

连续时间信号的傅里叶变换可以表示为：

$$F(\mathrm{j}\omega)=\int_{-\infty}^{\infty}f(t)\mathrm{e}^{-\mathrm{j}\omega t}\mathrm{d}t=\lim_{\tau\to 0}\sum_{n=-\infty}^{\infty}f(n\tau)\mathrm{e}^{-\mathrm{j}\omega n\tau}\tau \qquad (7\text{-}1)$$

当 τ 足够小时，对大多数信号上式的近似情况可以满足实际需要。

当 $f(t)$ 为时限信号时，或可以近似看作时限信号时，式（7-1）中的 n 值可认作是有限的，设为 $0\leqslant n\leqslant M$，则式（7-1）为

$$F(\mathrm{j}\omega)=\tau\sum_{n=0}^{M}f(n\tau)e^{-\mathrm{j}\omega n\tau} \qquad （7\text{-}2）$$

对式（7-2）中的角频率 ω 进行离散化，得到 N 个角频率采样点，即

$$\omega_k=\frac{2\pi \mathrm{k}}{N\tau} \qquad 0\leqslant k\leqslant N-1$$

则得到离散的傅里叶变换为

$$F(\mathrm{k})=\tau\sum_{n=0}^{M}f(n\tau)\mathrm{e}^{-\mathrm{j}\omega_{\mathrm{k}}n\tau}$$

算法步骤为：

① 生成 $f(t)$的 M 个样本 $f(n\tau)$， $0\leqslant n\leqslant M$ ；

② 对 ω 离散化，得到 $\mathrm{e}^{-\mathrm{j}\omega_k n\tau}$， $0\leqslant k\leqslant N-1$， $0\leqslant n\leqslant M$ ；

③ 将 $f(n\tau)$ 和 $\mathrm{e}^{-\mathrm{j}\omega_k n\tau}$ 进行内积，得到离散傅里叶变换 $F(k)$；

④ 将 $F(k)$的各个样值连接起来，得到 $F(\mathrm{j}\omega)$ 的近似表示。

例 7-5 试求信号 $\mathrm{e}^{-2t}u(t)$ 的傅里叶变换，并绘出该信号的时域波形和相应的傅里叶频谱图。

解 程序如下：

```
%连续时间信号的傅里叶变换的数值计算
dt＝0.015;
t＝0:dt:2;
f＝exp(-2*t);
N＝2000;
k＝-N:N;
W＝2*pi*k/(N*dt);
F＝f*exp(-j*t'*W)*dt;
F＝real(F);
subplot(211);
plot(t,f);
subplot(212);
plot(W,F);
axis([-6 6 0 0.6]);
%end
```

运行结果如图 7-4 所示。

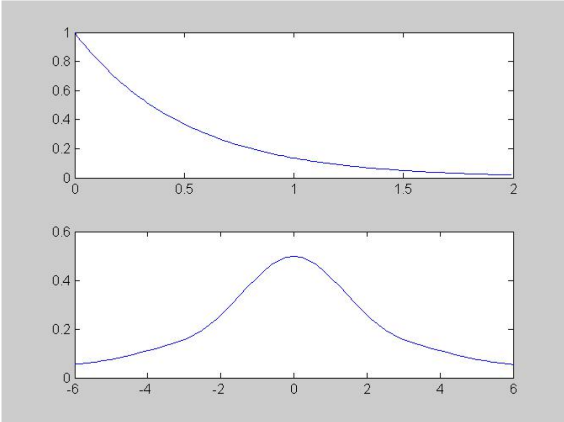


图 7-4　例 7-5 图（1）

MATLAB 还提供了很多数值计算的工具，可以用来进行信号的频域分析。quad8 和 quadl 是 MATLAB 中计算数值积分的函数。利用 quad8 和 quadl 可以计算非周期连续信号的频谱。

对例 7-5 中函数 $e^{-2t}u(t)$ 利用 quadl 函数重新计算单边指数信号的频谱。定义如下 MATLAB 函数：

```
function y＝dexp(t,w)
y＝(0<＝t<＝5).*exp(-2*t).*exp(-j*w*t);
```

函数 dexp 将计算傅里叶变换中被积函数的值。其中(0<＝t<＝5).*exp(-2*t)是单边指数信号的 MATLAB 表示。注意将上面的函数 dexp 用文件名 dexp.m 保存。近似计算非周期连续信号的频谱的程序如下：

```
% 用 quadl 函数计算非周期连续信号的频谱
w＝linspace(-6*pi,6*pi,512);
X＝1./(2+j*w);
N＝length(w);
F＝zeros(1,N);
for k＝1:N;
    F(k)＝quadl(@dexp,0,5,[],[],w(k));
end
subplot(311); f＝inline('(t>＝0).*exp(-2*t)');t＝0:0.1:5;
plot(t, f(t)); xlabel('t');ylabel('f(t)');title('单边指数信号');
subplot(312);plot(w,abs(F),'k',w,(abs(X)), 'r');
xlabel('\omega');ylabel('|F(j\omega)|');title('幅度频谱');
subplot(313);plot(w,abs(F)-abs(X), 'r');
xlabel('\omega');ylabel('error');title('误差');
%end
```

运行结果如图 7-5 所示。

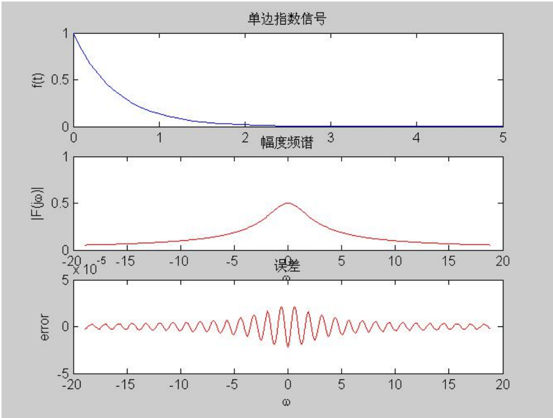


图 7-5　例 7-5 图（2）

由图 7-4、图 7-5 可知，用 quadl 函数计算非周期连续信号的频谱和理论值两者的计算误差非常小（10^{-5}数量级），说明 quadl 函数具有很高的数值求解精确度。

（五）连续时间系统频域分析的 MATLAB 实现

MATLAB 提供了 freqs 函数来实现连续时间系统函数 $H(\mathrm{j}\omega)$ 的频域特性分析。该函数可以求出 $H(\mathrm{j}\omega)$ 的数值解，并可绘出系统的幅频及相频响应曲线。

例 7-6　三阶归一化的 Butterworth 低通滤波器的频率响应为

$$H(\mathrm{j}\omega)=\frac{1}{(\mathrm{j}\omega)^3+2(\mathrm{j}\omega)^2+2\mathrm{j}\omega+1}$$

利用 MATLAB 绘出该系统的幅频响应和相频响应。

解　程序如下：

```
w=linspace(0,5,200);                %生成有 200 个元素的线性分布行向量
b=[1];
a=[1 2 2 1];
[h,w]=freqs(b,a,w);
subplot(2,1,1);
plot(w,abs(h));
set(gca,'xtick',[0 1 2 3 4 5]);
set(gca,'ytick',[0 0.4 0.707 1]);
xlabel('角频率(\omega)rad/s');
ylabel('幅度|H(j\omega)|');grid on;
subplot(2,1,2);
plot(w,angle(h));
set(gca,'xtick',[0 1 2 3 4 5]);
xlabel('角频率(\omega)rad/s');
ylabel('相位\phi(rad)');grid on;
```

运行结果如图 7-6 所示。

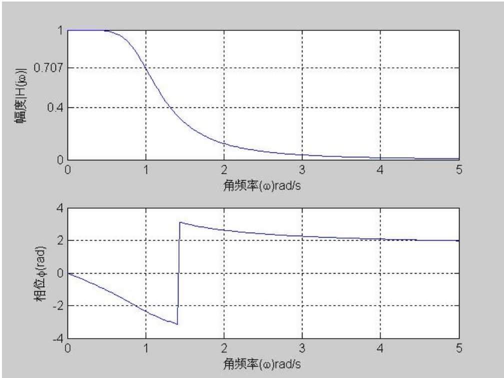


图 7-6　例 7-6 图

例 7-7　如图 7-7 所示，常见的用 R、L、C 元件构造的二阶高通滤波器，用 MATLAB 求其系统函数并绘制幅度响应和相位响应曲线。假设 $L = 0.4H$ ，$C = 0.05F$ ，$R = 2\Omega$ 。

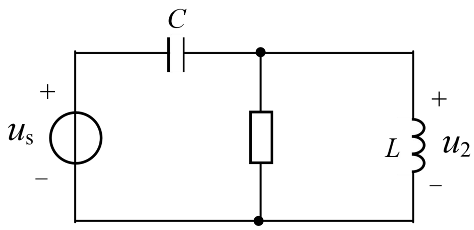


图 7-7　例 7-7 图（1）

解　电路的系统函数为

$$H(\mathrm{j}\omega) = \frac{U_2(\mathrm{j}\omega)}{U_s(\mathrm{j}\omega)} = \frac{RLC(\mathrm{j}\omega)^2}{RLC(\mathrm{j}\omega)^2 + \mathrm{j}\omega L + R} = \frac{0.04(\mathrm{j}\omega)^2}{0.04(\mathrm{j}\omega)^2 + 0.4\mathrm{j}\omega + 2}$$

截止频率为

$$\omega_c = \frac{1}{\sqrt{LC}} = 7.0711$$

$$|H(\mathrm{j}\omega)|_{\omega=\omega_c} = \frac{1}{\sqrt{2}} |H(\mathrm{j}\omega)|_{\omega=0} = \frac{1}{\sqrt{2}} \approx 0.707$$

实现以上分析的 MATLAB 程序如下：

```
%二阶高通滤波器的频率特性
b＝[0.04 0 0];
a＝[0.04 0.4 2];
[h,w]＝freqs(b,a,100);
h1＝abs(h);
h2＝angle(h);
```

```
subplot(211);
plot(w,h1);
hold on
plot([7.0711 7.0711], [0 0.707],':');
plot([0 7.0711], [0.707 0.707], ':');
axis([0,40,0,1.1]);
grid
xlabel('角频率(\omega)');
ylabel('幅度');
title('H(j\omega)的幅频特性');
subplot(212);
plot(w,h2*180/pi);
axis([0,40,0,200]);
grid
xlabel('角频率(\omega)');
ylabel('相位(度) ');
title('H(j\omega)的相频特性');
%end
```

程序运行结果如图 7-8 所示。

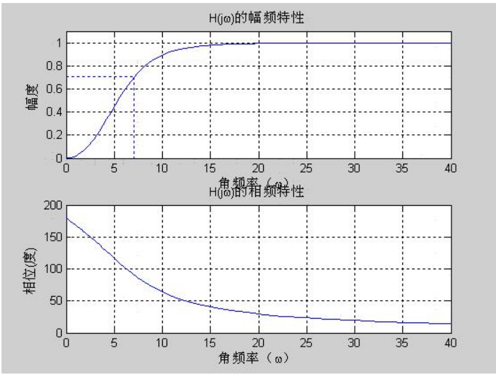


图 7-8　例 7-7 图（2）

由图 7-8 可以看出，当 ω 从 0 增大时，该高通滤波器的幅度从 0 开始上升，当 $\omega=\omega_c=7.0711$ 时，幅度等于 0.707，$\omega>\omega_c$ 后进入通带。

三、实验内容

7.1　已知某一连续时间信号为 $f(t)=\mathrm{e}^{-2|t|}$，试绘出它的时域波形及相应的频谱图。

7.2　若信号 $f(t)$ 的傅里叶变换 $F(\mathrm{j}\omega)=\tau\mathrm{Sa}\dfrac{\omega\tau}{2}$，试绘出该信号的时域波形和相应的频谱图。

