

```
1 sys
2 zero
3 pole, real (pole) , imag (pole)
4 roots
5 pzmap
6 zpk
7 tf2pz, pz2tf
8 d2c, c2d
9 series
10 parallel
11 feedback
12 cloop
13 residue
14 minreal零极点对消
15 impulse
16 lsim任意输入下系统的输出响应
17 ctrb
18 Obsv
19 acker
20 rlocus
21 systool
22 计算矩阵指数函数expm
```

# 一、传递函数

## 1、定义

### 传递函数

$$F(s) = \frac{4s^2 - 12s + 8}{s^3 + 6s^2 + 11s + 6}$$

```
1 num和den分别F(s)的分子和分母
2 num=[4 -12 8];
3 den=[1 6 11 6];
```

```

4 sys= tf (num, den)
5 sys.den{1}取分母系数向量
6 z=zero(sys)零点
7 p=pole(sys)极点
8 pzmap(sys)绘制复平面上的零-极点分布图
9 小技巧：还能在图像中直接看出超调、阻尼、频率

```

## 零-极-增益模型

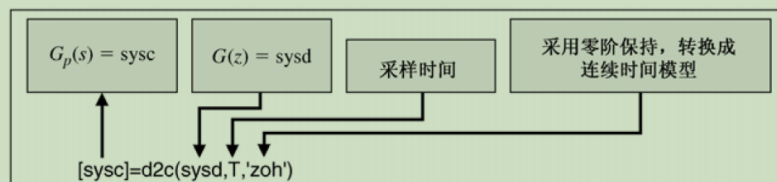
$$F(s) = \frac{4(s-1)(s-2)}{(s+1)(s+2)(s+3)}$$

```

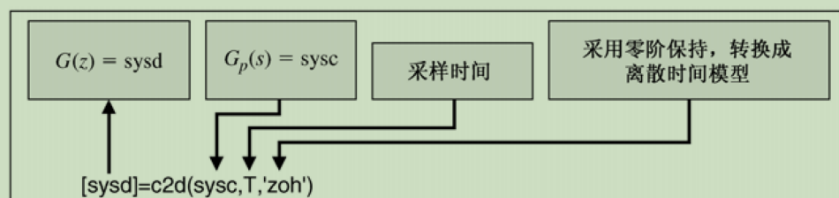
1 z=[1 2];
2 p=[-1 -2 -3];
3 k=4;
4 sys=zpk(z,p,k)
5 转换
6 [z, p, k] = tf2zp (num, den)
7 [num, den] = zp2tf (z, p, k)

```

## 2. 可以利用函数d2c将离散时间模型转换成连续时间模型



## 3. 可以利用函数c2d将连续时间模型转换成离散时间模型



## 2、运算

### 合并传递函数

```
1 串联: sys = series (sys1, sys2)
2 并联: sys = parallel (sys1, sys2)
3 反馈: sys = feedback (sys1, sys2, sign)
4      +1表示正反馈
5      -1表示负反馈（默认值）
6 错！实际负反馈sign=1，醉了不知道为啥
7
8 可以直接生成串并联后传递函数的系数
9 [numc,denc]=series(num1,den1,num2,den2)
10 sys=tf (numc, denc)
11 常数可直接当作sys用，以下等价
12 sys=series(4,sys1)
13 sys=series(tf(4,1),sys1)
```

**单位反馈:**  $\text{sys} = \text{feedback}(\text{sys1}, [1], \text{sign})$   
 $\text{sys} = \text{cloop}(\text{sys1}, -1)$



### 部分分式展开

- 分类讨论，是否有重极点
- r、p分别为展开后的留数及极点构成的列向量
- k为余项多项式行向量

**例** 求  $F(s) = \frac{s^5 + 10s^2 + 5s + 6}{s^4 + 5s^3 + 9s^2 + 7s + 2}$  的部分分式展开

```
num=[1 0 0 10 5 6];  
den=[1 5 9 7 2];  
[r, p, k]=residue(num, den)
```

r =  
-4.0000  
20.0000  
-20.0000  
10.0000

p =  
-2.0000  
-1.0000  
-1.0000  
-1.0000

k =  
1 -5

展开式为:  $F(s) = \frac{-4}{s+2} + \frac{20}{s+1} + \frac{-20}{(s+1)^2} + \frac{10}{(s+1)^3} + s - 5$

