

信号部分实验

目录

1. 连续信号的绘制
2. 连续信号的微积分和卷积
3. 非周期信号的频谱
4. 连续系统的零极点分析
5. 迭代法及离散卷积计算
6. 离散系统的时域和频域分析
7. 信号与系统综合实验项

实验 1 连续信号的绘制

1.1 实验目的

1. 掌握用 Matlab 绘制波形图的方法，学会常见波形的绘制。
2. 掌握用 Matlab 编写函数的方法。
3. 周期信号与非周期信号的观察。加深对周期信号的理解。

1.2 实验原理与计算示例

1、用 Matlab 绘制波形图的基本函数

Matlab 是一种基于矩阵和数组的编程语言，它将所有的变量都看成矩阵。它不仅有强大的计算功能，还有多种多样的画图功能。

这里主要介绍信号与系统分析中常用的几个 Matlab 函数，包括 Matlab 提供的内部函数和自定义函数。

我们可以在命令窗口中每次执行一条 Matlab 语句；或者生成一个程序，存为 M 文件供以后执行；或是生成一个函数，在命令窗口中执行。下面先定义几个基本函数。

➤ 单位阶跃函数

M 文件名：u.m

```
% 单位阶跃函数(连续或离散)
% 调用 y=u(t) 产生单位阶跃函数
function y=u(t)
y=(t>=0)
```

➤ 门函数

M 文件名：rectpuls.m, Matlab 的内部函数

```
调用格式 y=rectpuls(t) 产生高度为 1, 宽度为 1 的门函数
调用格式 y=rectpuls(t,W) 产生高度为 1, 宽度为 W 的门函数
```

➤ 三角脉冲函数

M 文件名：tripuls.m, Matlab 的内部函数

```
调用格式 y=tripuls(t) 产生高度为 1, 宽度为 1 的三角脉冲函数
调用格式 y=tripuls(t,w) 产生高度为 1, 宽度为 w 的三角脉冲函数
调用格式 y=tripuls(t,w,s)
产生高度为 1, 宽度为 w 的三角脉冲函数,  $-1 < s < 1$ ,  $s=0$  时为对称三角形;  $s=-1$ , 三角形顶点左
```

边。

➤ 抽样函数

M 文件名：Sa.m

```
% 抽样函数(连续或离散)
% 高度为 1,
% 调用 y=Sa(t) 产生高度为 1, 第一个过零点为  $\pi$ 
function f=Sa(t)
f=sinc(t./pi); % sinc(t)=sin( $\pi t$ )/( $\pi t$ ) 是 Matlab 内部函数
```

➤ 符号函数

M 文件名：sign.m 是 Matlab 的内部函数

➤ 周期方波

M 文件名：**square.m**, Matlab 的内部函数

调用格式 $y=\text{square}(w_0*t)$ 产生基频为 w_0 (周期 $T=2\pi/w_0$) 的周期方波。占空比为 50%。

调用格式 $y=\text{square}(w_0*t, \text{DUTY})$ 占空比 $\text{DUTY}=\tau/T*100$, τ 为一个周期中信号为正的时间长度。

➤ 周期锯齿波或三角波

M 文件名：**sawtooth.m**, Matlab 的内部函数

调用格式 $y=\text{sawtooth}(w_0*t)$ 产生基频为 w_0 (周期 $T=2\pi/w_0$) 的周期锯齿波。为正斜率。

调用格式 $y=\text{sawtooth}(w_0*t, \text{WIDTH})$ 参数 $\text{WIDTH}=0.5$, 产生周期三角波; $\text{WIDTH}=0$, 产生斜率为负的周期锯齿波。

例 2.1-1 画出下列信号的波形图。

- (a) 正弦信号 $\sin(0.5\pi t)$; (b) 门函数 $G_2(t)$; (c) 随机信号;
 (d) 离散信号 $\cos(0.1\pi k)$, $k = 0, \pm 1, \pm 2, \dots$
 (e) 周期 $T=10\text{s}$ 的周期锯齿波信号。
 (f) 指数衰减的余弦信号 $2e^{-t/6} \cos \pi t$ 。

解: 用Matlab计算的程序如下:

```
% 画正弦信号的程序 e2_1_1.m
t0=-2*pi;t1=2*pi;dt=0.02;
t=t0:dt:t1;
f=sin(pi/2*t);
max_f=max(f);
min_f=min(f);
plot(t,f,'linewidth',2);
grid;line([t0 t1],[0 0]);
axis([t0,t1,min_f-0.2,max_f+0.2])
xlabel('t(sec)'),title('正弦信号的波形')
```

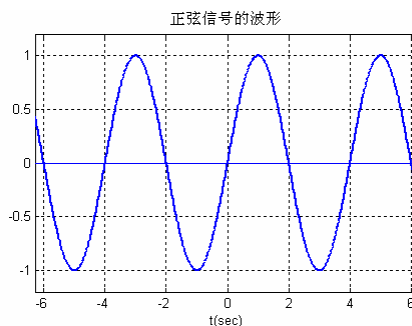


图 2.1-1 正弦信号

运行程序后显示的图形如图 2.1-1 所示。

```
% 画门函数的程序 e2_1_2.m
t0=-3;t1=3;dt=0.02;
t=t0:dt:t1;
f=rectpuls(t,2);
max_f=max(f);
min_f=min(f);
plot(t,f,'linewidth',2);
grid;line([t0 t1],[0 0]);
axis([t0,t1,min_f-0.2,max_f+0.2])
xlabel('t(sec)'),title('门函数的波形')
```

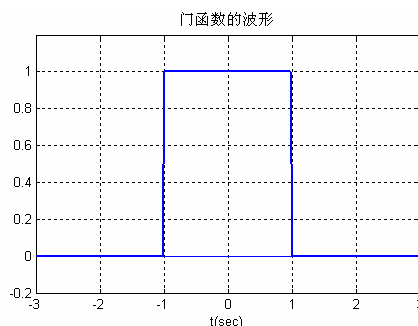


图 2.1-2 门函数的波形

运行程序后显示的图形如图 2.1-2 所示。

```
% 画随机信号的程序 e2_1_3.m
t0=-8;t1=8;dt=0.15;
t=t0:dt:t1;
f=randn(1,length(t));
max_f=max(f);
min_f=min(f);
plot(t,f,'linewidth',2);
grid;line([t0 t1],[0 0]);
axis([t0,t1,min_f-0.2,max_f+0.2])
xlabel('t(sec)'),title('随机信号的波形')
```

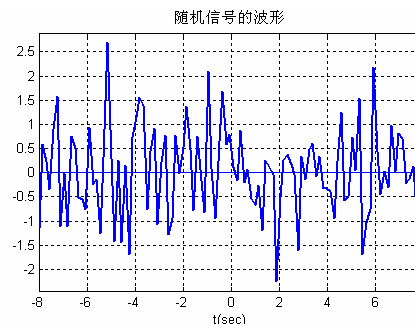


图 2.1-3 随机信

运行程序后显示的图形如图 2.1-3 所示。

```
% 画离散余弦信号的程序 e2_1_4.m
n0=-20;n1=20;
n=n0:n1;
f=cos(pi*n/10);
max_f=max(f);
min_f=min(f);
stem(n,f,'.');
axis([n0,n1,min_f-0.2,max_f+0.2])
xlabel('k'),title('离散余弦信号的波形')
```

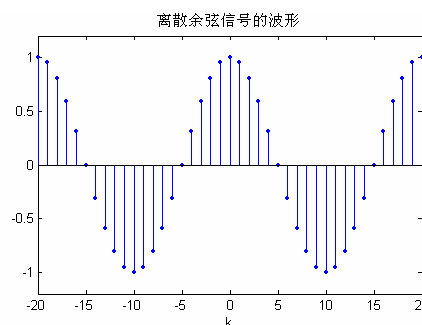


图 2.1-4 离散余弦信号

运行程序后显示的图形如图 2.1-4 所示。

```
% 画周期锯齿波的程序 e2_1_5.m
t0=-6*pi;t1=6*pi;dt=0.05;
t=t0:dt:t1;
f=sawtooth(pi/5*t,0);
max_f=max(f);
min_f=min(f);
plot(t,f,'linewidth',2);
grid;line([t0 t1],[0 0]);
axis([t0,t1,min_f-0.2,max_f+0.2])
xlabel('t(sec)'),title('周期锯齿波的波形')
```

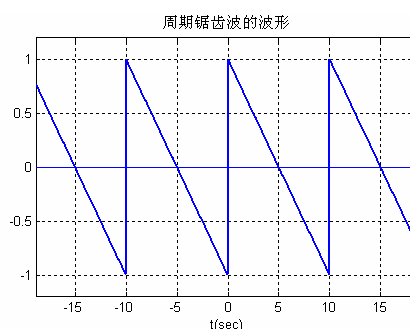


图 2.1-5 周期锯齿波

运行程序后显示的图形如图 2.1-5 所示。

```
% 画指数衰减余弦信号的程序 e2_1_6.m
t0=-4*pi;t1=2*pi;dt=0.01;
t=t0:dt:t1;
f1=2*exp(-t/6).*cos(pi*t);
f2=2*exp(-t/6);
f3=-2*exp(-t/6);
max_f=max(f1);
min_f=min(f1);
plot(t,f1,'linewidth',2);
hold on;
plot(t,f2,':');
hold on;
plot(t,f3,':');
line([t0 t1],[0 0]);
line([0 0],[min_f-0.5 max_f+0.2]);
axis([t0,t1,min_f-0.5,max_f+0.2])
xlabel('t(sec)')
hold off
gtext('2e^{-t/6}cos(\pi t)')
title('指数衰减余弦信号的波形')
```

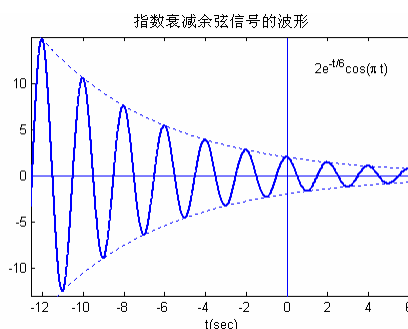


图 2.1-6 指数衰减余弦信号

运行程序后显示的图形如图 2.1-6 所示。

例 2.1-2 画出下列信号的波形图。

$$(a) f_1(t) = t\varepsilon(t) - \sum_{i=1}^{\infty} \varepsilon(t-i); \quad (b) f_2(t) = \sin[\pi t \operatorname{sgn}(t)];$$

$$(c) f_3(t) = (1-0.5|t|)G_4(t); \quad (d) f_4(t) = Sa(t/5-1)$$

解：为了方便地画图，可以编写一个通用的画波形的函数，这样每次画图时调用该函数就可以了。函数名为myplot.m，Matlab程序为

```

function myplot(x,y)
% x为横坐标数组,y为纵坐标数组.
x0=x(1);xe=x(end);
max_y=max(y);min_y=min(y);dy=(max_y-min_y)/10;
plot(x,y,'linewidth',2);grid;
axis([x0,xe,min_y-dy,max_y+dy])
set(gca,'FontSize',8)

```

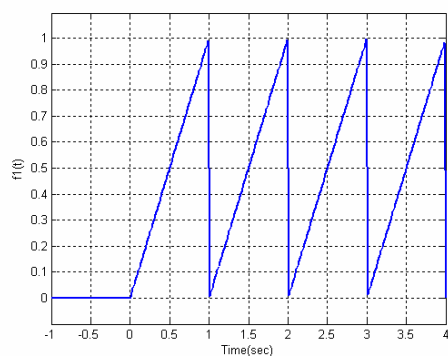
用Matlab画图的程序如下：

```

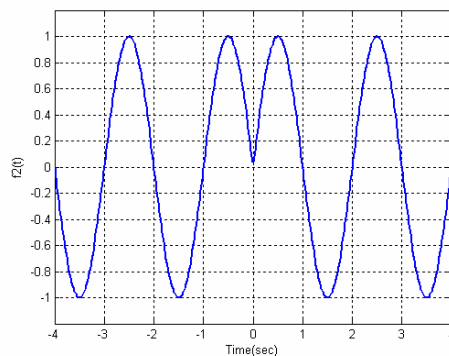
% 画波形图的程序    e2_1_7.m
t=linspace(-1,4,300);
f1=t.*u(t)-u(t-1)-u(t-2)-u(t-3)-u(t-4);
figure(1),myplot(t,f1)
ylabel('f1(t)'),xlabel('Time(sec)')
t=linspace(-4,4,300);
f2=sin(pi*t.*sign(t));
figure(2),myplot(t,f2)
ylabel('f2(t)'),xlabel('Time(sec)')
t=linspace(-4,4,300);
f3=(1-0.5*abs(t)).*rectpuls(t,4);
figure(3),myplot(t,f3)
ylabel('f3(t)'),xlabel('Time(sec)')
t=linspace(-18*pi,18*pi,300);
f4=Sa(t/5-1);
figure(4),myplot(t,f4)
ylabel('f4(t)'),xlabel('Time(sec)')

```

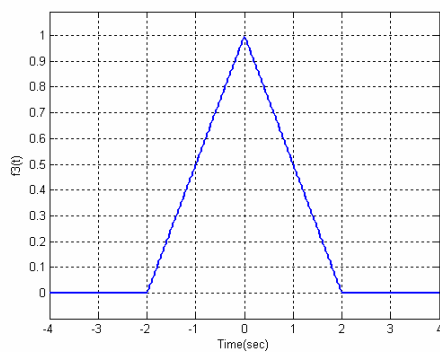
运行程序后显示的图形如图 2.1-7 所示。



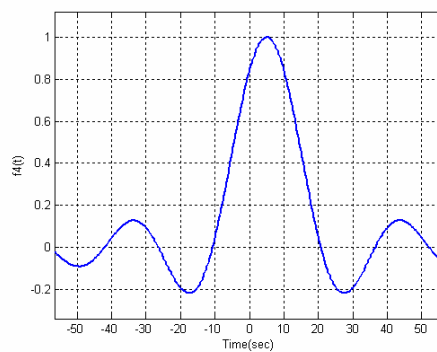
(a)



(b)



(c)



(d)

图 2.1-7 例 2.1-2 的波形

2、周期信号的判断

正弦信号是最典型的周期信号，对于任意给定的频率，正弦信号总是周期的。两个或多个正弦信号的和不一定是周期的，这取决于各个正弦信号的周期或频率之间的关系。正弦信号组合后的周期 T 是每个正弦信号完成整数个周期所用的最小持续时间。它由计算各周期的 LCM（最小公倍数）得出。基频 f_0 是 T 的倒数，它等于各频率的 GCD（最大公约数）。即它们的周期之比为有理数，或它们的频率是可约的，则它们的和是周期信号。

例 2.1-3 观察下面的信号是否是周期信号？

$$(a) f_1(t) = 2\sin\left(\frac{2}{3}\pi t\right) + 4\cos\left(\frac{1}{2}\pi t\right) + 4\cos\left(\frac{1}{3}\pi t - \frac{1}{5}\pi\right)$$

$$(b) f_2(t) = \sin(t) + 3\cos(\pi t)$$

解 (a) $f_1(t)$ 中每个分量的周期（以秒为单位）分别是 3、4 和 6。

$f_1(t)$ 的公共周期是 $T = \text{LCM}(3, 4, 6) = 12\text{ s}$ 。所以， $f_1(t)$ 是周期为 $T = 12\pi$ 的周期信号。

(b) 由于两个分量的频率 $\omega_1 = 1\text{ rad/s}$ 、 $\omega_2 = \pi\text{ rad/s}$ 的比值是无理数，因此无法找出公共周期。所以 $f_2(t)$ 是非周期的。

(a) 程序如下

```
% 观察周期信号的周期 e2_1_8.m
t=linspace(-13, 13, 400);
f=2*sin(2/3*pi*t)+4*cos(0.5*pi*t)+4*cos(1/3*pi*t-1/5*pi);
myplot(t, f)
xlabel('Time(sec)')
[x, y]=ginput(2) % 返回当前鼠标的位置
text(-9, 7, ['\bf 周期:T=', num2str(x(2)-x(1)), ' sec']) % 显示周期
```