7.3　已知 R、L、C 元件构造的二阶低通滤波器电路的系统函数为

$$H(\mathrm{j}\omega)=\frac{1}{0.08(\mathrm{j}\omega)^2+0.4\mathrm{j}\omega+1}$$

试用 MATLAB 绘制幅度响应和相位响应曲线，并分析该系统的频率特性。

7.4　已知电路的系统函数为

$$H(\mathrm{j}\omega)=\frac{(\mathrm{j}\omega)^2+22500}{(\mathrm{j}\omega)^2+200\mathrm{j}\omega+20000}$$

试用 MATLAB 绘制幅度响应和相位响应曲线，并分析该系统的频率特性。

7.5　求出门函数和抽样信号函数的傅里叶变换，观察并验证傅里叶变换的对称性质。

7.6　求出抽样信号函数的傅里叶变换，观察并验证傅里叶变换的尺度变换性质。

四、实验要求

1. 在 MATLAB 中输入程序，验证实验结果，并将实验结果存入指定存储区域。

2. 在实验报告中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，写出自编程序，并给出实验结果，总结实验得出的主要结论。

实验九　连续时间信号与系统的复频域分析

一、实验目的

掌握使用 MATLAB 软件来实现连续时间信号拉普拉斯变换和拉普拉斯逆变换的方法，掌握连续时间系统的复频域分析方法，熟悉 MATLAB 相应函数的调用格式和作用，掌握使用 MATLAB 来分析连续时间信号与系统的复频域特性，实现连续系统的零、极点分析、稳定性分析及系统传递函数的求解。

二、实验原理

（一）连续时间信号与系统的复频域分析方法

拉普拉斯变换是分析连续时间信号的有效手段，对于当 $t \to \infty$ 时信号幅度不衰减或增长的时间信号，其傅里叶变换不存在，但可以用拉普拉斯变换来分析它们。

1. 拉普拉斯变换

$$F(s)=\int_{-\infty}^{\infty} f(t)\,\mathrm{e}^{-st}\mathrm{d}t$$

其中 $s=\alpha+\mathrm{j}\omega$，若以 α 为横坐标（实轴），$\mathrm{j}\omega$ 为纵坐标（虚轴），复变量 s 就构成了一个复平面，称为 s 平面。$F(s)$ 是复变量 s 的复函数，为了便于理解和分析 $F(s)$ 随 s 的变化规律，将 $F(s)$ 写成：$F(s)=|F(s)|\mathrm{e}^{\mathrm{j}\varphi(s)}$，其中 $|F(s)|$ 为复信号 $F(s)$ 的模，而 $\varphi(s)$ 为 $F(s)$ 的相角。

选择 0_ 系统，单边拉普拉斯变换为

$$F(s)=\int_{0^-}^{\infty} f(t)\,\mathrm{e}^{-st}\mathrm{d}t$$

2. 拉普拉斯逆变换

$$f(t)=\frac{1}{2\pi\mathrm{j}}\int_{\sigma-\mathrm{j}\infty}^{\sigma+\mathrm{j}\infty} F(s)\mathrm{e}^{st}\mathrm{d}s$$

利用拉普拉斯逆变换的定义式求拉普拉斯逆变换比较复杂。如果象函数 $F(s)$ 是有理函数，

即 s 的多项式分式，可以通过部分分式展开法将它们展开成部分分式之和，对各分式进行拉普拉斯逆变换，求出 $f(t)$。

3. 系统的复频域分析

对系统的微分方程两边取拉普拉斯变换，代入初始条件和输入信号的拉普拉斯变换，可以直接求得系统的全响应；或根据系统传递函数 $H(s)=\dfrac{Y(s)}{F(s)}$，计算任意输入 $f(t)$ 的零状态响应 $y(t)$，先计算 $y(t)$ 的拉普拉斯变换 $Y(s)$，即 $Y(s)=H(s)F(s)$，然后再求拉普拉斯逆变换可得到 $y(t)$，零输入响应按时域方法求得。

4. 零、极点与单位冲激响应和稳定性

如果系统传递函数的所有极点满足 $\mathrm{Re}(P_i)=\sigma<0$，则单位冲激响应 $h(t)$ 为衰减函数，系统稳定。若 $\mathrm{Re}(P_i)=\sigma>0$ (只要有一个极点)，则 $h(t)$ 为增长函数，系统不稳定；或若 $\mathrm{Re}(P_i)=\sigma=0$，且在虚轴上有二阶（或以上）极点，系统不稳定。若 $\mathrm{Re}(P_i)=\sigma=0$ (但不能有重极点)，则 $h(t)$ 为非零数值或等幅振荡，系统临界稳定。

（二）MATLAB 函数介绍

1. laplace 函数

laplace 函数用来求解信号的拉普拉斯变换。

调用 laplace 函数的命令格式为：

① L＝laplace(F)：输入参量 F 为连续时间信号的符号表达式，默认为变量 t 的函数，输出参量 L 为默认符号变量 s 的关于 F 的拉普拉斯变换的符号表达式。

② L＝laplace(F,v)：输入参量 F 为连续时间信号的符号表达式，默认为变量 t 的函数，输出参量 L 为符号变量 v 的函数。

2. ilaplace 函数

ilaplace 函数用来求解信号的拉普拉斯逆变换。

调用 ilaplace 函数的命令格式为：

① F＝ilaplace(L)：输入参量 L 为默认符号变量 s 的符号表达式，输出参量 F 为默认符号变量 t 的关于 L 的拉普拉斯逆变换的符号表达式。

② F＝ilaplace(L,w)：输入参量 L 为默认符号变量 s 的符号表达式。输出参量 F 为符号变量 w 的关于 L 的拉普拉斯逆变换的符号表达式。

3. residue 函数

residue 函数用来求部分分式展开系数。

调用 residue 函数的格式为：

[k,p,c]＝residue(num,den)

其中输入参量 num 为 $F(s)$ 的分子多项式系数构成的行向量，den 为 $F(s)$ 的分母多项式系数构成的行向量。函数将返回三个输出参量 k，p，c，其中 p 为包含 $F(s)$ 所有极点位置的列向量，k 为包含 $F(s)$ 所有部分分式展开系数 k_i 的列向量，c 为包含 $F(s)$ 部分分式展开的多项式的系数 c_j 的列向量，若为真分式(m<n)，则 c 返回为空阵。

4. pole 函数

pole 函数用于计算 LTI 模型的极点。

调用格式为：

p＝pole (sys)：计算 LTI 模型 sys 的极点，p 是一个列向量。sys 是根据系统函数的分子和分母多项式的系数调用 tf 函数生成的系统函数对象。

5. zero 函数

zero 函数用于计算 LTI 系统的零点。

调用格式为：

z＝zero (sys)：返回 LTI 模型 sys 的零点。

6. roots 函数

roots 函数用于计算多项式的根。

① 当计算系统函数的零点时，调用格式为：

z＝roots (num)

② 当计算系统函数的极点时，调用格式为：

p＝roots (den)

7. pzmap 函数

pzmap 函数用于绘制系统函数零、极点分布图和计算系统零、极点。

调用格式如下：

① pzmap(sys)：直接绘制出系统函数零、极点分布图。

② [p,z]＝pzmap(sys)：输出参量 p 和 z 为系统函数零、极点的向量，并不绘制系统函数零、极点分布图。

（三）连续时间信号的复频域分析的 MATLAB 实现

MATLAB 的 Symbolic Math Toolbox 提供了能直接求解拉普拉斯变换及其逆变换的函数 laplace 和 ilaplace。

1. 拉普拉斯变换的 MATLAB 实现

如果连续时间信号 $f(t)$ 可用符号表达式表示，则可直接调用 MATLAB 的函数来实现 $f(t)$

的单边拉普拉斯变换。

例 9-1 已知单边衰减正弦振荡信号 $f(t)=\mathrm{e}^{-t}\sin(\pi t)u(t)$，调用 laplace 函数计算其拉普拉斯变换。

解 程序如下：

```
% 计算连续时间信号拉普拉斯变换
syms t;
F＝exp(-t)*sin(pi*t);
L＝laplace(F)
```

运行结果为：

```
L＝pi/((s+1)^2+pi^2)
```

2. 拉普拉斯逆变换的 MATLAB 实现

（1）可直接调用专用的符号函数 ilaplace 函数来实现拉普拉斯逆变换

例 9-2 已知某连续系统的系统函数 $H(s)=\dfrac{9s^3}{(s+1)(s^2+2s+10)}$，调用 ilaplace 函数计算其拉普拉斯反变换。

解 程序如下：

```
% 计算连续时间信号拉普拉斯变换
syms s;
H＝(9*s^3)/((s+1)* (s^2+2*s+10));
h＝ilaplace(H)
```

运行结果为：

```
h＝9*Dirac(t)-exp(-t)-26*exp(-t)*cos(3*t)-18*exp(-t)*sin(3*t)
```

即 $$h(t)=9\delta(t)-\mathrm{e}^{-t}u(t)+31.622\,8\mathrm{e}^{-t}\cos(3t+0.8072\,\pi)u(t)$$

（2）利用 MATLAB 求部分分式展开及拉普拉斯逆变换

设连续函数信号 $f(t)$ 的拉普拉斯变换为

$$F(s)=\frac{N(s)}{D(s)}=\frac{\sum\limits_{j=0}^{m}a_j s^j}{\prod\limits_{i=1}^{n}(s-p_i)}=\sum_{i=1}^{n}\frac{k_i}{s-p_i}+\sum_{j=0}^{m-n}c_j s^j$$

通过 residue 函数可直接求出 $F(s)$ 部分分式展开的系数 k_i、极点 p_i 及多项式系数 c_j，然后便可根据公式求出 $F(s)$ 的拉普拉斯逆变换 $f(t)$。

例 9-3 已知某连续系统的系统函数如下

$$H(s)=\frac{s-2}{s(s+1)^3}$$

试利用 MATLAB 对其进行部分分式展开，求拉普拉斯逆变换。

解 $H(s)$ 的分母不是多项式，可以利用 conv 函数先将其现在的因子相乘的形式转换为多项式的形式，然后调用 residue 函数求出系统函数 $H(s)$ 部分分式展开的系数及极点。对应的程序如下：

```
num=[1 -2];
a=conv([1 0],[1 1]);
b=conv([1 1],[1 1]);
den=conv(a,b);
[k,p,c]=residue(num,den)
```

运行结果为：

```
k'=2.0000      2.0000      3.0000       -2.0000
p'=-1.0000    -1.0000    -1.0000        0
c=[]
```

$H(s)$ 的部分分式展开为：

$$H(s)=\frac{2}{s+1}+\frac{2}{(s+1)^2}+\frac{3}{(s+1)^3}+\frac{-2}{s}$$

则 $H(s)$ 的反变换为：

$$h(t)=(2\mathrm{e}^{-t}+2t\mathrm{e}^{-t}+1.5t^2\mathrm{e}^{-t}-2)u(t)$$

注意，第一个出现的系数 2 对应 $H(s)$的极点 $p=-1$ 的 k1/(s+1)项，第二个出现的系数 2 对应极点 $p=-1$ 的 k2/(s+1)^2 项，第三个出现的系数 3 对应极点 $p=-1$ 的 k3/(s+1)^3 项，第四个出现的系数－2 对应极点 $p=0$ 的 k4/s 项。

例 9-4 已知某连续系统的系统函数如下

$$H(s)=\frac{9s^3}{(s+1)(s^2+2s+10)}$$

试利用 MATLAB 对其进行部分分式展开，求系统的单位冲激响应 $h(t)$，绘出其时域波形，并根据系统函数零、极点分布判断系统的稳定性。

解 对应的程序如下：

```
num=[9 0 0 0];
den=conv([1 1],[1 2 10]);
[k,p,c]=residue(num,den)
```

