第三章 控制系统数字仿真

第一节 控制系统的数学模型

- 一.控制系统模型的表示形式
- 二.控制系统模型之间的转换
- 三.模型的变换和简化
- 四.模型特性

第三章 控制系统数字仿真

第一节 控制系统的数学模型

- 一.控制系统模型的表示形式
- 二.控制系统模型之间的转换
- 三.模型的变换和简化
- 四.模型特性

第三章 控制系统数字仿真

第一节 控制系统的数学模型

一. 控制系统模型的表示形式

1. 连续系统
2. 离散系统

- 微分方程
- 传递函数
- 零极点增益
- 结构图模型
- 状态方程

连续系统

微分方程

设系统的输入为$u(t)$，输出为$y(t)$，

$$a_0\frac{d^n y(t)}{dt^n}+a_1\frac{d^{n-1}y(t)}{dt^{n-1}}+\cdots+a_{n-1}\frac{dy(t)}{dt}+a_n y(t)=b_0\frac{d^m u(t)}{dt^m}+$$

$$b_1\frac{d^{m-1}u(t)}{dt^{m-1}}+\cdots+b_{m-1}\frac{du(t)}{dt}+b_m u(t) \qquad (\mathrm{m}\le \mathrm{n})$$

用微分方程的系数向量（n+1维/m+1维）表示：

输入系统向量　　num＝[b_0，b_1，……，b_m]

输出系统向量　　den＝[a_0，a_1，……，a_n]

连续系统

- 传递函数模型 $$G(s)=\frac{num(s)}{den(s)}=\frac{b_1s^m+b_2s^{m-1}+\cdots+b_ms+b_{m+1}}{a_1s^n+a_2s^{n-1}+\cdots+a_ns+a_{n+1}}$$

  用分子/分母的系数向量（n+1维/m+1维）表示:

  num＝[b_1，b_2，…… b_m，b_{m+1}]

  den＝[a_1，a_2，…… a_n，a_{n+1}]

- 零极点增益模型 $$G(s)=k\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

  用[z，p，k]向量组来表示，即

  z＝[z_1，z_2，……，z_m]

  p＝[p_1，p_2，……，p_n]

  k＝[k]

- 状态方程模型 $$\begin{cases}\dot{x}=ax+bu\\ y=cx+du\end{cases}$$

  系统可用(a，b，c，d)矩阵组表示

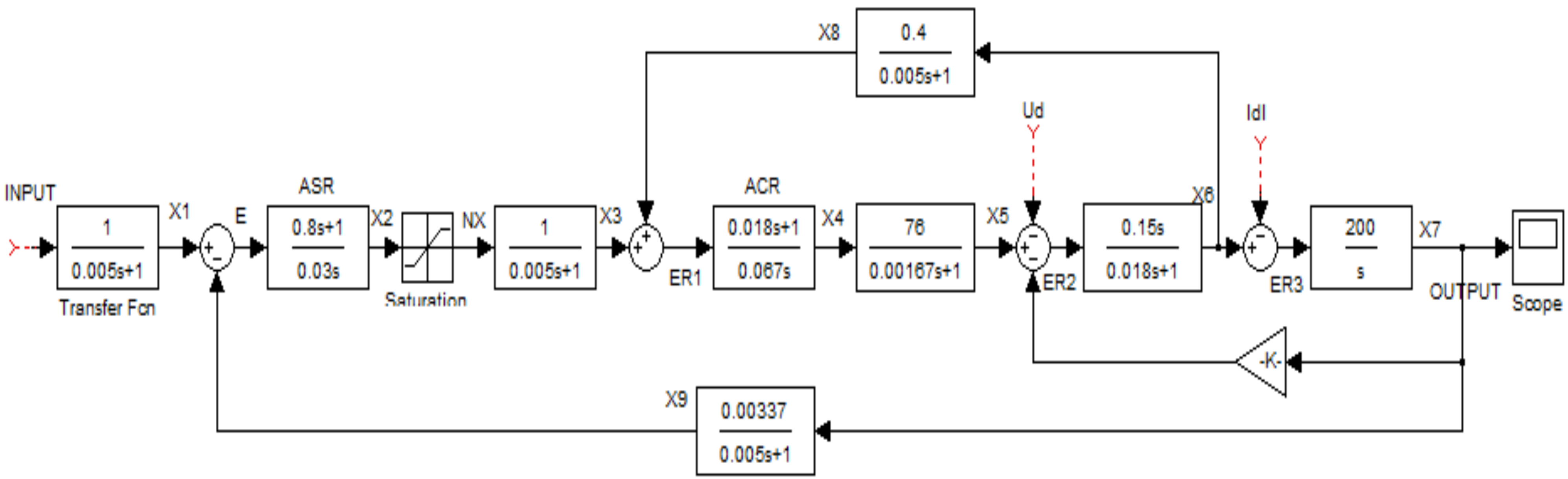
表达式不唯一：

控/观标准型，约当型

连续系统

◆结构图模型

系统中每个元件或环节的功能和信号流向的图解表示。



双闭环调速系统动态结构图

离散系统

◆差分方程　　设系统的采样周期为T，输出变量的初始条件为 $y(0), y(T), \ldots, y[(n-1)T]$

$$y[(k+n)T]+a_{n-1}yp(k+n-1)T]+\cdots+a_1y[(k+1)T]+a_0y[kT]=$$
$$b_mu[(k+m)T]+b_{m-1}u[(k+m-1)T]+\cdots+b_1u[(k+1)T]+b_0u[kT]$$

用输入/输出的系数向量表示：

num＝[b_m，b_{m-1}，……，b_0]

den ＝[1，a_{n-1}，a_{n-2}，……，a_0]

离散系统

- 传递函数模型 $G(z)=\dfrac{b_1z^m+b_2z^{m-1}+\cdots+b_mz+b_{m+1}}{a_1z^n+a_2z^{n-1}+\cdots+a_nz+a_{n+1}}$

  直接用分子/分母的系数表示，即

  num＝[b_1，b_2，……，b_{m+1}]

  den＝[a_1，a_2，……，a_{n+1}]

- 零极点增益模型 $G(z)=k\dfrac{(z-z_1)(z-z_2)\cdots(z-z_m)}{(z-p_1)(z-p_2)\cdots(z-p_n)}$

  用[z，p，k]向量组来表示，即

  z＝[z_1，z_2，……，z_m]

  p＝[p_1，p_2，……，p_n]

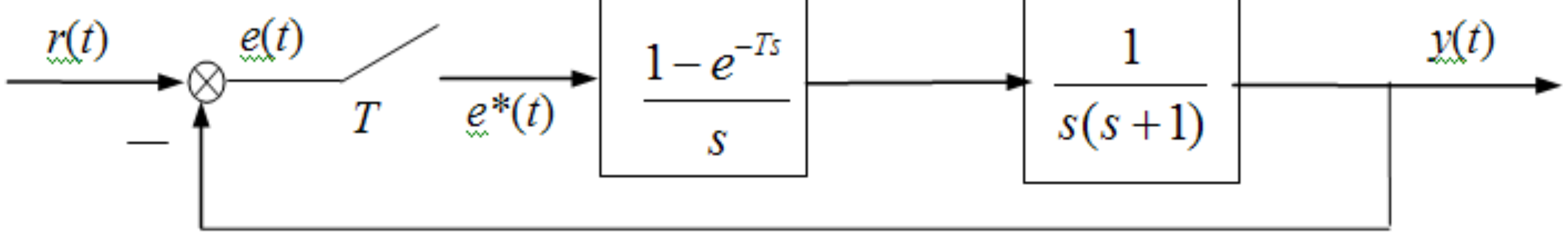
  k＝[k]