程序运行后会在图上出现可动的十字，这就是函数 `ginput(2)` 的作用，移动鼠标使纵线对准波形的最大值按下左键，再移动鼠标使纵线对准波形的另一最大值按下左键。周期就显示在图中。图形如图 2.1-8 所示。显然是周期的，周期 $T \approx 12\text{s}$ 。

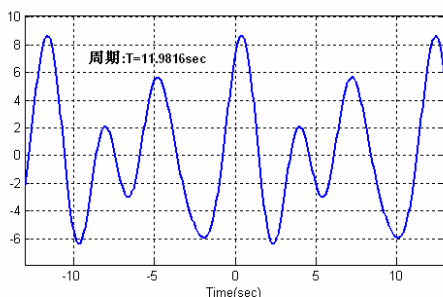


图 2.1-8 观察周期波形

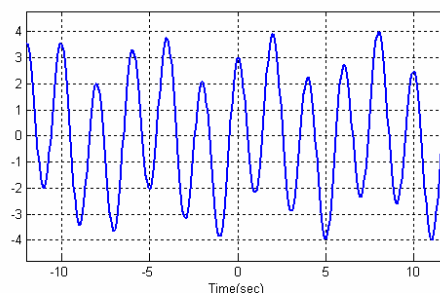


图 2.1-9 观察非周期波形

(b) 程序如下

```
% 观察周期信号的周期 e2_1_9.m
t=linspace(-12, 12, 400);
f=sin(t)+3*cos(pi*t);;
myplot(t, f)
xlabel('Time(sec)')
```

程序运行后的图形如图 2.1-9 所示。显然它不是周期的。

1.3 实验内容

1-1、用MATLAB画出下列信号的波形。

(a) $f_1(t) = \varepsilon[\cos t]$;

(b) $f_2(t) = \frac{|t|}{2}[\varepsilon(t+2) - \varepsilon(t-2)]$;

(c) $f_3(t) = \sin \pi t[\varepsilon(-t) - \varepsilon(2-t)]$;

(d) $f_4(t) = G_2(t)\text{sgn}(t)$;

(e) $f_5 = G_6(t)Q_2(t-2)$;

(f) $f_6(t) = \varepsilon(2-|t|)\sin(\pi t)$

1-2、用基本信号画出图2.1-10中的信号。

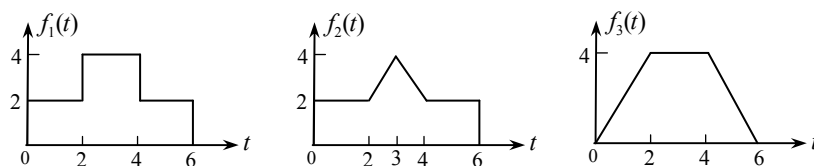


图 2.1-10

1-3 用MATALB画出图2.1-11所示信号。其中，K=10，A=5。

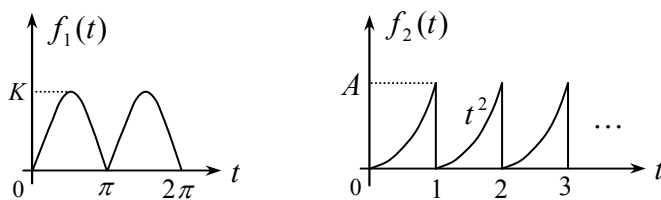


图 2.1-11

1-4 试用 Matlab 绘制出如下连续时间信号的时域波形，并观察信号是否为周期信号。若是周期信号，周期是多少？

(a) $f(t) = 3\sin(\frac{\pi}{2}t) + 2\sin(\pi t) + \sin(2\pi t)$

(b) $f(t) = \sin(t) + 2\cos(4t) + \sin(5t)$

(c) $f(t) = \sin(\pi t) + 2\cos(2t)$

1.4 实验步骤和方法

1、学习例 2.1-1 的基本函数波形的画图方法，将程序中的参数如门函数的宽度、频率的大小、三角或锯齿波的变形等，以便熟悉这些基本函数的用法。

2、仿照例 2.1-1 的方法，完成实验 1-1 的编程。上机调试程序，观察并判别波形的正确性。

3、仿照例 2.1-2 的方法，完成实验 1-2、1-3 的编程。上机调试程序，观察并判别波形的正确性。比较调用自编函数画图的优点。

4、仿照例 2.1-3 的方法，完成实验 1-4 的编程。上机调试程序，观察并判别信号的周期性。与理论分析结果比较。

1.5 预习要点

1. 学习有关 Matlab 的绘画函数的用法。主要绘图函数有：
plot, stem, grid, line, axis, xlabel, ylabel,
hold, title, text, gtext, ginput
以及曲线的颜色、线的粗细等。
自编画图函数：myplot
2. 学习有关基本信号的数学表示法和 MATLAB 表示法。
如门函数、三角波等。
3. 门函数的若干表示法。
4. 复习有关周期信号的判别方法。几个不同频率的周期信号组合后还是周期信号吗？
如何计算组合后周期信号的周期？

1.6 实验报告要求

1. 根据求出的数学表达式编写出的程序。绘出的各种波形图。
2. 上机调试程序的方法。
3. 根据实验归纳、总结出用 Matlab 绘图的方法。
4. 心得体会及其他。
5. 用 MATLAB 显示下列表达式均表示门函数。

数学表达式	MATLAB 表达式
$G_6(t)$	rectpuls(t, 6)
$\varepsilon(t) - \varepsilon(t-4)$	$u(t) - u(t-4)$ rectpuls(t-2, 4)
$\varepsilon(-t) - \varepsilon(-t-4)$	$u(-t) - u(-t-4)$
$\varepsilon(-t) - \varepsilon(-t+4)$	$u(-t) - u(t+4)$
$\varepsilon(2- t)$	$u(2-\text{abs}(t))$
$\varepsilon(-t)\varepsilon(t+4)$	$u(-t) .* u(t+4)$

实验 2 连续信号的微积分和卷积

2.1 实验目的

1. 学习 Matlab 的符号运算功能及编程方法。
2. 掌握用 Matlab 计算微积分的数值方法和解析方法。并应用于求解信号的功率或能量。
3. 掌握用 Matlab 进行卷积运算的数值方法和解析方法。加深对卷积积分的理解。

2.2 实验原理与计算示例

1、微分和积分的数值解

我们知道，连续信号的微分是用差分来近似的，当步长（时间间隔）越小时，用差分表示微分就越精确。见图 2.3-1 所示。

所以，求导数就是近似求差分与步长之比。

即

$$f'(t) \Big|_{t=k} \approx \frac{f(k) - f(k-1)}{h}$$

Matlab 中用 diff 函数来计算差分 $f(k) - f(k-1)$

其调用格式为

$$y = \text{diff}(f)$$

连续信号的定积分可由 Matlab 中的 quad 函数和 quadl 来实现，其调用格式为

quad('function_name', a, b) 采用自适应 Simpson 算法

quadl('function_name', a, b) 采用自适应 Lobatto 算法

其中，function_name 为被积函数名，a, b 为指定的积分区间。

2、微分和积分的解析解

Matlab 除了数值计算以外，还有强大的符号运算功能。在数值计算过程中，参与运算的变量都是被赋了值的数值。而在符号运算的整个过程中，参与运算的是符号变量。在符号运算中所出现的数字都是当做符号来处理的。

Matlab 中，对符号表达式微分的函数是 diff()。利用这个函数，可以求符号表达式的一阶导数、n 阶导数。该函数有三种调用格式

diff(f), diff(f, a), diff(f, n) 或 diff(f, a, n)

其中，f 为符号表达式，a 说明对其求导，n 表示求导次数。

积分运算的函数是 int()，它也有三种调用形式

int(f), int(f, v), int(f, a, b) 或 int(f, v, a, b)

其中，f 为符号表达式，v 说明对其求积分，a, b 表示积分的区间。显然前两个是求不定积分，后两个是求定积分。

例 2.3-1 计算定积分 $s = \int_0^{3\pi} e^{-0.5t} \sin(t + \pi/6) dt$

解 用三种积分函数计算如下

```
>> format long;
>> y=inline('exp(-0.5*t).*sin(t+pi/6)');
>> s=quad(y, 0, 3*pi)
s =
    0.90084081100646
```

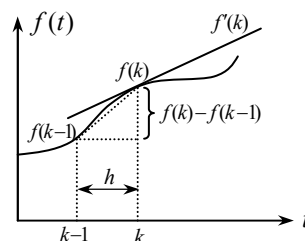


图 2.3-1 用差分表示微分

```
>> s1=quadl(y,0,3*pi)
s1 =
    0.90084078775646
>> sf=int('exp(-0.5*t)*sin(t+pi/6)',0,3*pi)
sf =
    .90084078781888619095323632923836
```

符号积分计算是精确值。

例 2.3-2 画出如图 2.3-2 所示信号的一阶导数和积分的波形图。

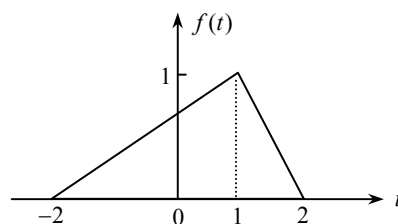


图 2.3-2

解: (a)用Matlab的数值计算方法的程序如下:

```
% 画微分和积分的程序(数值计算)  e2_3_1n.m
clear all
t0=-3;t1=3;dt=0.02;
t=t0:dt:t1;
f1=tripuls(t,4,0.5);
df=diff(f1)/dt;
f=inline('tripuls(t,4,0.5)');
for x=1:length(t)
    intf(x)=quad(f,-3,t(x));
end
subplot(3,1,1),myplot(t,f1);
ylabel('f(t)')
subplot(3,1,2),myplot(t(1:length(t)-1),df);
ylabel('df(t)/dt')
subplot(3,1,3),myplot(t,intf);
ylabel('f(t)的积分'),xlabel('Time(sec)')
```

% 定义三角波
% 求导
% 定义在线函数对象
% 求积分
% 画 f(t)
% 画 f(t)的导数
% 画 f(t)的积分

运行程序后显示的图形如图 2.3-3 所示。

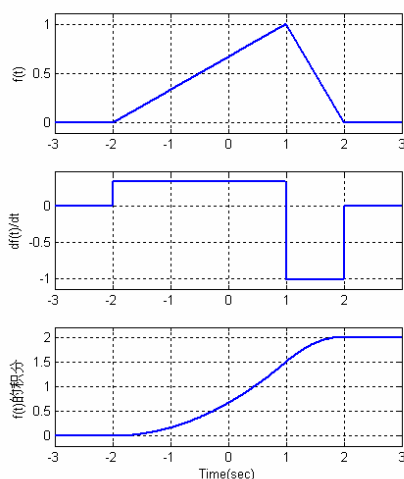


图 2.3-3 用数值计算的微积分波形

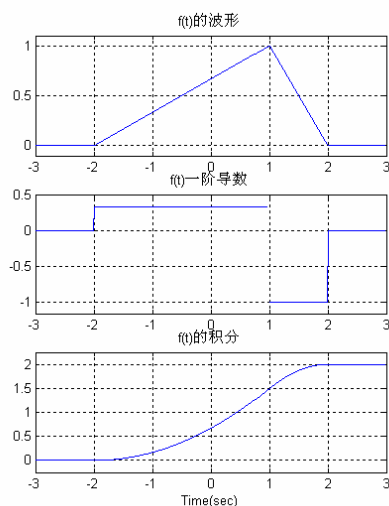


图 2.3-4 用符号计算的微积分

(b)用 Matlab 的符号计算方法的程序如下:

```
% 画微分和积分的程序(符号计算)  e2_3_1s.m
clear all;syms t
f1=sym('1/3*(t+2)*Heaviside(t+2)-4/3*(t-1)*Heaviside(t-1)+(t-2)*Heaviside(t-2)');
```

```
df=diff(f1);df=simple(df);
intf=int(f1);f2=simple(intf);
subplot(3,1,1),ezplot(f1,[-3,3]);xlabel('')
grid;title('f(t)的波形')
subplot(3,1,2),ezplot(df,[-3,3]);xlabel('')
grid;title('f(t)一阶导数')
subplot(3,1,3),ezplot(f2,[-3,3]);
grid;title('f(t)的积分')
```

运行程序后显示的图形如图 2.3-4 所示。其中 Heaviside(t) 表示符号形式的单位阶跃函数。

例 2.3-3 已知三种有用的脉冲波形的信号能量如图 2.3-5 所示, 试用 Matlab 的积分运算来证明。

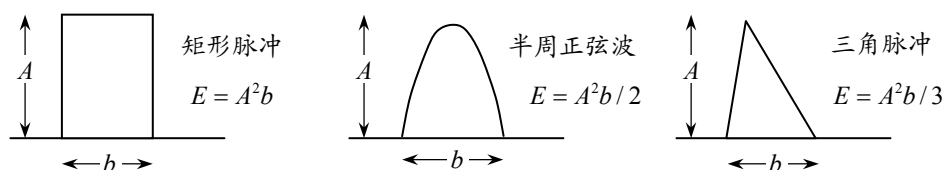


图 2.3-5 三种脉冲波形及能量

解 在 Matlab 的命令窗口输入以下命令

```
>> syms t b A a
>> E=int('A^2',t,0,b)
E =
A^2*b
>> E=int(' (A*sin(pi/b*t))^2',0,b)
E =
1/2*A^2*b
>> E=int(' (A/a*t)^2',t,0,a)+int(' (A/(b-a)*(t-b))^2',t,a,b)
E =
1/3*A^2*a+1/3*A^2/(b-a)^2*(b^3-a^3)-A^2/(b-a)^2*a*(b^2-a^2)+A^2/(b-a)*a^2
>> E=simple(E)
E =
1/3*A^2*b
```

3、卷积积分的数值计算

卷积积分计算实际上可用信号的分段求和来实现, 即

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

如果我们只求当 $t = n\Delta$ (n 为整数) 时 $f(t)$ 的值 $f(n\Delta)$, 则由上式可得:

$$f(n\Delta) = \Delta \sum_{k=-\infty}^{\infty} f_1(k\Delta) f_2[(n-k)\Delta]$$

当时间间隔足够小时, $f(n\Delta)$ 就是 $f(t)$ 的数值近似。Matlab 的 conv(x,h) 函数可以用来计算卷积积分的数值解。为此编写了计算卷积积分的通用函数如下。

```
function x=CSCONV(F1,t1_s,t1_e,F2,t2_s,t2_e);
% 计算有限长连续信号的卷积
% F1,F2 为两信号的字符串,t1_s,t2_s 为两信号的起始点,t1_e,t2_e 为两信号的终止点
% 同时显示 f1(t),f2(t) 和 f(t)=f1*f2 三个波形,并且三个波形自动显示在图形中间。
```

```

% 例: f1='2*rectpuls(t,4)'; f2='3*rectpuls(t-0.5,3)'; figure(1)
%      CSCONV(f1,-2,2,f2,-1,2)
%
t_s=t1_s+t2_s;t_e=t1_e+t2_e;
t0=min([t1_s,t2_s,t_s])-1;t1=max([t1_e,t2_e,t_e])+1;dt=0.005;
t=t0:dt:t1;
L=length(t);
tp=[2*t(1):dt:2*t(L)];
f1=eval(F1);f2=eval(F2);
y=dt*conv(f1,f2);y_max=max(y);y_min=min(y);dy=(y_max-y_min)/10;
f1_max=max(f1);f1_min=min(f1);df1=(f1_max-f1_min)/10;
f2_max=max(f2);f2_min=min(f2);df2=(f2_max-f2_min)/10;
subplot(3,1,1),plot(t,f1,'linewidth',2);title('信号 f1(t)的波形','color','b','FontSize',8)
axis([t(1) t(L) f1_min-df1 f1_max+df1]);grid;set(gca,'FontSize',8)
subplot(3,1,2),plot(t,f2,'linewidth',2);title('信号 f2(t)的波形','color','b','FontSize',8)
axis([t(1) t(L) f2_min-df1 f2_max+df1]);grid;set(gca,'FontSize',8)
subplot(3,1,3),plot(tp,y,'linewidth',2);
title('卷积 f(t)=f1*f2 的波形','color','b','FontSize',8)
if y_min>=0
    y0=(y_max-y_min)/2;
else
    y0=0;
end
axis([t(1),t(L),y_min-dy,y_max+dy]);grid;
set(gca,'Ytick',[y_min,y0,y_max],'FontSize',8)
xlabel('Time(sec)')