运行结果为：

```
k'=-13.0000 + 9.0000i    -13.0000 - 9.0000i    -1.0000
p'=-1.0000 + 3.0000i        -1.0000 - 3.0000i      -1.0000
c=9
```

由结果可以看出，系统函数部分分式展开的多项式的系数 $c=9$，有 3 个极点，其中有一对共轭极点 $p_{1,2}=-1\pm\mathrm{j}3$，一个实极点 $p_3=-1$。$H(s)$ 的部分分式展开为

$$H(s)=9+\frac{-1}{s+1}+\frac{-13+\mathrm{j}9}{s+1-\mathrm{j}3}+\frac{-13-\mathrm{j}9}{s+1+\mathrm{j}3}$$

由于系数 k 中有一对共轭复数，为了得到简洁的时域表达式，需调用 abs 函数和 angle 函数将共轭复数表示成模和辐角形式。求出共轭极点对应的部分分式展开项系数的模和辐角的程序如下：

```
r＝abs(k)
w＝angle(k)/pi        % angle 产生的值是以弧度计的复数相位
```

运行结果为：

```
r'＝15.8114    15.8114     1.0000
w'＝0.8072     -0.8072     1.0000
```

$H(s)$ 的部分分式展开为：

$$H(s)=9+\frac{-1}{s+1}+\frac{15.811\,4\mathrm{e}^{\mathrm{j}0.8072\pi}}{s+1-\mathrm{j}3}+\frac{15.811\,4\mathrm{e}^{-\mathrm{j}0.8072\pi}}{s+1+\mathrm{j}3}$$

则 $H(s)$ 的反变换为：

$$h(t)=9\delta(t)-\mathrm{e}^{-t}u(t)+31.6228\mathrm{e}^{-t}\cos(3t+0.8072\,\pi)u(t)$$

系统冲激响应 $h(t)$ 的时域波形可直接调用 impulse 函数来绘制，命令如下：

```
num＝[9 0 0 0];
den＝conv([1 1],[1 2 10]);
impulse(num,den);
```

上述命令绘制的系统冲激响应 $h(t)$ 的时域波形如图 9-1 所示。

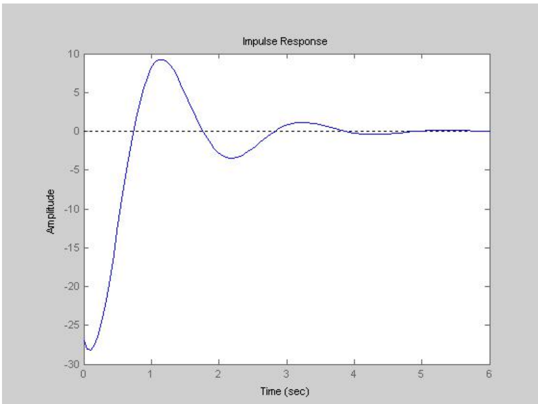


图 9-1　例 9-4 系统冲激响应的时域波形

由 residue 函数返回的结果可见，满足系统所有极点均位于 s 平面左半平面的条件，故该系统是稳定系统。另外，从系统的冲激响应曲线也可看出，$h(t)$ 是衰减振荡信号，满足绝对可积条件，也可得出该系统稳定的结论。

（四）连续时间系统的复频域分析的 MATLAB 实现

例 9-5　已知输入 $f(t)=\mathrm{e}^{-t}u(t)$，系统传递函数为 $H(s)=\dfrac{s+5}{s^2+5s+6}$，初始条件为 $y(0_-)=2$，

$y'(0_-)=1$，求系统全响应。

解 用拉普拉斯变换法求符号解。计算全响应的程序如下：

```
yx＝dsolve('D2y+5*Dy+6*y＝0','y(0)＝2','Dy(0)＝1','t')     %计算零输入响应 yx
syms t s;
H＝(s+5)/(s^2+5*s+6);
f＝exp(-t);
F＝laplace(f);
yf＝ilaplace(H*F)                              %计算零状态响应 yf
y＝yx+yf                                      %计算全响应 y
t＝(0:0.01:2)';
yx＝subs(yx);
yf＝subs(yf);
y＝subs(y);
plot(t,yx,'k-.', t,yf,'k--', t,y,'k');          %绘制响应波形
legend('零输入响应 yx', '零状态响应 yf', '全响应 y');
```

程序运行结果为：

```
yx＝
    -5*exp(-3*t)+7*exp(-2*t)
yf＝
    exp(-3*t)-3*exp(-2*t)+2*exp(-t)
y＝
    -4*exp(-3*t)+4*exp(-2*t)+2*exp(-t)
```

与计算结果比较，答案是一致的。响应的波形如图 9-2 所示。

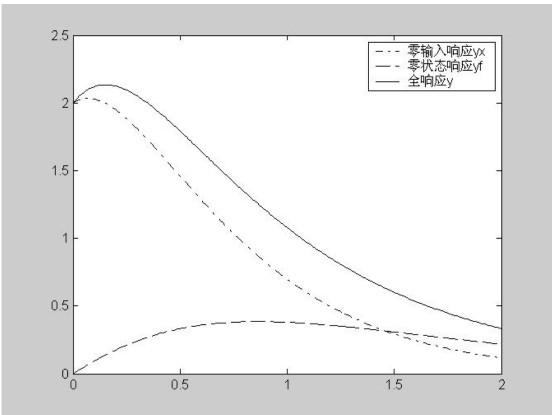


图 9-2　例 9-5 系统响应的时域波形

（五）连续时间系统的复频域特性的 MATLAB 实现

1. 系统函数零、极点分析

MATLAB 的控制系统工具箱(control system toolbox)为用户实现系统函数零、极点分析提供了丰富的函数。设连续系统的系统函数为

$$H(s)=\frac{N(s)}{D(s)}=k\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

即系统函数的分子和分母多项式已知，则可利用 MATLAB 的控制系统工具箱提供的 tf 函数、pole 函数、zero 函数、roots 函数和 pzmap 函数方便地求出系统函数零、极点，并绘出其零、极点分布图。

例 9-6　已知某连续系统的系统函数如下

$$H(s)=\frac{2s^2-12s+16}{s^3+4s^2+6s+3}$$

试利用 MATLAB：

① 求出零、极点，并绘出其零、极点分布图；

② 求出并绘制波形系统的单位冲激响应和幅频响应，判断系统是否稳定。

解　利用 MATLAB 的 pole 函数、zero 函数求零、极点，利用 pzmap 函数绘出其零、极点分布图，程序如下：

```
num＝[2 -12 16];
den＝[1 4 6 3];
sys＝tf(num,den);
set(gcf,'color','w');
pzmap(sys);
p＝pole(sys)
z＝zero(sys)
```

运行结果为：

```
p＝-1.5000 + 0.8660i
   -1.5000 - 0.8660i
   -1.0000
z＝4
   2
```

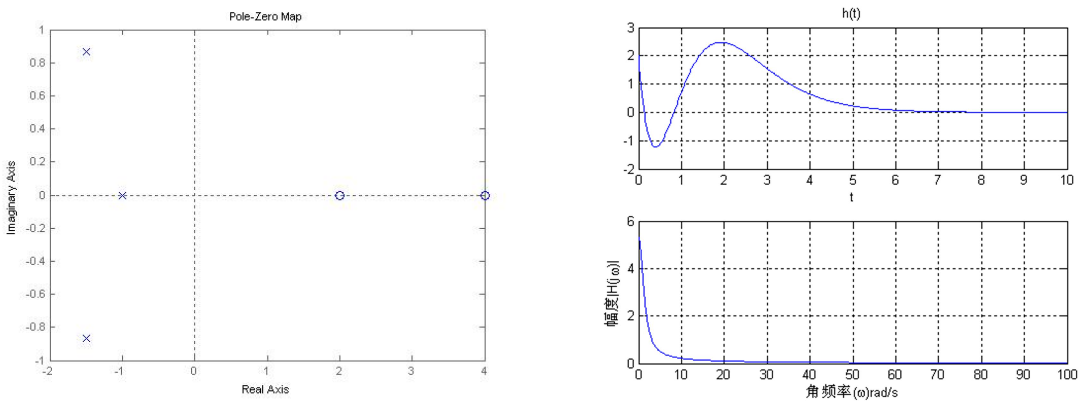
绘制的系统零、极点分布图如图 9-3(a)所示，其中符号“o”表示零点，符号“×”表示极点。

系统冲激响应 $h(t)$ 的时域波形可直接调用 impulse 函数来绘制，系统的频率响应利用前面介绍过的 freqs 函数，程序如下：

```
num＝[2 -12 16];
den＝[1 4 6 3];
t＝0:0.02:10;
subplot(2,1,1); h＝impulse(num,den,t);
```

```
plot(t,h);xlabel('t');title('h(t)'); grid on;
subplot(2,1,2);[H,w]＝freqs(num,den);
plot(w,abs(H));xlabel('角频率(\omega)rad/s');ylabel('幅度|H(j\omega)|');
grid on;
```

系统的单位冲激响应和幅频响应如图 9-3(b)所示。



(a) 零极点图　　　　(b) 单位冲激响应和幅频响应图

图 9-3　例 9-6 波形

从系统零极点分布图可看出，系统所有极点均位于 s 平面左半平面，故该系统是稳定系统。另外，从系统的冲激响应曲线也可看出，$h(t)$ 是衰减信号，可得出该系统稳定的结论。

2. 系统函数极点与冲激响应的关系

（1）当极点为单极点时，分为实数极点和复数极点两种情况

例 9-7　已知连续时间系统的系统函数如下：

① $H(s)=\dfrac{1}{s+2}$　　② $H(s)=\dfrac{1}{s}$

③ $H(s)=\dfrac{1}{s-2}$　　④ $H(s)=\dfrac{1}{(s+0.5)^2+4}$

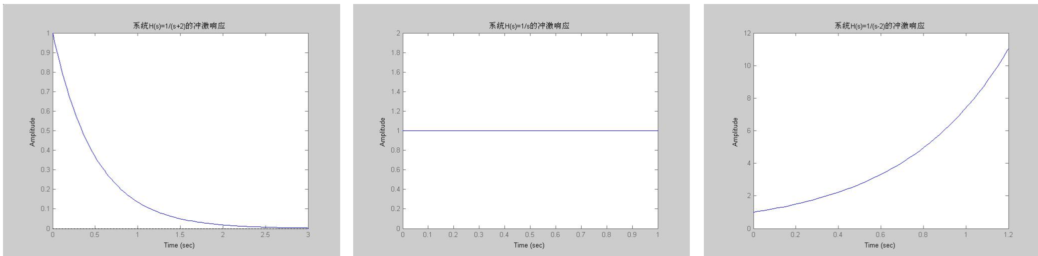
⑤ $H(s)=\dfrac{1}{s^2+0.09}$　　⑥ $H(s)=\dfrac{1}{(s-0.5)^2+4}$