## 部分分式合并

**例** 求  $F(s) = \frac{1}{s+1} + \frac{2}{s+2} + \frac{3}{s+3} + \frac{4}{s+4}$  的合并式。

```
r = [1 2 3 4]'; p = [-1 -2 -3 -4]'; k = 0;  
[num, den] = residue(r, p, k)
```

```
num = 10    70    150    96  
den = 1     10     35     50     24
```

合并式为:  $F(s) = \frac{10s^3 + 70s^2 + 150s + 96}{s^4 + 10s^3 + 35s^2 + 50s + 24}$

## 零极点对消

### 约分

**例**

$\text{sys} = \frac{s^5 + 4s^4 + 6s^3 + 6s^2 + 5s + 2}{12s^6 + 205s^5 + 1066s^4 + 2517s^3 + 3128s^2 + 2196s + 712}$

```
>>num=[1 4 6 6 5 2]; den=[12 205 1066 2517 3128 2196 712];  
>>sys1=tf(num,den);  
>>sys=minreal(sys1);
```

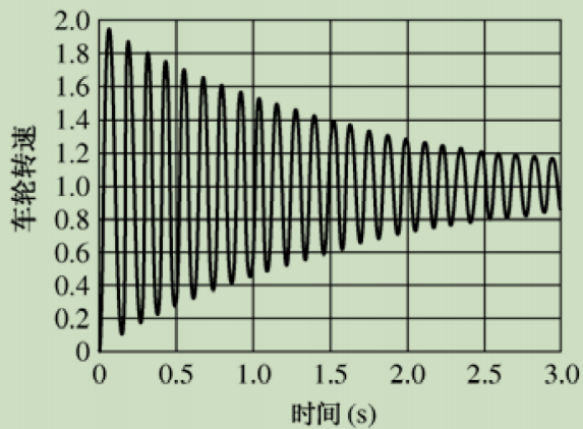
消除公因式

Transfer function:

$$\frac{0.08333 s^4 + 0.25 s^3 + 0.25 s^2 + 0.25 s + 0.1667}{s^4 + 16.08 s^3 + 72.75 s^2 + 137 s + 123.7 s + 59.33}$$

## 3、画图

### 计算系统的单位阶跃响应



(a) 牵引电机中车轮转速的阶跃响应

```
% This script computes the step
% response of the traction motor
% wheel velocity
%
num=[5400]; den=[2 2.5 5402]; sys=tf(num,den);
t=[0:0.005:3];
[y,t]=step(sys,t);
plot(t,y),grid
xlabel('Time (s)')
ylabel('Wheel velocity')
```

(b) m 脚本程序

- 幅值：最大值

```
1 创建阶跃
2 x=[-2:0.01:2]
3 t = linspace(-2,2,200)
4 y=heaviside(x);
5 y=stepfun(t,3) 发生阶跃的点: 3
6 plot(x,y)
```

## 画图只需取消赋值

```
1 单位阶跃
2 y=step (sys, t) 或 [y, t]=step (sys)
3 y=step (num, den, t) 或 [y, t]=step (num, den)
4 单位脉冲
5 y=impulse (sys, t) 或 [y, t]= impulse (sys)
6 指定输入
7 y=lsim (sys, u, t) 或[y, t]= lsim (sys, u)
8 %u为给定输入构成的列向量，元素个数和t一致
```

## 单位斜坡

在MATLAB中没有斜坡响应命令，可利用阶跃响应命令求斜坡响应

例如，考虑下列闭环系统：
$$\frac{Y(s)}{R(s)} = \frac{50}{25s^2 + 2s + 1}$$

对于单位斜坡输入量  $R(s) = \frac{1}{s^2}$

则

$$Y(s) = \frac{50}{25s^2 + 2s + 1} \cdot \frac{1}{s^2} = \frac{50}{(25s^2 + 2s + 1)s} \cdot \frac{1}{s} = \frac{50}{25s^3 + 2s^2 + s} \cdot \frac{1}{s}$$