```

调用以上函数可以方便地计算画出两个任意有限长波形和卷积波形。用下面的例子说明。

例 2.3-4 已知信号 $f_1(t)$ 和 $f_2(t)$ 的波形如图 2.3-6(a)、(b)所示(其中令 $A=2, B=3$)。计算卷积积分 $f(t) = f_1(t) * f_2(t)$ 。

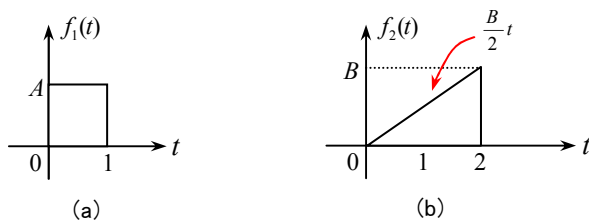


图 2.3-6 两信号波

解 MATLAB 程序如下:

```

f1='2*rectpuls(t-0.5)';
f2='3*tripuls(t-1,2,1)';
figure(1)
CSCONV(f1,0,1,f2,0,2)

```

程序运行后显示的波形如图 2.3-7 所示。

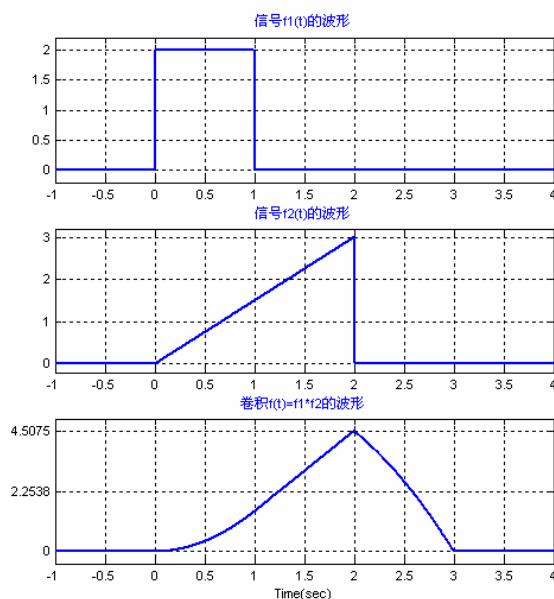


图 2.3-7 信号的卷积波形

4、卷积积分的符号计算

Matlab 不仅有数值计算功能，还有强大的符号计算功能。即有推理功能可得解析表达式。卷积积分用 Matlab 的积分 `int()` 函数，可以计算不定积分和定积分。调用格式为

$$\text{int}(f), \text{int}(f, a, b)$$

其中， f 为被积函数， a, b 为积分的上下限。为此编写了计算卷积积分解析解的通用函数如下：

```
function y=CSCONVS(f,h,t_s,t_e,a,b);
% 计算卷积积分的解析解
% f 激励信号, 含有阶跃函数 Heaviside(t);
% h 冲激响应, 卷积时要反折的信号, 不可含阶跃函数, 默认起始点为 0;
% t_s,t_e 为系统零状态响应 y=f*h 波形的起始点和终止点,
% a,b 为卷积积分的上, 下限. 例: 计算 exp(-2t)u(t+1) 与 u(t-3) 卷积.
% syms t
% h=sym('exp(-2*t+2)');
% f=sym('Heaviside(t-2)');
% CSCONVS(f,h,1,5,2,t)
%
syms t tao
ftao=subs(f,t,tao);
ht_tao=subs(h,t,t-tao);
y=simple(int(ftao*ht_tao,tao,a,b));
t=t_s:0.02:t_e;
yt=subs(y);y_max=max(yt);y_min=min(yt);dy=(y_max-y_min)/10;
plot(t,yt,'linewidth',2),
axis([t_s t_e y_min-dy y_max+dy]);%axis([-2 3 0 11]);
xlabel('t(sec)');grid;
if y_min>=0
    y0=(y_max-y_min)/2;
else
    y0=0;
end
set(gca,'Ytick',[y_min,y0,y_max],'FontSize',8)
title('卷积的波形');
disp('零状态响应'),y
```

现用实例说明其计算方法。

例 2. 3-5 线性非时变系统的输入信号 $f(t)$ 和冲激响应 $h(t)$ 由下列各式给出，试求系统的零状态响应 $y_{zs}(t)$ 。

(a) $f(t) = e^{-0.5t} [\varepsilon(t) - \varepsilon(t-2)]$, $h(t) = e^{-t} \varepsilon(t)$;

(b) $f(t) = e^{-2t} \varepsilon(t+3)$, $h(t) = e^{-2t} \varepsilon(t-1)$ 。

解 (a) 系统的零状态响应为

$$\begin{aligned} y_{zs}(t) &= f(t) * h(t) = \int_{-\infty}^{\infty} e^{-0.5\tau} [\varepsilon(\tau) - \varepsilon(\tau-2)] e^{-(t-\tau)} \varepsilon(t-\tau) d\tau \\ &= \int_{-\infty}^{\infty} e^{-0.5\tau} e^{-(t-\tau)} \varepsilon(\tau) \varepsilon(t-\tau) d\tau - \int_{-\infty}^{\infty} e^{-0.5\tau} e^{-(t-\tau)} \varepsilon(\tau-2) \varepsilon(t-\tau) d\tau \\ &= [e^{-t} \int_0^t e^{0.5\tau} d\tau] \varepsilon(t) - [e^{-t} \int_2^t e^{0.5\tau} d\tau] \varepsilon(t-2) \\ &= 2(e^{-0.5t} - e^{-t}) \varepsilon(t) - 2(e^{-0.5t} - e^{-(t-1)}) \varepsilon(t-2) \end{aligned}$$

(b) 系统的零状态响应为

$$\begin{aligned} y_{zs}(t) &= f(t) * h(t) = \int_{-\infty}^{\infty} e^{-2\tau} \varepsilon(\tau+3) [e^{-2(t-\tau)} \varepsilon(t-\tau-1)] d\tau \\ &= [e^{-2t} \int_{-3}^{t-1} d\tau] \varepsilon(t+2) = (t+2) e^{-2t} \varepsilon(t+2) \end{aligned}$$

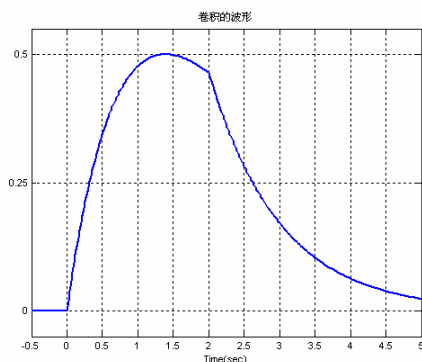
为了用 CSCONVS () 函数计算，利用卷积的时移性质将被积函数要变换如下：

$$\begin{aligned} y_{zs}(t) &= e^{-2t} \varepsilon(t+3) * e^{-2t} \varepsilon(t-1) \\ &= e^{-2(t-1)} \varepsilon(t+2) * e^{-2(t+1)} \varepsilon(t) \end{aligned}$$

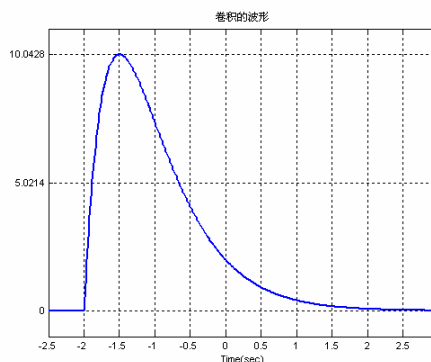
用 MATLAB 并调用 CSCONVS () 函数的计算程序为

```
syms t
h=sym('exp(-t)');
f=sym('exp(-t/2)*(Heaviside(t)-Heaviside(t-2))');
figure(1)
y1=CSCONVS(f,h,-0.5,5,0,t)
h=sym('exp(-2*t-2)');
f=sym('exp(-2*t+2)*(Heaviside(t+2))');
figure(2)
y2=CSCONVS(f,h,-2.5,3,-2,t)
disp('零状态响应'),y1,y2,
```

运行程序后显示的图形如图 2.3-8(a)、(b)所示。



(a) $y_1(t)$ 的波形



(b) $y_2(t)$ 的波形

图 2.3-8

在命令窗口显示其卷积的解析式如下：

>>零状态响应

```
y1 =
2*Heaviside(t)*exp(-1/2*t)-2*Heaviside(t)*exp(-t)-2*Heaviside(t-2)*exp(-1/2*t)+2*Heaviside(t-2)*exp(1-t)
y2 =
Heaviside(t+2)*(t+2)*exp(-2*t)
```

其中，Heaviside(t)表示阶跃函数 $\varepsilon(t)$ ，所以，

$$y_1(t) = 2(e^{-t/2} - e^{-t})\varepsilon(t) - 2(e^{-t/2} - e^{-(t-1)})\varepsilon(t-2)$$

$$y_2(t) = (t+2)e^{-2t}\varepsilon(t+2)$$

与理论计算结果一致。

2.3 实验内容

3-1、周期信号如题图2.3-9所示，试计算信号的功率。

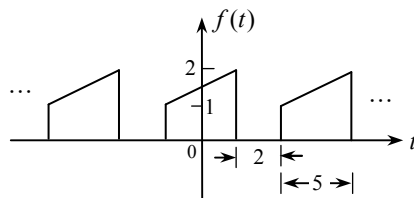


图 2.3-9

3-2、求图2.3-10中的信号的能量。

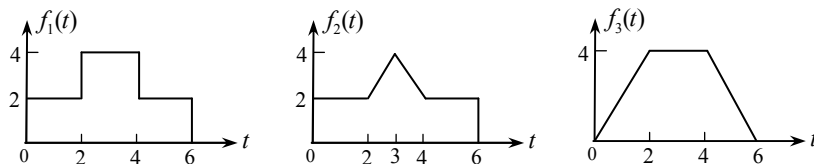


图 2.3-10

3-3、用 Matlab 画出图 2.3-11 中的信号的卷积波形。

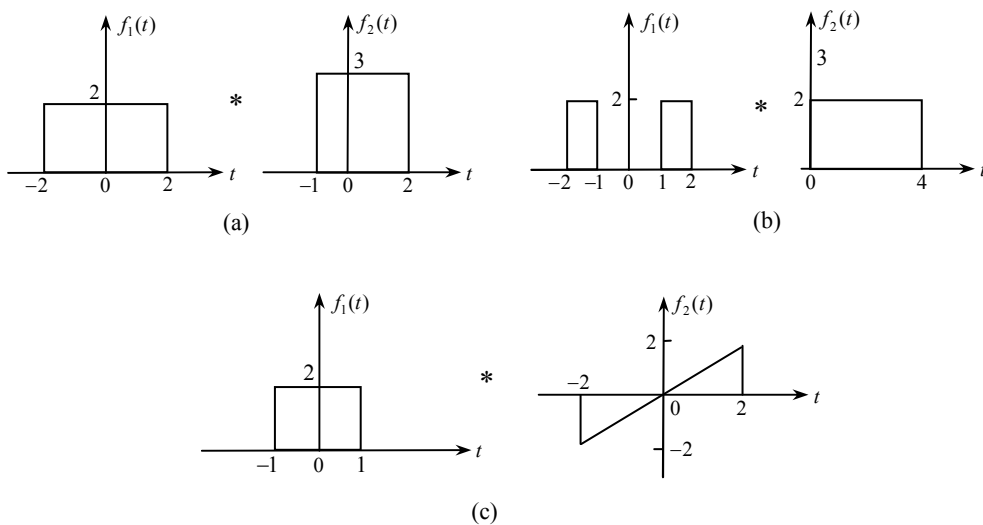


图 2.3-11

3-4、用符号积分求下列函数的卷积积分：

- (a) $f_1(t) = t\varepsilon(t)$, $f_2(t) = \varepsilon(t)$;
- (b) $f_1(t) = e^{-2t}\varepsilon(t+1)$, $f_2(t) = \varepsilon(t-3)$;
- (c) $f_1(t) = t\varepsilon(t+1)$, $f_2(t) = (t+1)\varepsilon(t)$;
- (d) $f_1(t) = e^{-2t}\varepsilon(t)$, $f_2(t) = e^{-3t}\varepsilon(t)$ 。

2.4 实验步骤和方法

1、对实验 3-1，找出图 2.3-9 周期信号的周期 T ，用 Matlab 的数值积分或符号积分计算一个周期的能量 E ，再计算功率 $P=E/T$ 。

2、用 Matlab 的数值积分或符号积分计算实验 3-2 信号的能量 E 。用符号积分时可以分段积分再相加。

3、仿照例 2.3-4 的卷积的数值计算方法，调用函数 CSCONV 完成实验 3-3 的编程。上机调试程序，观察并判别卷积波形的正确性。

3、仿照例 2.3-5 的卷积的符号计算方法，调用函数 CSCONVS 完成实验 3-4 的编程。上机调试程序，观察并判别卷积波形和解析表达式的正确性。并与理论计算加以比较。

2.5 预习要点

- 学习有关 Matlab 的微积分运算函数的用法。主要函数有：
quad, quadl, diff, int, conv。
学习常用函数
length, Heaviside, syms, sym, subs, disp, inline.
format long, ezplot
- 复习有关计算信号的能量或功率的方法。观察图 2.3-10 的波形具有对称性，能否用其对称性来计算信号的能量？即只要计算信号一半的能量再乘 2。
- 复习有关计算卷积的方法。用符号积分法计算无时限信号的卷积，其积分上下限如何确定？对超前或滞后的波形如何处理？
- 函数 CSCONV 是如何将画图坐标的刻度设置成自动坐标的刻度？即使波形始终处于图形的中间。
- 学习自编函数 CSCONV 和 CSCONVS 的编程方法。

2.6 实验报告要求

- 根据求出的数学模型所编写出的程序。以及绘出的各种波形图。
- 上机调试程序的方法。
- 根据实验归纳、总结出用 Matlab 计算积分和卷积的方法。
- 心得体会及其他。

实验 3 非周期信号的频谱

3.1 实验目的

1. 学习 Matlab 的符号运算功能中计算傅里叶变换和反变换的方法。
2. 掌握用 Matlab 绘制非周期信号频谱的数值方法和符号方法。
3. 通过对非周期信号频谱的绘制。加深对傅里叶变换性质的理解。

3.2 实验原理与计算示例

1、傅里叶变换和反变换的符号运算

MATLAB 的符号运算工具箱中，专门提供了傅里叶变换和反变换的函数。正变换的调用格式为

$$F=\text{fourier}(f)$$

式中， f 为时间函数的符号表达式， F 为傅里叶变换式，也是符号表达式。

反变换的调用格式为

$$f=\text{ifourier}(F)$$

式中， F 为傅里叶变换式的符号表达式， f 为时间函数，是符号形式。

为了改善公式的可读性，MATLAB 提供了 pretty 函数，调用格式为

$$\text{Pretty}(f)$$

式中， f 为符号表达式。

如已知 $f(t) = e^{-2t}\varepsilon(t)$ ，求其频谱 $F(j\omega)$ 。

```
>> syms t w
>> f=sym('exp(-2*t)*Heaviside(t)')
f =
exp(-2*t)*Heaviside(t)
>> F=fourier(f)
F =
1/(2+i*w)
>> pretty(F)
```

$$\frac{1}{2 + i w}$$

```
>> f1=ifourier(F,t)
f1 =
exp(-2*t)*Heaviside(t)
```

例 3.2-1 求如图 3.2-1 所示信号的傅里叶变换。

解 用 Matlab 的符号运算功能可以很方便地求出傅里叶变换，本题计算如下

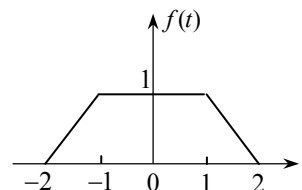


图 3.2-1

梯形波可表示为

$$f(t) = (t+2)\varepsilon(t+2) - (t+1)\varepsilon(t+1) - (t-1)\varepsilon(t-1) + (t-2)\varepsilon(t-2)$$