- 状态空间模型 $\begin{cases}\dot{x}(k+1)=ax(k)+bu(k)\\ y(k+1)=cx(k)+du(k)\end{cases}$

  系统可用(a，b，c，d)矩阵组表示

离散系统

◆结构图模型



线性时不变系统的对象数据类型描述

- 建立线性时不变（Linear Time Invariant）模型对象

G=tf(num,den)	利用传递函数二对组生成LTI对象模型
G=zpk(Z,P,K)	利用零极点增益三对组生成LTI对象模型
G=ss(A,B,C,D)	利用状态方程四对组生成LTI对象模型

LTI对象模型G一旦生成，就可以用单一变量名G描述系统的数学模型，而不必每次调用系统都输入模型参数组各向量或矩阵数据。

线性时不变系统的对象数据类型描述

- 线性时不变（Linear Time Invariant）模型对象转换

  G1=tf(G)　将LTI对象模型转换为传递函数模型

  G2=zpk(G)　将LTI对象模型转换为零极点增益模型

  G3=ss(G)　将LTI对象模型转换为状态方程模型

- 通过以下函数获得不同要求下的模型参数组向量或矩阵数据

[num,den]=tfdata(G)　从LTI对象获得传递函数二对组模型参数

[Z,P,K]=zpkdata(G)　从LTI对象获取零极点增益三对组模型参数

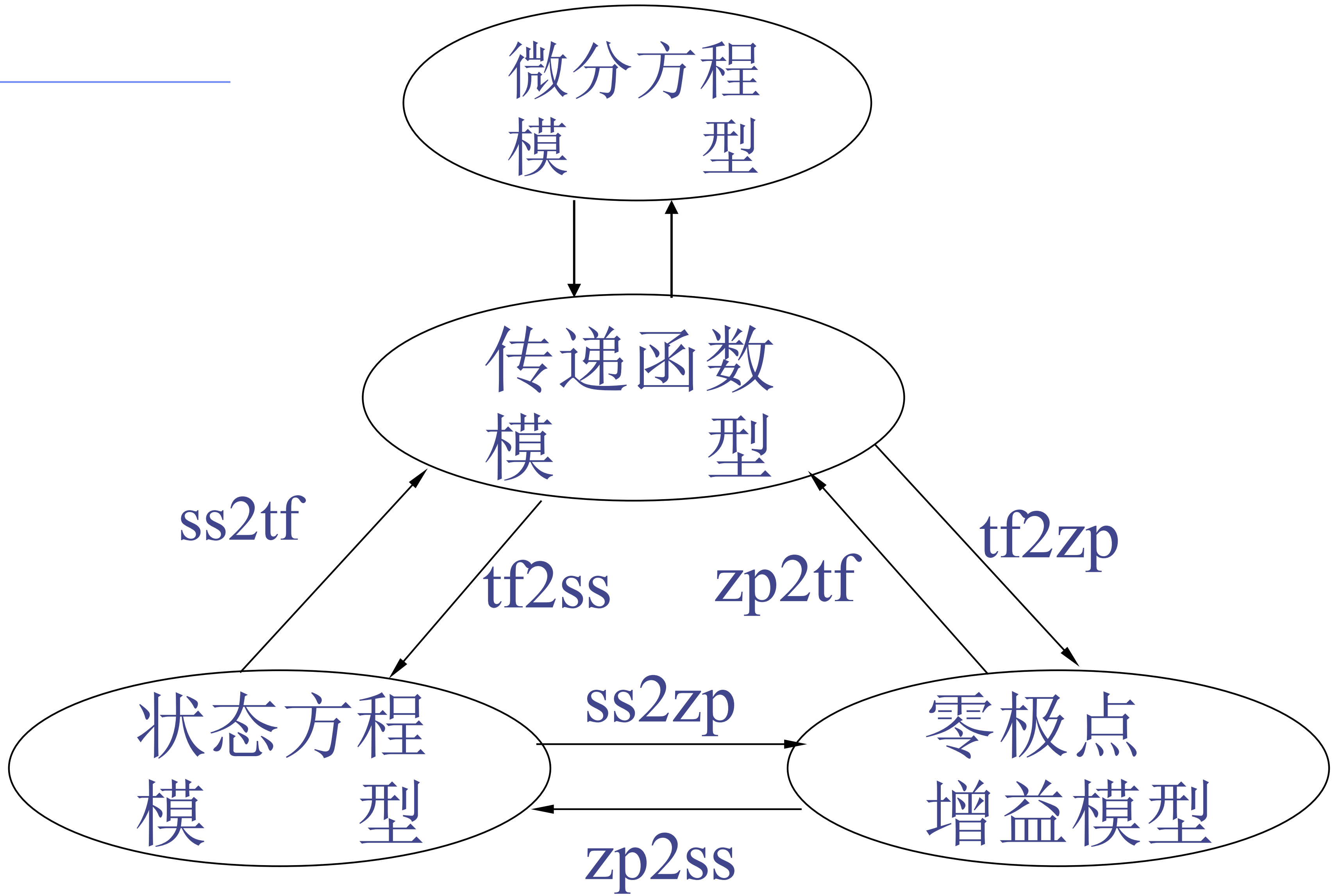
[A,B,C,D]=ssdata(G)　从LTI对象获取状态方程四对组模型参数

第三章 控制系统数字仿真

第一节 控制系统的数学模型

- 一.控制系统模型的表示形式
- 二.控制系统模型之间的转换
- 三.模型的变换和简化
- 四.模型特性

二．控制系统模型之间的转换



模型之间的转换

控制系统模型之间的转换

◆ss2tf

功能： 变系统状态空间形式为传递函数形式。

格式： [num，den]=ss2tf(A，B，C，D，iu)

说明：

- 可将状态空间表示变换成相应的传递函数表示，iu用于指定变换所使用的输入量。
- 生成所有输出对于输入的传递函数（分子系数被返回到num，其行数等于输出数目）。
- ss2tf函数还可以应用于离散时间系统，这时得到的是Z变换表示。

第三章 控制系统数字仿真

第一节 控制系统的数学模型

◆例题：现在讨论具有2个输入和1个输出的系统

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

对这个系统可以求得2个传递函数。其中一个传递函数描述输出y和输入u_1的关系，另一个传递函数描述输出y和输入u_2的关系（在考虑输入u_1时，假设输入u_2为零，反之亦然）。

第三章 控制系统数字仿真

第一节 控制系统的数学模型

◆例题：

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

$$y = [1 \quad 0]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0 \quad 0]\begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

```
A=[0 1;-2 -3];
B=[1 0 ;0 1];
C=[1 0];
D=[0 0];
[num,den1]=ss2tf(A,B,C,D,1)
[num,den2]=ss2tf(A,B,C,D,2)
```

$$\frac{Y(s)}{U_1(s)} = \frac{s+3}{s^2+3s+2}$$

$$\frac{Y(s)}{U_2(s)} = \frac{1}{s^2+3s+2}$$

第三章 控制系统数字仿真

第一节 控制系统的数学模型

◆例题： 下面考虑具有多个输入和多个输出的系统。由

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}0 & 1\\ -25 & -4\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix}$$

$$\begin{bmatrix}y_1\\ y_2\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix}$$

描述的系统。 这个系统包含2个输入和2个输出，求其包括的四个传递函数：$Y_1(s)/U_1(s)$, $Y_2(s)/U_1(s)$, $Y_1(s)/U_2(s)$, $Y_2(s)/U_2(s)$。