试利用 MATLAB 绘出各单位冲激响应的时域波形，并分析系统函数极点位置对冲激响应的影响。

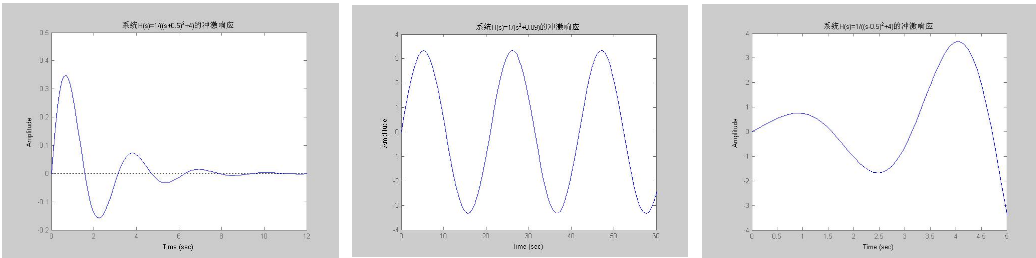
解　系统冲激响应 $h(t)$ 的时域波形可直接调用 impulse 函数来绘制，程序如下：

```
① num＝[1];den＝[1 2]; impulse(num,den); title('系统 H(s)＝1/(s+2)的冲激响应');
② num＝[1];den＝[1 0];impulse(num,den); title('系统 H(s)＝1/s 的冲激响应');
③ num＝[1];den＝[1 -2]; impulse(num,den); title('系统 H(s)＝1/(s-2)的冲激响应');
④ num＝[1];den＝[1 1 4.25]; impulse(num,den);title('系统 H(s)＝1/((s+0.5)^2+4)的冲激响应');
⑤ num＝[1];den＝[1 0 0.09]; impulse(num,den);title('系统 H(s)＝1/(s^2+0.09)的冲激响应');
⑥ num＝[1];den＝[1 -1 4.25];impulse(num,den);title('系统 H(s)＝1/((s-0.5)^2+4)的冲激响应');
```

上述命令绘制的系统冲激响应 $h(t)$ 的时域波形如图 9-4(a)～(f)所示。可见，图(a)和图(d)中极点位于 s 平面的左半平面，对应的冲激响应为衰减函数，系统稳定；图(b)和图(e)中极点位于 s 平面的虚轴上，对应的冲激响应分别为常数和等幅振荡，系统临界稳定；图(c)和图(f)中极点位于 s 平面的右半平面，对应的冲激响应为增长函数，系统不稳定。



(a) (b) (c)



(d) (e) (f)

图 9-4　例 9-7 波形

（2）当极点为重极点时，分为实数极点和复数极点两种情况

例 9-8　已知连续时间系统的系统函数如下：

① $H(s)=\dfrac{1}{s^3}$　　② $H(s)=\dfrac{1}{(s+0.5)^2}$

③ $H(s)=\dfrac{1}{(s-0.5)^2}$　　④ $H(s)=\dfrac{1}{(s^2+0.09)^2}$

⑤ $H(s)=\dfrac{1}{((s+0.5)^2+4)^3}$　　⑥ $H(s)=\dfrac{1}{((s-0.5)^2+4)^3}$

试利用 MATLAB 绘出各单位冲激响应的时域波形，并分析系统函数极点位置对冲激响应的影响。

解　程序如下：

① num＝[1];den＝[1 0 0 0];impulse(num,den);title('系统 H(s)＝1/(s^3)的冲激响应');

上述命令绘制的系统冲激响应 $h(t)$ 的时域波形如图 9-5(a)所示。可见，三重极点位于 s 平面的原点，对应的冲激响应为增长函数，系统不稳定。

② num＝[1];den＝[1 1 0.25];impulse(num,den);title('系统 H(s)＝1/(s+0.5)^2 的冲激响应');

系统冲激响应 $h(t)$ 的时域波形如图 9-5(b)所示。可见，二重极点位于 s 平面的负实轴上，对应的冲激响应为衰减函数，系统稳定。

③ num＝[1];den＝[1 -1 0.25];impulse(num,den);title('系统 H(s)＝1/(s-0.5)^2 的冲激响应');

系统冲激响应 $h(t)$ 的时域波形如图 9-5(c)所示。可见，二重极点位于 s 平面的正实轴上，对应的冲激响应为增长函数，系统不稳定。

④ num＝[1];den＝conv([1 0 0.09],[1 0 0.09]); impulse(num,den);

title('系统 H(s)＝1/(s^2+0.09)^2 的冲激响应');

系统冲激响应 $h(t)$ 的时域波形如图 9-5(d)所示。可见，二重共轭极点位于 s 平面的虚轴上，对应的冲激响应为增长函数，系统不稳定。

⑤ num＝[1];den1＝conv([1 1 4.25], [1 1 4.25]); den2＝[1 1 4.25];

den＝conv(den1,den2);

impulse(num,den);title('系统 H(s)＝1/((s+0.5)^2+4)^3 的冲激响应');

系统冲激响应 $h(t)$ 的时域波形如图 9-5(e)所示。可见，三重共轭复极点位于 s 平面的左半部分，对应的冲激响应为衰减函数，系统稳定。

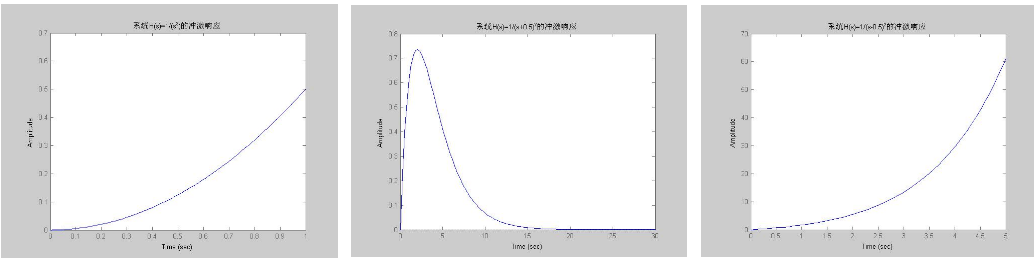
⑥ num＝[1];

den1＝conv([1 -1 4.25], [1 -1 4.25]); den2＝[1 -1 4.25];

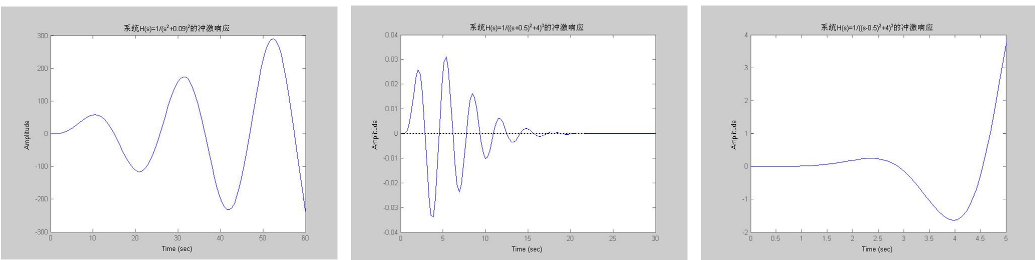
den＝conv(den1,den2);

impulse(num,den);title('系统 H(s)＝1/((s-0.5)^2+4)^3 的冲激响应');

系统冲激响应 $h(t)$ 的时域波形如图 9-5(f)所示。可见，三重共轭复极点位于 s 平面的右半部分，对应的冲激响应为增长函数，系统不稳定。



(a) (b) (c)



(d) (e) (f)

图 9-5　例 9-8 波形

（六）连续时间系统的系统函数求解的 MATLAB 实现

复杂的系统框图或信号流图有着很多元件和各种交叉连接，可以利用方框图简化和梅森

(Mason)公式得到整个系统的激励与响应的比值，即系统的传递函数。在 MATLAB 中，可以使用 append 函数和 connect 函数实现系统传递函数的求解。

append 函数的调用格式为：

sys＝append (sys1,sys2, ...)

其中，sys 是系统矩阵(未连接)，sys1,sys2, ...是各模块传递函数，sys1＝tf(num,den)，num 是模块传递函数分子对应系数，den 是分母对应系数，例如下例中第 3 个模块(图中第 3 条支路)为 G3＝tf(4,1)。

connect 函数的调用格式为：

G＝connect(sys,Q, INPUT,OUTPUT)

其中，Q 是输入、输出连接关系矩阵，第一列是模块编号，从第二列开始分别为进入该模块的所有通路编号，INPUT 指输入信号加入的通路编号，OUTPUT 指输出信号加入的通路编号。

例 9-9 求图 9-6 所示的信号流图的系统函数 $H(s)=\dfrac{Y(s)}{F(s)}$。

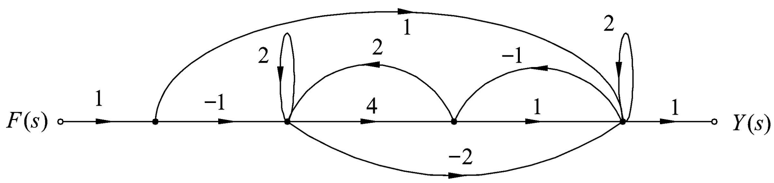


图 9-6 例 9-9 图（1）

解 ① 将各模块(支路)编号，如图 9-7 所示。

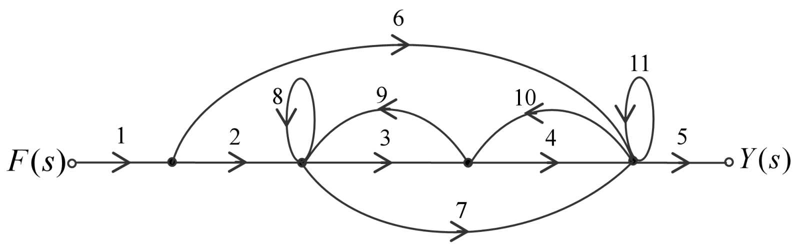


图 9-7 例 9-9 图（2）

② 使用 append 函数实现各模块(未连接)的系统矩阵，程序如下：

```
G1＝tf(1,1);G2＝tf(-1,1);G3＝tf(4,1);G4＝tf(1, 1); G5＝tf(1,1);G6＝tf(1,1);G7＝tf(-2,1);
G8＝tf(2,1);G9＝tf(2,1); G10＝tf(-1,1);G11＝tf(2,1);
sys＝append(G1,G2,G3,G4,G5,G6,G7,G8,G9,G10,G11);
```

③ 指定各模块之间的连接关系，程序如下：

```
Q＝[1 0 0 0 0;2 1 0 0 0;3 2 8 9 0;4 3 10 0 0;5 4 6 7 11;6 1 0 0 0;
7 2 8 9 0;8 2 8 9 0;9 3 10 0 0;10 4 6 7 11;11 4 6 7 11]
INPUT＝1;
OUTPUT＝5;
Q＝
```

1	0	0	0	0
2	1	0	0	0
3	2	8	9	0
4	3	10	0	0
5	4	6	7	11
6	1	0	0	0
7	2	8	9	0
8	2	8	9	0
9	3	10	0	0
10	4	6	7	11
11	4	6	7	11

④ 使用 connect 函数构造整个系统的模型，程序如下：

```
G＝connect(sys,Q, INPUT,OUTPUT)
```

运行此程序可以求出整个系统的传递函数，运行结果为：

```
Transfer function:
-2.75
```

三、实验内容

9.1 已知单边余弦信号 $f(t)=\cos(2t)u(t)$，试调用 laplace 函数计算其拉普拉斯变换。

9.2 已知某连续系统的系统函数如下：

① $H_1(s)=\dfrac{(s+1)(s+4)}{s(s+2)(s+3)}$

② $H_2(s)=\dfrac{s^3+5s^2+9s+7}{s^2+3s+2}$

③ $H(s)=\dfrac{s}{(s+1)^2(s+3)}$

④ $H(s)=\dfrac{s^3}{(s+1)^3}$

试利用 MATLAB 对其进行部分分式展开，并求出其拉普拉斯逆变换。

9.3 已知某连续系统的系统函数如下：

$$H(s)=\frac{2s+4}{s^3+4s}$$

试利用 MATLAB 对其进行部分分式展开，求系统的单位冲激响应 $h(t)$，绘出其时域波形，并根据系统函数零、极点分布判断系统的稳定性。

9.4 已知某连续系统的系统函数如下：

$$H(s)=\frac{s^2+1}{s^5+2s^4-3s^3+3s^2+3s+2}$$

① 利用 MATLAB 的 pzmap 函数绘出其零、极点分布图，时域波形，并判断系统是否稳定。

② 使用 MATLAB 的函数[z,p,k]＝tf2zp(num,den)及函数 roots 判断系统的稳定性。

9.5 已知系统函数如下：

$$H(s)=\frac{s+5}{s^2+5s+6}$$

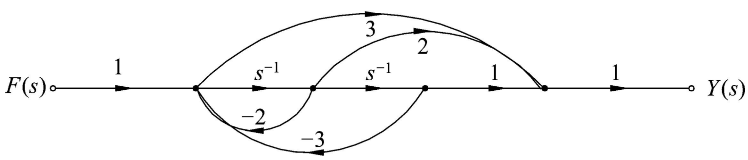
利用 MATLAB 求系统的：① 单位冲激响应；② 单位阶跃响应；③ 输入为 $f(t)=\sin(2t)u(t)$ 的零状态响应；④ 输入为 $f(t)=\mathrm{e}^{-t}u(t)$ 零状态响应。

