

铜陵学院

《自动控制理论》课程设计 指导书

自动化教研室

目 录

1	控制系统的数学描述·····	1
1.1	微分方程·····	1
1.2	传递函数·····	6
1.3	状态空间描述·····	9
1.4	模型转换·····	11
2	控制系统的校正·····	15
2.1	单变量系统的两种主要校正方式·····	15
2.2	PI、PD、PID 校正·····	15
2.3	串联校正举例·····	18
3	MATLAB 在自动控制系统中的应用·····	24
3.1	概述·····	24
3.2	控制系统函数全集·····	30
3.3	应用实例·····	31
4	SIMULINK 简介·····	41
4.1	引导·····	41
4.2	SIMULINK 模型的构建·····	45
5	自动控制系统设计任务·····	50
5.1	任务一·····	50
5.2	任务二·····	52
5.3	任务三·····	53
5.4	任务四·····	54
6	附录·····	56
6.1	时域分析例题及程序·····	56
6.2	根轨迹分析例题及程序·····	64
6.3	频域分析例题及程序·····	66

第一章 控制系统的数学描述

1.1 微分方程

1.1.1 物理系统的微分方程

利用机械学、电学、流体力学和热力学等的物理规律，我们可以得到物理系统的动态方程。它们通常用常系数线性微分方程来描述。

1.1.2 数值解

通过拉普拉斯变换和反变换，可得到线性时不变方程的解析解，也可用状态转移矩阵 $\Phi(t)$ 求解。这些分析方法通常只限于常系数的线性微分方程。解析解是精确的，然而通常寻找解析解是困难的，甚至是不可能的。而数值分析方法直接在时域里求解微分方程，不仅适用于线性时不变方程，也适用于非线性以及时变微分方程。

MATLAB 提供了两个求微分方程数值解的函数，它们采用龙格-库塔(Runge-kutta)法。Ode23 和 ode45 分别表示采用 2 阶和 4 阶龙格-库塔公式，后者具有更高的精度。

n 阶微分方程必须化为 n 个首 1 的一阶微分方程组，且放入 M-文件中，以便返回方程状态变量的导数，下面的例子介绍这些函数的用法。

例 1.1 对图 1-1 的机械系统，已知三个量——拉力、摩擦力、以及弹簧力都影响质量 M 的加速度。

利用牛顿运动定理，建立系统的力平衡方程式

$$M \frac{d^2x}{dt^2} + B \frac{dx}{dt} + Kx = f(t)$$

令 $x_1 = x, x_2 = \frac{dx}{dt}$ ，有

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = \frac{1}{M} [f(t) - Bx_2 - Kx_1]$$

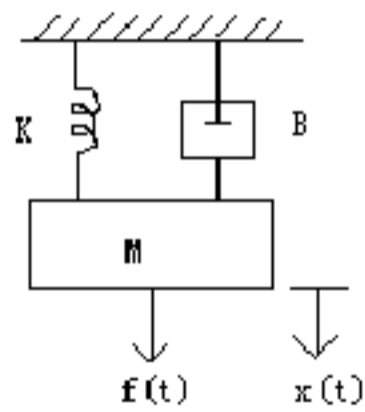


图1-1

设质量 $M=1\text{kg}$ ，摩擦系数 $B=5\text{N/m/sec}$ ，弹簧常数 $K=25\text{N/m}$ 。在 $t=0$ 时刻，施加 25N 的拉力。上述方程及已知量在 M-文件 mechsyst.m 中定义如下：

```
function xdot=mechsyst(t, x);
F=25;
M=1;B=5;K=25;
xdot=[x(2);1/M*(F-B*x(2)-K*x(1))];
```

下面的 M-文件使用 ode23 对系统在零初始条件下进行仿真：

```
t0=0;    tfinal=3;           %时间间隔 0~3 秒
x0=[0,0];           %零初始条件
tol=0.001;         %精度
trace=0;           %如果非零，则打印出每一步的计算值
[t, x]=ode23('mechsys',t0,tfinal,x0,tol,trace) subplot(211),plot(t, x);
title('Time response of mechanical translational system') xlabel
('Time-sec')
text(2,1.2,'displacement') text
(2,2,'velocity')
d=x(:,1);v=x(:,2);
subplot(212),plot(d,v);
title('velocityversusdisplacement')
xlabel('displacement')
ylabel('velocity')
subplot(111)
```

仿真结果如图 1-2。

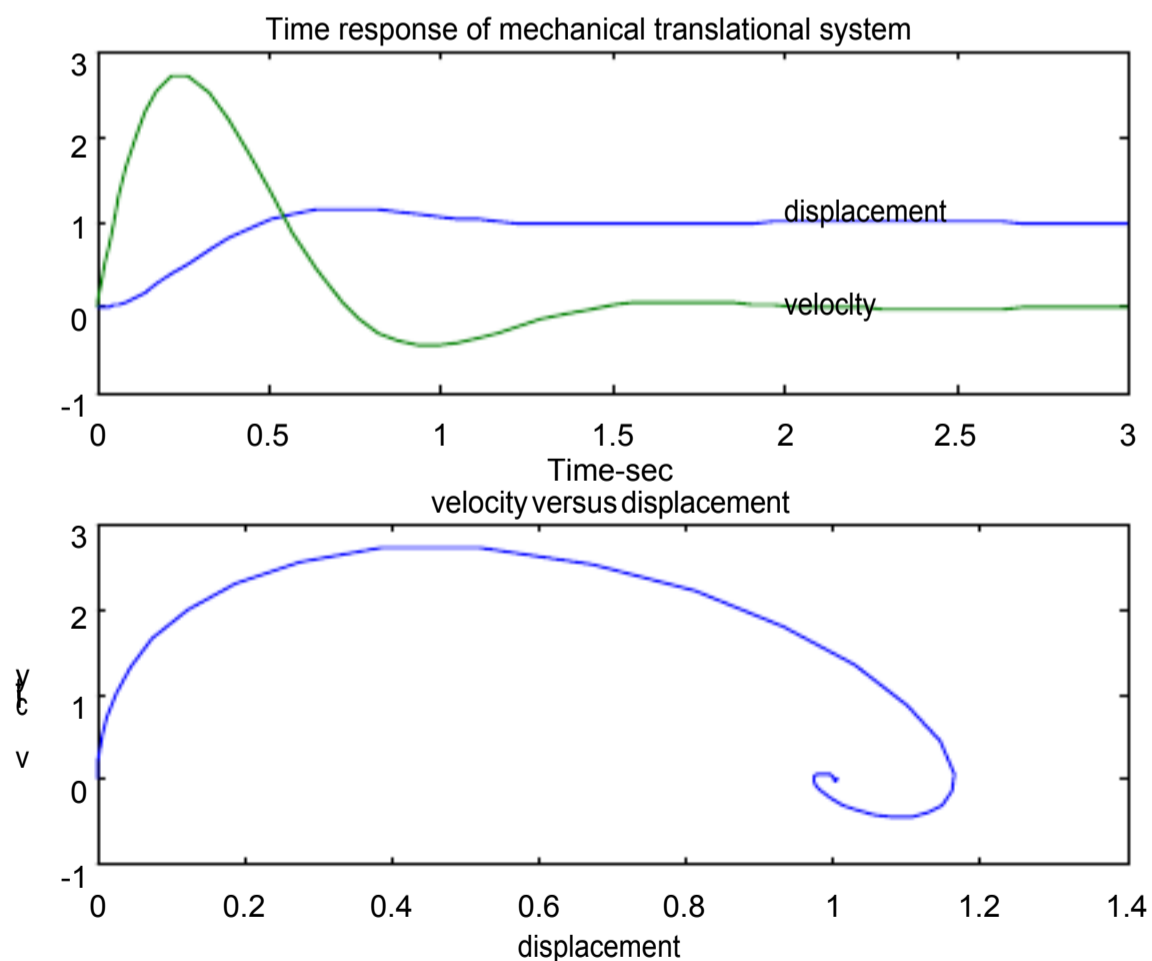


图 1-2

例 1.2 电路图见图 1-3。R=1.4Ω，L=2H，C=0.32F。初始状态：电感电流为零，电容电压为 0.5V。t = 0 时刻接入 1 伏的电压。求 0<t<15 秒时 i(t)和 v(t)的值，并且画出电流与

电容电压的关系曲线。

根据基尔霍夫定律[RLC]有：

$$Ri + L \frac{di}{dt} + v_0 = v_s$$

和

$$i = C \frac{dv_0}{dt}$$

令

$$x_1 = v_0, x_2 = i$$

得

$$\dot{x}_1 = \frac{1}{C} x_2 \quad \dot{x}_2 = \frac{1}{L} (v_s - x_1 - R x_2)$$

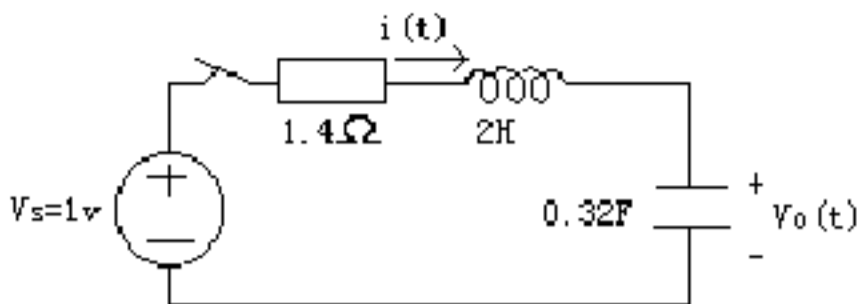


图 1-3

以上方程式在M-文件 electsys.m 中定义如下：

```
function xdot=electsys(t, x); %状态导数
V=1; %阶跃输入
R=1.4; L=2; C=0.32;
xdot=[x(2)/C; 1/L*(V-x(1)-R*x(2))];
下面的 M-文件使用函数 ode23 对系统进行仿真。
t0=0;tfinal=15; * 时间间隔 0~15 秒
x0=[0.5,0]; %初值条件x1=0.5,x2=0
tol=0.001; %精度
trace=0; %若非零值则打印出每一步的计算值
[t, x]=ode23('electsys',t0,tfinal,x0,tol,trace);
subplot(211), plot(t, x);
title('TimeresponseofaRLCseriescircuit')
xlabel('Time-sec')
text(8,1.05,'Capacitor voltage'),text(8,0.05,'current')
vc=x(:,1), i=x(:,2);
subplot(212),plot(vc, i);
title('current versus capacitor voltage')
xlabel('Capacitor Voltage')
ylabel('current')
subplot(111)
```