用 MATLAB 命令计算如下：

```
>> ft=sym('(t+2)*Heaviside(t+2)-(t+1)*Heaviside(t+1)-(t-1)*Heaviside(t-1)+(t-2)*Heaviside(t-2)')
ft =
(t+2)*Heaviside(t+2)-(t+1)*Heaviside(t+1)-(t-1)*Heaviside(t-1)+(t-2)*Heaviside(t-2)
```

```

>> F=fourier(ft)
F =
i*(2*i*exp(2*i*w)*(pi*Dirac(w)-i/w)+exp(2*i*w)*(pi*Dirac(1,w)+i/w^2))+2*exp(2*i*w)*(pi*
Dirac(w)-i/w)-i*(i*exp(i*w)*(pi*Dirac(w)-i/w)+exp(i*w)*(pi*Dirac(1,w)+i/w^2))-exp(i*w)*(pi*D
irac(w)-i/w)-i*(-i*exp(-i*w)*(pi*Dirac(w)-i/w)+exp(-i*w)*(pi*Dirac(1,w)+i/w^2))+exp(-i*w)*(p
i*Dirac(w)-i/w)+i*(-2*i*exp(-2*i*w)*(pi*Dirac(w)-i/w)+exp(-2*i*w)*(pi*Dirac(1,w)+i/w^2))-2*exp
(-2*i*w)*(pi*Dirac(w)-i/w)
>> F1=simple(F)
F1 =
-2*(cos(2*w)-cos(w))/w^2

```

其中 $\text{Dirac}(t)$ 表示冲激函数 $\delta(t)$, $\text{Heaviside}(t)$ 表示阶跃函数 $\varepsilon(t)$ 。simple(F) 表示对函数 F 进行化简。经过化简后, 傅里叶变换为

$$F(j\omega) = \frac{-2[\cos(2\omega) - \cos(\omega)]}{\omega^2} \quad \text{与理论计算结果相}$$

同。

例 3.2-2 已知信号 $f(t)$ 的频谱 $F(j\omega)$, 求时间函数 $f(t)$ 。

$$(a) \quad F(j\omega) = \frac{2\sin[3(\omega - 2\pi)]}{\omega - 2\pi}; \quad (b) \quad F(j\omega) = \cos(4\omega + \frac{\pi}{3})$$

解 用 Matlab 的符号运算功能可以很方便地求出傅里叶反变换, 本题计算如下:

$$(a) \quad F(j\omega) = \frac{2\sin[3(\omega - 2\pi)]}{\omega - 2\pi} \text{ 的傅里叶反变换}$$

```

>> syms w t
>> F='2*sin(3*(w-2*pi))/(w-2*pi)'
F =
2*sin(3*(w-2*pi))/(w-2*pi)
>> f=ifourier(F,t)
f =
1/2*exp(2*i*pi*(t+3))*Heaviside(t+3)-1/2*exp(2*i*pi*(t+3))*Heaviside(-t-3)-1/2*exp(2*i*
pi*(t-3))*Heaviside(t-3)+1/2*exp(2*i*pi*(t-3))*Heaviside(-t+3)
>> f1=simple(f)
f1 =
exp(2*i*pi*(t+3))*Heaviside(t+3)-1/2*exp(2*i*pi*(t+3))-exp(2*i*pi*(t-3))*Heaviside(t-3)
+1/2*exp(2*i*pi*(t-3))
>> f1=simple(f1)
f1 =
exp(i*pi*t)^2*Heaviside(t+3)-exp(i*pi*t)^2*Heaviside(t-3)
>> f1=simple(f1)
f1 =
-exp(i*pi*t)^2*(-Heaviside(t+3)+Heaviside(t-3))
>> f1=simple(f1)
f1 =
exp(2*i*pi*t)*(Heaviside(t+3)-Heaviside(t-3))

```

经过 4 次化简, 原函数为 $f(t) = e^{j2\pi t}[\varepsilon(t+3) - \varepsilon(t-3)]$ 与理论计算结果相同。

$$(b) \quad F(j\omega) = \cos(4\omega + \frac{\pi}{3}) \text{ 的傅里叶反变换}$$

```

>> syms w t
>> F='cos(4*w+pi/3)'
F =
cos(4*w+pi/3)
>> f=ifourier(F,t)
f =
1/4*Dirac(t+4)+1/4*i*3^(1/2)*Dirac(t+4)+1/4*Dirac(t-4)-1/4*i*3^(1/2)*Dirac(t-4)

```

表达式为 $f(t) = \frac{1}{4}\delta(t+4)(1+j\sqrt{3}) + \frac{1}{4}\delta(t-4)(1-j\sqrt{3})$ 与理论计算结果相同。

2、傅里叶变换的数值计算

严格说来,若要分析连续信号,必须使用 symbolic box (符号工具箱)。为了更好地体会 MATLAB 的数值计算功能,特别是其强大的矩阵运算能力,我们给出连续信号傅立叶变换的数值计算方法。先给出算法的理论依据如下:

$$F(j\omega) = \int_{-\infty}^{+\infty} f(t)e^{-j\omega t} dt = \lim_{\tau \rightarrow 0} \sum_{n=-\infty}^{n=\infty} f(n\tau)e^{-j\omega n\tau} \quad (3)$$

对于一大类信号,当取 τ 足够小时,上式的近似情况可以满足实际需要。若信号 $f(t)$ 是时限的,或当 $|t|$ 大于某个给定值时, $f(t)$ 的值已衰减得很厉害,可以近似地看成时限信号时,则 (3) 式中的 n 取值就是有限的,设为 N ,有:

$$F(k) = \tau \sum_{n=0}^{N-1} f(n\tau)e^{-j\omega_k n\tau}, \quad 0 \leq k \leq N \quad (4)$$

上式是对 (3) 中的频率进行取样,通常:

$$\omega_k = \frac{2\pi}{N\tau} k$$

采用 MATLAB 实现 (3) 式时,其要点是要正确生成 $f(t)$ 的 N 个样本 $f(n\tau)$ 的向量 f 及向量 $e^{-j\omega_k n\tau}$, 两向量的内积 (即两矩阵的乘积) 结果即完成 (3) 式的计算。

为此编写了计算任意非周期信号频谱的通用函数 CXHFT()。只要输入信号的表达式,以及原信号的时间范围和频谱的频率范围。就可以画出原时间波形以及信号的幅度频谱和相位频谱。

```
function y=CXHFT(x,tn,wn);
% 计算连续非周期信号的频谱
% x 为非周期信号的字符串表达式, tn 为时间信号的起始时间和终止时间, wn 为显示频谱的范围。
% 例 1: 计算任意三角波的频谱
% f='tripuls(t,4,0.5)';t=[-3,3];w=[-15,15];
% CXHFT(f,t,w)
t1=tn(1);
t2=tn(2);
w1=wn(1);
w2=wn(2);
t=t1:0.01:t2;
N=500;W=6*pi*2;k=-N:N;w=k*W/N;
ft=eval([x]);
fmax=max(ft);
fmin=min(ft);df=(fmax-fmin)*0.1;
F=ft*exp(-j*t'*w)*0.01;
F1=abs(F);Fmax=max(F1);Fmin=min(F1);dF=(Fmax-Fmin)*0.1;
P1=angle(F)*180/pi;Pmax=max(P1);Pmin=min(P1);
subplot(3,1,1),plot(t,ft,'linewidth',2),grid;ylabel('f(t)'),
title('连续信号 f(t) 的波形', 'FontSize',8);
axis([t1,t2,fmin-df,fmax+df]);set(gca,'FontSize',8);
subplot(3,1,2),plot(w,F1,'linewidth',2),grid;ylabel('F(jw) 的模'),
title('连续信号的幅度频谱', 'FontSize',8);
axis([w1,w2,Fmin-dF,Fmax+dF]);
if Fmin>0
    F0=(Fmax-Fmin)/2;
else
    F0=0;
end
```

```

set(gca,'Ytick',[Fmin,F0,Fmax],'FontSize',8)
subplot(3,1,3),plot(w,P1,'linewidth',2),grid;xlabel('\omega'),ylabel('相位(度)');
axis([w1,w2,Pmin-45,Pmax+45]);title('连续信号的相位频谱','FontSize',8);
if Pmin>0
    P0=(Pmax-Pmin)/2;
else
    P0=0;
end
if round(Pmin)~=round(Pmax)
    set(gca,'Ytick',[round(Pmin),P0,round(Pmax)],'FontSize',8)
end
end

```

调用以上函数可以方便地计算画出任意非周期信号的频谱。

例 3.2-3 已知非周期信号如图 3.2-2 所示。用 Matlab 画出其频谱图。

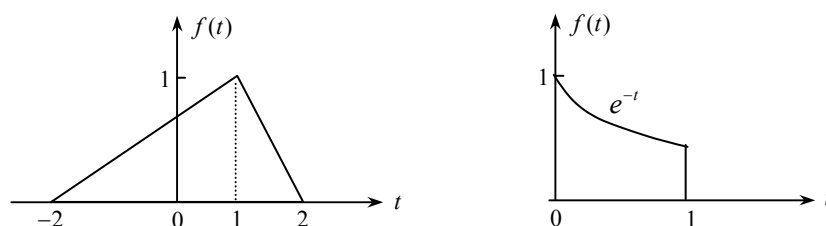


图 3.2-2 两个非周期信号

程序如下。

```

% 非周期信号的频谱图 e3_2_3.m
f='tripuls(t,4,0.5)';t=[-3,3];w=[-15,15]; % 信号的表达式，时间区间，频率区间。
figure(1)
CXHFT(f,t,w)
f='exp(-t).*(u(t)-u(t-1))';t=[-1,2];w=[-35,35]; % 信号的表达式，时间区间，频率区间。
figure(2)
CXHFT(f,t,w)

```

运行程序后显示的图形如图 3.2-3 所示。

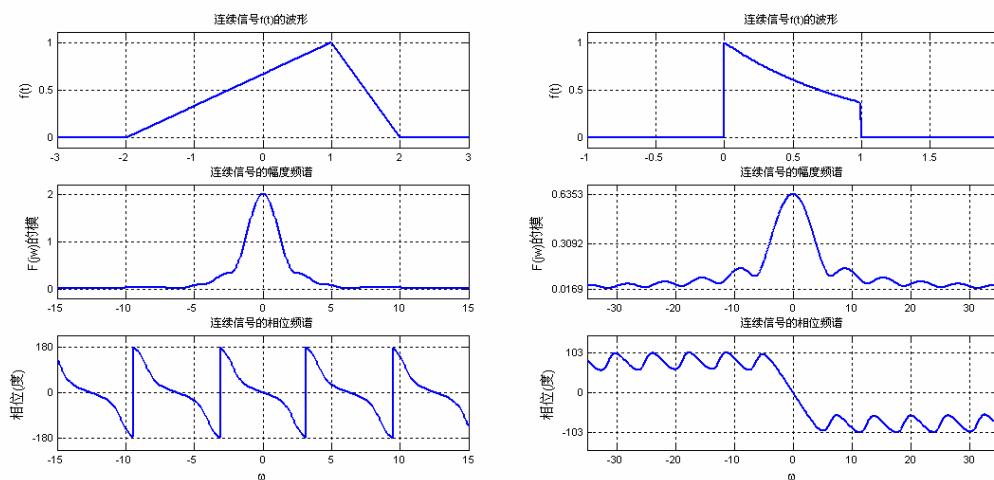


图 3.2-3 非周期信号的频谱图

3.3 实验内容

6-1、考虑非周期连续时间信号如单边指数，将其平移后观察其频谱的变化。验证傅里叶变换的时移性质。

6-2、考虑非周期连续时间信号如三角波，将其用调制为 $f(t) = Q_{\tau}(t) \cos \omega_0 t$ 。观察其频谱的变化。验证傅里叶变换的调制性质。

6-3、考虑非周期连续时间信号如门函数，将其进行尺度变换后观察其频谱的变化。验证傅里叶变换的尺度变换性质。

6-4、求下列信号的傅里叶变换。画出 $f_1(t)$ 和 $f_2(t)$ 的幅度谱和相位谱，并进行比较。画出 $f_3(t)$ 和 $f_4(t)$ 的幅度谱和相位谱，并进行比较。（令 $A=2$, $a=3$ ）

$$\begin{aligned} \text{(a)} \quad f_1(t) &= Ae^{-at} \varepsilon(t); & \text{(b)} \quad f_2(t) &= Ae^{at} \varepsilon(-t); \\ \text{(c)} \quad f_3(t) &= e^{-a|t|}; & \text{(d)} \quad f_4(t) &= Ae^{-at} \varepsilon(t) - Ae^{at} \varepsilon(-t)。 \end{aligned}$$

6-5、用 Matlab 的符号运算功能计算下列信号的傅里叶变换 $F(j\omega)$ 。

$$\begin{aligned} \text{(a)} \quad f(t) &= e^{-2|t-1|}; & \text{(b)} \quad f(t) &= e^{-2t} \cos(2\pi t) \varepsilon(t); \\ \text{(c)} \quad f(t) &= \frac{\sin 2\pi(t-2)}{\pi(t-2)}; & \text{(d)} \quad f(t) &= G_1(t-0.5)。 \end{aligned}$$

6-6、用 Matlab 的符号运算功能计算下列频谱 $F(j\omega)$ 的傅里叶反变换 $f(t)$ 。

$$\begin{aligned} \text{(a)} \quad F(j\omega) &= \frac{j\omega}{1+\omega^2}; & \text{(b)} \quad F(j\omega) &= \frac{e^{-j2\omega}}{1+\omega^2}; \\ \text{(c)} \quad F(j\omega) &= G_2(\omega+5) + G_2(\omega-5); & \text{(d)} \quad F(j\omega) &= \frac{2\sin^2 \omega}{\omega^2}; \end{aligned}$$

3.4 实验步骤和方法

1、仿照例 3.2-3 做实验 6-1，先画单边指数 $e^{-t} \varepsilon(t)$ 的频谱图，再画平移后的 $e^{-(t+1)} \varepsilon(t+1)$ 的频谱图。比较两个频谱图，加深对傅里叶变换时移性质的认识。

2、仿照例 3.2-3 做实验 6-2，先画三角波 $Q_{2.5}(t)$ 的频谱图，再画调制后的 $Q_{2.5}(t) \cos 10t$ 的频谱图。比较两个频谱图，加深对傅里叶变换调制性质的认识。

3、仿照例 3.2-3 做实验 6-3，先画门函数 $G_2(t)$ 的频谱图，再画调制后的 $G_4(t)$ 的频谱图。比较两个频谱图，加深对傅里叶变换尺度变换性质的认识。

4、仿照例 3.2-3 的方法，完成实验 6-4 的编程。上机调试程序，根据题目要求对幅度谱和相位谱加以比较。

5、仿照例 3.2-1、2 的符号计算方法，完成实验 6-5、6-6 的计算。并与理论计算进行比较。

3.5 预习要点

1. 学习有关 Matlab 函数的用法。主要函数有：

fourier, ifourier, pretty, Heaviside, Dirac, round
abs, angle, simple.