第三章 控制系统数字仿真

第一节 控制系统的数学模型

- 例题：

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}0 & 1\\ -25 & -4\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}1 & 1\\ 0 & 1\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix}$$

$$\begin{bmatrix}y_1\\ y_2\end{bmatrix}=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix}$$

```
A=[0 1;-25 -4];
B=[1 1 ;0 1];
C=[1 0;0 1];
D=[0 0;0 0];
[num,den]=ss2tf(A,B,C,D,1)
[num,den]=ss2tf(A,B,C,D,2)
```

$$\frac{Y_1(s)}{U_1(s)}=\frac{s+4}{s^2+4s+25},\quad \frac{Y_1(s)}{U_2(s)}=\frac{s+5}{s^2+4s+25}$$

$$\frac{Y_2(s)}{U_1(s)}=\frac{-25}{s^2+4s+25},\quad \frac{Y_2(s)}{U_2(s)}=\frac{s-25}{s^2+4s+25}$$

控制系统模型之间的转换

◆ss2zp

功能： 变系统状态空间形式为零极点增益形式。

格式：[z,p,k]=ss2zp（A，B，C，D，iu）

说明：

❖[z,p,k]=ss2zp（A，B，C，D，iu）可将状态空间表示转换成零极点增益表示，iu用于指定变换所用的输入量。

❖ss2zP函数还可以应用于离散时间系统，这时得到的是Z变换表示。

控制系统模型之间的转换

◆tf2ss

功能：变系统传递函数形式为状态空间形式。

格式：［A，B，C，D］＝ tf2ss（num，den）

说明：

❖tf2ss函数可将给定系统的传递函数表示成等效的状态空间表示。在[A，B，C，D]＝tf2ss(num,den)格式中，矢量den按s的降幂顺序输入分母系数，矩阵num每一行为相应于某输出的分子系数，其行数为输出的个数。tf2ss得到控制器正则形式的A,B,C,D矩阵。

❖ tf2ss也可以用于离散系统中，但这时必须在分子多项式中补零使分子分母的长度相同。

控制系统模型之间的转换

◆tf2zp

功能：变系统传递函数形式为零极点增益形式。
格式：[z，P，k]=tf2zp(num，den)
说明：

❖tf2zp函数可找出多项式传递函数形式的系统的零点、极点和增益。

❖tf2zP函数类似于ss2zP函数。

控制系统模型之间的转换

zp2ss

功能：变系统零极点增益形式为状态空间形式。

格式：［A，B，C，D］＝zp2ss（z，p，k）

说明：

- ［A，B，C，D］＝zp2ss（z，P，k）可将以z，P，k表示的零极点增益形式变换成状态空间形式。

控制系统模型之间的转换

◆zp2tf

功能：变系统零极点增益形式为传递函数形式。

格式：[num，den]=zp2tf（z，p，k）

说明：

➢[num，den]= zp2tf(z，P)可将以z，p，k表示的零极点增益形式变换成传递函数形式。

控制系统模型之间的转换

◆ss2ss

$$\begin{cases}\dot{x} = Ax + Bu \\ y = Cx + Du\end{cases}$$

功能：相似变换。

格式：

［at，bt，ct，dt］＝ss2ss（a，b，c，d，T）

说明：

［at，bt，ct，dt］＝ss2ss（a，b，c，d，T）可完成相似变换z=Tx以此得到状态空间系统为

$$\begin{cases}\dot{z} = TaT^{-1}z + Tbu \\ y = cT^{-1}z + du\end{cases}$$

控制系统模型之间的转换

◆c2d，c2dt

功能：变连续时间系统为离散时间系统。

格式：

[ad,bd]＝c2d(a,b,Ts)

[ad,bd,cd,dd]＝c2dt(a,b,c,Ts,lambda)

说明：c2d和c2dt完成将状态空间模型从连续时间到离散时间的转换

控制系统模型之间的转换

◆residue

功能：求两个多项式之比的部分分式展开式中的留数极点和直接项

格式：

[r,p,k]＝residue(num,den)

[num,den]＝residue(r,p,k)

说明：

$$\frac{B(s)}{A(s)} = \frac{num}{den} = \frac{b_1 s^m + b_2 s^{m-1} + \cdots + b_m s + b_{m+1}}{a_1 s^n + a_2 s^{n-1} + \cdots + a_n s + a_{n+1}}$$

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

第三章 控制系统数字仿真

◆例题：

例 3.1.2.1 设系统的零极点增益模型为

$$G(s)=\frac{6(s+3)}{(s+1)(s+2)(s+5)}$$

求系统的传递函数及状态空间模型。

第三章 控制系统数字仿真

◆例题：

例 3.1.2.2 给定离散系统状态空间方程

$$\begin{cases} \mathrm{x}(\mathrm{k}+1) = \begin{bmatrix} -2.8 & -1.4 & 0 & 0 \\ 1.4 & 0 & 0 & 0 \\ -1.8 & -0.3 & -1.4 & -0.6 \\ 0 & 0 & 0.6 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} u(k) \\ y(k) = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} x(k) \end{cases}$$

求传递函数模型和零极点模型，并判断其稳定性。

第三章 控制系统数字仿真

◆例题：

例 3.1.2.3 对例 3.1.2.1 中的连续模型，采用 MATLAB 提供的各种函数进行离散化处理，从而得稍有差异的状态方程。

第三章 控制系统数字仿真

◆例题：

例 3.1.2.5 求系统 $G(s)=\dfrac{s^2-0.5s+2}{s^2+0.4s+1}$ 的零点、极点和增益。

第三章 控制系统数字仿真

第一节 控制系统的数学模型

- 一.控制系统模型的表示形式
- 二.控制系统模型之间的转换
- 三.模型的变换和简化
- 四.模型特性

三．模型的建立和简化

◆augstate

功能： 将状态增广到状态空间系统的输出中。

格式： [ab,bb,cb,db]=augstate(a,b,c,d)

说明： 将状态加到状态空间系统的输出中。

$$\begin{cases} \dot{x} = ax + bu \\ y = cx + du \end{cases} \Rightarrow \begin{aligned} &\dot{x} = ax + bu \\ &\begin{bmatrix} y \\ x \end{bmatrix} = \begin{bmatrix} c \\ I \end{bmatrix} x + \begin{bmatrix} d \\ 0 \end{bmatrix} u \end{aligned}$$

这个命令是为feedback函数作准备，这样构成的系统可采用feedback命令构成全状态反馈的闭环系统。

三．模型的建立和简化

◆append

功能：两个状态空间系统的组合。

格式：

[a,b,c,d]=append(a1,b1,c1,d1,a2,b2,c2,d2)

说明：append函数可将两个状态空间系统组合。

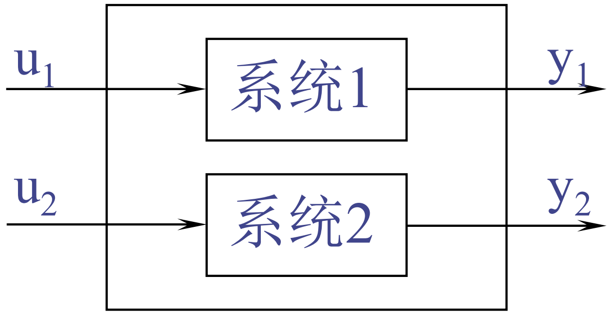


图3.1 两系统的组合

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}a_1 & 0\\ 0 & a_2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}b_1 & 0\\ 0 & b_2\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix}$$

$$\begin{bmatrix}y_1\\ y_2\end{bmatrix}=\begin{bmatrix}c_1 & 0\\ 0 & c_2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}d_1 & 0\\ 0 & d_2\end{bmatrix}\begin{bmatrix}u_1\\ u_2\end{bmatrix}$$