仿真结果见图 1-4。

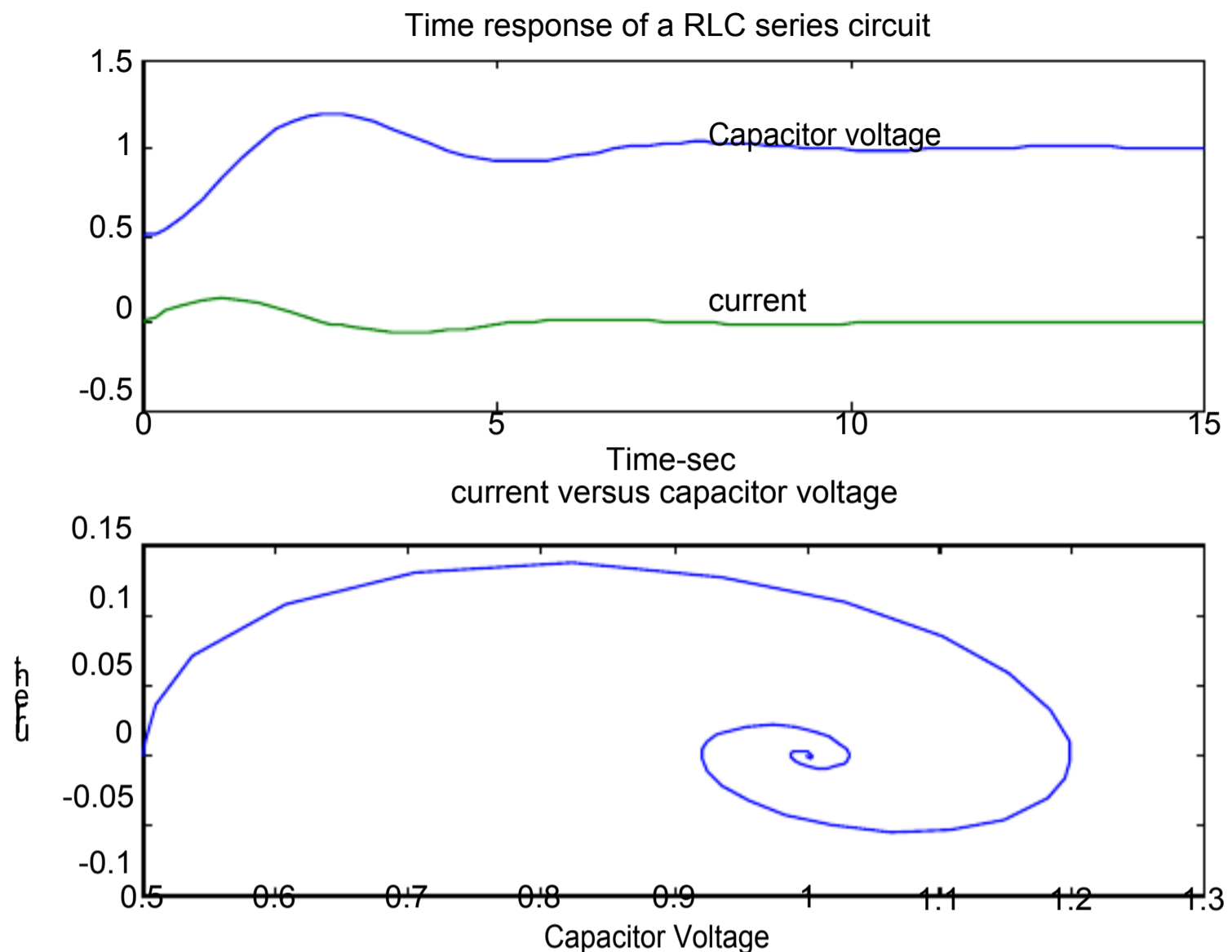


图 1-4

1.1.1 非线性系统

在变量的一定范围内大多数物理系统是线性的。然而，随着变量范围的延伸，所有的系统最终都是非线性的。对于非线性系统，叠加原理不再成立。Ode23 和 ode45 可以求解非线性微分方程，见例 1-3。

例1.3 图 1-5 的单摆，一长 L 米的无重量绳悬挂重为 $W=mg$ 千克的物体，阻尼系数为 B 千克/米/秒。通常，该系统用线性微分方程近似描述，实际上它是非线性的。

如果 θ 为绳的倾角，物体的速度将为 $L\dot{\theta}$ ，那么离心力为

$$F = -W\sin\theta - BL\ddot{\theta}$$

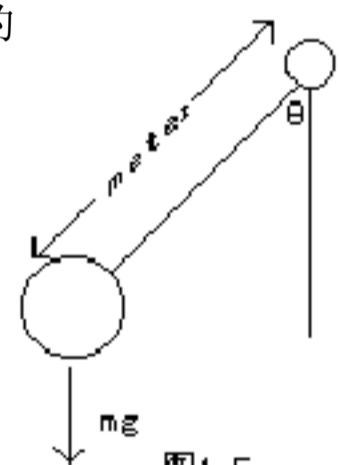
根据牛顿定理有

$$F = mL\ddot{\theta}$$

联结上两式，得

$$mL\ddot{\theta} + BL\dot{\theta} + W\sin\theta = 0$$

令 $x_1 = \theta, x_2 = \dot{\theta}$ 得



$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -\frac{B}{m}x_2 - \frac{W}{mL}\sin x_1$$

上面方程式在 M-文件 pendulum.m 中定义如下：

```
function xdot=pendulum(t,x); %状态导数
W=2; L=.6; B=0.02; g=9.81; m=W/g;
xdot=[x(2); -B/m*x(2)-W/(m*L)*sin(x(1))];
下面的 M-文件对系统进行仿真。
t0=0; tfinal=5; %时间间隔 0~5 秒
x0=[1,0]; %初始条件 x1=1,x2=0
tol=0.0001; trace=0; %精度, 不打印每一步的计算值
[t, x]=ode23('pendulum',t0,tfinal,x0,tol,trace);
subplot(211),plot(t, x)
title('Time response of a rigid pendulum'),xlabel('Time-sec')
text(3.2,3.5,'Velocity'),text(3.2,1.2,'Angle-Rad')
th=x(:,1);w=x(:,2);
subplot(212),plot(th, w)
title('Phase plane plot of pendulum') xlabel('Position-
Rad'),ylabel('Angular velocity') subplot(111)
```

仿真结果见图 1-6。

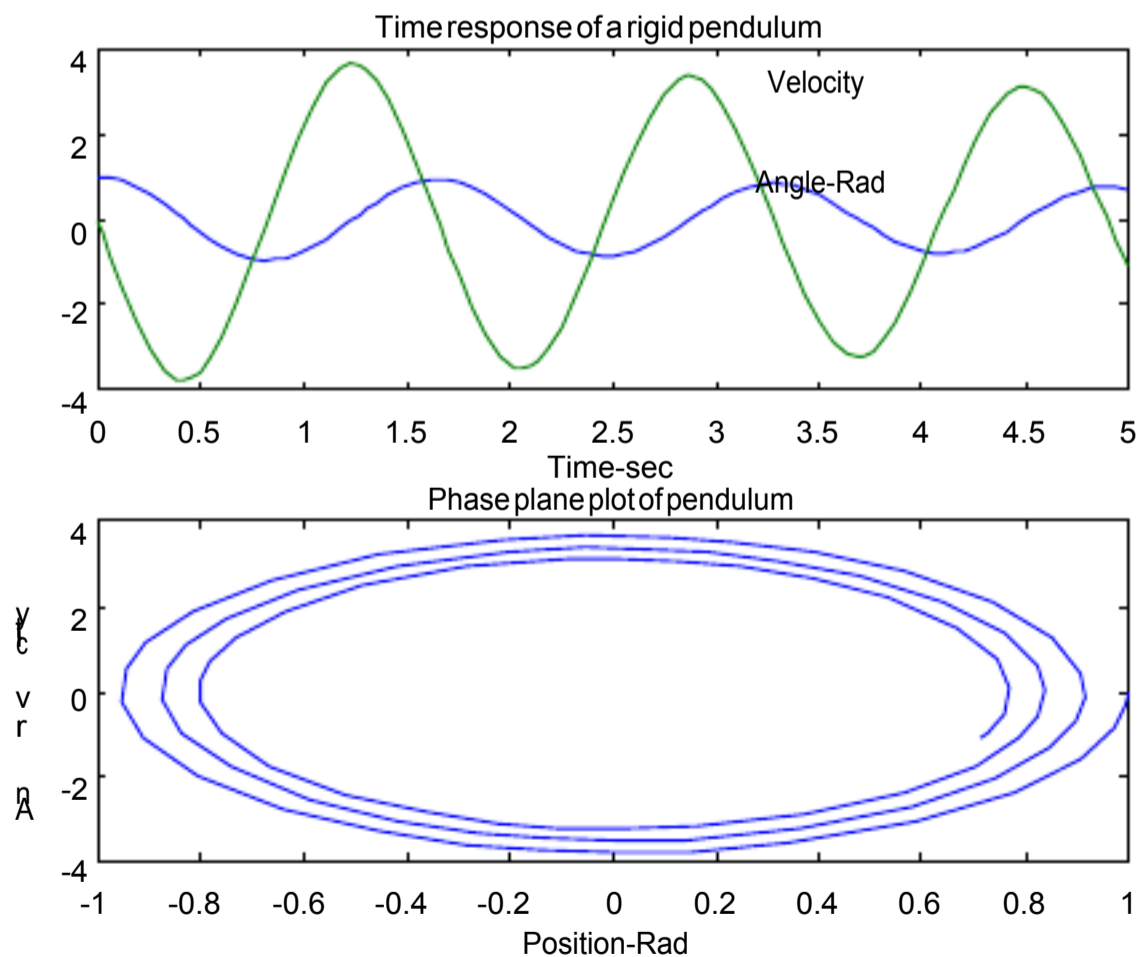


图 1-6

1.1.2 线性化

在小信号条件下，非线性系统可以线性化。例 1.3 中单摆的运动方程在初始角较小的情况下，可被线性化。设 $\theta = \theta_0 + \Delta\theta$ ，单摆运动方程如下：

$$mL(\ddot{\theta}_0 + \Delta\ddot{\theta}) + BL(\dot{\theta}_0 + \Delta\dot{\theta}) + W\sin(\theta_0 + \Delta\theta) = 0 \quad (1.1)$$