9.6 已知系统函数如下：

$$H(s)=\frac{1}{s^2+5s+4}$$

输入信号为 $f(t)=(-t+1)[u(t)-u(t-2)]-[u(t-2)-u(t-3)]$，利用 MATLAB 求解并绘出零状态响应。

9.7 求题 9.7 图所示的信号流图的系统函数 $H(s)=\dfrac{Y(s)}{F(s)}$。



题 9-7 图

四、实验要求

1. 在 MATLAB 中输入程序，验证实验结果，并将实验结果存入指定存储区域。

2. 在实验报告中简述实验目的及原理，写出自编程序，并给出实验结果，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。

实验十　离散时间系统的时域分析

一、实验目的

通过使用 MATLAB 软件对离散时间 LTI 系统的时域特性进行仿真分析，熟悉离散时间 LTI 系统在典型激励下的响应及特征，熟悉 MATLAB 相应函数的调用格式和作用，熟悉并掌握系统单位脉冲响应的 MATLAB 实现方法，掌握差分方程迭代法、零输入响应、零状态响应及全响应的 MATLAB 求解方法。

二、实验原理

（一）离散系统时域分析方法

已知线性时不变离散系统的差分方程、系统初始值和输入序列的条件下，在时域中求解系统响应的常用方法有迭代法、零输入和零状态响应法及经典法等。

1. 迭代法

由系统的差分方程得到系统的递推公式，根据系统的递推公式得到系统输出在每一时刻的样值。迭代法可以求离散系统的零输入响应、零状态响应和全响应，方法简单，适合计算机求解，但仅能求出序列离散点的样值，一般不能得到系统输出解的数学表达式。

2. 零输入和零状态响应法

利用齐次解得到离散系统的零输入响应，再利用卷积和得到零状态响应，然后将零输入响应和零状态响应加起来得到系统的全响应。其中离散系统的单位脉冲响应由系统的传输算子求出，系统的零状态响应用输入序列和单位脉冲响应的卷积和求得。

单位脉冲响应 $h(k)$是初始条件为零、输入信号为单位脉冲序列$\delta(k)$时的系统响应。求解 $h(k)$对离散时间系统的分析具有重要意义。

3. 经典法

与连续系统的经典解法类似，分别求出离散系统差分方程的齐次解和特解，然后根据初始值确定全响应中的系数，最后得到离散系统的全响应。

（二）MATLAB 函数介绍

1. impz 函数

impz 函数用来求离散系统单位脉冲响应并绘制时域波形。

调用 impz 函数时，与连续系统类似，也需要用向量来对离散系统进行表示。

描述离散时间 LTI 系统的输入 $f(k)$与输出 $y(k)$关系的差分方程为

$$\begin{aligned}&a_n y(k-n)+a_{n-1}y(k-n+1)+\cdots+a_1 y(k-1)+a_0 y(k)\\&=b_m f(k-m)+b_{m-1}f(k-m+1)+\cdots+b_1 f(k-1)+b_0 f(k)\end{aligned}$$

则可以用向量 a 和 b 表示该系统，即：

$$a=[a_0,a_1,\cdots,a_{n-1},a_n]\text{，}\quad b=[b_0,b_1,\cdots,b_{m-1},b_m]$$

在用向量来表示差分方程描述的离散系统时，缺项要用 0 来补齐。

上式的离散系统算子方程为

$$\begin{aligned}&a_n E^{-n}y(k)+a_{n-1}E^{-n+1}y(k)+\cdots+a_1E^{-1}y(k)+a_0 y(k)\\&=b_m E^{-m}f(k)+b_{m-1}E^{-m+1}f(k)+\cdots+b_1E^{-1}f(k)+b_0 f(k)\end{aligned}$$

传递算子为：

$$H(E)=\frac{y(k)}{f(k)}=\frac{b_mE^{-m}+b_{m-1}E^{-m+1}+\cdots+b_1E^{-1}+b_0}{a_nE^{-n}+a_{n-1}E^{-n+1}+\cdots+a_1E^{-1}+a_0}$$

用向量 a 和 b 表示该系统，$a=[a_0,a_1,\cdots,a_{n-1},a_n]$，$b=[b_0,b_1,\cdots,b_{m-1},b_m]$，这些系数都是从 E^0 按 E 的降幂排列。

函数 impz 能绘出向量 a 和 b 定义的离散系统在指定时间范围内单位脉冲响应 h(k)的时域波形，并能求出系统单位脉冲响应 h(k)在指定时间范围内的数值解。

调用格式为：

① impz(b,a)：该调用格式以默认方式绘出向量 a 和 b 定义的离散系统单位脉冲响应 h(k)的时域波形。

② impz(b,a,n)：绘出向量 a 和 b 定义的离散系统在 0～n（n 必须为整数）离散时间范围内单位脉冲响应 h(k)的时域波形。

③ impz(b,a,n1:n2)：绘出向量 a 和 b 定义的离散系统在 n1～n2（n1、n2 必须为整数，且 n1<n2）离散时间范围内单位脉冲响应 h(k)的时域波形。

④ y＝impz(b,a,n1:n2)：不绘出系统的 h(k)的时域波形，而是求出向量 a 和 b 定义的离散系统在 n1～n2（n1、n2 必须为整数，且 n1<n2）离散时间范围内单位脉冲响应 h(k)的数值解。

2. dimpulse 函数

函数 dimpulse 可以求解线性离散时间系统的单位脉冲响应。

调用格式为：

① dimpulse(A,B,C,D,IU)：绘出单输入 IU 作用下的单位脉冲响应，数据点数自动确定。

x[n+1]＝Ax[n] + Bu[n]

y[n]＝Cx[n] + Du[n]

② dimpulse(num,den)：绘出传输函数 H(z)＝num(z)/den(z)的单位脉冲响应。

③ dimpulse(A,B,C,D,IU,N)或 dimpulse(num,den,N)：绘出单输入 IU 作用下的单位脉冲响应，数据点数为 N。

④ [Y,X]＝dimpulse(A,B,C,D,…)：返回输出和状态向量 Y 和 X，不绘出单位脉冲响应的图形。

⑤ [Y,X]＝dimpulse(num,den,...)：返回输出和状态向量 Y 和 X，不绘出单位脉冲响应的图形。

3. stepz 函数

在信号处理工具箱中提供了求解离散时间系统单位阶跃响应的函数 stepz。

调用格式为：

y＝stepz(b,a,k)

式中参数与 impz 函数相同，如果没有输出参数，直接调用 stepz(b,a,k)，则 MATLAB 也将会在当前绘图窗口中自动画出系统单位阶跃响应的图形。

4. dstep 函数

MATLAB 还提供了求解线性离散时间系统单位阶跃响应的函数 dstep。

调用格式为：

① dstep(A,B,C,D,IU)：绘出单输入 IU 作用下的单位阶跃响应，数据点数自动确定。

x[n+1]＝Ax[n] + Bu[n]

y[n]＝Cx[n] + Du[n]

② dstep(num,den)：绘出传输函数 H(z)＝num(z)/den(z)的单位阶跃响应。

③ dstep(A,B,C,D,IU,N)或 dstep(num,den,N)：绘出单输入 IU 作用下的单位阶跃响应，数据点数为 N。

④ [Y,X]＝dstep(A,B,C,D,…)：返回输出和状态向量 Y 和 X，不绘出单位阶跃响应的图形。

⑤ [Y,X]＝dstep(num,den,...)：返回输出和状态向量 Y 和 X，不绘出单位阶跃响应的图形。

5. filter 函数

filter 函数用来计算常系数线性差分方程的系统响应。

调用格式为：

① filter(b,a,f)：计算系统在输入信号 f 作用下的零状态响应。

② y＝filter(b,a,f)：计算在输入 f 作用下的系统零状态响应 y 的数值解。filter 函数的最简形式需要三个输入参数，系数向量 a、b 和输入向量 f，由于没有给定任何初始条件，输出对应于系统的零状态响应。

③ y＝filter(b,a,f,zi)：计算系统在输入 f 和初始状态作用下的全响应 y 的数值解。

6. filtic 函数

filtic 函数为 filter 函数选择初始条件。filtic 函数将传统的初始条件[y(-1),y(-2),y(-3),…]转换成 filter 函数可以使用的条件。

调用格式为：

zi＝filtic(b,a,Y0)

其中，Y0 为系统的初始状态，Y0＝[y(-1),y(-2),y(-3),…]。

（三）离散时间系统响应迭代法的 MATLAB 实现

采用差分方程的迭代解法，求离散时间系统的响应。

例 10-1 已知系统的差分方程为

$$y(k+2)-0.7y(k+1)+0.1y(k)=7f(k+2)-2f(k+1)$$

初始条件 $y(-1)=10$， $y(-2)=-10$，计算在输入信号为 $f(k)=u(k)$ 时的系统全响应 $y(k)$。

解 程序如下：

```
%用迭代法计算系统的全响应
clear                                          %清除当前工作空间的全部变量
k＝-2:15;
n＝0:15;
y＝[-10;10;zeros(length(k)-2,1)];              %y（1）＝-10, y（2）＝10
f＝[0;0;(1.^n)'];
for i＝1:length(k)-2
    y(i+2)＝0.7*y(i+1)-0.1*y(i)+7*f(i+2)-2*f(i+1);   %用迭代法计算 y（3）~ y(18)
end
disp('k                    y');                      % 在屏幕上显示提示信息
disp(num2str([k',y]));
stem(k,y);xlabel('k'); ylabel('y(k)');axis([-3,15,-11,16]);
```

运行结果为：

```
k          y
-2       -10
-1        10
 0        15
 1       14.5
 2      13.65
 3     13.105
 4    12.8085
 5    12.6554
 6     12.578
```

7	12.539
8	12.5195
9	12.5098
10	12.5049
11	12.5024
12	12.5012
13	12.5006
14	12.5003
15	12.5002

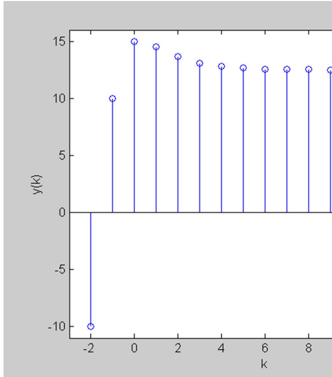


图 10-1　例 10-1

波形如图 10-1 所示。

注意：MATLAB 要求所有的向量序号从 1 开始，或下标为正整数。编程中若有 y(0)和 y(-1)，会出现提示：Subscript indices must either be real positive integers or logicals。因此这个程序是在初始条件 y（1）＝-10, y（2）＝10 下，系统全响应 y（3）~y(18)的值根据迭代式计算，i 从 1 开始。