三．模型的建立和简化

◆parallel

功能： 系统的并联连接。

格式：

［a,b,c,d］＝parallel（a1,b1,c1,d1,a2,b2,c2,d2）
［a,b,c,d］＝parallel（a1,b1,c1,d1,a2,b2,c2,d2,inp1,inp2,out1,out2）
［num，den］＝parallel（numl，denl，num2，den2）

说明： parallel函数按并联方式连接两个状态空间系统，它即适合于连续时间系统也适合于离散时间系统。

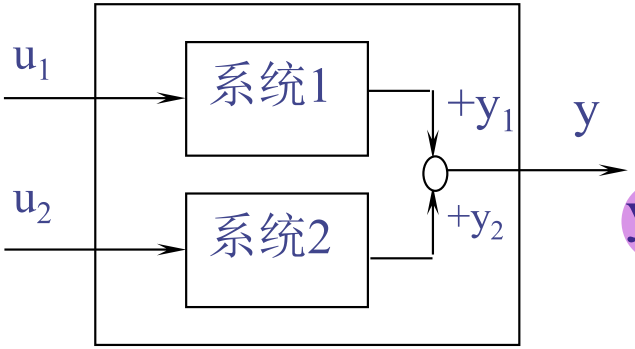


图3.2 系统的并联连接

$y=y_1+y_2$

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}a_1 & 0\\ 0 & a_2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}b_1 & 0\\ 0 & b_2\end{bmatrix}u$$

$$y=y_1+y_2=\begin{bmatrix}c_1 & c_2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix}+\begin{bmatrix}d_1 & d_2\end{bmatrix}u$$

三．模型的建立和简化

◆series

功能：系统的串联连接。

格式：

［a,b,c,d］＝series（a1,b1,c1,d1,a2,b2,c2,d2）

［a,b,c,d］＝series（a1,b1,c1,d1,a2,b2,c2,d2, outputs1, input2）

［num,den］＝series（numl,denl,num2,den2）

说明：series函数可以将两个系统按串联方式连接，它即适合于连续时间系统，也适合于离散时间系统。

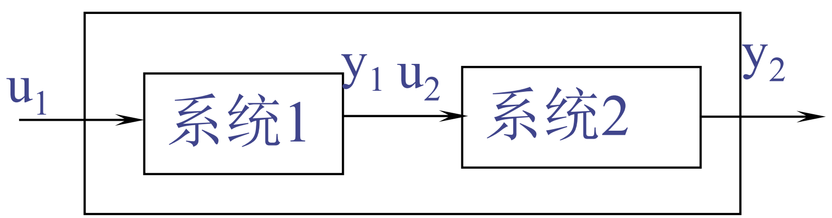


图3.4 系统的串联连接

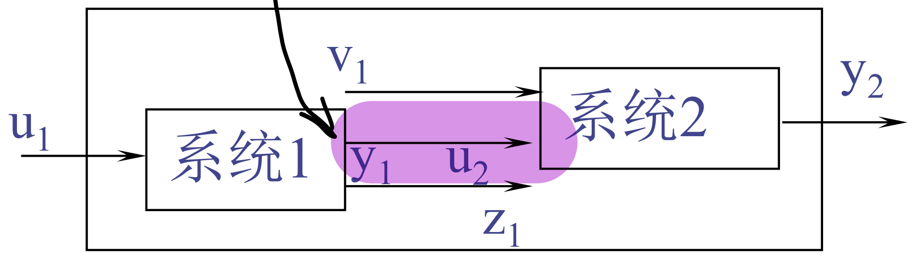


图3.5部分串联连接的系统

三. 模型的建立和简化

◆feedback

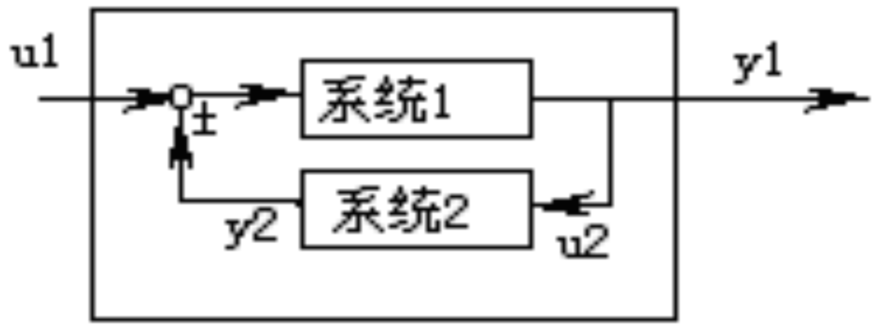


图3.7系统的反馈连接

功能：两个系统的反馈连接。

格式：

［a,b,c,d］＝feedback(a1,b1,c1,d1,a2,b2,c2,d2)

［a,b,c,d］＝feedback(a1,b1,c1,d1,a2,b2,c2,d2,sign)

［a,b,c,d］＝feedback(a1,b1,c1,d1,a2,b2,c2,d2, inp1,out1)

［num,den］＝ feedback(numl,denl,num2,den2)

［num,den］＝ feedback(numl,denl,num2,den2,sign)

说明：feedback可将两个系统接按反馈形式连接。

sign符号用于指示y_2到u_1连接的符号，缺省为负，即sign＝－1。

三．模型的建立和简化

◆etsim，destim

功能： 生成连续/离散状态估计器或观测器

格式：

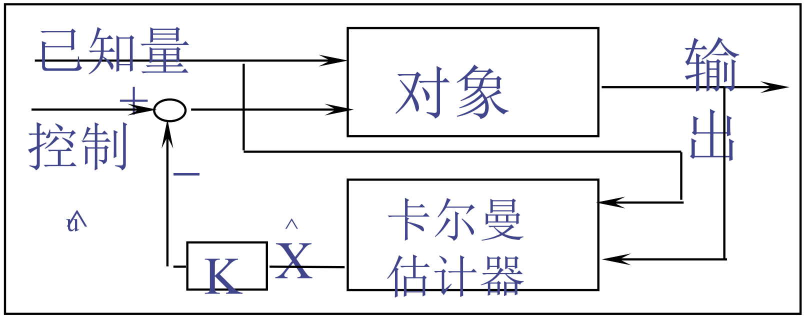
[ae,be,ce,de]=estim(a,b,c,d,l)

[ae,be,ce,de]=estim(a,b,c,d,l,sensors,known)

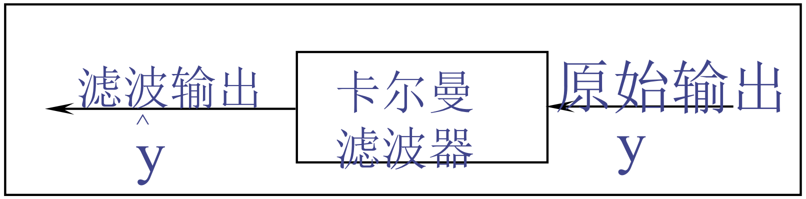
[ae,be,ce,de]=destim(a,b,c,d,l)

[ae,be,ce,de]=destim(a,b,c,d,l,sensors,known)

说明： estim和destim可从状态空间系统和增益矩阵*l*中生成稳态卡尔曼估计器。



(a) 估计器



(b) 滤波器

图3.10 卡尔曼滤波器

三．模型的建立和简化

◆reg，dreg

功能：生成控制器/估计器

格式：[ae,be,ce,de]=reg(a,b,c,d,k,l)