2. 复习有关傅里叶变换的性质。信号平移则幅度频谱和相位频谱会改变吗？什么是调制性质？信号的尺度变换对频谱带宽有何影响？

3. 学习有关 Matlab 编写函数的方法。如何将使用率较高的程序改编成通用函数？

4. 进一步学习 Matlab 的符号运算功能，学习化简函数的方法、奇异函数的表示方法。

3.6 实验报告要求

1. 根据实验内容编写出的程序。以及绘出的各种波形图和频谱图。
2. 对各种频谱图加以比较说明。并用理论计算验证。
3. 根据实验归纳、总结出用 **Matlab** 绘制信号频谱图的方法。
4. 心得体会及其他。

实验 4 连续系统的零极点分析

4.1 实验目的

- 1、学习用 Matlab 绘制连续系统零极点分布图、冲激响应波形、频率响应曲线图。
- 2、通过运行系统零极点分布与冲激响应的关系的演示程序，加深系统零极点分布对时域响应的影响。从而建立系统稳定性的概念。
- 3、研究系统零极点分布与频率响应的关系，学习用 Matlab 研究频率响应的方法。

4.2 实验原理与计算示例

1、系统函数及其曲面图

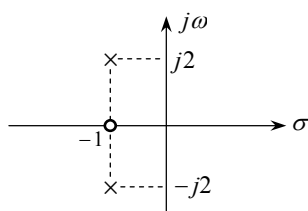
系统函数 $H(s)$ 是复变量 s 的复函数，为了便于理解和分析 $H(s)$ 随 s 的变化规律， $H(s)$ 可以写成

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

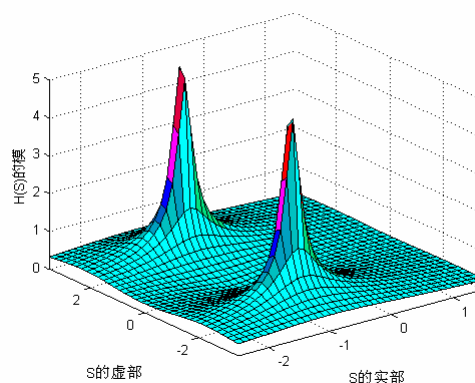
其中， $|H(s)|$ 是复信号 $H(s)$ 的模，而 $\varphi(s)$ 为 $H(s)$ 的相角。而 $s = \sigma + j\omega$ ，即 $|H(s)|$ 和 $\varphi(s)$ 同时也是 σ 和 ω 两个变量的函数，所以可在三维空间中把它表示为随 σ 和 ω 变化的曲面，即曲面图。如已知系统函数

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

$H(s)$ 的零极点图和 $|H(s)|$ 的曲面图 4.2-1 所示。



(a) $H(s)$ 的零极点分布图



(b) $|H(s)|$ 的曲面图

图 4.2-1 $H(s)$ 的零极点分布图和曲面图

从图中所示的曲面图我们可以明确看出，曲面图在 $s = -1 \pm j2$ 处有二个峰点，对应着系统函数的极点位置，而在 $s = -1$ 处有一个谷点，对应着系统函数的零点位置。因此，系统的零极点位置，决定了其曲面图的峰点的谷点位置。

2、系统零极点分布与冲激响应的关系

已知系统函数为

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

若分子多项式 $N(s)$ 的阶次高于分母多项式, 即 $m \geq n$, 则 $H(s)$ 可分解为 s 的有理多项式与 s 的有理真分式之和。有理多项式部分比较容易分析, 故我们讨论 $H(s)$ 为有理真分式的情况, 即上式中 $m < n$ 的情况。

设系统函数 $H(s)$ 具有单极点时, 系统函数 $H(s)$ 可按部分分式法展开为

$$H(s) = \sum_{i=1}^n \frac{K_i}{s - p_i}$$

系统的冲激响应 $h(t)$ 为

$$h(t) = \sum_{i=1}^n K_i e^{p_i t} \varepsilon(t)$$

从上式可见, 冲激响应 $h(t)$ 的性质完全由系统函数 $H(s)$ 的极点 p_i 决定。 p_i 称为系统的自然频率或固有频率。而待定系数 K_i 由零点和极点共同决定。

系统零极点分布与冲激响应有如下关系

- 极点决定了冲激响应 $h(t)$ 的形式, 而各系数 K_i 则由零、极点共同决定。
- 系统的稳定性由极点在 s 平面上的分布决定, 而零点不影响稳定性。
- 极点分布在 s 左半平面, 系统是稳定的。极点在虚轴上有单极点, 系统是临界稳定。

极点在 s 右半平面或在虚轴上有重极点, 系统不稳定。

Matlab 提供了画系统零极点图的函数。其一般调用形式为

pzmap(num, den)

画系统冲激响应的函数。其一般调用形式为

impz(num, den, t)

画系统阶跃响应的函数。其一般调用形式为

step(num, den, t)

式中的 num 为系统函数 $H(s)$ 的有理多项式中分子多项式的系数向量, den 为分母多项式的系数向量。t 为时间抽样点向量。

例 4.2-1 已知系统函数为

$$H(s) = \frac{s^2 - 2s + 0.8}{s^3 + 2s^2 + 2s + 1}$$

试用 Matlab 画出系统的零极点分布图、冲激响应波形、阶跃响应波形。

解 Matlab 的程序如下

```
% 画零极点分布图, 冲激响应, 阶跃响应  e4_2_1.m
num=[1 -2 0.8];
den=[1 2 2 1];
subplot(1,3,1);
pzmap(num, den);           % 计算零极点并画其分布图
t=0:0.02:15;
subplot(1,3,2);
impz(num, den, t);         % 计算冲激响应并画其波形
subplot(1,3,3);
step(num, den, t);         % 计算阶跃响应并画其波形
```

运行程序后图形显示如图 4.2-2 所示。

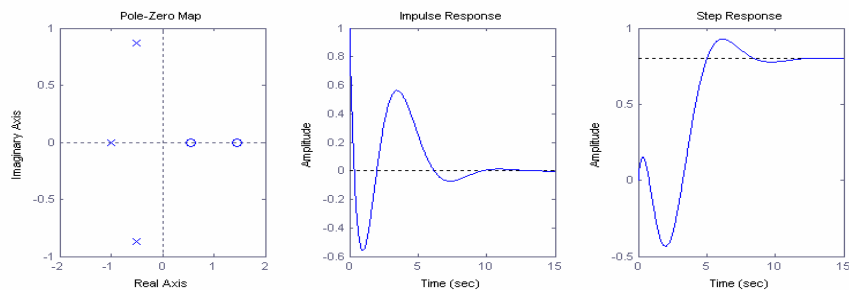


图 4.2-2 零极点分布图、冲激响应和阶跃响应

3、系统零极点分布与频率响应的关系

系统的零极点分布包含了系统的频率特性。几何矢量法是通过系统零极点分布来分析连续系统频率响应 $H(j\omega)$ 的一种直观的方法。但是对于零极点较多的系统，用这种方法就比较麻烦。

Matlab 提供了专用绘制频率响应的函数。信号处理工具箱提供的 freqs 函数可直接计算系统的频率响应，其一般调用形式为

$$H = \text{freqs}(b, a, w)$$

式中的 b 为系统函数 $H(s)$ 的有理多项式中分子多项式的系数向量， a 为分母多项式的系数向量， w 为需计算的频率抽样点向量，单位为 rad/s。如果没有输出参数，直接调用

$$\text{freqs}(b, a)$$

则 Matlab 会在当前绘图窗口中自动画出幅频和相频响应曲线图形。不过横坐标频率将取对数刻度，幅频特性的纵坐标取对数刻度，相频特性的纵坐标取度数。

Matlab 还提供了另外一个绘制频率响应波特图的函数，其调用形式为

$$\begin{aligned} &\text{bode}(\text{sys}) \quad \text{bode}(\text{sys}, w) \\ &[\text{mag}, \text{phase}, w] = \text{bode}(\text{sys}) \end{aligned}$$

其中， $\text{sys} = \text{tf}(b, a)$ ， mag 表示幅值， phase 表示相位。

例 4.2-2 用 MATLAB 画出例 4.2-1 所示系统的频率响应图。

解 Matlab 程序如下：

```
% 画频率响应图 e4_2_2.m
num=[1 -2 0.8];
den=[1 2 2 1];
figure(1)
freqs(num, den);
figure(2)
bode(num, den);
```

运行程序后，显示的频率响应图如图 4.2-3 所示。

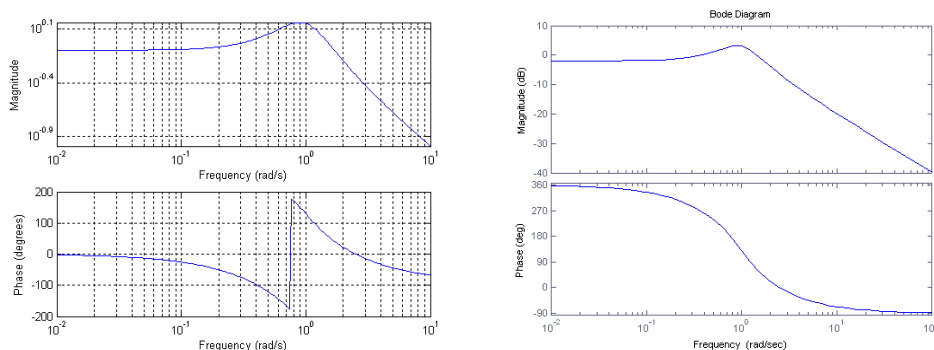


图 4.2-3 两种函数画出的频率响应图

为了方便地画出系统的零极点图和系统的频率响应图，编写了以下通用函数，画零极点图的函数如下：

```
function x=zpplot(b,a);
% 画零极点分布图
N_a=length(a)-1;
N_b=length(b)-1;
zs=roots(b); % 求零点
ps=roots(a); % 求极点
rzs=real(zs); % 求零点的实部
izs=imag(zs); % 求零点的虚部
rps=real(ps); % 求极点的实部
ips=imag(ps); % 求极点的虚部
dz=diff(zs);dp=diff(ps);
R_max=max(abs([rzs',rps']));R_max=R_max+0.3*R_max; % 求实部的绝对值最大值
I_max=max(abs([izs',ips']));I_max=I_max+0.3*I_max; % 求虚部的绝对值最大值
U_max=max(R_max,I_max);
plot(rzs,izs,'o',rps,ips,'kx','markersize',8,'linewidth',2);
if dz==0
    text(rzs+0.1,izs+0.1,'\fontsize{8}(2)','color','r');
end
if dp==0
    text(rps+0.1,ips+0.1,'\fontsize{8}(2)','color','r');
end
line([-U_max U_max],[0 0],'color','r');
line([0 0],[-U_max U_max],'color','r');
axis([-U_max U_max -U_max U_max]);
title('零极点分布图','FontSize',8)
set(gca,'FontSize',8)
```

画频率响应的函数如下：

```
function x=freresp(b,a,w);
% 画系统零极点图,幅频和相频特性图
w0=w(1);M=length(w);w1=w(M);
H=freqs(b,a,w);
F1=abs(H);
P1=unwrap(angle(H));
subplot(1,3,1),zpplot(b,a);
Fmin=min(F1);Fmax=max(F1);dF=(Fmax-Fmin)*0.1;
Pmin=min(P1);Pmax=max(P1);dP=(Pmax-Pmin)*0.1;
subplot(1,3,2),myplot(w,F1);
p=roots(a);z=roots(b);R_p=real(p);I_p=imag(p);R_z=real(z);I_z=imag(z);
if (length(R_p)==length(R_z)) & (R_p==R_z) & (I_p==I_z)
    axis([w0,w1,0,Fmax+0.5]);
else
    axis([w0,w1,Fmin-dF,Fmax+dF]);
end
ylabel(' |F(j\omega)| ');xlabel('\omega'),title('幅频特性图')
subplot(1,3,3),myplot(w,P1*180/pi);
ylabel(' 相位(度) ');xlabel('\omega'),title('相频特性图')
axis([w0,w1,(Pmin-dP)*180/pi,(Pmax+dP)*180/pi]);
```

例 4.2-3 已知下列系统函数，画出它们的零极点分布图、幅频和相频特性。并说明滤波器的类型。

$$(a) H(s) = \frac{s}{(s+20-j40)(s+20+j40)}; \quad (b) H(s) = \frac{s^2 + 22500}{s^2 + 200s + 20000};$$

$$(c) H(s) = \frac{s^2 - 40s + 2000}{s^2 + 40s + 2000}.$$

解 用 Matlab 的编写的画零极点图的函数如下：

```
% 例 4.2-3 的零极点图, 幅频和相频特性图 e4_2_3.m
figure(1)
w=linspace(0, 200, 200);
p=[-20-j*40 -20+j*40];a=poly(p);b=[1 0];
freresp(b, a, w)
figure(2)
w=linspace(0, 600, 200);
b=[1 0 22500];a=[1 200 20000];
freresp(b, a, w)
figure(3)
w=linspace(0, 200, 200);
b=[1 -40 2000];a=[1 40 2000];
freresp(b, a, w);
```

运行程序后, 显示的图形如图 4.2-4、5、6 所示。从幅频特性图上可看出, (a) 为带通滤波器; (b) 为带阻滤波器; (c) 为全通滤波器。

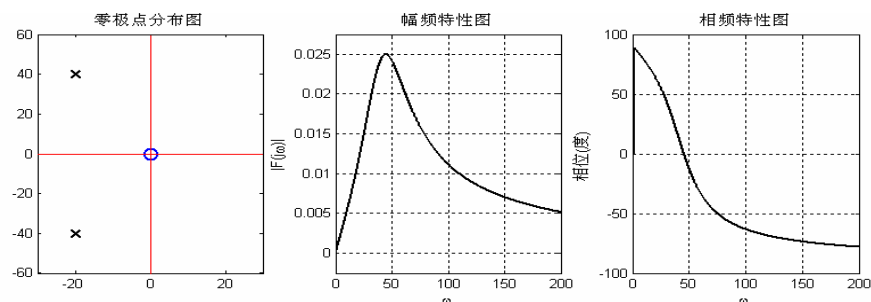


图 4.2-4 例 4.2-3(a) 的零极点分布图、幅频特性和相频特性

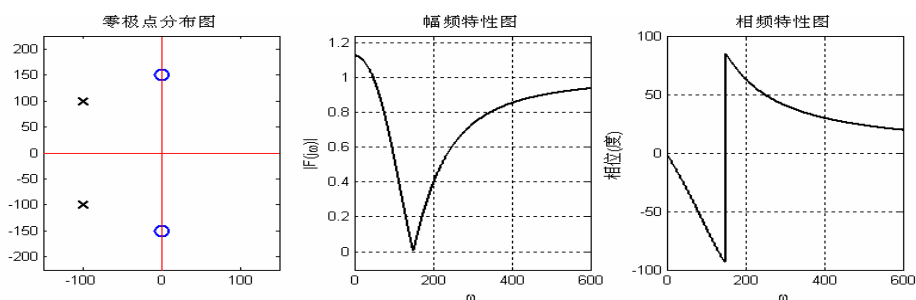


图 4.2-5 例 4.2-3(b) 的零极点分布图、幅频特性和相频特性

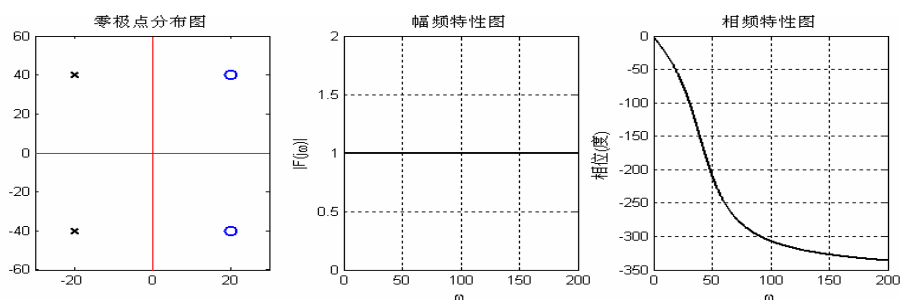


图 4.2-6 例 4.2-3(c) 的零极点分布图、幅频特性和相频特性

4.3 实验内容

9-1、用“拉普拉斯变换和系统函数的曲面图演示”程序，观察零极点三维图，加深对系统零极点的理解。考虑以下系统函数：

$$(a) H(s) = \frac{1}{(s+2)(s+4)}; \quad (b) H(s) = \frac{s}{(s+2)(s+4)}; \quad (c) H(s) = \frac{(s+1)(s+4)}{s(s+2)(s+3)}.$$