当 $\Delta\theta \ll \infty$ 时，有 $\sin\Delta\theta \approx \Delta\theta$, $\cos\Delta\theta \approx 1$ ，又因为 θ_0 较小，有 $\sin\theta_0 \approx \Delta\theta$ ，那么有线性方程

$$mL\Delta\ddot{\theta} + BL\Delta\dot{\theta} + W\Delta\theta = 0 \quad (1.2)$$

当 $\Delta\theta$ 很小时，线性方程 (1.2) 的解与 (1.1) 一致。

1.2 传递函数

线性时不变系统的传递函数定义为零初始条件下输出量的拉普拉斯变换与输入量的拉普拉斯变换像函数之比。尽管传递函数只能用于线性系统，但它比微分方程提供更为直观的信息。令传递函数的分母多项式等于零，便得到特征方程。特征方程的根是系统的极点，分子多项式的根是系统的零点。那么传递函数便可由常数项与系统的零、极点确定。常数项，通常记作 k ，是系统的增益。利用传递函数，我们可以方便的研究系统参数的改变对响应的影响。通过拉普拉斯反变换可得到系统在时域响应。通常需要用有理函数的部分分式展开。

在这部分举几个例子介绍 MATLAB 中求特征多项式的根，求传递函数的零、极点，部分分式展开以及已知零、极点求传递函数等函数的功能。

1.2.1 多项式的根和特征多项式

如果 P 是包含多项式系数的行向量，`roots(P)` 得到一个列向量，其元素为多项式的根。如果 r 是包含多项式根的一个行/列向量，`poly(r)` 得到一个行向量，其元素为多项式的系数。

例1.4 求多项式

$$s^6 + 9s^5 + 31.25s^4 + 61.25s^3 + 67.75s^2 + 14.75s + 15 \text{ 的根。}$$

多项式系数以降幂次序排列在一行向量中。用 `roots` 求根。

```
>>P=[1 9 31.25 61.25 67.75 14.75 15];
```

```
>>r=roots(P)
```

多项式的根从列向量 r 中得到

```
r =  
-4.0000  
-3.0000  
-1.0000 + 2.0000i  
-1.0000 - 2.0000i  
-0.0000 + 5.0000i  
-0.0000 - 5.0000i
```


例 1.5 多项式的根为 $-1, -2, -3 \pm j4$ 。求多项式方程。

为了输入复数，必须首先建立虚数单位。然后在行/列向量中输入根。使用 `poly` 得到多项式方程。

```
>> i=sqrt(-1);  
>> r=[-1;-2;-3+4*i;-3-4*i];  
>> p=poly(r)  
多项式的系数从行向量中得到  
p =  
    1    9   45   87   50
```

因此，多项式方程为

$$s^4+9s^3+45s^2+87s+50=0$$

例 1.6 求下列矩阵的特征方程的根

$$A = \begin{bmatrix} 0 & 1 & -1 \\ -6 & -11 & 6 \\ -6 & -11 & 5 \end{bmatrix}$$

用 `poly` 求矩阵的特征方程的根。用 `roots` 求方程的根。

```
>> A=[0 1 -1;-6 -11 6;-6 -11 5];  
>> P=poly(A)  
>> r=roots(P)  
结果如下：  
P=  
    1.0000    6.0000   11.0000    6.0000  
r =  
   -3.0000  
   -2.0000  
   -1.0000
```

1.2.2 传递函数的零点和极点

函数 `tf2zp` 求传递函数的零点，极点和增益。

例 1.7 求下列传递函数的零点，极点和增益。

$$H(s) = \frac{s^3 + 11s^2 + 30s}{s^4 + 9s^3 + 45s^2 + 87s + 50}$$

```
>> num=[1 11 30 0];  
>> den=[1 9 45 87 50];  
>> [z, p, k]=tf2zp(num, den)
```

z =

0
-6.0000
-5.0000

p =

-3.0000 + 4.0000i
-3.0000 - 4.0000i
-2.0000
-1.0000

因而有

$$H(s) = \frac{s(s+5)(s+6)}{(s+1)(s+2)(s+3+j4)(s+3-j4)}$$

函数 `zp2tf` 根据给定零点，极点和增益求传递函数。

例1.8 系统的零点为-6,-5,0,极点为-3±j4,-2,-1,增益为 1。求其传递函数。

```
>> z=[-6;-5;0];k=1;  
>> i=sqrt(-1);  
>> p=[-3+4*i;-3-4*i;-2;-1];  
>> [num, den]=zp2tf(z, p, k)
```

上面程序的结果为

```
num =  
    0    1   11   30    0  
den =  
    1    9   45   87   50
```

因此，传递函数为

$$H(s) = \frac{s^3 + 11s^2 + 30s}{s^4 + 9s^3 + 45s^2 + 87s + 50}$$

1.2.3 部分分式展开

函数 `[r, p, k]=residue(b, a)`，对两个多项式的比进行部分分式展开，如

$$\frac{P(s)}{Q(s)} = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{a_n s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0} \quad (1.3)$$

向量 **b**, **a** 以 **s** 的降幂顺序排列多项式的系数。部分分式展开后余数送入列向量**r**，极点送入列向量**p**，常数项送入**k**。

例1.9 对 **F(s)**进行部分分式展开

$$F(s) = \frac{2s^3 + 9s + 1}{s^2 + 4s + 4}$$

>>b=[2091];

>>a=[1144];

>> [r, p, k]=residue(b,a)

结果如下:

r =

0.0000 - 0.2500i

0.0000 + 0.2500i

-2.0000

p =

-0.0000 + 2.0000i

-0.0000 - 2.0000i

-1.0000

k =

2

因而, 部分分式展开为

$$2 + \frac{-2}{s+1} + \frac{j0.25}{s+j2} + \frac{-j0.25}{s-j2} = 2 + \frac{-2}{s+1} + \frac{1}{s^2+4}$$

函数[b, a]=residue(r, p, k)将部分分式转化为多项式比 P(s)/Q(s)。

1.3 状态空间描述

集总参数的线性网络可用微分方程表示为

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (1.4)$$

该系统的一阶微分方程即为状态方程, X 是状态向量。状态空间方法易采用数字或模拟计算机求解。另外, 状态空间方法容易拓展到非线性系统。状态方程可从 n 阶微分方程得到, 或者在系统模型中选用合适的状态变量直接写出。

1.3.1 将微分方程化成状态方程

设一个 n 阶线性系统由微分方程描述, 我们讨论如何选择状态变量, 使该系统由状态方程描述。

$$\frac{d^n y}{dt^n} + a_{n-1} \frac{d^{n-1} y}{dt^{n-1}} + \dots + a_1 \frac{dy}{dt} + a_0 y = u(t) \quad (1.5)$$

y(t),u(t)分别为系统的输出、输入。

状态模型不是唯一的，它依赖于状态变量的选择。一个有用的选择方法是定义

$$x_1 = y, x_2 = \dot{y}, x_3 = \ddot{y}, \dots, x_n = y^{(n-1)}$$

那么，y 的 n 阶导数可用各状态变量的线性组合来表示。

于是有

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\vdots \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= -a_0 x_1 - a_1 x_2 - \dots - a_{n-1} x_n + u(t) \end{aligned}$$

写成矩阵形式为

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \Lambda & 0 \\ 0 & 0 & 1 & \Lambda & 0 \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ -a_0 & -a_1 & -a_2 & \Lambda & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(t) \quad (1.7)$$

输出方程为

$$y = [1 \ 0 \ 0 \ \dots \ 0]x \quad (1.8)$$

即

$$\begin{aligned} A &= \begin{bmatrix} 0 & 1 & 0 & \Lambda & 0 \\ 0 & 0 & 1 & \Lambda & 0 \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ & & & & \\ -a_0 & -a_1 & -a_2 & \Lambda & -a_{n-1} \end{bmatrix} & B &= \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \\ C &= [1 \ 0 \ 0 \ \Lambda \ 0] & D &= 0 \end{aligned} \quad (1.9)$$

例1.10 给定系统的微分方程描述为

$$y^{(3)} + 2y^{(2)} + 3y^{(1)} + 4y = 5u(t)$$

利用(1.9)即可写出其相应的一个状态空间描述为：

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -4 & -3 & -2 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 5 \end{bmatrix} \quad C = [1 \ 0 \ 0]$$

上述状态方程，称为能控标准型。

1.3.2 矩阵的对角化

为什么要研究 A 矩阵的对角化问题呢？原因之一是将矩阵对角化后，对角元即为矩阵 A

的特征值 $\lambda_1, \lambda_2, \dots, \lambda_n$ 。从而，状态转移矩阵亦可化为对角元为 $e^{\lambda_1 t}, e^{\lambda_2 t}, \dots, e^{\lambda_n t}$ 的对角矩阵。
 (假定 A 有 n 个互不相同的特征值)。

给定线性系统。当 A 有 n 个互不相同的特征值时，我们可以找到非奇异变换矩阵 P ，令

$$x(t) = py(t) \quad (1.10)$$

将上面状态方程化为对角线规范型

$$\dot{y} = ay + bu(t) \quad (1.11)$$

其中

$$a = p^{-1}AP, \quad b = p^{-1}B$$

有很多方法可以求得 P ，如利用 A 的特征向量可构造 P 。

例1.11 定系统的状态空间描述为

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & -1 \\ -6 & -11 & 6 \\ -6 & -11 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u$$

求它的对角规范型和变换矩阵 P 。

```
>> A=[0 1 -1;-6 -11 6;-6 -11 5]; B=[0;0;1]
```

```
>> [P,L]=eig(A); %L 是一个对角元为特征值的对角矩阵
                %P 是一个变换矩阵，其列是相应于特征值的特征向量
```

```
>> a=inv(P)*A*P %A 矩阵对角化
```

```
>> b=inv(P)*B
```

结果为：

```
p =
    -0.7071    -0.2182    -0.0921
    -0.0000     0.4364    -0.5523
    -0.7071     0.8729    -0.8285
```

```
a =
    -1.0000    -0.0000    -0.0000
     0.0000    -2.0000    -0.0000
    -0.0000     0.0000    -3.0000
```

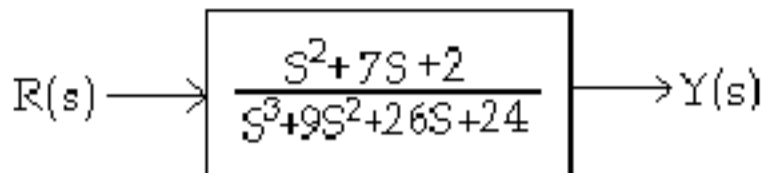
```
b =
     2.8284
    13.7477
    10.8628
```