[ae,be,ce,de]=reg(a,b,c,d,k,l,sensors,known,controls)

[ae,be,ce,de]=dreg(a,b,c,d,l)

[ae,be,ce,de]=dreg(a,b,c,d,l,sensors,known,controls)

说明：reg和dreg可从状态空间系统、反馈增益矩阵k及估计器增益矩阵*l* 中形成控制器/估计器。

第三章 控制系统数字仿真

◆例题：

例 3.1.3.1 已知两个系统为

$$G_1(s)=\frac{5}{s+1};\quad G_2(s)=\frac{5s+2}{s^2+3s+10}$$，求将两者并联连接所得系统。

例 3.1.3.2 已知两个系统为

$$G(s)=\frac{s^2+5s+2}{s^2+3s+1};\quad H(s)=\frac{5s+3}{s+5}$$，求将两者 反馈连接所得系统。

第三章 控制系统数字仿真

◆例题：

例 3.1.3.3 已知两个系统

$$\begin{cases} \dot{x}_1 = \begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix} x_1 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_1 \\ y_1 = \begin{bmatrix} 1 & 3 \end{bmatrix} x_1 + u_1 \end{cases}$$

$$\begin{cases} \dot{x}_2 = \begin{bmatrix} 0 & 1 \\ -1 & -3 \end{bmatrix} x_2 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_2 \\ y_2 = \begin{bmatrix} 1 & 4 \end{bmatrix} x_2 \end{cases}$$

求按串联、并联、单位负反馈、单位正反馈连接时的系统状态方程。

第三章 控制系统数字仿真

第一节 控制系统的数学模型

- **例题：** 已知系统如图2.7所示，试用MATLAB求闭环传递函数$Y(s)/U(s)$。

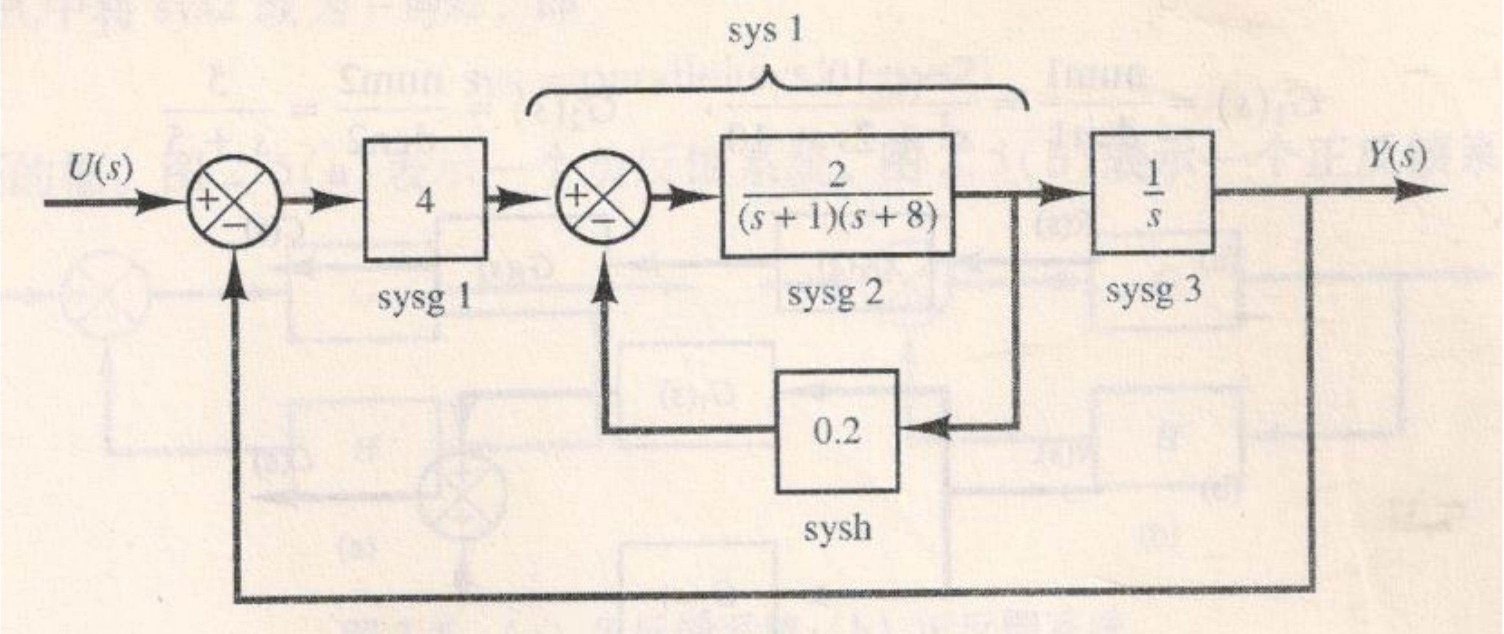


图 2.7 闭环系统

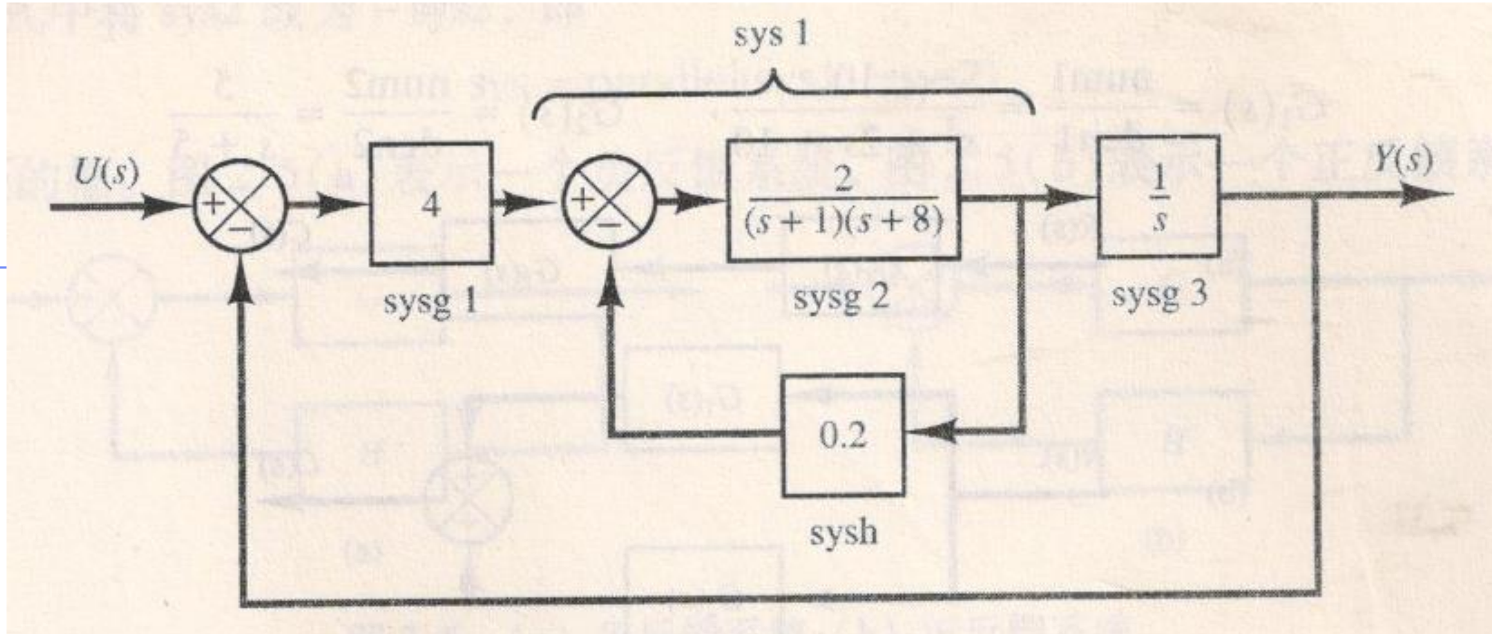


图 2.7　闭环系统

```
sysg1=[4];
numg2=[2];deng2=[1 9 8];sysg2=tf(numg2,deng2);
numg3=[1];deng3=[1 0];sysg3=tf(numg3,deng3);
sysh = [0.2];
sys1=feedback(sysg2,sysh);
sys2=series(sys1,sysg3);
sys3=series(sysg1,sys2);
sys =feedback(sys3,[1])
```

```
Transfer function:
              8
----------------------------
s^3 + 9 s^2 + 8.4 s + 8
```