9-2、用“连续系统零极点和冲激响应的关系”程序，观察零极点对冲激响应的影响，加深对系统稳定性的理解。

画出下列系统的零极点分布图和冲激响应，确定系统的稳定性。

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

$$(c) H(s) = \frac{s-2}{s(s+1)}; \quad (d) H(s) = \frac{2(s^2+4)}{s(s+2)(s^2+1)};$$

$$(e) H(s) = \frac{16}{s^2(s+4)}; \quad (f) H(s) = \frac{2(s+1)}{s(s^2+1)^2}$$

9-3、图 4.2-7 所示为 $H(s)$ 的零极点分布图，试判别它们是低通、高通、带通、带阻中哪一种网络？零点和极点的数据自己设定。

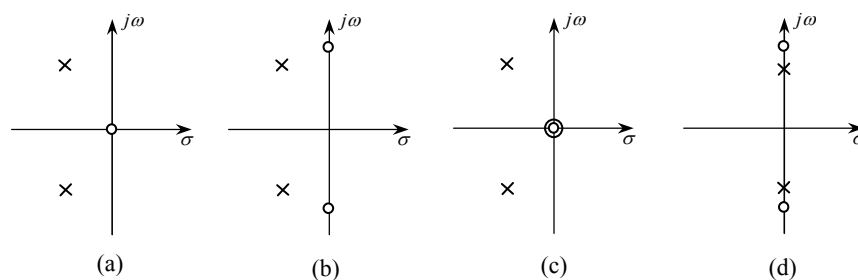


图 4.2-7

9-4、考虑如下所示的稳定系统的系统函数：

$$(a) H_1(s) = \frac{(s+1)(s+2)}{(s+3)(s+4)(s+5)}; \quad (b) H_2(s) = \frac{(s-1)(s+2)}{(s+3)(s+4)(s+5)}$$

$$(c) H_3(s) = \frac{(s-1)(s-2)}{(s+3)(s+4)(s+5)}$$

试判断它们是否是最小相位系统。

4.4 实验步骤和方法

1、在 MATLAB 的命令窗口输入：

```
>>ZPQMT
```

屏幕显示“拉普拉斯变换和系统函数的曲面图演示”图形用户界面。

(1) 观察基本信号的曲面图，以及曲面图的剖面图。理解拉普拉斯变换与傅里叶变换的关系。

(2) 在“零极点与曲面图的关系”中，可以输入任意系统函数的零点和极点，适当调整三个坐标值，就可显示出它的曲面图。

2、在 MATLAB 的命令窗口输入：

```
>>CSZPH
```

屏幕显示“连续系统零极点与冲激响应的关系”图形用户界面。

(1) 观察极点分布的三种情况：左半平面、虚轴、右半平面，以及四种极点的组合：单极点、重极点、实极点、复极点。理解零极点分布与冲激响应的关系。加深系统稳定性的认识。

(2) 在“零极点与冲激响应的关系”中，可以输入任意系统函数的零点和极点或分子、分母，适当调整时间坐标值，就可显示出它的零极点图和相应的冲激响应。将实验内容 2 用 CSZPH 演示程序观察。

3、仿照例 4.2-3 的方法，完成实验 9-3 的编程。上机调试程序，画出零极点图、幅频和相频特性图，根据题目要求说明滤波器的类型。

4、对实验 9-4，仿照例 4.2-3 的方法，将三个系统的幅频特性和三个相频特性进行比较，进一步理解最小相位系统的含义。

4.5 预习要点

1. 学习有关 Matlab 函数的用法。主要函数有：
freqs, bode, pzmap, roots, real, imag, unwrap,
figure, linspace.
2. 为什么拉普拉斯变换能画出曲面图？曲面图的剖面图是什么含义？曲面图中的峰点和谷点分别表示什么？
3. 系统的零点和极点对系统冲激响应有何影响？什么是系统的稳定性？稳定系统有什么条件？
4. 如何用 Matlab 画系统的幅频特性和相频特性？画系统函数的零极点分布图？
5. 滤波器有哪些类型？什么是全通系统？什么是最小相位系统？

4.6 实验报告要求

1. 实验 9-1、9-2 只要求观察图形，图形不写在实验报告内。
2. 根据实验 9-3、9-4 编写出的程序。以及绘出的各种波形图。根据题目要求对各种频率响应图加以比较说明。
3. 上机调试程序的方法。
4. 心得体会及其他。

实验 5 迭代法及离散卷积的计算

5.1 实验目的

- 1、学习并掌握用迭代法求解差分方程的方法。
- 2、掌握用 Matlab 进行离散卷积运算的数值方法和解析方法。加深对离散卷积的理解。

5.2 实验原理与计算示例

1、差分方程的迭代解法

在连续系统中,可用常系数线性微分方程来表示其输出与输入关系,在离散系统中,则用常系数线性差分方程来描述,其一般形式为

$$\sum_{i=0}^N a_i y(k-i) = \sum_{i=0}^M b_i f(k-i) \quad (1)$$

式中, a 和 b 为常数, $f(k)$ 的最大移位阶次为 M , $y(k)$ 的最大移位阶次为 N 。

令(1)式上的 $a_0 = 1$, 则常系数线性差分方程为

$$y(k) = -\sum_{i=1}^N a_i y(k-i) + \sum_{i=0}^M b_i f(k-i) \quad (2)$$

令上式中 $k=0$, 有

$$y(0) = -a_1 y(-1) - a_2 y(-2) - \cdots - a_N y(-N) + b_0 f(0) + b_1 f(-1) + \cdots + b_M f(-M)$$

即 $y(0)$ 是差分方程的系数与 $y(-1)$, $y(-2)$, $\cdots$ $y(-N)$ 和 $f(0), f(-1), \cdots, f(-M)$ 的线性组合。令上式中 $k=1$, 有

$$y(1) = -a_1 y(0) - a_2 y(-1) - \cdots - a_N y(-N+1) + b_0 f(1) + b_1 f(0) + \cdots + b_M f(-M+1)$$

所以, $y(1)$ 是差分方程的系数与 $y(0)$, $y(-1)$, $\cdots$ $y(-N+1)$ 和 $f(1), f(0), \cdots, f(-M+1)$ 的线性组合。

以此类推,通过反复迭代,就可以求出任意时刻的响应值。这种迭代方法最适合用计算机计算,下面我们用 Matlab 来实现这种计算。

为了找出迭代计算的一般规律,(2)式中的求和计算可写成矩阵的形式,如第一项可写为

$$\sum_{i=1}^N a_i y(k-i) = [a_N \quad a_{N-1} \quad \cdots \quad a_1] \begin{bmatrix} y(k-N) \\ y(k-N+1) \\ \vdots \\ y(k-1) \end{bmatrix}$$

第二项求和与上式类似。用 Matlab 编写的计算迭代法计算差分方程的函数如下:

```
function y=recur(a,b,n,f,f0,y0);
% recur 是用迭代法计算差分方程的解
% 其中 a 是差分方程左边除第一项外的系数
%      b 是差分方程右边的系数,n 计算的点数
%      f 输入信号,f0 输入信号的初始值
%      y0 系统的初始值
%
N=length(a);y=[y0 zeros(1,length(n))];
```

```

M=length(b)-1;f=[f0 f];
a1=a(N:-1:1); % a 的元素反转
b1=b(M+1:-1:1); % b 的元素反转
for i=N+1:N+length(n),
    y(i)=-a1*y(i-N:i-1)'+b1*f(i-N:i-N+M)';
end
y=y(N+1:N+length(n));

```

例 5.2-1 求下述差分方程的解

$$y(k) - 1.5y(k-1) + y(k-2) = 2f(k-2)$$

其中输入信号 $f(k) = \varepsilon(k)$ ，初始条件 $y(-1) = 1$ ， $y(-2) = 2$ 。

解 Matlab 程序如下：

```

% 计算例 5.2-1 的程序 e5_2_1.m
a=[-1.5 1];b=[0 0 2];
y0=[2 1];f0=[0 0];
n=0:30;
f=u(n);
y=recur(a,b,n,f,f0,y0);
mystem(n,y);
ylabel('y(k)')

```

运行程序后，系统响应波形如图 5.2-1 所示。显然用迭代法不仅可以求全响应，还可以求零输入响应和零状态响应。

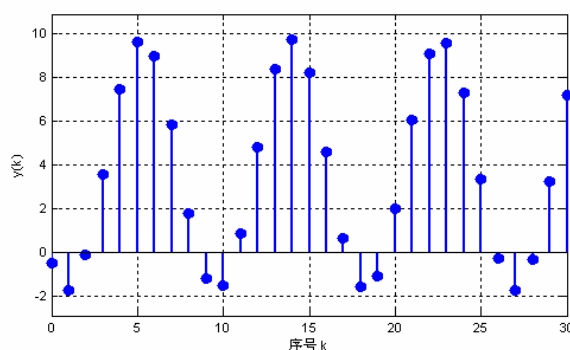


图 5.2-1 例 5.2-1 的系统响应波

2、离散卷积的计算

Matlab 信号处理工具箱提供了一个计算两个离散序列卷积和的函数 `conv()`，其调用格式为

$$y = \text{conv}(f, h)$$

式中， f 、 h 分别为待卷积的两序列的向量表示， y 是卷积的结果。

如

```

>> f1=[2 2 2];
>> f2=[1 4 9];
>> y=conv(f1,f2)
y =
    2    10    28    26    18

```

对于有限长序列，我们建立一个通用函数，它可以计算并画出两个有限长序列卷积的结果和波形。能使三个波形的横坐标统一，间隔相同。卷积结果显示在横坐标的中间位置。这个函数取名为 `DSCONV()`，程序如下：

```

function y=DSCONV(f1,n1,f2,n2,M);
% 计算有限长离散卷积

```

```

% n1, f1, n2, f2 为序列的起始点, 序列值
% 将卷积值显示在中间, 左右插入 M 点
% 例 1: n1=-2; f1=[2 2 2 2 2 2]; n2=0; f2=[1 1 1 1 1 1];
%      M=6; dsconv(f1, n1, f2, n2, M)
% 例 2: n1=-2; f1=[1 1 1 1 1]; n2=1; f2=n2;
%      M=6; dsconv(f1, n1, f2, n2, M)
%
y=conv(f1, f2); % 两序列的卷积值
ny0=n1(1)+n2(1); % 卷积值 y 的起始点
n0=ny0-M; n10=n0-n1(1); n20=n0-n2(1); % 计算 y, f1, f2 要插入的点数
L=length(y); k=n0:ny0+L+M-1; % 计算所有序列的起始点和终止点
f=[zeros(1, M) y zeros(1, M)]; % 在卷积值左右插入 M 个零点
% 在 f1, f2 左右插入零点, 与自变量 k 的点数一致
f11=[zeros(1, abs(n10)) f1 zeros(1, L+2*M-length(f1)-abs(n10))];
f22=[zeros(1, abs(n20)) f2 zeros(1, L+2*M-length(f2)-abs(n20))];
% 画出 f1, f2 和 f1*f2 的波形
subplot(3, 1, 1), mystem(k, f11), ylabel('f1(k)');
subplot(3, 1, 2), mystem(k, f22), ylabel('f2(k)');
subplot(3, 1, 3), mystem(k, f), ylabel('f1(k)*f2(k)');

```

例 5.2-2 用 Matlab 求下列序列的卷积和。

(a) $f_1(k) = [2, 2, 2, 2, 2, 2]$, $f_2(k) = [1, 1, 1, 1, 1]$ 。

(b) $f_1(k) = \begin{cases} 1 & -2 \leq k \leq 2 \\ 0 & \text{其它} \end{cases}$, $f_2(k) = \begin{cases} k & 1 \leq k \leq 5 \\ 0 & \text{其它} \end{cases}$

解 用 Matlab 并调用 DSCONV() 函数, 程序如下:

```

% 计算离散信号的卷积 e5_2_2.m
n1=-2; f1=[2 2 2 2 2 2]; % 序列的起始点, 序列值
n2=0; f2=[1 1 1 1 1 1]; % 序列的起始点, 序列值
M=6; % 将卷积值显示在中间, 左右插入 M 点
figure(1)
y1=dsconv(f1, n1, f2, n2, M)
n1=-2; f1=[1 1 1 1 1]; % 序列的起始点, 序列值
n2=1; f2=n2; % 序列的起始点, 序列值
M=4; % 将卷积值显示在中间, 左右插入 M 点
figure(2)
y2=dsconv(f1, n1, f2, n2, M)

```

程序运行后显示的图形如图 5.2-2 所示。

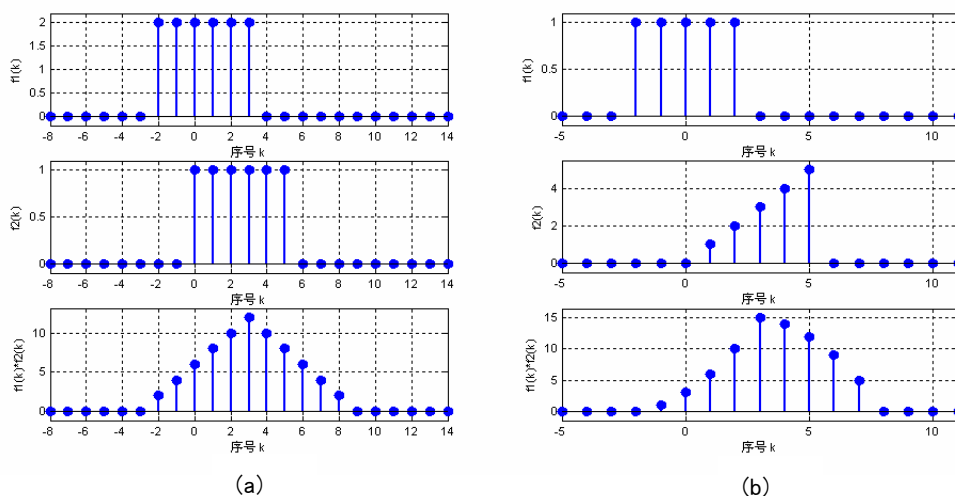


图 5.2-2 离散卷积的图形

在命令窗口显示的卷积结果

```
>> y1 =
      2      4      6      8     10     12     10      8      6      4      2
y2 =
      1      3      6     10     15     14     12      9      5
```

例 5.2-3 用 Matlab 求下列系统的零状态响应 $y(k) = h(k) * f(k)$ 。

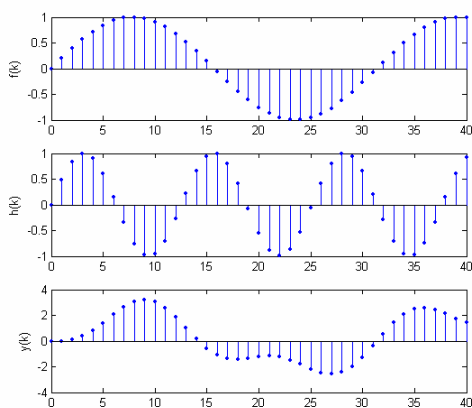
(a) $f(k) = \sin(0.2k)\varepsilon(k)$, $h(k) = \sin(0.5k)\varepsilon(k)$ 。