（四）离散系统的单位脉冲响应和单位阶跃响应的 MATLAB 实现

1. 离散系统的单位脉冲响应

MATLAB 为用户提供专门求离散系统单位脉冲响应 h(k) 的函数 impz。

例 10-2　描述离散系统的差分方程为

$$y(k+2)+0.4y(k+1)-0.12y(k)=f(k+2)+2f(k+1)$$

计算系统的单位脉冲响应 $h(k)$。

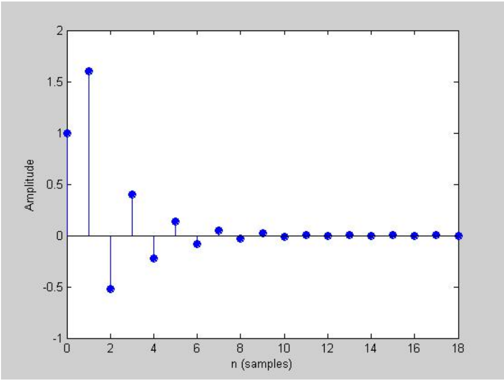
解　运行如下 MATLAB 命令：

```
a＝[1 0.4 -0.12];
b＝[1 2];
impz(b,a);
```

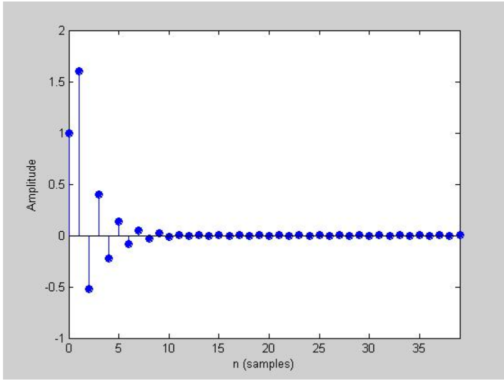
则绘出系统的脉冲响应波形，如图 10-2(a)所示。

对上例，若运行命令 impz(b,a,N)，则绘出系统在 0～N 离散时间范围内脉冲响应的时域波形,如图 10-2(b)所示。

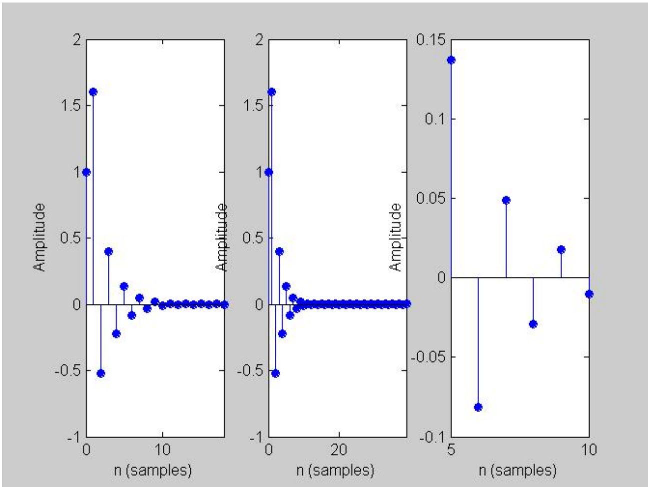
```
N＝40;
a＝[1 0.4 -0.12];
b＝[1 2];
impz(b,a,N);
```



(a)



(b)



(c)

图 10-2　系统的单位脉冲响应

若运行命令 impz(b,a,5:1:10)，则绘出 5～10 离散时间范围内脉冲响应的时域波形，如图 10-2(c)所示。运行程序如下：

```
N＝40;a＝[1 0.4 -0.12];b＝[1 2];
subplot(1,3,1);impz(b,a);
subplot(1,3,2);impz(b,a, N);
subplot(1,3,3);impz(b,a, 5:1:10);
```

若运行命令 y＝impz(b,a,5:1:10)，则求出在 5～10 离散时间范围内的系统脉冲响应的数值解。

```
a＝[1 0.4 -0.12];b＝[1 2];y＝impz(b,a,5:1: 10);
```

则运行结果为：

```
y＝
    0.1370
   -0.0815
    0.0490
```

-0.0294

0.0176

-0.0106

MATLAB 还提供了求解线性离散时间系统单位脉冲响应的函数 dimpulse。

```
a＝[1 0.4 -0.12]; b＝[1 2 0];
subplot(121);impz(b,a,6);
subplot(122);dimpulse(b,a,6);
```

波形如图 10-3 所示。

2. 离散系统的单位阶跃响应

在信号处理工具箱中提供了求解离散时间系统单位阶跃响应的函数 stepz 和求解线性离散时间系统单位阶跃响应的函数 dstep。

计算例 10-2 的离散系统差分方程的单位阶跃响应的程序如下：

```
%单位阶跃响应的求解
a＝[1 0.4 -0.12];
b＝[1 2 0];
k＝0:5;
y1＝stepz(b,a,6);
subplot(221);stem(k,y1);
y2＝dstep(b,a,6);
subplot(222);stem(k,y2);
subplot(223);stepz(b,a,6)
subplot(224);dstep(b,a,6)
```

波形如图 10-4 所示。

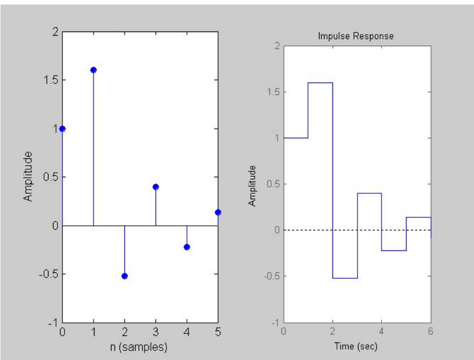


图 10-3　系统的单位脉冲响应

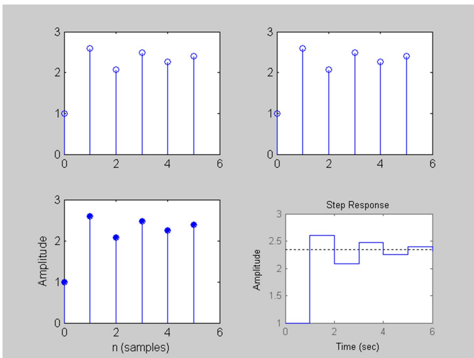


图 10-4　系统的单位阶跃响应

（五）利用函数 filter 求离散时间系统的响应

MATLAB 提供了求线性时不变离散时间系统响应的专用函数 filter。filter 函数可以求出

由离散时间系统的差分方程描述的系统响应，也能求出系统在指定的任意时间范围内系统响应的数值解。

lsim 函数也可以求解离散系统的系统响应，该函数在连续系统分析实验中已介绍。

例 10-3 已知系统传递算子为 $H(E)=\dfrac{E(7E-2)}{(E-0.5)(E-0.2)}$，初始条件 $y(-1)=10$，$y(-2)=-10$，计算在输入信号为 $f(k)=u(k)$ 时的系统零状态响应、零输入响应和完全响应 $y(k)$。

解 $H(E)=\dfrac{E(7E-2)}{(E-0.5)(E-0.2)}=\dfrac{7E^2-2E}{E^2-0.7E+0.1}$

① 计算系统零状态响应，程序如下：

```
%计算系统零状态响应
N＝30;
a＝[1 -0.7 0.1];
b＝[7 -2];
f＝ones(1,N);
k＝0:1:N-1;
y＝filter(b,a,f);
stem(k,y);xlabel('k'); title('系统零状态响应 y(k)')
```

② 计算系统零输入响应。零输入可以由 zeros 函数产生，使用 size 函数设置这个零输入的维数与向量 k 匹配。程序如下：

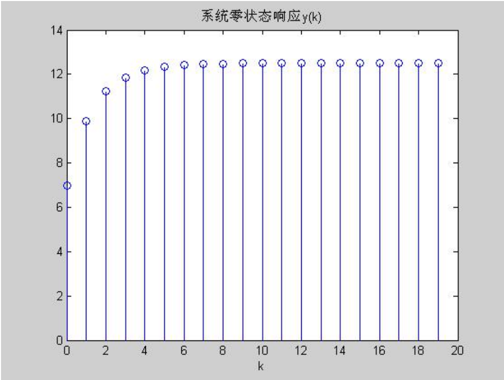
```
%计算系统零输入响应
N＝20;
a＝[1 -0.7 0.1];
b＝[7 -2];
k＝0:1:N-1;
zi＝filtic(b,a, [10, -10]);
y＝filter(b,a,zeros(size(k)),zi);
stem(k,y);
xlabel('k');
title('系统零输入响应')
```

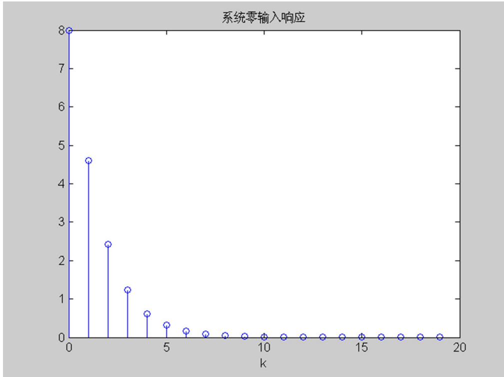
③ 计算系统全响应。y＝filter(b,a,f,zi)中的初始值 zi 不是 $y(-1)=10$， $y(-2)=-10$，它可以由 filtic 函数计算。程序如下：

```
%计算系统全响应
N＝20;
a＝[1 -0.7 0.1];
b＝[7 -2];
f＝ones(1,N);
k＝0:1:N-1;
zi＝filtic(b,a,[10, -10]);
y＝filter(b,a,f,zi);
```

stem(k,y);xlabel('k');title('系统全响应 y(k)')

系统响应时域波形如图 10-5 所示。





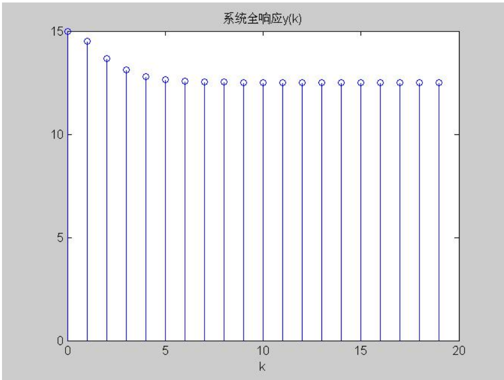


图 10-5　例 10-3 图

本例可以利用 conv 函数卷积和实现系统零状态响应，详细方法见实验四。

三、实验内容

10.1　已知系统的差分方程为

$$y(k)+y(k-1)=f(k)+2f(k-2)$$

初始条件 $y(0)=2$，输入信号为 $f(k)=ku(k)$。试用迭代法计算系统全响应，并画出系统全响应的波形。

10.2　已知某离散系统的差分方程为

$$y(k)-y(k-1)-2y(k-2)=f(k)-f(k-2)$$

试用 MATLAB 完成：

① 以默认方式绘出系统单位脉冲响应 $h(k)$ 的时域波形。

② 绘出系统在 0～50 取样点范围内 $h(k)$ 的时域波形。

③ 绘出系统在－20～30 离散时间范围内 $h(k)$ 的时域波形。

④ 求出系统在－5～10 离散时间范围内 $h(k)$ 的数值解。

10.3　已知某离散系统的差分方程为

$$y(k)-y(k-1)-2y(k-2)=f(k)-f(k-2)$$

试用 MATLAB 计算并绘出单位阶跃响应的时域波形。

10.4　已知系统的差分方程为

$$6y(k)-y(k-1)-y(k-2)=f(k)$$

初始条件 $y(0)=15$，$y(2)=4$，试用 MATLAB 计算在输入信号为 $f(k)=\cos(k\pi)u(k)$ 时的系统全响应，并画出系统全响应的波形。

10.5　已知系统 $y(k+2)+0.4y(k+1)-0.12y(k)=f(k+2)+2f(k+1)$，初始条件 $y(-1)=1$，$y(-2)=2$，计算在输入信号为 $f(k)=u(k)$ 时的系统零状态响应和全响应 $y(k)$。