1.4 模型转换

1.4.1 传递函数向状态空间描述的转换

控制系统工具箱包含一组模型转换的函数。[A,B,C,D]=tf2ss(num, den)将传递函数转换为状态空间描述。

例 1.12 求下面传递函数的状态空间描述



```
>> num=[1 7 2]; den=[1 9 26 24];
```

```
>> [A, B, C, D]=tf2ss(num, den)
```

状态方程各矩阵如下：

$$A = \begin{bmatrix} -9 & -26 & -24 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad C = [1 \ 7 \ 2]$$

$$B = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \quad D = 0$$

1.4.2 状态空间描述向传递函数的转换

已知状态方程和输出方程

$$\dot{x} = Ax + Bu \tag{1.12}$$

$$y = Cx + Du \tag{1.13}$$

采用拉普拉斯变换有

$$Y(s) = C(sI - A)^{-1}Bu(s) + Du(s)$$

则

$$G(s) = \frac{Y(s)}{U(s)} = C(sI - A)^{-1}B + D \tag{1.14}$$

函数 ss2tf(A,B,C,D,i)是将状态空间描述转换为对第一个输入的传递函数。

[num,den]=ss2tf(A,B,C,D,i)是将状态空间描述化为分子、分母多项式形式的传递函数。

[z,p,k]=ss2zp(A,B,C,D,i)将状态空间描述化为零极点形式表示的传递函数。

例 1.13 一个系统的状态空间描述如下

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -2 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 10 \\ 0 \\ 0 \end{bmatrix} u$$

$$y = [1 \ 0 \ 0]x$$

求传递函数G(s)=Y(s)/U(s)


```
>>A=[0 1 0;0 00 1;-1 -2 -3]; B=[10; 0; 0];
>>C=[1 0 0];D=[0];
>>[num,den]=ss2tf(A,B,C,D,1)
>>[z,p,k]=ss2zp(A,B,C,D,1)
```

其中，ss2tf(A,B,C,D,1)中“1”表示对第一个输入。

传递函数的分子、分母多项式系数如下：

```
num=
    0    10.0000    30.0000    20.0000
den=
    1.0000    3.0000    2.0000    1.0000
```

传递函数的零、极点如下：

```
z=
   -1
   -2
p=
-0.3376+0.5623i
-0.3376-0.5623i
-2.3247
```

k=10

因而传递函数为

$$G(s) = \frac{10(s^2 + 3s + 2)}{s^3 + 3s^2 + 2s + 1} = \frac{10(s+1)(s+2)}{(s+0.3376-0.5623i)(s+0.3376+0.5623i)(s+2.3247)}$$

1.4.3 由方框图求状态空间描述和传递函数

控制系统工具箱中提供了函数[A,B,C,D]=connect(a, b, c, q, iu, iy)。将方框图描述转换成状态空间描述和传递函数。其中 q 矩阵规定了各框之间的连接关系。其每一行的第一个元素是框号，其余的元素依次是于该框连接的框号，iu,iy 分别表示输入，输出施加的框号。

例 1.14 将图 1-7 由框图表示的系统转换成状态空间描述和传递函数。

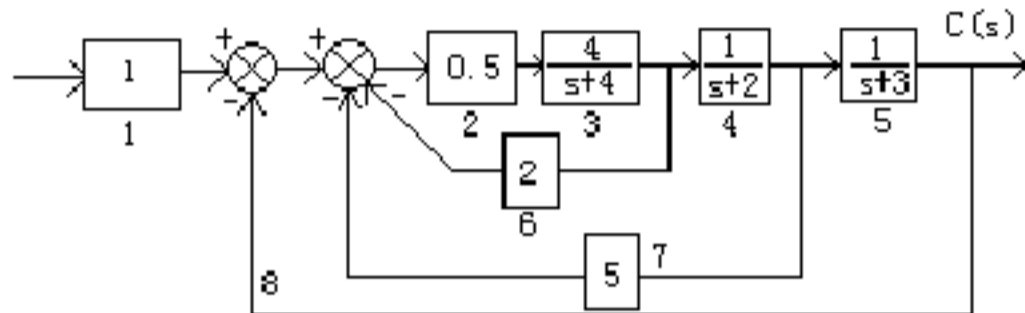
```
>>n1=1;d1=1; n2=0.5;d2=1; n3=4;d3=[1 4];
>>n4=1;d4=[1 2]; n5=1;d5=[1 3]; n6=2;d6=1;
>>n7=5;d7=1; n8=1;d8=1;
>>nblocks=8; blkbuild
>>q=[1 0 0 0 0          %q 矩阵表示框图的结构。
    2 1 -6 -7 -8        如第 2 个框于第 1 个框按
```

```

3  2  0  0  0
4  3  0  0  0
5  4  0  0  0
6  3  0  0  0
7  4  0  0  0
8  5  0  0  0];

```

1 的关系连接，于第 6.7.8
个框按-1 关系连接，依次类推。



```

>>iu=[1];           %输入施加于第 1 个框上
>>iy=[8];           %由第 8 个框输出
>>[A, B, C, D]=connect(a, b, c, d, q, iu, iy)
>>[num,den]=ss2tf(A,B,C,D,1)           %转换成传递函数

```

结果为

A=

```

-8.0  -2.5  -0.5
0.4   -2.0   0
0     1.0  -3.0

```

B=

```

0.5
0
0

```

C=

```

0 0 1

```

D=

```

0

```

num=

```

0 0 0 2

```

den=

```

1.0 13.0 56.0 80.0

```

即

$$\frac{C(s)}{R(s)} = \frac{2}{s^3 + 13s^2 + 56s + 80}$$

第二章 控制系统的校正

2.1 单变量系统的两种主要校正方式

单变量系统常用的校正主要有两种方式。一种是校正装置与被控对象串联，如图2-1所示。这种校正方式称为串联校正。另一种校正方式是从被控对象中引出反馈信号，与被控对象或其一部分构成反馈回路，并在局部反馈回路设置校正装置。这种校正方式称为局部反馈校正，如图2-2所示。

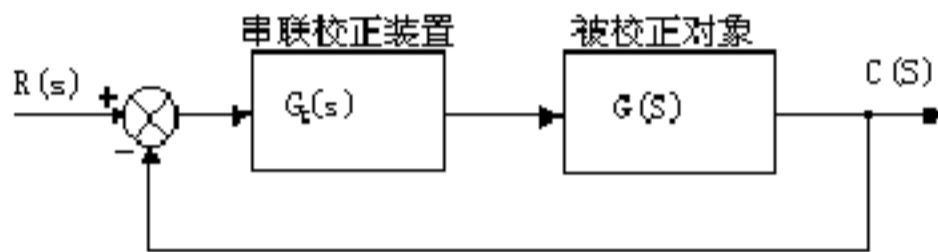


图 2-1

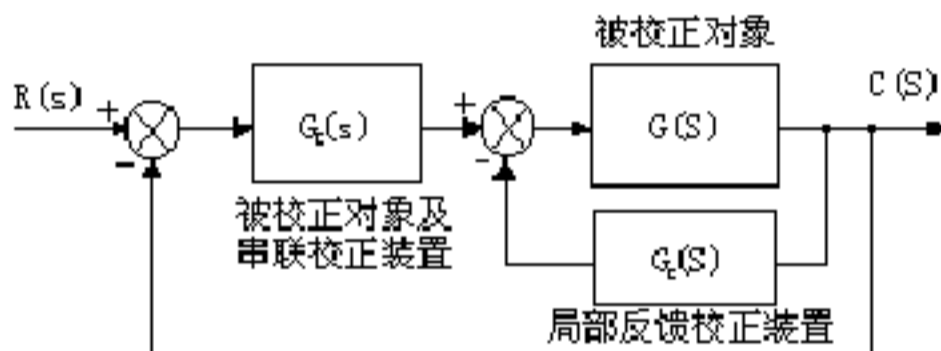


图 2-2

串联校正和局部反馈校正应用都相当普遍，究竟选择哪一种，取决于系统中信号的性质，可供采用的元件以及其他条件。两种校正方式结合起来可以收到更好的效果。

2.2 PI、PD、PID 校正

2.2.1 PD 校正（超前校正）

超前校正（亦称PD校正）的传递函数为

$$G(s) = \frac{\alpha Ts + 1}{Ts + 1}, (\alpha > 1) \quad (2.1)$$

其对数频率特性如图2-3所示。超前校正能够产生相位超前角，它的强度可由参数 α 表征。

超前校正的相频特性函数是

$$\theta(\omega) = \arctg \alpha \omega T - \arctg \omega T \quad (2.2)$$

最大相移点位于对数频率的中心点，即

$$\omega_m = \frac{1}{\sqrt{\alpha}} \cdot \frac{1}{T} \quad (2.3)$$

最大相移量为

$$\theta_m = \theta(\omega_m) = \arctg \sqrt{\alpha} - \arctg \frac{1}{\sqrt{\alpha}} = \arcsin \frac{\alpha - 1}{\alpha + 1} \quad (2.4)$$

或者

$$\sin \theta_m = \frac{\alpha - 1}{\alpha + 1}$$

$$\alpha = \frac{1 + \sin \theta_m}{1 - \sin \theta_m} \quad (2.5)$$

容易求出，在 ω_m 点有

$$L(\omega_m) = 10 \lg \alpha \quad (2.6)$$

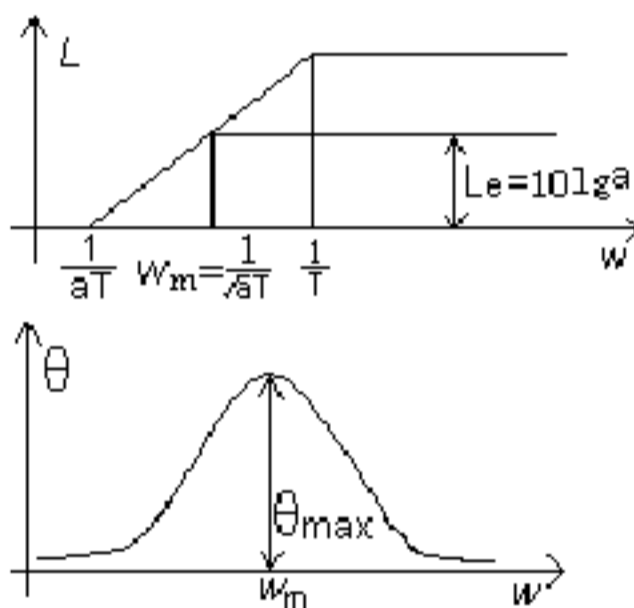


图 2-3

基于频率法综合超前校正的步骤是：

1. 首先根据静态指标要求，确定开环比例系数 K ，并按已确定的 K 画出系统固有部分的 Bode 图。
2. 根据静态指标要求预选 ω_c ，从 Bode 图上求出系统固有部分在 ω_c 点的相角。
3. 根据性能指标要求的相角裕量，确定在 ω_c 点是否需要提供相角超前量。如需要，算出需要提供的相角超前量 θ_m 。
4. 如果所需相角超前量不大于 60° ，按 (2.5) 求出超前校正强度 α 。
5. 令 $\omega_m = \omega_c = \frac{1}{(\sqrt{\alpha} T)}$ ，从而求出超前校正的两个转折频率 $1/\alpha T$ 和 $1/T$ 。
6. 计算系统固有部分在 ω_c 点的增益 $L_g(\text{dB})$ 及超前校正装置在 ω_c 点的增益 $L_c(\text{dB})$ 。如