(b) $f(k) = (0.9)^k \varepsilon(k)$, $h(k) = (0.8)^k \varepsilon(k)$ 。

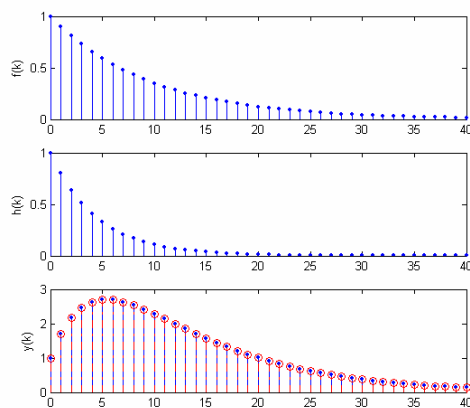
解 用 Matlab 编写程序如下：

```
% 计算无限长离散卷积 e5_2_3a.m
n=0:40;
f=sin(.2*n);
h=sin(.5*n);
y=conv(f,h)
subplot(3,1,1),stem(n,f,'. '),ylabel('f(k)');
subplot(3,1,2),stem(n,h,'. '),ylabel('h(k)');
subplot(3,1,3),stem(n,y(1:length(n)),'. '),ylabel('y(k)');
```

运行后显示的波形如图 5.2-3(a)所示。



(a)



(b)

图 5.2-3 离散卷积的图形

```
% 计算无限长离散卷积 e5_2_3b.m
n=0:40;
f=.9.^n;
h=.8.^n;
y=conv(f,h)
subplot(3,1,1),stem(n,f,'. '),ylabel('f(k)');
subplot(3,1,2),stem(n,h,'. '),ylabel('h(k)');
subplot(3,1,3),stem(n,y(1:length(n)),'. '),ylabel('y(k)');
% 用理论计算结果验证
hold on;y1=9*.9.^n-8*.8.^n;
subplot(3,1,3),stem(n,y1,'r-')
hold off
```

运行后显示的波形如图 23-3(b)所示。用理论计算的结果画出的波形与用 conv 计算出的波形完全重合。

5.3 实验内容

12-1、系统的差分方程为

$$y(k) + 0.7y(k-1) - 0.45y(k-2) - 0.6y(k-3) = 0.8f(k) - 0.44f(k-1) + 0.36f(k-2) + 0.22f(k-3)$$

已知激励为 $f(k) = [(0.5)^k + 1]\varepsilon(k)$ ，初始值为 $y(-1) = 1$, $y(-2) = -1$, $y(-3) = 2$ 。用迭代法求系统零输入响应、零状态响应和全响应。

12-2、求下列序列的卷积和 $y(k) = f_1(k) * f_2(k)$ 。

- (a) $f_1(k) = (0.3)^k \varepsilon(k)$, $f_2(k) = (0.5)^k \varepsilon(k)$;
- (b) $f_1(k) = \{\frac{1}{4}, 2, 0, 1\}$, $f_2(k) = \{\frac{2}{3}, 2, 3\}$
- (c) $f_1(k) = \varepsilon(k+2)$, $f_2(k) = \varepsilon(k-3)$
- (d) $f_1(k) = (0.5)^k \varepsilon(k)$, $f_2(k) = (0.5)^k [\varepsilon(k+3) - \varepsilon(k-4)]$

5.4 实验步骤和方法

1、用迭代法计算实验 12-1，注意计算零输入响应、零状态响应和全响应用迭代法时有什么不同。仿照例 5.2-1 的方法调用迭代法函数 `recur()`。

2、计算实验 12-2 的卷积和时，首先区分有限长序列和无限长序列。对有限长序列的卷积和采用例 5.2-2 的方法，调用计算离散卷积的函数 `dsconv()` 编程。对无限长序列的卷积和采用例 5.2-3 的方法编程。上机调试程序，并与理论计算加以比较。

5.5 预习要点

1、学习有关 Matlab 运算的用法。主要有：

`a1=a(N:-1:1)`, `y(i-N:i-1)`, `y=[y0 zeros(1,length(n))]`, `conv`, `sum`,

2、复习有关迭代法解差分方程的方法。仔细阅读迭代法函数 `recur()`。

3、复习有关计算离散卷积的方法。仔细阅读计算有限离散信号卷积的函数 `dsconv()`。有限离散信号的卷积和无限离散信号的卷积有何区别？对超前或滞后的波形如何处理？

5.6 实验报告要求

1、根据实验内容所给出的习题，编写出的程序或命令。并绘出的各种波形图。

2、总结用迭代法计算差分方程的方法，对零输入响应、零状态响应和全响应分别进行计算时应注意的问题。

3、根据实验归纳、总结出用 Matlab 计算离散卷积的方法。指出计算有限长序列卷积的 `dsconv()` 函数有什么特点。

4、心得体会及其他。

实验 6 离散系统的时域和频域分析

6.1 实验目的

- 1、学习用 Matlab 绘制离散系统零极点分布图、冲激响应波形、频率响应曲线图。
- 2、通过运行系统零极点分布与冲激响应的关系的演示程序，加深系统零极点分布对时域响应的影响。从而建立系统稳定性的概念。
- 3、学习用 Matlab 计算离散系统响应的数值方法，包括冲激响应、零输入响应、零状态响应和全响应。
- 4、研究系统零极点分布与频率响应的关系，学习用 Matlab 研究频率响应的方法。

6.2 实验原理与计算示例

1、系统零极点分布与冲激响应的关系

已知线性非时变因果系统的系统函数为

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

若分子多项式 $N(z)$ 的阶次高于分母多项式，即 $m > n$ ，则 $H(z)$ 可分解为 z 的有理多项式与 z 的有理真分式之和。有理多项式部分比较容易分析，故我们讨论 $H(z)$ 为有理真分式的情况，即上式中 $m \leq n$ 的情况。

设系统函数 $H(z)$ 具有单极点时，系统函数 $H(z)$ 可按部分分式法展开为

$$H(z) = \sum_{i=1}^n \frac{zK_i}{z - p_i}$$

系统的冲激响应 $h(k)$ 为

$$h(k) = \sum_{i=1}^n C_i (p_i)^k \varepsilon(k)$$

从上式可见，冲激响应 $h(k)$ 的性质完全由系统函数 $H(z)$ 的极点 p_i 决定。 p_i 称为系统的自然频率或固有频率。而待定系数 C_i 由零点和极点共同决定。

系统零极点分布与冲激响应有如下关系

- 极点决定了冲激响应 $h(k)$ 的形式，而各系数 C_i 则由零、极点共同决定。
- 系统的稳定性由极点在 z 平面上的分布决定，而零点不影响稳定性。
- 极点分布在 z 平面的单位圆内，系统是稳定的。极点在单位圆上有单极点，系统是临界稳定。极点在 z 平面的单位圆外或在单位圆上有重极点，系统不稳定。

MATLAB 画 Z 平面零极点图的函数为：

$$\text{zplane}(b, a)$$

计算离散系统的冲激响应的函数 $\text{impz}()$ ，有下面几种格式：

$$[h \ n] = \text{impz}(b, a) \quad \text{或} \quad h = \text{impz}(b, a)$$

$$[h \ n] = \text{impz}(b, a, n) \quad \text{或} \quad h = \text{impz}(b, a, n)$$

其中， b 和 a 分别是系统函数 $H(z)$ 的分子多项式和分母多项式的系数矩阵，若分子多项式和分母多项式的系数的个数不等，则用零补充； n 为抽样时间点数。 $\text{impz}(b, a)$ 则自动绘制单位冲激响应的图形。

2、系统零极点分布与频率响应的关系

系统的零极点分布包含了系统的频率特性。几何矢量法是通过系统零极点分布来分析连续系统频率响应 $H(e^{j\omega})$ 的一种直观的方法。但是对于零极点较多的系统，用这种方法就比较麻烦。

Matlab 提供了专用绘制频率响应的函数。信号处理工具箱提供的 `freqz` 函数可直接计算系统的频率响应，其一般调用形式为

$$[H, w] = \text{freqz}(b, a)$$

式中的 b 为系统函数 $H(z)$ 的有理多项式中分子多项式的系数向量， a 为分母多项式的系数向量， w 为需计算的频率抽样点向量(取 $\omega = 0 \sim \pi$)，单位为 rad/s。如果没有输出参数，直接调用

$$\text{freqz}(b, a)$$

则 Matlab 会在当前绘图窗口中自动画出幅频和相频响应曲线图形。不过横坐标频率将取线性刻度，幅频特性的纵坐标取分贝刻度，相频特性的纵坐标取度数。

例 5.4-1 已知一离散因果线性时不变系统系统的系统函数为

$$H(z) = \frac{z^2 + 2z + 1}{z^3 - 0.5z^2 - 0.005z + 0.3}$$

试画出系统的零极点分布图，求系统的单位冲激响应 $h(k)$ 和频率响应 $H(e^{j\omega})$ ，并判断系统是否稳定。

解 如果已知系统函数，求系统的单位冲激响应和频率响应，可分别使用 Matlab 中的函数 `impz` 和 `freqz`。

根据已知的 $H(z)$ ，用 `zplane` 即可画出系统的零极点分布图。而利用 `impz` 函数和 `freqz` 函数求系统的单位冲激响应和频率响应时需要将 $H(z)$ 改写成

$$H(z) = \frac{z^{-1} + 2z^{-2} + z^{-3}}{1 - 0.5z^{-1} - 0.005z^{-2} + 0.3z^{-3}}$$

程序如下：

```
% 画例5.4-1的冲激响应和频率响应      e5_4_1.m
b=[0, 1, 2, 1];                        % 分子系数
a=[1, -0.5, -0.005, 0.3];              % 分母系数
figure(1); zplane(b, a);
n=0:25;
[h, n]=impz(b, a, n);                  % 求冲激响应
figure(2); system(h)
xlabel('k')
title('冲激响应')
[H, w]=freqz(b, a);                    % 求频率响应
figure(3); myplot(w/pi, abs(H))
xlabel('频率\omega (x\pi rad/sample)')
title('幅度响应')
figure(4); myplot(w/pi, unwrap(angle(H)*180/pi))
xlabel('频率\omega (x\pi rad/sample)'), title('相位响应')
```

程序运行结果如图5.4-1所示。图5.4-1 (a)为系统函数的零极点分布图，由图知极点都在单位圆内，故系统是稳定的。

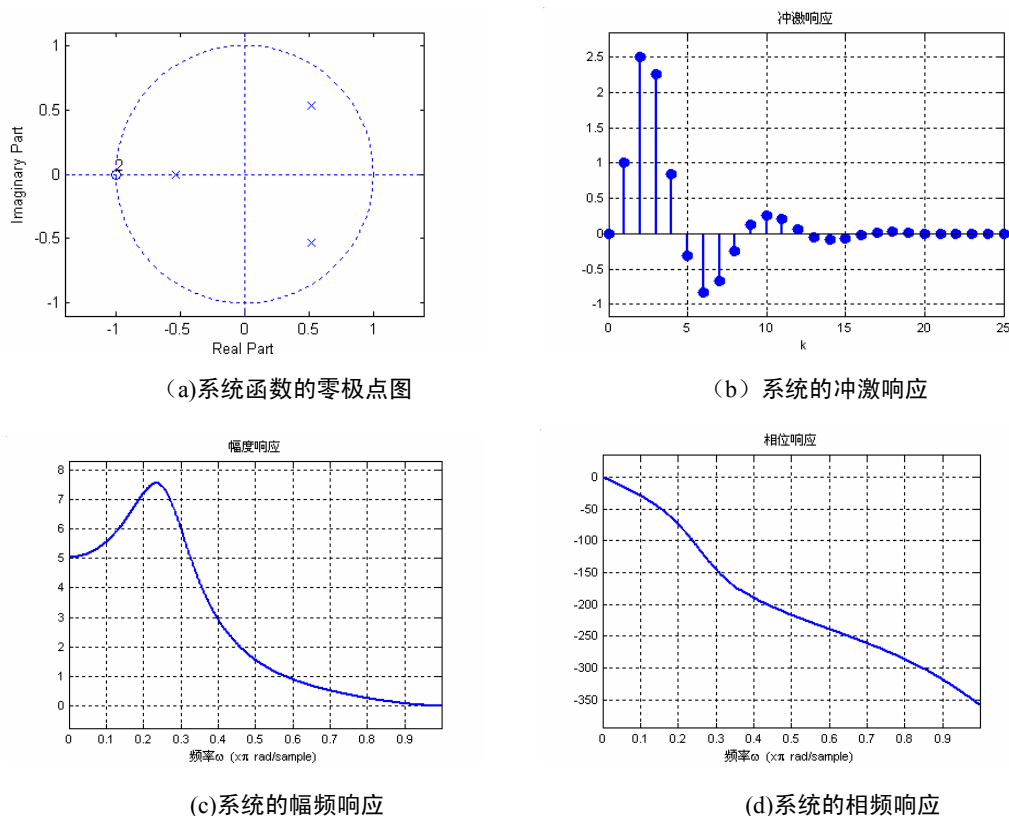


图5.4-1 系统的零极点图、冲激响应和频率响应

3、离散系统时域分析的数值解

用 MATLAB 分析离散系统的常用函数是：

画零极点分布图的函数 `zplane(b, a)`

计算系统冲激响应的函数 `impz(b, a)`

计算系统全响应的函数 `filter(b, a, x, zi)`

其中， b 为系统函数 $H(z)$ 的分子多项式系数向量， a 为分母多项式系数向量。 x 为输入信号， zi 为系统的初始值。注意， zi 并不是 $y(-1)$ ， $y(-2)$ 。它可以由以下推导求得：

设输入 $f(k) = 0$ ，二阶差分方程为

$$y_{zi}(k) + a_1 y_{zi}(k-1) + a_2 y_{zi}(k-2) = 0$$

对上式进行 z 变换：

$$Y(z) + a_1 [z^{-1}Y(z) + y(-1)] + a_2 [z^{-2}Y(z) + z^{-1}y(-1) + y(-2)] = 0$$

零输入响应为

$$Y_{zi}(z) = \frac{-[a_1 y(-1) + a_2 y(-2)] - a_2 y(-1) z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

因此， $zi = [-[a_1 y(-1) + a_2 y(-2)], -a_2 y(-1)]$ ，它可以用函数 `filtic` 求得。

$$\text{filtic}(b, a, y0, x0)$$

其中， $y0$ 为 $y(k)$ 的初始值。 $x0$ 为 $f(k)$ 的初始值。

例 5.4-2 求下述差分方程的数值解

$$y(k) - 1.5y(k-1) + y(k-2) = 2f(k-2)$$

其中输入信号 $f(k) = \varepsilon(k)$ ，初始条件 $y(-1) = 1$ ， $y(-2) = 2$ 。

解 Matlab 程序如下:

```
% 离散系统的数值解          e5_4_2.m
a=[1 -1.5 1];b=[0 0 2];
k=0:30;f=u(k);
zi=filtic(b,a,[1 2]);
y=filter(b,a,f,zi);
mystem(k,y),xlabel('k'),ylabel('y(k)')
```

运行程序后, 系统响应波形如图 5.4-2 所示。可见与迭代法得出的结果一致。

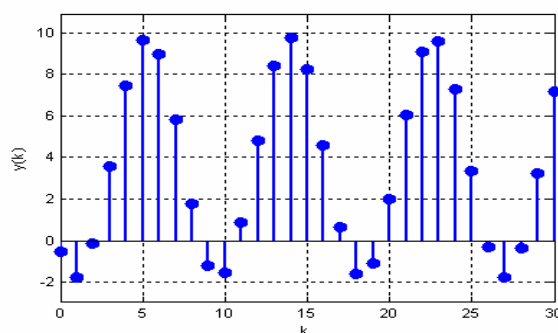


图 5.4-2 例 5.4-2 的系统响应波

例 5.4-3 已知差分方程为

$$y(k) - 1.143y(k-1) + 0.4128y(k-2) = 0.0675f(k) + 0.1349f(k-1) + 0.0675f(k-2)$$

(a) 初始条件 $y(-1) = 1$, $y(-2) = 2$, 求系统的零输入响应并作图。

(b) 有三个信号分别通过系统: $f_1(k) = \cos(\frac{\pi}{10}k)\varepsilon(k)$, $f_2(k) = \cos(\frac{\pi}{5}k)\varepsilon(k)$,