- 法二

```
1 u=t;%斜坡信号r(t)=t
2 lsim(sys_cl,u,t)
```

## 循环画图

**例** 对于二阶系统  $G(s) = \frac{1}{T^2s^2 + 2\zeta Ts + 1}$ ，分别就  $T=1$  和  $T=0.1$ ， $\zeta=0, 0.2, 0.5, 0.7, 1, 10$  时，画出系统的单位阶跃响应曲线。

```
for T=[1 0.1]
    figure;
    for Zeta=[0 0.2 0.5 0.7 1 10]
        num=1;
        den=[T*T, 2*Zeta*T, 1];
        step(num, den, T*30);
        hold on;
    end
    grid
end
```

- figure+hold on+end

为每个T新开一个窗口

## 4、劳斯判据

### 验证基于劳斯判定表的结果

```
1 pole(sys)极点
2 real(pole(sys))实部
3 imag(pole(sys))虚部
```

## 劳斯判据特征方程

```
1 q=[1 2 4 K(i)];
2 p=roots(q);
3 if max(real(p))>0, x(i)=K(i);
4 break;end;%最大实部为正，不满足，记录
```

- 斜坡信号 $r(t)=At$ ，信号斜率是A

## 根轨迹

$$q(s) = s^3 + 2s^2 + 4s + K = 0$$

```
1 K=[0:0.5:20];限制K范围
2 for i=1:length(K)
3     q=[1 2 4 K(i)];%K的第i维数据
4     p(:,i)=roots(q);
5 end
6 plot(real(p),imag(p),'x'),grid%复数实部和虚部
```

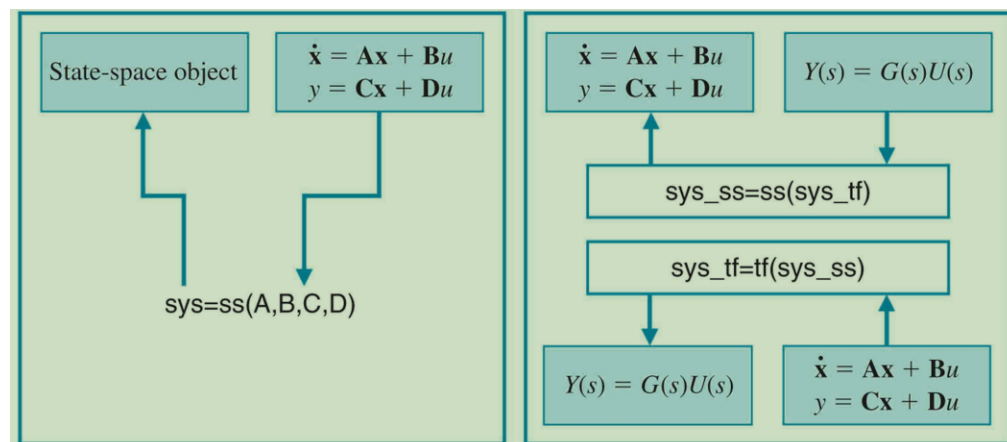
## 计算使稳定的参数范围

- 如果只给了开环传递函数

```
1 K=[0:0.5:20];限制K范围
2 for i=1:length(K)
3     sys=tf(1,[1 2 4 K(i)]);
4     sys=feedback(sys,1);
5     p=pole(sys);
6     realmax=max(real(p));
7     if realmax<0
8         K(i-1)输出临界且稳定的K值
9         break;
10    end
11 end
```

## 与状态空间互换

注意矩阵表示，列向量用；分隔



## 矩阵指数函数

**例** 求  $e^{0.2A}$

```
>>A=[0 -2; 1 -3]; dt=0.2; Phi=expm(A*dt)
```

Phi =

0.9671	-0.2968
0.1484	0.5219

State transition matrix for a  $\Delta t$  of 0.2 second

区别：函数 **exp(A)** 针对矩阵  $A$  中的每个元素  $a_{ij} \in A$  分别求解  $e^{a_{ij}}$

## 时间响应

- x0不需要加分号，竖向量也不用（不一定）
- 用sys\_ss画



## 利用函数 **lsim** 分别求解零输入和非零初始条件下的时间响应

$$\mathbf{A} = \begin{bmatrix} 0 & -2 \\ 1 & -3 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 2 \\ 0 \end{bmatrix},$$