果 $Lg+Lc>0$ ，则校正后系统的截止角频率 ω_c' 比预选的值要高。如果高出较多，应采用滞后超前校正，如果只是略高一些，则只需核算 ω_c' 点的相角裕量，若满足要求，综合完毕，否则转第 3 步。

如果 $Lg+Lc<0$ ，则实际的 ω_c' 低于预选的 ω_c 。可将系统的开环增益提高到 $Lg+Lc=0$ （即将系统的开环比例系数提高 $lg^{-1}[-(Lg+Lc)]/20$ 倍）。

超前校正的主要作用是产生超前相角，可用于补偿系统固有部分在截止角频率 ω_c 附近的相角滞后，以提高系统的相角稳定裕量，改善系统的动态特性。

2.2.2 PI 校正（滞后校正）

滞后校正（亦称PI校正）的传递函数的

$$G(s) = \frac{Ts+1}{\beta Ts+1}, (\beta > 1) \quad (2.7)$$

其对数频率特性如图 2-4 所示。参数 β 表征滞后校正的强度。

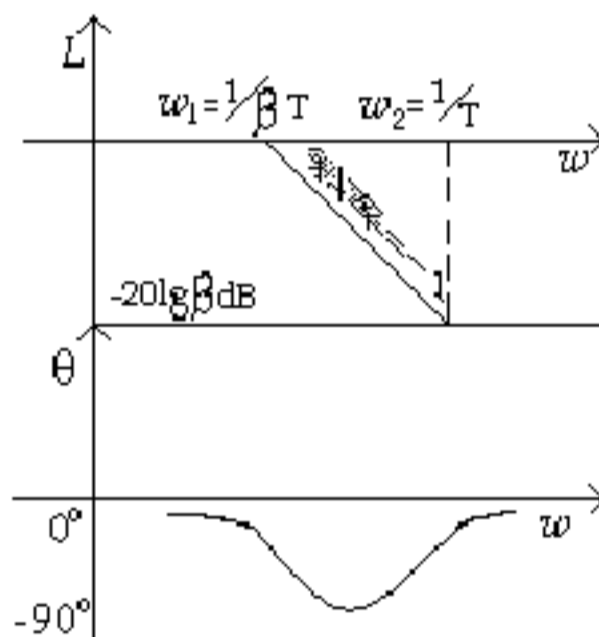


图 2-4

基于频率法的滞后校正指标的综合步骤是：

1. 首先根据静态指标要求确定开环比例系数 K ，按照所确定的 K 画出系统固有部分的 Bode 图。
2. 根据动态指标要求试选 ω_c ，从图上求出在试选的 ω_c 点的相角，判断是否满足相位裕量的要求（注意计入滞后校正将会带来的 $50^\circ \sim 120^\circ$ 的滞后量），如果满足，转向下一步。否则，如果允许降低 ω_c ，就适当重选较低的 ω_c 。
3. 从图上求出系统固有部分在 ω_c 点的开环增益 $Lg(\omega_c)$ 。如果 $Lg(\omega_c) > 0$ ，令 $Lg(\omega_c) = 20lg \beta$ ，求出 β ，就是滞后校正的强度，如果 $Lg(\omega_c) < 0$ ，则无须校正，且可将开环比例系数提高。
4. 选择 $\omega_2 = 1/T = (1/5 \sim 1/10) \omega_c$ ，进而确定 $\omega_1 = 1/(\beta T)$ 。
5. 画出校正后系统的 Bode 图，校核相位裕量。

滞后校正的主要作用是降低中频段和高频段的开环增益，但同时使低频段的开环增益不受影响，从而达到坚固静态性能和稳定性。它的副作用是会在 ω_c 点产生一定的相角滞后。

2.2.3 PID 校正（滞后超前校正）

超前校正的主要作用是增加相角裕量，改善系统的动态响应特性。滞后校正的主要作用是改善系统的静态特性，两种校正结合起来就能同时改善系统的动态和静态特性。滞后超前校正（亦称PID校正）综合了前面两种校正的功能。

滞后超前校正的传递函数为

$$G(s) = \frac{(T_2s+1)(\alpha T_1s+1)}{(\beta T_2s+1)(T_1s+1)}, (\beta \geq \alpha > 1), (T_2 > \alpha T_1) \quad (2.8)$$

它相当于一个滞后校正与一个超前校正相串联，其对数频率特性如图2-5所示。

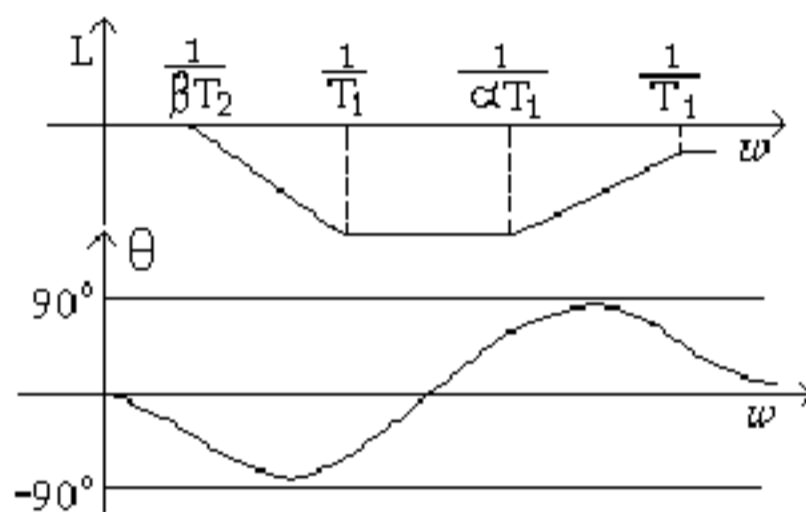


图 2-5

基于频率法的滞后超前校正的综合步骤是：

1. 首先根据静态指标要求确定开环比例系数，按照所确定的 K 画出系统固有部分的 Bode 图。
2. 按指标要求确定 ω_c ，检查系统固有部分在 ω_c 的对数幅频的斜率是否为-2，如果是，求出 ω_c 点的相角。
3. 按综合超前校正的步骤 3~6 综合超前部分 $G_{c1}(s)$ (注意在确定 θ_m 时要计入滞后校正将造成的 $50 \sim 120$ 的相角滞后量)。在第 6 步时注意，通常 $Lg(\omega_c) + Lc(\omega_c)$ 比 0 高出很多，所以要引进滞后校正。
4. 令 $20 \lg \beta = Lg(\omega_c) + Lc(\omega_c)$, 求出 β 。
5. 按综合校正的步骤 4~5 综合滞后部分 $G_{c2}(s)$ 。
6. 将滞后校正与超前校正串联在一起，构成滞后超前校正： $G_c(s) = G_{c1}(s) * G_{c2}(s)$ 。

2.3 串联校正举例

对于一结构如图 2-6 所示的系统，给定固有部分的传递函数 $G_g(s)$ 和性能指标要求，试设计串联校正装置 $K(s)$ 。设

$$G(s) = \frac{100}{s(0.1s+1)(0.01s+1)}$$

1. 若要求开环比例系数 $K \geq 100$ 1/秒，相角裕量 $\gamma \geq 30^\circ$ ， $\omega_c \geq 45$ 1/秒。
2. 若要求开环比例系数 $K \geq 100$ 1/秒，相角裕量 $\gamma \geq 40^\circ$ ， $\omega_c = 5$ 1/秒。
3. 若要求开环比例系数 $K \geq 100$ 1/秒，相角裕量 $\gamma \geq 40^\circ$ ， $\omega_c \geq 20$ 1/秒，分别设计校正装置，并比较它们的作用效果。

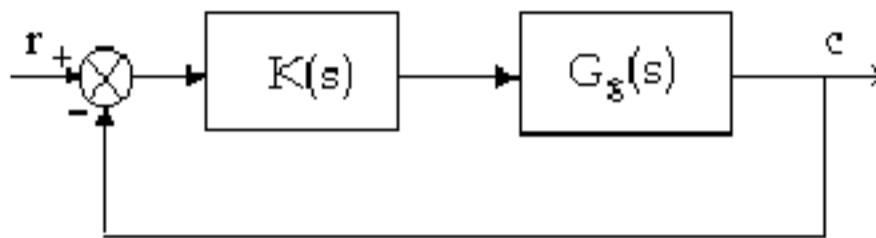


图 2-6

对于 1，需进行超前校正，这里给出一个参考的 $K(s) = (0.05s+1)/(0.005s+1)$ ，于是系统的开环传递函数为

$$G(s) = \frac{0.05s+1}{0.005s+1} \frac{100}{s(0.1s+1)(0.01s+1)} = \frac{1000,000(s+20)}{s(s+10)(s+100)(s+200)}$$

闭环传递函数为

$$G_0(s) = \frac{1000,000s+20,000,000}{s(s+10)(s+100)(s+200)+1000,000s+20,000,000}$$

$$= \frac{1000,000s+20,000,000}{s^4+310s^3+23,000s^2+1200,000s+20,000,000}$$

其单位阶跃响应的实现为：

```
num=[1000,000    20,000,000];
den=[1    310    23,000    1200,000    20,000,000];
t=xx:xx:xx; (起始时间, 步长, 终止时间)
c=step(num, den, t);
plot(t, c);
xlabel('t-sec. '), ylabel('c(t)'), grid, pause
```