$f_3(k) = \cos(\frac{7\pi}{10}k)\varepsilon(k)$, 分别计算零状态响应并作图。

(c) 作出系统的频率特性, 指出这是一个什么类型的系统。

解 用 Matlab 的编写的程序如下:

```
% 离散系统的数值解          e5_4_3.m
a=[1 -1.143 0.4128];b=[0.0675 0.1349 0.0675];
k=0:30;
f1=cos(pi/10.*k);f2=cos(pi/5.*k);f3=cos(7*pi/10.*k);
zi=filtic(b,a,[1 2]);
f=zeros(1,26);
yzi=filter(b,a,f,zi);
figure(1),mystem([0:25],yzi)
xlabel('k'),title('零输入响应')
y1=filter(b,a,f1);y2=filter(b,a,f2);y3=filter(b,a,f3);
figure(2),mystem(k,y1)
xlabel('k'),title('f1(k)的零状态响应')
figure(3),mystem(k,y2)
xlabel('k'),title('f2(k)的零状态响应')
figure(4),mystem(k,y3)
xlabel('k'),title('f3(k)的零状态响应')
[H,w]=freqz(b,a);
figure(5),myplot(w/pi,abs(H))
xlabel('频率\omega (x\pi rad/sample)'),title('幅度响应')
figure(6),myplot(w/pi,angle(H)*180/pi)
xlabel('频率\omega (x\pi rad/sample)'),title('相位响应')
```

运行程序后, 显示的图形如图 5.4-3 所示。

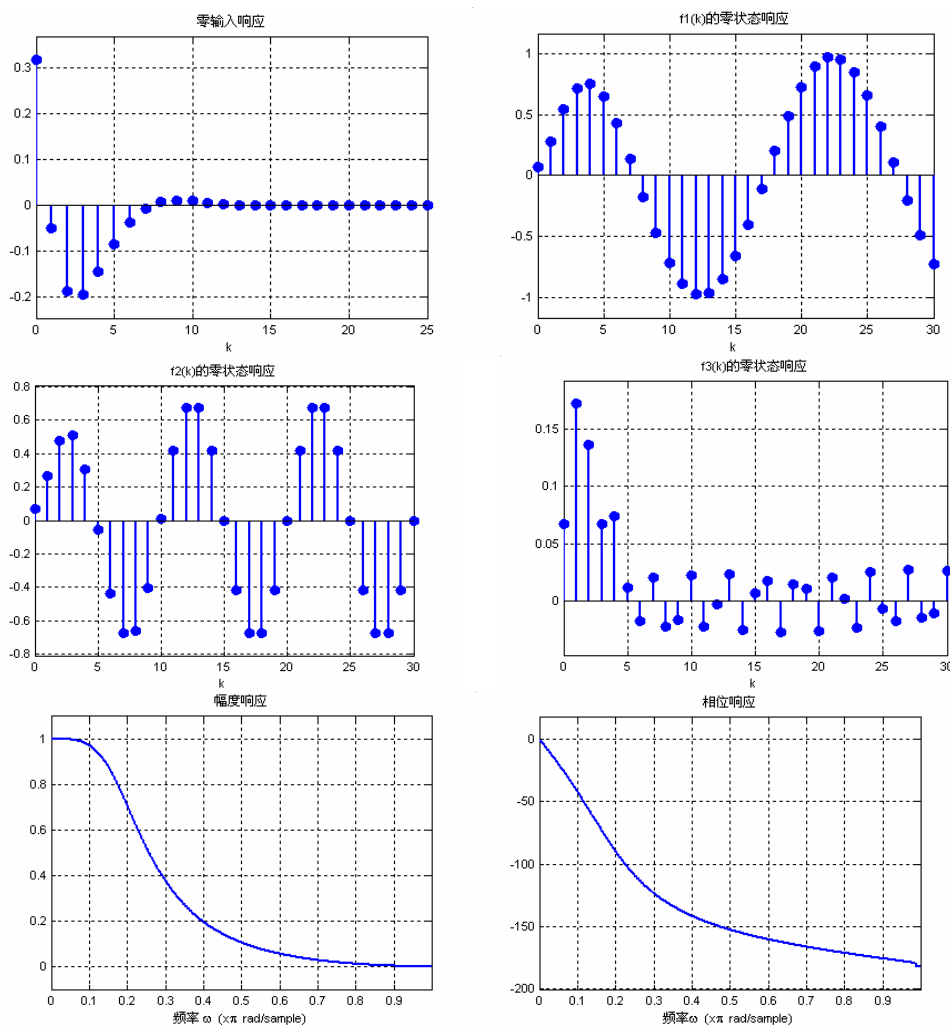


图 5.4-3 例 5.4-3 的系统响应、幅频特性和相频特性

从幅频特性图上可看出，该系统是低通滤波器。从三个激励的响应来看， $f_1(k)$ 是角频率为 0.1π 的正弦信号，经过低通滤波器后的输出几乎没有衰减； $f_2(k)$ 是角频率为 0.2π 的正弦信号，经过低通滤波器后的输出有些衰减； $f_3(k)$ 是角频率为 0.7π 的正弦信号，经过低通滤波器后的输出衰减最多。充分显示了其低通特性。

6.3 实验内容

14-1、用“离散系统零极点和冲激响应的关系”程序，观察零极点对冲激响应的影响，加深对系统稳定性的理解。

画出下列系统的零极点分布图和冲激响应，确定系统的稳定性。

$$(a) H(z) = \frac{1+z^{-2}+2z^{-3}}{2-0.5z^{-2}+0.25z^{-3}};$$

$$(b) H(z) = \frac{z^{-1}+1.5z^{-2}+4z^{-3}}{1+0.5z^{-2}+1.5z^{-3}};$$

$$(c) H(z) = \frac{0.15(z^2-1)}{z^2+0.7};$$

$$(d) H(z) = \frac{2(z^2+4)}{z(z+0.5)(z^2+1)};$$

14-2、已知系统函数为 $H(z) = \frac{1+2z^{-1}}{1+0.4z^{-1}-0.12z^{-2}}$ ，求

(a) 系统的冲激响应 $h(k)$ 的波形;

- (b) 输入 $f(k) = \varepsilon(k)$ ，求系统的零状态响应 $y_{zs}(k)$ 的波形；
- (c) 输入 $f(k) = \varepsilon(k)$ ，初始条件 $y(-1) = 1$, $y(-2) = 2$ ，求系统的全响应的波形。
- 14-3、对每一个滤波器，画出其幅度和相位频谱图，并说明滤波器的类型：
- (a) $h(k) = \delta(k) - \delta(k-2)$
- (b) $y(k) - 0.25y(k-1) = f(k) - f(k-1)$

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

14-4、已知一个因果离散系统的系统函数为：

$$H(z) = \frac{0.03571 + 0.14281z^{-1} + 0.2143z^{-2} + 0.1428z^{-3} + 0.03571z^{-4}}{1 - 1.035z^{-1} + 0.8264z^{-2} - 0.2605z^{-3} + 0.04033z^{-4}};$$

- (a) 作出系统的零极点图，绘出幅频特性和相频特性曲线。
- (b) 作出系统的冲激响应波形。
- (c) 已知输入为 $f(k) = \left[1 + \cos\left(\frac{\pi}{4}k\right) + \cos\left(\frac{\pi}{2}k\right) \right] \varepsilon(k)$ ，计算系统的零状态响应 $y_{zs}(k)$ 。并绘出输入和输出的波形。

6.4 实验步骤和方法

1、在 MATLAB 的命令窗口输入：

```
>>DSZPH
```

屏幕显示“离散系统零极点与冲激响应的关系”图形用户界面。

(1) 观察极点分布的三种情况： z 平面的单位圆内、单位圆上、单位圆外，以及四种极点的组合：单极点、重极点、实极点、复极点。理解零极点分布与冲激响应的关系。加深系统稳定性的认识。

(2) 在“零极点与冲激响应的关系”中，可以输入任意系统函数的零点和极点或分子、分母，适当调整时间坐标值，就可显示出它的零极点图和相应的冲激响应。将实验 14-1 用 DSZPH 演示程序观察。

2、学习理解离散系统数值解的原理，用 `impz`、`filtic`、`filter` 等函数计算实验 14-2。

3、仿照例 5.4-1 的方法，完成实验 14-3 的编程。上机调试程序，画出零极点图、幅频和相频特性图，根据题目要求说明滤波器的类型。

4、参考例 5.4-3，对实验 14-4 进行计算。

6.5 预习要点

- 学习有关 Matlab 函数的用法。主要函数有：
`freqz`, `filtic`, `impz`, `filter`, `zplane`。
- 系统的零点和极点对系统冲激响应有何影响？什么是系统的稳定性？稳定系统有什么条件？
- 如何用 Matlab 画离散系统的幅频特性和相频特性？画系统函数的零极点分布图？
- 数字滤波器有哪些类型？

6.6 实验报告要求

- 实验 14-1 只要求观察图形，图形不写在实验报告内。
- 根据实验 14-2、3、4 编写出的程序。以及绘出的各种波形图。根据题目要求对各种频率响应图加以比较说明。
- 上机调试程序的方法。
- 心得体会及其他。

信号与系统综合设计实验项目

实验指导

项目一 用 MATLAB 验证时域抽样定理

目的:

通过 MATLAB 编程实现对时域抽样定理的验证, 加深抽样定理的理解。同时训练应用计算机分析问题的能力。

任务:

连续信号 $f(t) = \cos(8\pi t) + 2\sin(40\pi t) + \cos(24\pi t)$, 经过理想抽样后得到抽样信号 $f_s(t)$, 通过理想低通滤波器后重构信号 $f(t)$ 。

方法:

1、确定 $f(t)$ 的最高频率 f_m 。对于无限带宽信号, 确定最高频率 f_m 的方法: 设其频谱的模降到 10^{-5} 左右时的频率为 f_m 。

2、确定 Nyquist 抽样间隔 T_N 。选定两个抽样时间: $T_s < T_N$, $T_s > T_N$ 。

3、MATLAB 的理想抽样为

$$n = -200:200; nTs = nTs; \quad \text{或} \quad nTs = -0.04:Ts:0.04$$

4、抽样信号通过理想低通滤波器的响应

理想低通滤波器的冲激响应为

$$h(t) = T_s \frac{\omega_c}{\pi} \text{Sa}(\omega_c t) \Leftrightarrow T_s G_{2\omega_c}(\omega) + H(j\omega)$$

系统响应为

$$y(t) = f_s(t) * h(t)$$

由于

$$f_s(t) = f(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) = \sum_{n=-\infty}^{\infty} f(nT_s) \delta(t - nT_s)$$

所以

$$\begin{aligned} y(t) &= \sum_{n=-\infty}^{\infty} f(nT_s) \delta(t - nT_s) * T_s \frac{\omega_c}{\pi} \text{Sa}(\omega_c t) \\ &= T_s \frac{\omega_c}{\pi} \sum_{n=-\infty}^{\infty} f(nT_s) \text{Sa}[\omega_c(t - nT_s)] \end{aligned}$$

MATLAB 计算为

$$ft = fs * Ts * wc / pi * sinc((wc / pi) * (ones(length(nTs), 1) * t - nTs' * ones(1, length(t))));$$

要求 (画出 6 幅图):

当 $T_s < T_N$ 时:

- 1、在一幅图中画原连续信号 $f(t)$ 和抽样信号 $f_s(t)$ 。 $f(t)$ 是包络线, $f_s(t)$ 是离散信号。
- 2、画出重构的信号 $y(t)$ 。
- 3、画出误差图, 即 $\text{error} = \text{abs}(f(t) - y(t))$ 的波形。

当 $T_s > T_N$ 时同样可画出 3 幅图。

项目二 连续系统的频域分析

目的:

周期信号输入连续系统的响应可用傅里叶级数分析。由于计算过程烦琐，最适合用 MATLAB 计算。通过编程实现对输入信号、输出信号的频谱和时域响应的计算，认识计算机在系统分析中的作用。

任务:

线性连续系统的系统函数为 $H(j\omega) = \frac{1}{j\omega + 1}$ ，输入信号为周期矩形波如图 1 所示，用 MATLAB 分析系统的输入频谱、输出频谱以及系统的时域响应。

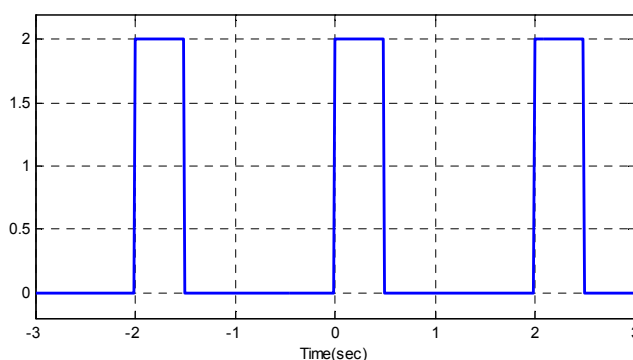


图 1

方法:

- 1、确定周期信号 $f(t)$ 的频谱 $\dot{F}_n$ 。基波频率。
- 2、确定系统函数 $H(jn\Omega)$ 。
- 3、计算输出信号的频谱

$$\dot{Y}_n = H(jn\Omega)\dot{F}_n$$

- 4、系统的时域响应

$$y(t) = \sum_{n=-\infty}^{\infty} \dot{Y}_n e^{jn\Omega t}$$

MATLAB 计算为

$$y=Y_n*\exp(j*w0*n'*t);$$

要求（画出 3 幅图）:

- 1、在一幅图中画输入信号 $f(t)$ 和输入信号幅度频谱 $|F(j\omega)|$ 。用两个子图画。
- 2、画出系统函数的幅度频谱 $|H(j\omega)|$ 。
- 3、在一幅图中画输出信号 $y(t)$ 和输出信号幅度频谱 $|Y(j\omega)|$ 。用两个子图画。

项目三 连续系统的复频域分析

目的:

周期信号输入连续系统的响应也可用拉氏变换分析。用 MATLAB 的符号计算功能，通过编程实现对系统瞬态响应和稳态响应的分析，加深理解拉氏变换在分析系统中的作用。

任务:

线性连续系统的系统函数为 $H(s) = \frac{1}{s+1}$ ，输入信号为周期矩形波如图 2 所示，用 MATLAB 分析系统的响应和稳态响应。

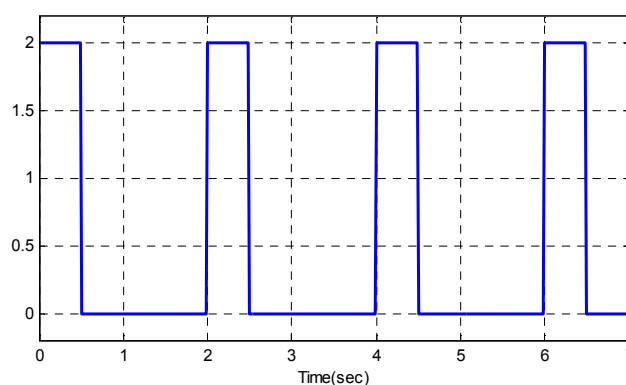


图 2

方法:

- 1、确定第一个周期拉氏变换 $F_0(s)$ 。
- 2、确定前 6 个周期的拉氏变换 $F(s)$ 。
- 3、计算输出信号的拉氏变换

$$Y(s) = H(s)F(s)$$

- 4、系统的时域响应

$$y(t) \Leftrightarrow Y(s)$$

MATLAB 计算为

$$y = \text{ilaplace}(Y);$$

- 5、系统的稳态响应和稳态值，即经过 4 个周期后，系统响应趋于稳态，两个稳态值可取为 $t=8s$ 和 $t=8.5s$

要求:

- 1、画出输入信号 $f(t)$ 波形。
- 2、画出系统输出信号 $y(t)$ 的波形。
- 3、画出系统稳态响应 $y_{ss}(t)$ 的波形，4 个周期后。并计算出稳态值。

注：信号与系统实验中的第 7 个实验为综合性设计实验，每个同学可在以上三个实验项目中任选做一个实验项目。本实验项目评分为 40 分。第 16 周五前交纸质实验报告。