四、实验要求

1. 在 MATLAB 中输入程序，验证实验结果，并将实验结果存入指定存储区域。

2. 在实验报告中简述实验目的及原理，写出自编程序，并给出实验结果，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。

实验十一　离散时间信号与系统的 z 域分析

一、实验目的

掌握离散时间信号的 z 变换和逆 z 变换的实现方法，掌握离散时间系统的 z 域分析方法，熟悉 MATLAB 相应函数的调用格式和作用，掌握使用 MATLAB 来分析离散时间信号与系统的 z 域特性、实现离散系统的零、极点分析及稳定性分析的方法。

二、实验原理

（一）离散时间信号与系统的 z 域分析方法

z 变换是离散信号与系统分析的重要方法和工具。类似于连续信号与系统分析中的拉普拉斯变换，它将离散系统的数学模型差分方程转化为简单的代数方程，使其求解过程得以简化。

1. 单边 z 变换定义

$$F(z)=\sum_{k=0}^{\infty}f(k)z^{-k}$$

2. z 逆变换定义

$$f(k)=\frac{1}{2\pi\mathrm{j}},\quad F(z)\quad {}^{-1}\mathrm{d}z$$

z 逆变换采用部分分式展开法和幂级数展开法。

（1）部分分式展开法

已知函数 $F(z)$，用部分分式展开法求 z 逆变换的步骤为：先将 $F(z)$ 除以 z，得到 $\frac{F(z)}{z}$；然后将 $\frac{F(z)}{z}$ 展开成部分分式；再将展开的每个分式乘以 z，即得到 $F(z)$ 的表达式；最后对各

分式进行 z 逆变换，求出序列 $f(k)$。

（2）幂级数展开法

根据 z 变换的定义，$F(z)=\sum_{k=0}^{\infty} f(k)z^{-k}$，可见 $F(z)$ 是 z 的负幂次级数。如果 $F(z)$ 是 z 的多项式分式，可以用长除法将它展开成 z 的负幂次级数，级数各项系数就是序列 $f(k)$的各项数值。

3. 离散系统的 z 域分析

对系统的差分方程两边取 z 变换，代入初始条件和输入的 z 变换，可以直接求得系统的全响应；或根据离散系统的系统传递函数 $H(z)$，$H(z)=\dfrac{Y_{\mathrm{f}}(z)}{F(z)}$，计算任意输入 $f(k)$的零状态响应 $y(k)$，先计算 $y(k)$的 z 变换 $Y(z)$，即 $Y(z)=F(z)H(z)$，然后再求 z 逆变换可得到 $y(t)$，零输入响应按时域方法求得。

4. 系统传递函数的零极点及系统的稳定性

① 离散时间系统稳定的充分必要条件是：

$$\sum_{k=-\infty}^{\infty} |h(k)| \leqslant M \quad （M \text{ 为有界正常数}）$$

② 当系统传递函数 $H(z)$ 的极点全位于单位圆内时，系统是一个稳定的系统；反之，系统是一个不稳定的系统。

（二）MATLAB 函数介绍

1. ztrans 函数

ztrans 函数用来求离散序列的 z 变换。

调用 ztrans 函数的命令格式为：

① F＝ztrans(f)：实现函数 f(n)的 z 变换，默认返回函数 F 是关于 z 的函数，即
　f＝f(n)＝> F＝F(z)

② F＝ztrans(f,w)：实现函数 f(n)的 z 变换，默认返回函数 F 是关于 w 的函数，即
　f＝f(n)＝> F＝F(w)

③ F＝ztrans(f,k,w)：实现函数 f(k)的 z 变换，默认返回函数 F 是关于 w 的函数，即
　f＝f(k)＝> F＝F(w)

2. iztrans 函数

iztrans 函数用来实现 z 逆变换。

调用 iztrans 函数的命令格式为：

① f＝iztrans(F)：实现函数 F(z)的 z 逆变换，默认返回函数 f 是关于 n 的函数。

② f＝iztrans(F,k)：实现函数 F(z)的 z 逆变换，默认返回函数 f 是关于 k 的函数。

③ f＝iztrans(F,w,k)：实现函数 F(w)的 z 逆变换，默认返回函数 f 是关于 k 的函数。

3. residuez 函数

residuez 函数用来进行 z 域的部分分式展开。

设离散系统的 z 域函数用下列有理分式来表示：

$$F(z)=\frac{B(z)}{A(z)}=\frac{b_m z^{-m}+b_{m-1}z^{-m+1}+\cdots+b_1 z^{-1}+b_0}{a_n z^{-n}+a_{n-1}z^{-n+1}+\cdots+a_1 z^{-1}+a_0}$$

可以将 $F(z)$ 展开成部分分式之和的形式，再对其求 z 逆变换。MATLAB 的信号处理工具箱提供了对 $F(z)$ 进行部分分式展开的函数 residuez。

调用格式为：

① [r,p,k]＝residuez(b,a)

其中，输入参量 b 为 $F(z)$ 的分子多项式系数构成的行向量，a 为 $F(z)$ 的分母多项式系数构成的行向量，b 和 a 都按 z^{-1} 升幂排列。函数将返回三个输出参量 r、p、k，其中 r 为 $F(z)$ 部分分式展开系数的列向量，p 为极点的列向量，k 为多项式的系数的行向量，若为真分式(m<n)，则 k 返回为空阵。

利用 residuez 函数可以将有理分式展开为

$$F(z)=\frac{B(z)}{A(z)}=\frac{r(1)}{1-p(1)z^{-1}}+\cdots+\frac{r(n)}{1-p(n)z^{-1}}+k(1)+k(2)z^{-1}+\cdots$$

如果 $P(j)=\cdots=P(j+m-1)$是 m 重极点，则展开式包含下面的结果

$$\frac{r(j)}{1-p(j)z^{-1}}+\frac{r(j+1)}{(1-p(j)z^{-1})^2}+\cdots+\frac{r(j+m-1)}{(1-p(j)z^{-1})^m}$$

② [b,a]＝residuez(r,p,k)：将部分分式展开式转换成 B/A 形式。

4. zplane 函数

zplane 函数用来绘制离散系统 z 平面的零、极点分布图。

调用格式为：

① zplane(Z,P)：以单位圆为参考圆绘制 Z 为零点列向量、P 为极点列向量的零极点图。每个零点用'o'表示，每个极点用'x'表示。若有重复点，在重复点右上角以数字标出重数。

② zplane(B,A)：B 和 A 分别是传递函数 $H(Z)=B(z)/A(z)$按 z^{-1} 的升幂排列的分子分母系数行向量。注意，当 B 和 A 同为标量时，如 B 为零点，则 A 为极点。

（三）离散时间信号 z 域分析的 MATLAB 实现

1. 利用 MATLAB 的符号运算实现 z 变换

如果离散序列 $f(n)$ 可用符号表达式表示，则可直接调用 MATLAB 的 ztrans 函数来求离散序列 $f(n)$ 的单边 z 变换。如果得到的结果不是最简形式，可以调用 simplify 函数对其简化。

例 11-1 试利用 MATLAB 的符号运算实现下列序列的 z 变换。

① 单边指数序列 $f(n)=\left(\frac{1}{2}\right)^{n}u(n)$；

② 阶跃序列 $f(k)=u(k)$；

③ 单位样值序列 $f(k)=\delta(k)$。

解 调用 ztrans 函数计算其 z 变换的 MATLAB 命令如下：

①
```
syms a n;
f＝a^n;
F＝ztrans(f)
'f＝',pretty(f);                    % pretty 函数是排版格式输出函数
'F＝',pretty(F);
```
运行结果为：
```
F＝z/a/(z/a-1)
ans＝
f＝
                  n
                a
ans＝
F＝
                  z
            -----------
            a (z/a - 1)
```
②
```
syms n;
f＝sym('Heaviside(n)');            % 也可输入 f＝1^n 或 f＝sym('1')
F＝ztrans(f);
simplify(F)
```
运行结果为：
```
F＝z/(z-1)
```
③
```
f＝sym('charfcn[0](n)');
F＝ztrans(f)
```
运行结果为：
```
F＝1
```
注意，在符号工具箱中，单位阶跃序列 u(n)是用 Heaviside(n)表示，而单位样值序列δ(n)是用 charfcn[0](n)表示。

2. z 逆变换的 MATLAB 实现

MATLAB 求 z 逆变换可直接调用专用的符号函数 iztrans 函数来实现 z 逆变换，或利用

residue 函数求得部分分式展开式再求 z 逆变换，也可以利用 dimpulse 函数求 z 逆变换的幂级数展开式。下面给出这三种方法的实现。

（1）调用 iztrans 函数来实现 z 逆变换

例 11-2　已知函数① $F(z)=1$，② $F_1(z)=\dfrac{1}{(z-2)(z-3)}$，③ $F_2(z)=\dfrac{z^2}{(z-2)(z-3)^3}$，利用 MATLAB 求单边 z 变换。

解　MATLAB 命令如下：

①
```
syms z;
F=sym('1');
f=iztrans(F)
```
运行结果为：
```
f=charfcn[0](n)
```
②
```
syms z;
F=1/((z-2)*(z-3));
f=iztrans(F)
```
运行结果为：
```
f=1/6*charfcn[0](n)-1/2*2^n+1/3*3^n
```
③
```
syms z;
F=z^2/((z-2)* ((z-3)^3));
f=iztrans(F)
```
运行结果为：
```
f=-2*2^n+2*3^n-5/6*3^n*n+1/6*3^n*n^2
```
（2）利用 MATLAB 求部分分式展开及 z 逆变换

例 11-3　利用 MATLAB 求下列函数的单边 z 逆变换。已知：

① $F_1(z)=\dfrac{z(7z-2)}{(z^2-0.7z+0.1)(z-0.4)}$；

② $F_2(z)=\dfrac{z^2}{(z-2)(z-3)^3}$。

解

① z 逆变换可用 residuez 函数进行部分分式分解求得。

程序如下：
```
num=[0 7 -2 0];
den=conv([1 -0.7 0.1],[1 -0.4]);
[r,p,k]=residuez(num,den)
```
运行结果为：
```
r=                p=                k=
      50.0000          0.5000             0
     -40.0000          0.4000
```

-10.0000 0.2000

$$F(z) = -\frac{10z}{z-0.2} - \frac{40z}{z-0.4} + \frac{50z}{z-0.5}$$

则 $$f(n) = \left[-10(0.2)^n - 40(0.4)^n + 50(0.5)^n\right]u(n)$$

② 由已知条件得：

$$\frac{H(z)}{z} = \frac{z}{(z-2)(z-3)^3}$$

首先调用 residue 函数对 $\frac{H(z)}{z}$ 进行部分分式展开。命令如下：

```
num＝[0 0 0 1 0]
den＝poly([2 3 3 3])
[r,p,k]＝residue(num,den)
```

其中，poly 函数的功能是由多项式的根创建多项式。调用格式为：P＝poly(V) 向量 V 为包含多项式 n 个根的行向量，向量 P 则为返回该多项式对应的系数向量。

程序运行结果为：

```
num＝
    0     0     0     1     0
den＝
    1   -11    45   -81    54
r＝                p＝                k＝
2.0000             3.0000             []
   -2.0000            3.0000
   3.0000             3.0000
   -2.0000            2.0000
```

由上述程序运行结果可得：

$$\frac{H(z)}{z} = \frac{2}{z-3} + \frac{-2}{(z-3)^2} + \frac{3}{(z-3)^3} + \frac{-2}{z-2}$$