对于 2 需进行滞后校正，这里给出一参考的

$$K(s) = \frac{0.5s+1}{10s+1}$$

仿照上述，可求出系统的阶跃响应。

对于 3，需进行滞后超前校正，这里给出一参考的

$$K(s) = \frac{(0.25s+1)(0.1s+1)}{(1.25s+1)(0.02s+1)}$$

同理可得到系统的阶跃响应。

比较上述三种不同的校正装置 $K(s)$ 所产生的不同效果，可以看出三种校正方式的作用。

这里需要特别指出阶跃响应的起始时间，步长及终止时间的给定问题。

起始时间通常给 0，终止时间要根据系统的过渡过程时间的长短来给定。我们可按估计过渡过程时间的经验公式

$$t_s \approx (4 \sim 9) / \omega_c$$

给定此值：如 2， $\omega_c = 5$ 1/秒，则过渡过程时间大约在 0.8~1.8 秒之间，于是终止时间可给 2 秒，我们希望在 0 至 2 秒内显示出 20 个点，则步长为 0.1 秒。若已知系统的 ω_c ，建议步长按 $1/(2\omega_c)$ 给定。对 ω_c 未知，也不便估计过渡过程时间的系统只好用试探法，待曲线显示出以后再调整步长和终止时间。

图 2-7 为系统固有部分的频率特性图（幅频特性和相频特性）可以看出该系统已接近临界稳定状态了。

图 2-8 至图 2-11 分别表示不加校正，加超前校正，加滞后校正，和加滞后超前校正的系统阶跃响应。

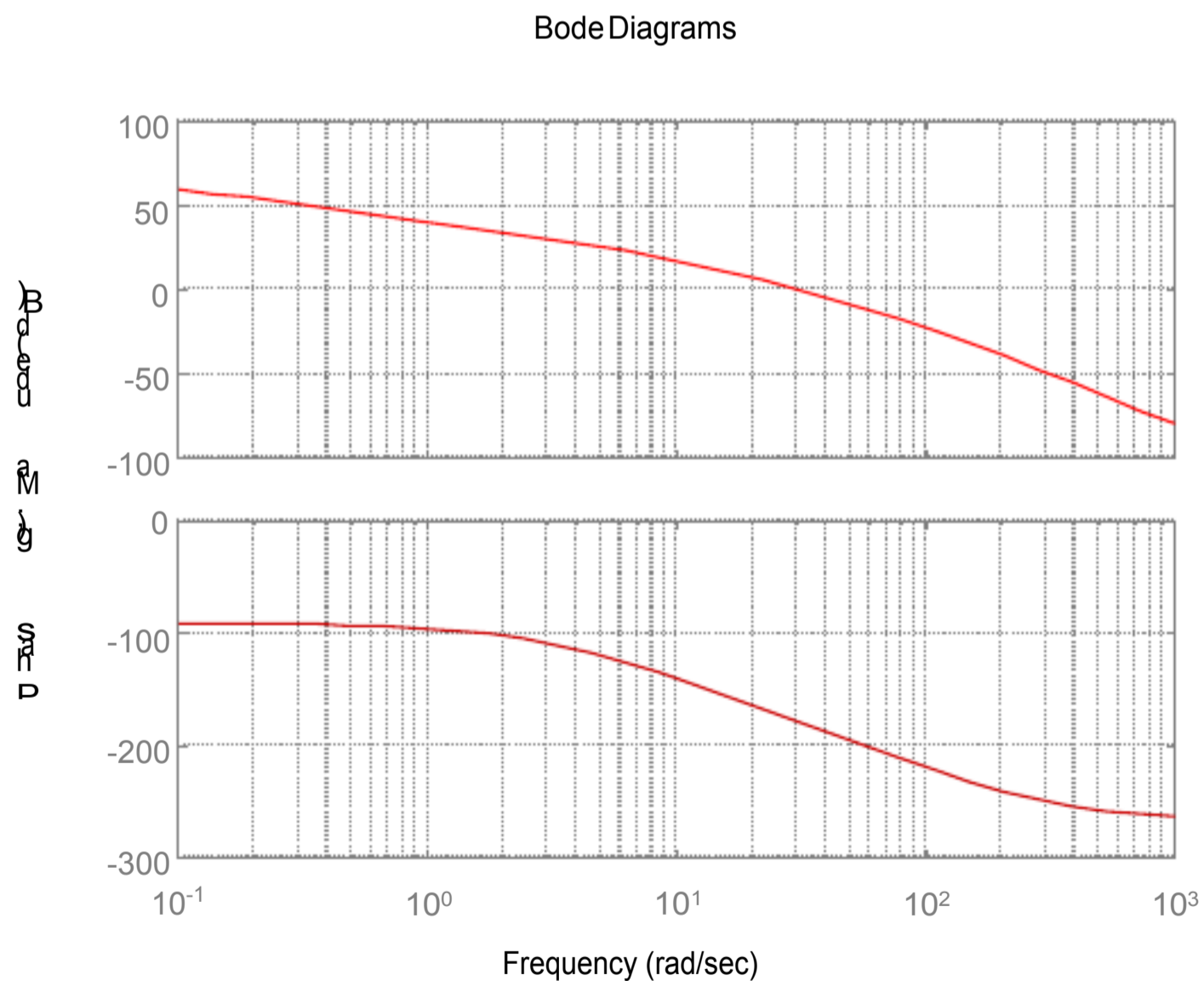


图 2-7

```
num=100;  
den=conv([1,0], conv([0.1,1], [0.01,1]));  
bode(num, den)
```

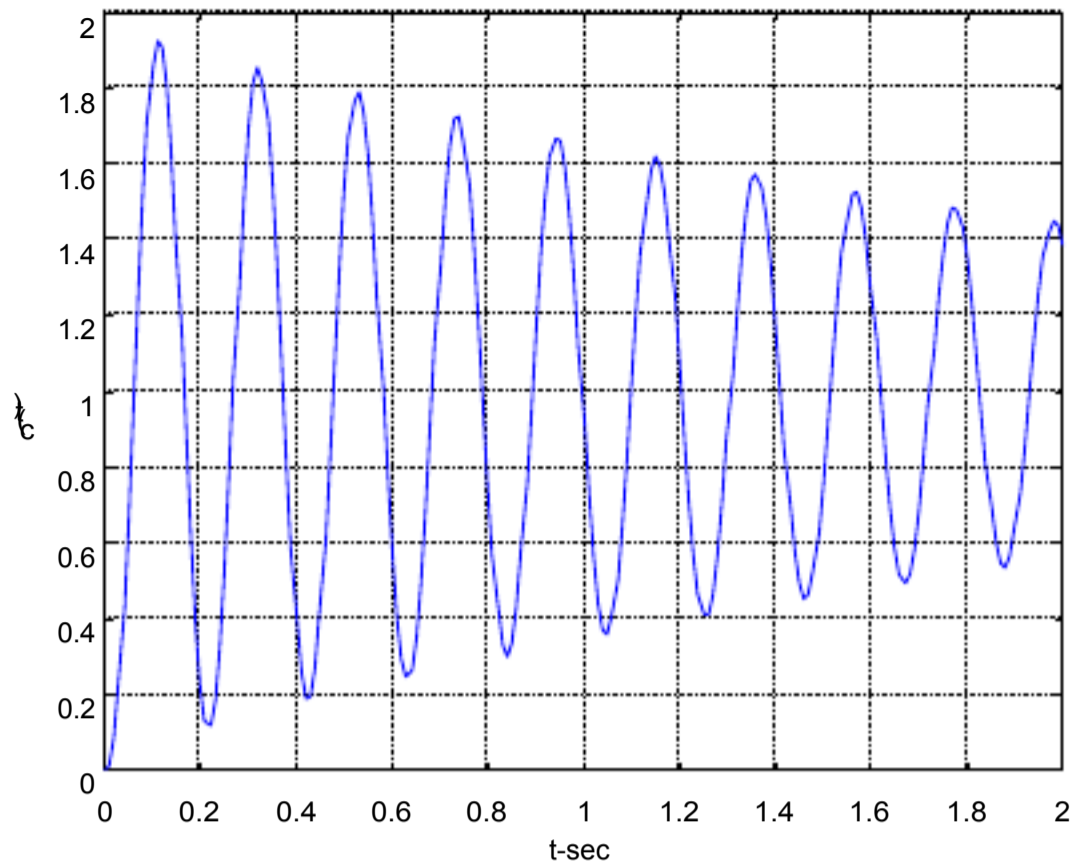



图 2-8

```

numo=100;
deno=conv([1,0],conv([0.1,1],[0.01,1]));
t=0:0.01:2;
numc=numo;
denc=[zeros(1,length(deno)-length(numo)),numo]+deno; c=step(numc,denc,t);
plot(t,c)
xlabel('t-sec'),ylabel('c(t)'),grid,pause