第三章 控制系统数字仿真

第一节 控制系统的数学模型

- **例题：** 已知系统如图2.8所示。求闭环传递函数Y(s)/U(s),同时利用sys_ss=ss(sys)求该传递函数系统的状态空间表达式。

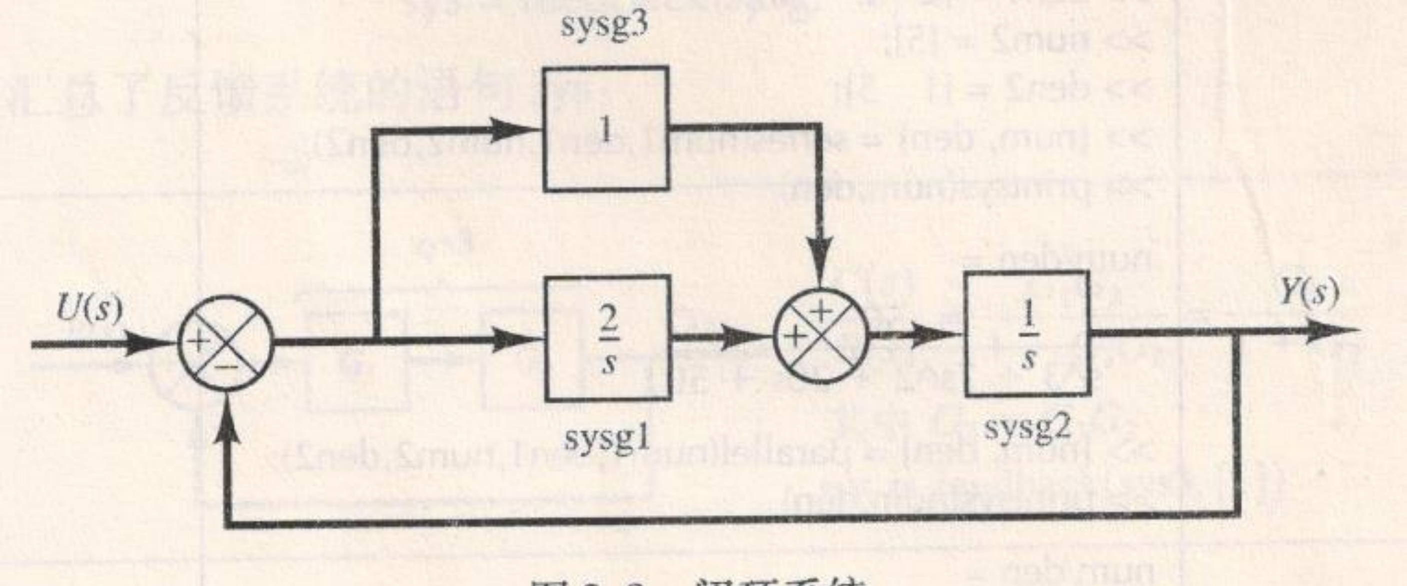


图 2.8　闭环系统

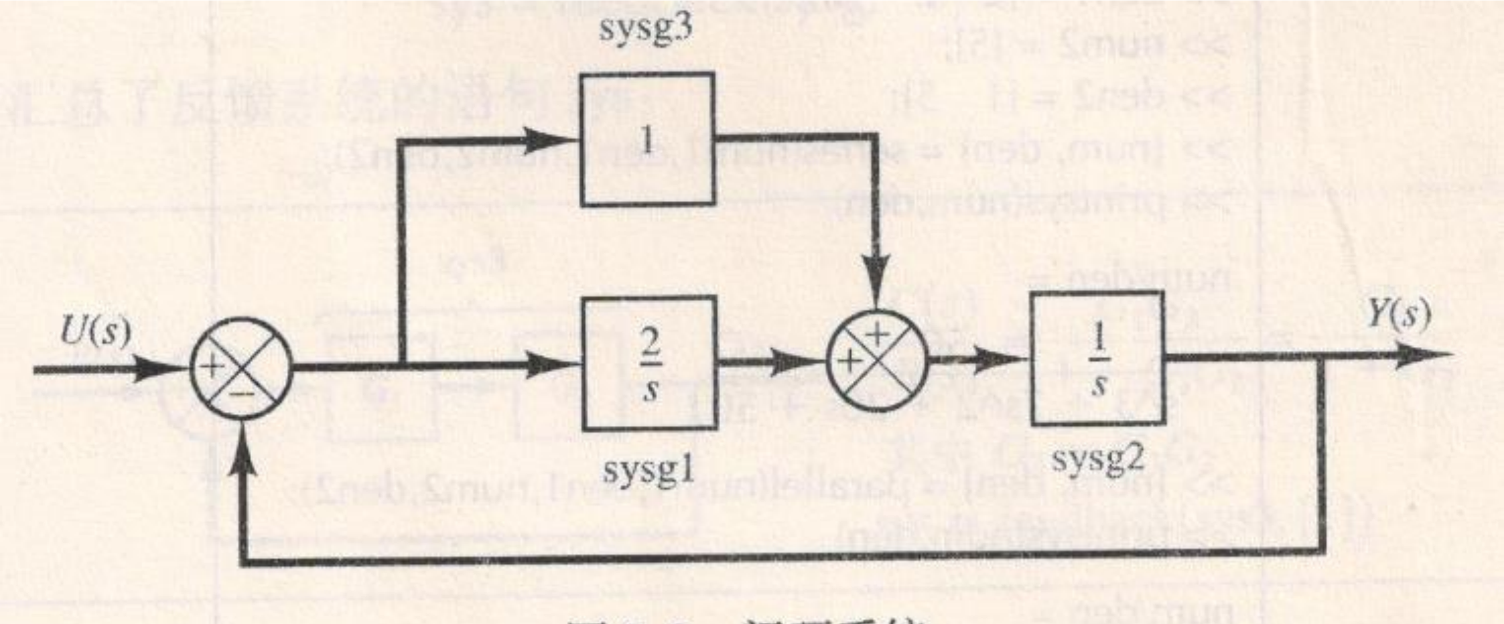


图 2.8　闭环系统

```
numg1=[2];deng1=[1 0];sysg1=tf(numg1,deng1);
numg2=[1];deng2=[1 0];sysg2=tf(numg2,deng2);
sysg3=[1];
sys1 = parallel(sysg1,sysg3);
sys2 = series(sys1,sysg2);
sys = feedback(sys2,[1]);
sys_ss=ss(sys)
```

```
Transfer function:
   s + 2
-----------
s^2 + s + 2
```