$$\mathbf{C} = [1 \ 0], \text{ and } \mathbf{D} = 0.$$

初始条件为  $x_1(0) = x_2(0) = 1$

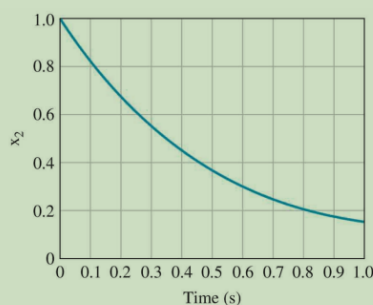
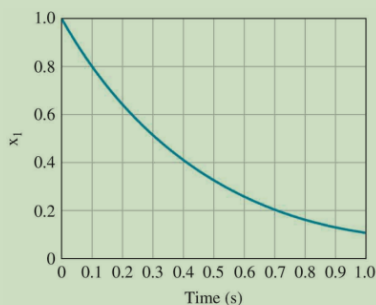
输入信号为  $u(t) = 0$

```
A=[0 -2;1 -3]; B=[2;0]; C=[1 0]; D=[0];
sys=ss(A,B,C,D);
x0=[1 1];
t=[0:0.01:1];
u=0*t;
[y,T,x]=lsim(sys,u,t,x0);
subplot(121), plot(T,x(:,1))
xlabel('Time (s)'), ylabel('x_1')
subplot(122), plot(T,x(:,2))
xlabel('Time (s)'), ylabel('x_2')
```

State-space model

Initial conditions

Zero input



## 能控能观性

判断系统是否能控和能观，并计算从输入  $u$  到输出  $y$  之间的传递函数。

```
a=[0 1;-6 -5]; b=[0;6]; c=[1 0]; d=[0];
sys_ss = ss(a,b,c,d);
Pc=ctrb(sys_ss);
dt_Pc=det(Pc);
Ob=obsv(sys_ss);
dt_Ob=det(Ob);
sys_tf=tf(sys_ss)
```

dt\_Pc = -36  
dt\_Ob = 1

Transfer function:  
6  
-----  
 $s^2 + 5s + 6$

## 计算增益矩阵

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t).$$

为该系统设计状态变量反馈控制器，其控制信号为  $u = -Kx$ 。试确定合适的增益矩阵  $K$ ，将闭环系统的极点配置为  $s_{1,2} = -1 \pm j$ 。

```
A=[0 1;0 0];
B=[0;1];
P=[-1+j; -1-j];
K=acker(A,B,P)
```

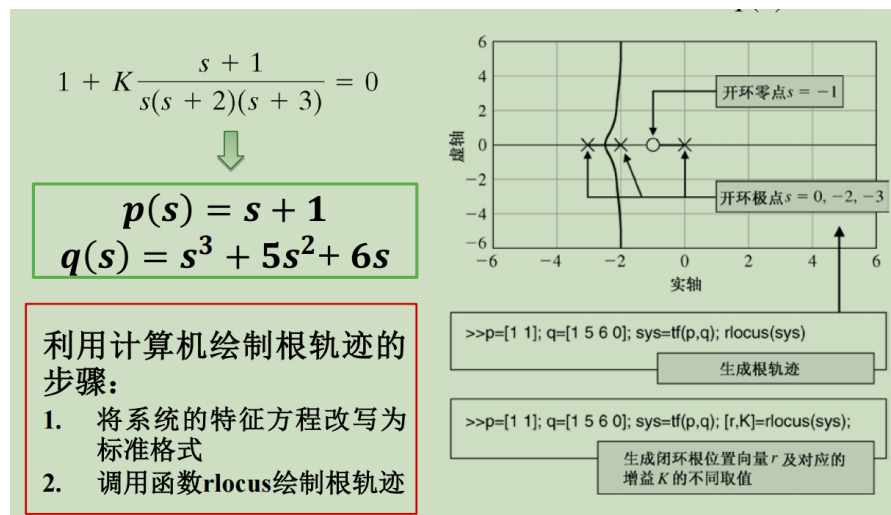
$K =$

2 2

The feedback gain matrix.

## 二、根轨迹

### 画根轨迹



#### • 特定点分析

```
1 rlocus(sys)
2 rlocfind(sys)
```

### sistool系统

```
1 sisotool(sys)
```