```

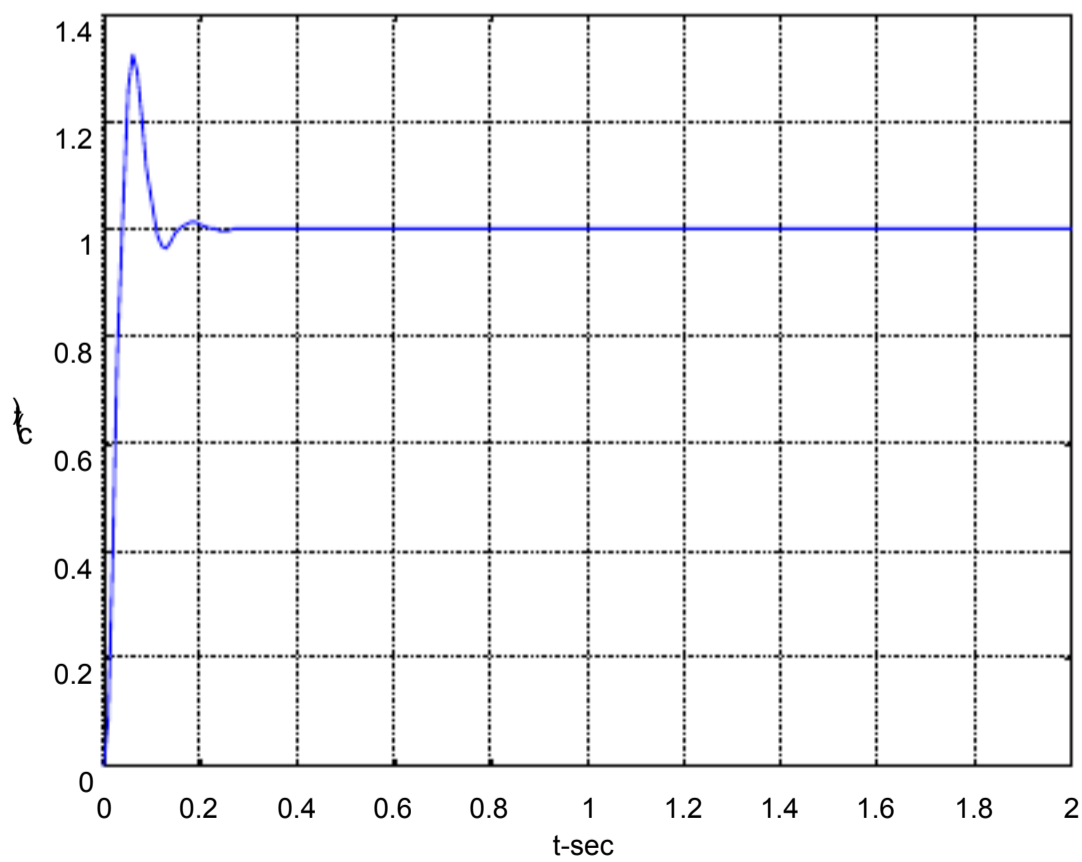


图 2-9

```

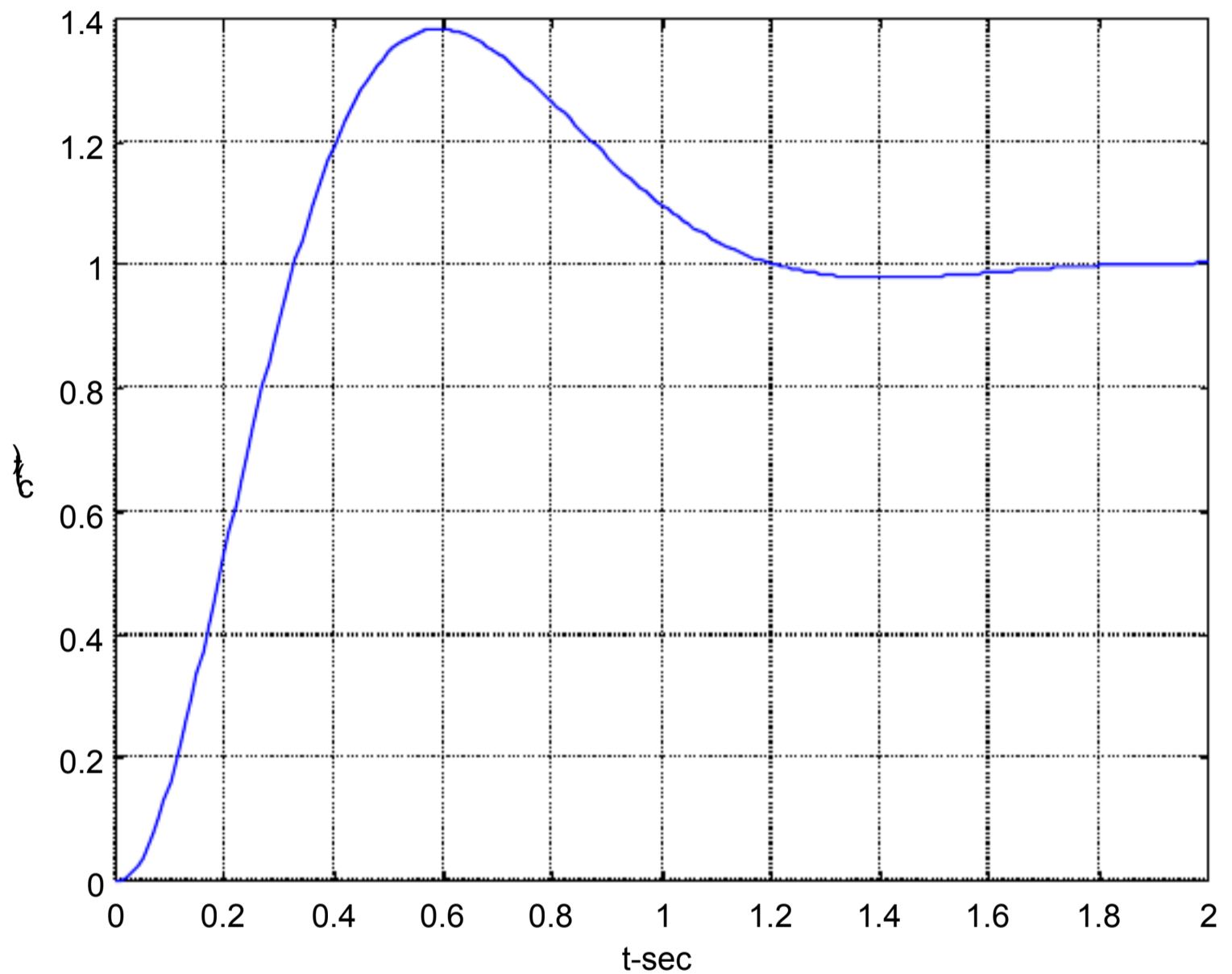
numo=conv(100,[0.05,1]);
deno=conv([0.005,1],conv([1,0],conv([0.1,1],[0.01,1])));
t=0:0.01:2;
numc=numo;
denc=[zeros(1,length(deno)-length(numo)),numo]+deno; c=step(numc,denc,t);
plot(t,c)

```

```

xlabel('t-sec'),ylabel('c(t)'),grid,pause

```



```

numo=conv(100,[0.5,1]);
deno=conv([10,1],conv([1,0],conv([0.1,1],[0.01,1]))); t=0:0.01:2;
numc=numo;
denc=[zeros(1,length(deno)-length(numo)),numo]+deno; c=step(numc,denc,t);
plot(t,c)
xlabel('t-sec'),ylabel('c(t)'),grid,pause

```

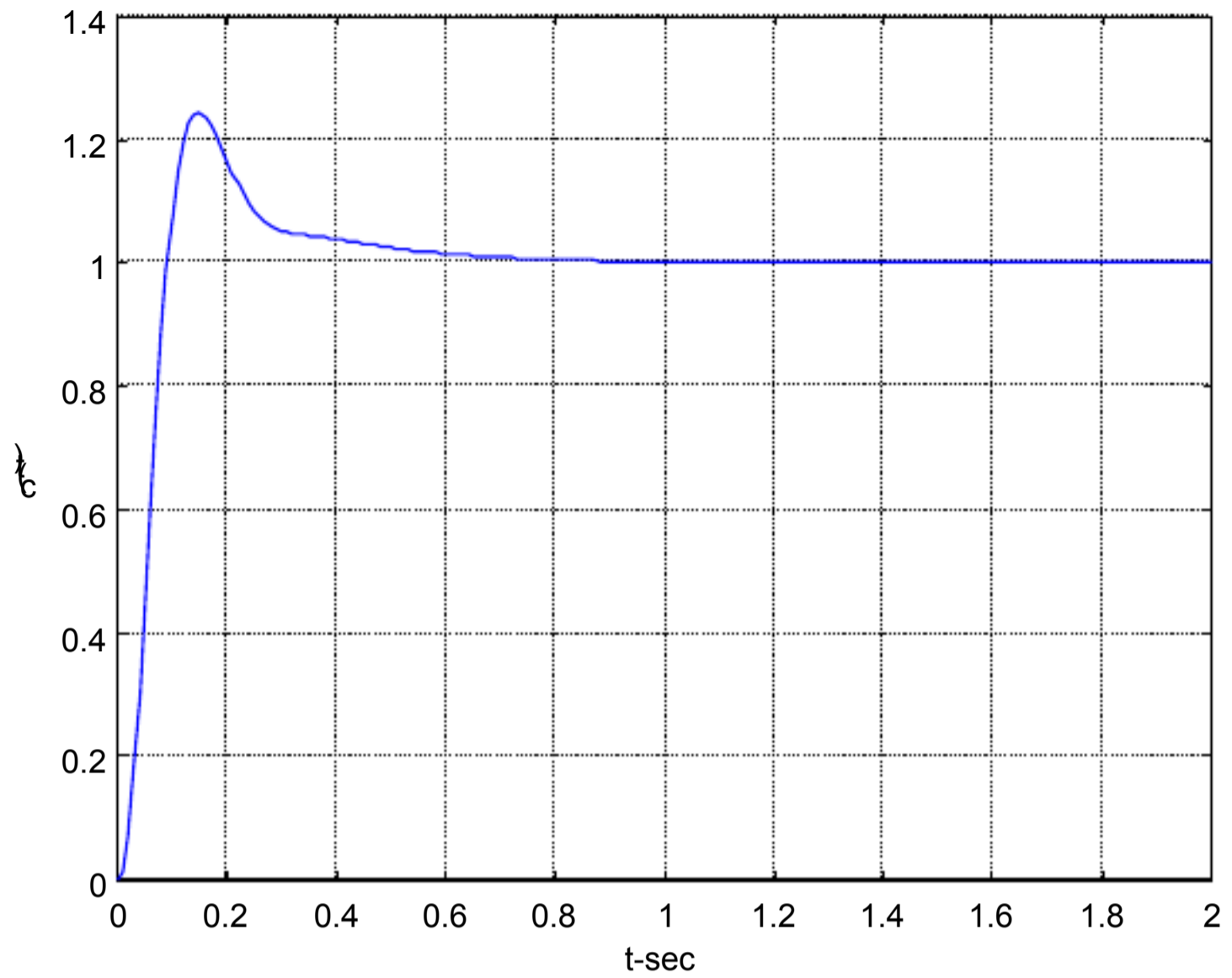


图 2-11

```

numo=conv(conv([0.25,1],[0.1,1]],[100]);
deno=conv(conv([1.25,1],[0.02,1]),conv([1,0],conv([0.1,1],[0.01,1]))); t=0:0.01:2;
numc=numo;
denc=[zeros(1,length(deno)-length(numo)),numo]+deno; c=step(numc,denc,t);
plot(t,c)
xlabel('t-sec'),ylabel('c(t)'),grid,pause

```

第三章 MATLAB 在控制理论中的应用

3.1 概述

MATLAB 提供了大量的控制工程计算、设计库函数。其中，控制系统软件包包括复数运算、特征值计算、方程求解、矩阵变换以及 FFT 等重要计算工具及举例。MATLAB 的线性代数处理，矩阵运算和数值分析的能力为控制系统工程设计及其它学科研究提供了可靠的基础和强有力的研究工具。