第三章 控制系统数字仿真

第一节 控制系统的数学模型

- 一.控制系统模型的表示形式
- 二.控制系统模型之间的转换
- 三.模型的变换和简化
- 四.模型特性

四．模型特性

◆ctrbf，obsvf

功能： 可控性和可观性阶梯形式。

格式：

［ab，bb，cb，T，k］＝ctrbf（a，b，c）

［ah，bb，cb，T，k］＝ctrbf（a，b，c，tol）

［ab，bb，cb，T，k］＝obsvf（a，b，c）

［ab，bb，cb，T，k］＝obsvf（a，b，c，tol）

说明：

❖函数［ab,bb,cb,T,k］＝ctrbf（a,b,c）可将系统分解为可控/不可控两部分。

❖函数［ab, bb, cb, T, k］＝obsvf（a, b, c）可将系统分解为可观/不可观两部分。

四．模型特性

◆ctrb，obsv

功能： 可控性和可观性矩阵。

格式： co＝ctrb(a,b)　　　　　ob＝obsv(a,c)

说明：

❖**ctrb**和obsv函数可求出状态空间系统的可控性和可观性矩阵。

❖对n×n矩阵a，n×m矩阵b和p×n矩阵c，ctrb(a,b)可得到n×nm的可控性矩阵co＝$[b\ ab\ a^2b\ a^3b......a^{n-1}b]$

❖obsv(a,b)可得到mn×n的可观性矩阵

$$ob=[c\ ca\ ca^2\ ......ca^{n-1}]'$$。

当co的秩为n时,系统可控；当ob的秩为n时,系统可观。

四．模型特性

◆gram，dgram

功能：求可控和可观的gram矩阵。

格式：gc=gram(a,b)

go=gram(a’,c’)

gc=dgram(a,b)

go=dgram(a’,c’)

说明：gram函数可计算出可控与可观的gram矩阵, gram阵可用于研究状态空间系统的可控性与可观性，它们比由ctrb与obsv形成的可控性与可观性有更好的特性。

四．模型特性

◆dcgain，ddcgain

功能：计算系统的稳态（DC）增益。

格式：

k＝ dcgain（a ，b，c，d）

k＝dcgain（num，den）

k＝ddcgain（a，b，c，d）

k＝ ddcgain（num，den）

说明：dcgain函数可计算出一个系统的稳态（DC或低频）增益。

第三章 控制系统数字仿真

◆例题：

例 3.1.5.1 时不变线性系统：

$$\begin{cases} \dot{x} = \begin{bmatrix} -3 & 1 \\ 1 & -3 \end{bmatrix} x + \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} x \end{cases}$$

判别系统的能控性和能观性。

rank( ) 求秩

第三章 控制系统数字仿真

◆例题：

例 3.1.5.2 线性系统

$$H(s)=\frac{s+a}{s^3+10s^2+27s+18}$$

当 a 分别取-1，0，+1 时，判别系统的可控性和可观性，并求出相应的状态方程。

第三章 控制系统数字仿真

第一节 控制系统的数学模型

- 例题：零极点对消指令

  minreal

分子分母中存在公因子，可以利用这个函数实现极点-零点对消，并产生最小阶传递函数。

$$\frac{Y(s)}{U(s)} = \frac{(s+1)(s+2)}{(s+3)(s+4)(s+1)} = \frac{s^2+3s+2}{s^3+8s^2+19s+12}$$

$$\Rightarrow \quad \frac{Y(s)}{U(s)} = \frac{s+2}{s^2+7s+12}$$

```
>> num=[1 3 2];den=[1 8 19 12];sys=tf(num,den);
>> sys

Transfer function:
     s^2 + 3 s + 2
-----------------------
s^3 + 8 s^2 + 19 s + 12

>> sys_min=minreal(sys)

Transfer function:
    s + 2
--------------
s^2 + 7 s + 12
```

第三章 控制系统数字仿真

第二节 控制系统分析与设计

- 一．时域分析
- 二．频域分析
- 三．根轨迹分析

第三章 控制系统数字仿真

第二节 控制系统分析与设计

一．时域分析

- step

功能：求连续系统的单位阶跃响应。

格式：

[y,x,t]＝step(a，b，c，d，iu)

[y,x,t]＝step(a，b，c，d，iu，t)

[y,x,t]＝step(num，den)

[y,x,t]＝step(num，den，t)

- step函数可计算出线性系统的单位阶跃响应，当不带输出变量引用时，step函数可在当前图形窗口中绘出系统的阶跃响应曲线。
- step(a，b，c，d) 可得到一组阶跃响应曲线，每条曲线对应于连续LTI系统

$$\dot{x} = ax + bu$$

$$y = cx + du$$

的输入／输出组合，其时间矢量由函数自动设定。
- step(a，b，c，d，iu) 可绘制出从第iu个输入到所有输出的单位阶跃响应曲线。
- step(a,b,c,d,iu,t)或step(num,den,t) 可利用用户指定的时间矢量t来绘制单位阶跃响应。t为显示从时间0开始到时间t之间的单位阶跃响应。
- [y,t,x] = step(SYS), [y,t] = step(SYS) 返回输出响应y与时间t的向量，x为状态轨迹矩阵

dstep

功能： 求离散系统的单位阶跃响应。

格式： [y，x]＝dstep（a，b，c，d）

[y，x]＝dstep（a，b，c，d，iu）

[y，x]＝dstep（a，b，c，d，iu，n）

[y，x]＝dstep（num，den）

[y，x]＝dstep（num，den，n）

说明： dstep函数可计算出离散时间线性系统的单位阶跃响应，当不带输出变量引用时，dsetp可在当前图形窗口中绘出系统的阶跃响应曲线。

◆impulse

功能： 求连续系统的单位冲激响应

格式：

［y,x,t］＝impulse（a，b，c，d）

［y,x,t］＝impulse（a，b，c，d，iu）

［y,x,t］＝impulse（a，b，c，d，iu，t）

［y,x,t］＝impulse（num，den）

［y,x,t］＝impulse（num，den，t）

说明： impulse函数用于计算线性系统的单位冲激响应，当不带输出变量时，impulse可在当前图形窗口中直接绘出系统的单位冲激响应。

dimpulse

功能： 求离散系统的单位冲激响应

格式： [y,x,t] = dimpulse（a，b，c，d）

[y,x,t] = dimpulse（a，b，c，d，iu）

[y,x,t] = dimpulse（a，b，c，d，iu，n）

[y,x,t] = dimpulse（num，den）

[y,x,t] = dimpulse（num，den，n）

说明： dimpulse函数用于计算离散时间线性系统的单位冲激响应，当不带输出变量时，dimpulse可在当前图形窗口中直接绘出系统的单位冲激响应。

◆initial

功能： 求连续系统的零输入响应。

格式：

[y，x，t] = initial（a，b，c，d，x0）

[y，x，t] =initial（a，b，c，d，xo，t）

说明： initial函数可计算出连续时间线性系统由于初始状态所引起的响应（故而称零输入响应）。当不带输出变量引用函数时，initial函数在当前图形窗口中直接绘出系统的零输入响应。

◆dinitial

功能：求离散系统的零输入响应。

格式：

［y，x］＝（a，b，c，d，x0）

［y，x］＝dinitial（a，b，c，d，x0，n）

说明：dinitial函数可计算出离散时间线性系统由于初始状态所引起的响应（故而称零输入响应）。当不带输出变量引用函数时，dinitial可在当前图形窗口中直接绘制出系统的零输入响应。