注意，第一个系数 2 对应于重极点 p＝3 的 c1/(z-3)项，第二个系数-2 对应于重极点 p＝3 的 $c2/(z-3)^2$ 项，第三个系数 3 对应于重极点 p＝3 的 $c3/(z-3)^3$ 项，第四个系数-2 对应于单极点 p＝2 的 c4/(z-2)项。则：

$$H(z) = \frac{2z}{z-3} + \frac{-2z}{(z-3)^2} + \frac{3z}{(z-3)^3} + \frac{-2z}{z-2}$$

系统单位脉冲响应

$$h(n) = \left[2(3)^n - 2n(3)^{n-1} + 3\frac{n(n-1)}{2}(3)^{n-2} - 2(2)^n\right]u(n)$$

即 $$h(n)=\left[2(3)^n-\frac{5}{6}n(3)^n-\frac{1}{6}n^2(3)^n-2(2)^n\right]u(n)$$

下面求解 n＝0 到 n＝9 的信号 $h(n)$的值，程序如下：

```
syms n;
h＝-2*2^n+2*3^n-5/6*3^n*n+1/6*3^n*n^2;
hn＝subs(h,[0 1 2 3 4 5 6 7 8 9])
```

运行结果为：

```
hn＝
  Columns 1 through 7
0        0        1        11        76        422        2059
Columns 8 through 10
9221     38854    156440
```

（3）利用 MATLAB 的 dimpulse 函数求 z 逆变换

dimpulse 函数的实质是系统单位脉冲响应的数值计算，由于在 z 域中单位脉冲响应定义为：

$$h(k)=z^{-1}\left[H(z)\right]=z^{-1}\left[\frac{b_m z^m+b_{m-1}z^{m-1}+\cdots+b_0}{a_n z^n+a_{n-1}z^{n-1}+\cdots+a_0}\right]$$

可见，求解离散时间脉冲响应 $h(k)$ 的过程，也可以用来求解用两个多项式的比值表示的复杂函数的逆 z 变换。

考虑具有下面形式的 z 变换：

$$F(z)=\frac{b_m z^m+b_{m-1}z^{m-1}+\cdots+b_0}{a_n z^n+a_{n-1}z^{n-1}+\cdots+a_0}$$

其中 $m\leqslant n$。应用 dimpulse 函数求解 $F(z)$ 的 z 逆变换，相当于长除法得到的输出序列。

运行下列命令求解 n＝0 到 n＝q-1 的信号 f(n)的值：

```
num＝[bm bm-1 ... b0];
den＝[an an-1 ... a0];
f＝dimpulse(num,den,q)
```

重新求例 11-3，应用下列命令求解 n＝0 到 n＝9 的信号 $h(n)$的值：

```
num＝[0 0 1 0 0];
den＝poly([2 3 3 3]);
hn＝dimpulse(num,den,10)
```

运行结果为：

```
hn＝
        0
        0
        1
       11
       76
```

422
2059
9221
38854
156440

（四）离散时间系统的 z 域分析

例 11-4 已知系统的差分方程为：

$$y(k+2)-0.7y(k+1)+0.1y(k)=7f(k+1)-2f(k)$$

初始条件 y(－1)＝－4，y(－2)＝－38，输入 $f(k)=(0.4)^k u(k)$。求系统的零输入响应 $y_x(k)$、零状态响应 $y_f(k)$和全响应 $y(k)$。

分析：系统的零输入响应 $y_x(k)$、零状态响应 $y_f(k)$和全响应 $y(k)$可以用 filter 函数实现，其中零状态响应 $y_f(k)$还可以用符号法求解。

解 由差分方程得系统函数

$$H(z)=\frac{7z-2}{z^2-0.7z+0.1}$$

用 filter 函数实现系统的零输入响应 $y_x(k)$、零状态响应 $y_f(k)$和全响应 $y(k)$程序如下：

```
num＝[0 7 -2];
den＝[1 -0.7 0.1];
n＝0:15;
f＝(0.4).^n;
zi＝filtic(num,den,[-4, -38]);
yx＝filter(num,den, zeros(size(n)),zi);
yf＝filter(num,den,f);
y＝filter(num,den,f, zi);
subplot(131);stem(n,yx);xlabel('n');title('系统零输入响应');
subplot(132);stem(n,yf);xlabel('n');title('系统零状态响应');
subplot(133);stem(n,y);xlabel('n');title('系统全响应');
```

响应如图 11-1(a)所示。

用符号法求系统的零状态响应 $y_f(k)$程序如下：

```
syms n z;
f＝(0.4)^n;
F＝ztrans(f);
H＝(7*z-2)/((z-0.2)*(z-0.5));
yf＝iztrans(H*F)
```

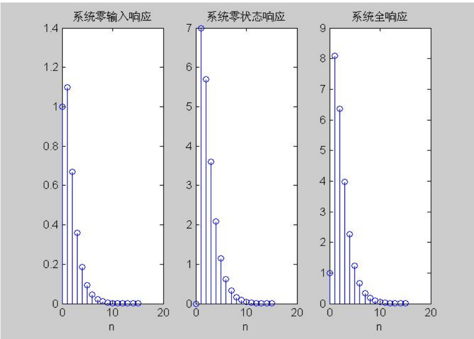
运行结果为：

```
yf＝-10*(1/5)^n+50*(1/2)^n-40*(2/5)^n
```

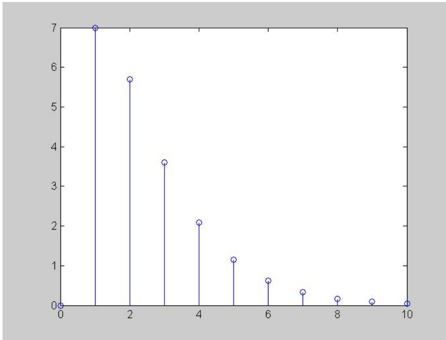
绘制零状态响应的程序为：

```
n＝0:15;
yf＝-10*(1/5).^n+50*(1/2).^n-40*(2/5).^n
stem(n,yf);
```

零状态响应如图 11-1(b)所示。



(a)



(b)

图 11-1　例 11-4 图

（五）离散时间系统的 z 域特性

离散系统的系统函数可以表示为零极点形式

$$H(s)=\frac{N(s)}{D(s)}=\frac{b_m z^{-m}+b_{m-1}z^{-m+1}+\cdots+b_1 z^{-1}+b_0}{a_n z^{-n}+a_{n-1}z^{-n+1}+\cdots+a_1 z^{-1}+a_0}$$

$$=k\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

可利用 MATLAB 提供的 roots 函数计算系统的零、极点，或使用 zplane 函数绘制离散系统的零、极点分布图。

例 11-5　已知某离散时间系统的系统函数如下：

$$H(z)=\frac{z^2}{(z-0.5)(z-0.25)}$$

① 利用 MATLAB 求系统的单位脉冲响应 $h(n)$，绘出 $h(n)$ 的时域波形，观察其时域特性，并根据 $h(n)$ 的时域波形判断系统是否稳定。

② 利用 MATLAB 计算系统的零、极点，并画出系统的零、极点分布图，判断系统是否稳定。

解

① 由已知条件得

$$\frac{H(z)}{z}=\frac{z}{(z-0.5)(z-0.25)}$$

调用 residue 函数将$\frac{H(z)}{z}$进行部分分式展开，命令如下：

```
%计算逆z变换
num=[1 0 0];
den=poly([0 0.5 0.25]);
[r,p,k]=residue(num,den)
```

运行结果为：

```
r=2          p=0.5000          k=
 -1            0.2500            []
  0            0
```

由上述程序运行结果可得：

$$\frac{H(z)}{z}=\frac{2}{z-0.5}+\frac{-1}{z-0.25}$$

则

$$H(z)=\frac{2z}{z-0.5}+\frac{-z}{z-0.25}$$

系统单位序列响应

```
f1=2*(1/2)^n-(1/4)^n
```

$$h(n)=\left[4\left(\frac{1}{2}\right)^n-2\left(\frac{1}{4}\right)^n\right]u(n)$$

绘制系统单位序列响应$h(n)$的时域波形的 MATLAB 命令如下：

```
%绘制系统单位序列响应时域波形
n=-10:20;
h=(2*(1/2).^n-(1/4).^n).*(n>=0);
stem(n,h,'filled')
title('单位序列响应 h(n)')
xlabel ('n')
```

上述命令绘制的系统单位序列响应$h(n)$的时域波形如图 11-2(a)所示。由图可以看出，系统的单位序列响应是随时间增加而按指数规律衰减的因果序列，满足时域绝对可和条件，故该系统是稳定系统。

② 程序如下：

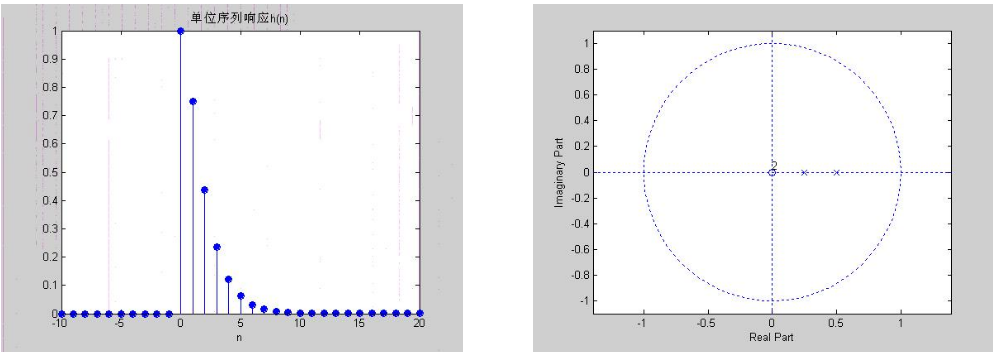
```
num=[1 0 0];
den=conv([1 -0.5], [1 -0.25]);
z=roots(num)
p=roots(den)
zplane(z,p);          %或 zplane (num,den);
```

运行结果为：

```
z=0          0
```

p＝0.5000　　0.2500

绘制的系统零、极点分布图如图 11-2(b)所示，系统传递函数 $H(z)$ 的极点全位于单位圆内时，系统是一个稳定的系统。



(a)　　　　　　　　　　(b)

图 11-2　例 11-5 图

三、实验内容

11.1　用 ztrans 函数求下列序列的 z 变换。

① $f_1(n)=nu(n)$　　　　② $f_2(n)=a^n u(n)$

③ $f_3(n)=0.5n[u(n)-u(n-5)]$　　　　④ $f_4(n)=a^n\cos\left(\dfrac{n\pi}{2}\right)u(n)$

⑤ $f_5(n)=\dfrac{n(n-1)}{2}u(n)$

11.2　用 z 域分析法求解下列系统的零状态响应。

① 已知线性离散时间系统的激励信号为 $f(n)=(-1)^n u(n)$，单位脉冲响应为 $\mathrm{h}(n)=\left[\dfrac{1}{3}(-1)^n+\dfrac{2}{3}3^n\right]u(n)$。

② 已知线性离散时间系统的激励信号为 $f(n)=u(n)$，系统传递函数为 $H(z)=\dfrac{z(7z-2)}{(z-0.2)(z-0.5)}$。

③ 已知线性离散时间系统的激励信号为 $f(n)=\cos\left(\dfrac{n\pi}{2}\right)u(n)$，系统传递函数为 $H(z)=\dfrac{z}{z-0.4}$。

11.3　已知某离散时间系统的系统函数如下：

$$H(z)=\frac{z^2}{z^2+\sqrt{2}z+1}$$

试利用 MATLAB：

① 求系统的单位序列响应 $h(n)$，绘出 $h(n)$ 的时域波形。

② 计算并绘出系统的零极点分布图。

四、实验要求

1. 在 MATLAB 中输入程序，验证实验结果，并将实验结果存入指定存储区域。

2. 在实验报告中简述实验目的及原理，写出自编程序，并给出实验结果，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。