控制系统软件包利用 MATLAB 矩阵功能提供了适用于控制工程的专用函数，这些函数大部分用 M 文件表示。控制系统软件包可以方便地用于控制系统设计、分析和建模。

在控制系统软件包中，控制系统通常采用传递函数与状态空间两种形式建模，允许“经典”和“现代”技术并用，既可处理连续时间系统也可处理离散时间系统，并且可以进行不同模型表示形式之间的相互转换，也可以计算和绘制时间响应、频率响应及根轨迹图。此外 M 文件还能够进行极点配置和最优控制器的参数计算。即使在软件包中没有提供的功能，也可以通过编写新的 M 文件方式来构造。

3.1.1 系统模型

控制系统软件包可用于线性时不变（简称LTI）系统模型。时不变系统模型包括：状态空间模型；传递函数模型；零一极点增益模型；部分分式模型；离散时间模型。

下面介绍如何用矩阵表示不同类型的线性时不变系统模型。

(1) 状态空间模型

一个 LTI 微分方程系统总可以用一组一阶微分方程组来表示。按矩阵或状态空间表示形式，LTI系统模型的一般形式如下

$$\dot{x}=Ax+Bu$$

$$y=Cx+Du$$

其中： u 是一个 n_u 维控制输入向量， x 是一个 n_s 维状态向量， y 是 n_y 维输出向量。

采用 MATLAB 表示状态空间系统十分容易。 A ， B ， C ， D 都是矩阵，均作为独立变量处理。

例 设有一个具有两个极点的二阶系统，自然角频率 $\omega_n=1.5$ ，阻尼系数 $\xi=0.2$ 。按下面状态空间模型表示形式输入该系统

$$W_n=1.5;$$

$$z=0.2;$$

$$a=[0 \quad 1$$

$$\begin{aligned}
 & \quad -Wn^2 \quad -2*z*Wn]; \\
 b = & [0 \\
 & \quad Wn^2]; \\
 c = & [1 \quad 0]; \\
 d = & 0;
 \end{aligned}$$

在 MATLAB 中，采用状态空间表示系统模型是最通用的 LTI 系统模型的表示形式。对于多输入多输出（简称 MIMO）系统，状态空间表示是唯一容易处理的模型表示形式之一。

(2) 传递函数模型

状态空间系统模型的等效表示是拉氏变换的传递函数形式如下

$$Y(S) = H(S) U(S)$$

$$\text{其中: } H(s) = C(SI - A)^{-1}B + D$$

在通常情况下，H(S) 需要用三维矩阵表示，即H(S)的维数是ny行，nu列，ns+1阶，其中 ns+1 是多项式的系数。由于 MATLAB 变量都是两维的，因此，只限于用单输入（输入是一维的 u）多输出（简称为 SIMO）系统描述方式表示该系统

$$H(s) = \frac{N(s)}{q(s)} = \frac{N(1)s^{n-1} + N(2)s^{n-2} + \dots + N(n)}{q(1)s^{n-1} + q(2)s^{n-2} + \dots + q(n)}$$

行向量 q 是按 s 降幂排列的分母系数。矩阵 N 包含了分子系数，输出向量的个数与向量行数相等。

例 考虑单输入多输出（简称 SIMO）系统

$$H(S) = \frac{\begin{bmatrix} 3s+2 & s^3+2s+5 \end{bmatrix}}{3s^3 + 5s^2 + 2s + 1}$$

按下面形式输入此系统到 MATLAB 中

$$\begin{aligned}
 \text{num} = & [0 \quad 0 \quad 3 \quad 2 \\
 & \quad 1 \quad 0 \quad 2 \quad 5]; \\
 \text{den} = & [3 \quad 5 \quad 2 \quad 1];
 \end{aligned}$$

系统输入表明，多项式中的缺省项系数值用零替补。

(3) 零-极点增益模型

传递函数可以表示为系数或零-极点增益形式，SIMO 系统表示为

$$H(s) = \frac{Z(s)}{P(s)} = K \frac{(s + Z(1))(s + Z(2))\dots(s + Z(n))}{(s + P(1))(s + P(2))\dots(s + P(n))}$$

MATLAB 中规定，把多项式的根存放到列向量中，而行向量包含多项式系数。所以在因子形式下，列向量 P 包含有传递函数分母的极点分布，分子零点存放到矩阵 Z 的列向量中。矩阵的列数与向量 y 的输出数相等。分子传递函数的增益存放到列向量 k 中、对于 SIMO 系

统，k 是标量。

在 MATLAB 中，把多项式的根存储于列向量中，而多项式的系数存储于行向量中，两者可以互相转换。

```

例   if
      P= [1 3 5 2]
      then
          r=roots (P)
          r=
              -1.2267+1.4671i
              -1.2267-1.4671i
              -0.5436
      and
          PP=poly (r)  PP=
              1.0000  3.0000  5.0000  2.0000
  
```

使其又返回到原多项式。以后将会看到，对于 SIMO 的 LTI 系统，tf2zp 将多项式传递函数形式转换为因子零-极点增益形式，而 zp2tf 则进行相反形式的转换。

例 考虑一个 SIMO 系统

$$H(S) = \frac{Z(s)}{P(s)} = \frac{\begin{bmatrix} 3(S+12) \\ 4(S+1)(S+2) \end{bmatrix}}{(s+3)(s+4)(s+5)}$$

输入形式如下

```

k=[3; 4];
z=[-12  -1
    inf  -2];
P= [-3
    -4
    -5];
  
```

在这个多输出系统中，第 1 个分子比第 2 个分子阶数低，因此，采用 inf 来补充第 1 个分子的那个无限零点，使第 1 个分子与第 2 个分子有相同的阶数。

4) 部分分式模型

传递函数也可以采用部分分式展开式或留数的形式表示。对于 SIMO 系统表示为

$$H(s) = \frac{r(1)}{s-p(1)} + \frac{r(2)}{s-p(2)} + \dots + \frac{r(n)}{s-p(n)} + k(s)$$

行向量 P 表示极点，行向量 r 表示对应于 P 中极点的留数，而且行向量 k 包含原传递函数中

分子阶次大于分母阶次的部分。

传递函数形式与部分分式形式可以互相转换。

4) 离散时间系统模型

在 MATLAB 中，离散时间线性时不变系统与连续时间线性时不变系统有相同的表示方法，即状态空间法，多项式传递函数法，零-极点增益法与部分分式法。

离散时间 LTI 系统模型可以表示为一组一阶差分方程组形式。在状态空间或矩阵表示形式中可以写成如下形式

$$\begin{aligned}x(n+1) &= Ax(n) + Bu(n) \\ y(n) &= Cx(n) + Du(n)\end{aligned}$$

其中： u 是控制输入向量， x 是状态向量， y 是输出向量。另

一种等效表示是 Z 变换传递函数描述法，其形式如下

$$Y(z) = H(z) U(z) \quad H(z) = C(zI - A)^{-1}B + D$$

在单输入为 u 的 SIMO 系统中，Z 变换传递函数描述形式为

$$H(z) = \frac{N(z)}{q(z)} = \frac{N(1) + N(2)z^{-1} + \dots + N(n)z^{-(n-1)}}{q(1) + q(2)z^{-1} + \dots + q(n)z^{-(n-1)}}$$

其中：向量 q 中包含按单位延迟 $1/z$ 升幂排列的分母系数，矩阵 N 包含分子的系数，其行数与 y 输出的数目相等。

零一极点增益表示形式为

$$H(z) = \frac{Z(z)}{P(z)} = K \frac{(z^{-1} + Z(1))(z^{-1} + Z(2))\dots(z^{-1} + Z(n))}{(z^{-1} + P(1))(z^{-1} + P(2))\dots(z^{-1} + P(n))}$$

3.1.1 模型转换

下面一组函数允许 LTI 系统模型的不同表示形式之间可以互相转换。

- (1) 状态空间模型到传递函数模型的转换。

$$[\text{num}, \text{den}] = \text{ss2tf}(a, b, c, d, u)$$

- (2) 状态空间模型到零-极点增益模型的转换。

$$[z, p, k] = \text{sstzp}(a, b, c, d, u) \quad (3)$$

传递函数模型到状态空间模型的转换。

$$[a, b, c, d] = \text{tf2ss}(\text{num}, \text{den})$$

- (4) 传递函数模型到零-极点增益模型的转换。

$$[z, p, k] = \text{tf2zp}(\text{num}, \text{den})$$

- (5) 零一极点增益模型到状态空间模型的转换。

$$[a, b, c, d] = \text{zp2ss}(z, p, k)$$

- (6) 零-极点增益模型到传递函数模型的转换。

$$[\text{num}, \text{den}] = \text{zp2tf}(z, p, k)$$

(7) 传递函数模型到部分分式模型的转换。

$$[z, p, k] = \text{residue}(\text{num}, \text{den})$$

(8) 部分分式模型到传递函数模型的转换。

$$[\text{num}, \text{den}] = \text{residue}(z, p, k)$$

(9) 连续系统模型到离散系统模型的转换。

$$[\text{ad}, \text{bd}] = \text{c2d}(a, b, T_s)$$

(10) 离散系统模型到连续系统模型的转换。

$$[a, b] = \text{d2c}(\text{ab}, \text{bd}, T_s)$$

3.1.2 分析函数

控制系统软件包提供了控制系统工程需要的基本的时域与频域分析工具函数。

连续时间系统分析函数

impulse	脉冲响应
step	阶跃响应
lsim	任意输入的仿真
bode	波特图
nyquist	奈奎斯特图
lyap	李雅普诺夫方程
gram	可控性与可观性

离散时间系统分析函数

dimpulse	单位采样响应
dstep	阶跃响应
filter	SISO 系统 z 变换仿真
dbode	离散波特图
freqz	SISO 系统 Z 变换频域响应
dlyap	李雅普诺夫方程
dgram	离散可控性与可观性

3.1.3 闭环系统建模

上节给出的函数为连续系统和离散系统提供了频域和时域分析工具，适合于 4 参数 (A,B,C,D)系统、3 参数 (z, p, k) 系统和 2 参数 (N, q)。但是还没有用于分析闭环系统的专门命令工具。闭环系统必须满足完全的闭环系统动力学。

例 设状态空间描述的开环系统模型如下:

$$\dot{x} = Ax +$$

$$Bu \quad y = Cx +$$

$$Du$$

并具有参考输入 r 的全状态反馈控制准则

$$u = -Kx + Nr$$

为了给该闭环系统建模，求解该闭环系统矩阵

$$\dot{x} = Ax + Bu = Ax - BKx + BNr = (A - BK)x + BNr$$

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/936053212121010145>