◆lsim，dlsim

功能：对任意输入的连续/离散系统进行仿真。

格式： [y，x] =lsim（a，b，c，d，u，t）

[y，x] =lsim（a，b，c，d，u，t，x0）

[y，x] =lsim（num，den，u，t）

[y，x] =dlsim（a，b，c，d，u）

[y，x] =dlsim（a，b，c，d，u，x 0）

[y，x] =dlsim（num，den，u）

说明：lsim /dlsim函数可对任意输入的连续/离散时间线性系统进行仿真，在不带输出变量引用函数时，lsim可在当前图形窗口中绘制出系统的输出响应曲线。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.1.3 求三阶系统

$$H(s)=\frac{5(s^2+5s+6)}{s^3+6s^2+10s+8}$$

的单位阶跃响应。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.1.4 求典型二阶系统

$$H(s)=\frac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2}$$

当 ξ =0. 7， ω_n =6 时的单位冲激响应。

第三章 控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.1.1 典型二阶系统

$$H(s)=\frac{\omega_n^{\ 2}}{s^2+2\xi\omega_n s+\omega_n^{\ 2}}$$

其中 ω_n 为自然频率（无阻尼振荡频率），ξ 为相对阻尼系数。试绘制出当

ω_n=6，ξ 分别为 0.1，0.2，……，1.0，2.0 时的单位阶跃响应。

第三章 控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.1.2 对例 3.2.1.1 中的典型二阶系统，绘制出当ξ=0. 7， ω_n 取 2，4，6，8，10，12 时的单位阶跃响应。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.1.5 有高阶系统

$$\begin{cases} \dot{x} = \begin{bmatrix} -1.6 & -0.9 & 0 & 0 \\ 0.9 & 0 & 0 & 0 \\ 0.4 & 0.5 & -5.0 & -2.45 \\ 0 & 0 & 2.45 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} u \\ y = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} x \end{cases}$$

求单位阶跃响应，单位冲激响应及零输入响应（设初始状态 $x_0=[1 \quad 1 \quad 1 \quad -1]^T$）。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.1.6 多输入多输出系统

$$\begin{cases} \dot{x} = \begin{bmatrix} 2.25 & -5 & -1.25 & -0.5 \\ 2.25 & -4.25 & -1.25 & -0.25 \\ 0.25 & -0.5 & -1.25 & -1 \\ 1.25 & -1.75 & -0.25 & -0.75 \end{bmatrix} x + \begin{bmatrix} 4 & 6 \\ 2 & 4 \\ 2 & 2 \\ 0 & 2 \end{bmatrix} u \\ y = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 2 & 0 & 2 \end{bmatrix} x \end{cases}$$

求单位阶跃响应和单位冲激响应。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.1.7 将例 3.2.1.5 中的连续系统，以 t=0.5 取样周期，采用双线性变换算法转换成离散系统，然后求出离散系统的单位阶跃响应、单位冲激响应及零输入响应（设初始状态 $x_0=[1\quad 1\quad 1\quad -1]^T$）。

第三章 控制系统数字仿真

第二节 控制系统分析与设计

- 一．时域分析
- 二．频域分析
- 三．根轨迹分析

二．频域分析

◆bode

功能：求连续系统的Bode（波特）频率响应。

格式：

［mag，phase，w］＝bode（a，b，c，d）
［mag，phase，w］＝bode（a，b，c，d，iu）
［mag，phase，w］＝bode（a，b，c，d，iu，w）
［mag，phase，w］＝bode（num，den）
［mag，phase，w］＝bode（num，den，w）

说明：bode函数可计算出连续时间LTI系统的幅频和相频响应曲线（bode图）。bode图可用于分析系统的增益裕度、相位裕度、直接增益、带宽、扰动抑制及其稳定性等特性。

◆dbode

功能： 求离散系统的Bode（波特）频率响应。

格式：

［mag，phase，w］＝dbode（a，b，c，d，Ts）

［mag，phase，w］＝dbode（a，b，c，d，Ts，iu）

［mag，phase，w］＝dbode（a，b，c，d，Ts，iu，w）

［mag，phase，w］＝dbode（num，den，Ts）

［mag，phase，w］＝bode（num，den，Ts，w）

说明： dbode函数可计算出离散时间LTI系统的幅频和相频响应曲线（即Bode图），当不带输出变量引用函数时，dbode函数可在当前图形窗口中直接绘制出系统的Bode图。

◆nyuest

功能：求连续系统的Nyquist（乃氏）频率曲线。

格式：

[re，im，w]＝nyquist（a，b，c，d）

[re，im，w]＝nyquist（a，b，c，d，iu）

[re，im，w]＝nyquist（a，b，c，d，iu，w）

[re，im，w]＝nyquist（num，den）

[re，im，w]＝nyquist（num，den，w）

说明：nyquist函数可计算连续时间LTI系统的乃氏(nyquist)频率曲线，myquist曲线可用来分析包括增益裕度、相位裕度及稳定性在内的系统特性。当不带输出变量引用函数时，nyquist函数会在当前图形窗口中直接绘制出Nyquist曲线。

nyquist函数可以确定单位负反馈系统的稳定性。

dnyuest

功能：求离散系统的Nyquist频率曲线。

格式：

[re，im，w]＝dnyquist（a，b，c，d，Ts）

[re，im，w]＝dnyquist（a，b，c，d，，Ts，iu）

[re，im，w]＝dnyquist（a，b，c，d，Ts，iu，w）

[re，im，w]＝dnyquist（num，den，Ts）

[re，im，w]＝dnyquist（num，den，Ts，w）

说明：dnyquist函数可计算出离散时间LTI系统的nyquist频率响应曲线，当不带输出变量引用函数时，dnyquist函数会在当前图形窗口中直接绘制出Nyquist曲线。

nyquist函数可以确定单位负反馈系统的稳定性。

◆margin

功能： 求增益和相位裕度。

格式：

［gm，pm，wcp，wcg］＝margin（mag，phase，w）

［gm，pm，wcm，wcg］＝margin（num，den）

［gm，pm，wcm，wcg］＝margin（a，b，c，d）

说明： margin函数可从频率响应数据中计算出增益、相位裕度以及有关的交叉频率。增益和相位裕度是针对开环SISO系统而言的，它指示出当系统闭环时的相对稳定性。当不带输出变量引用时，margin可在当前图形窗口中绘制出裕度的Bode图。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.2.1 典型二阶系统

$$H(s)=\frac{\omega_n^2}{s^2+2\xi\omega_n s+\omega_n^2}$$

绘制出 ξ 取不同的值时的 Bode 图。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.2.2 高阶系统

$$H(s)=\frac{100(s+4)}{s(s+0.5)(s+50)^2}$$，试绘制出系统的 Bode 图。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.2.3 试绘制系统 $G(s)=\dfrac{50}{(s+5)(s-2)}$ 的 Nyquist 曲线，并判断闭环系统的稳定性，最后求出闭环系统的单位脉冲响应。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.2.4 试绘制系统 $G(s)=\dfrac{50}{(s+1)(s+5)(s-2)}$ 的 Nyquist 曲线，并判断闭环系统的稳定性，最后求出闭环系统的单位脉冲响应。

第三章 控制系统数字仿真

第二节 控制系统分析与设计

- 一．时域分析
- 二．频域分析
- 三．根轨迹分析

三．根轨迹分析

pzmap

功能： 绘制系统的零极点图。

格式： ［p，z］＝pzmap（a，b，c，d）

［p，z］＝pzmap（num，den）

［p，z］＝pzmap（p，z）

说明： pzmap函数可绘出 LTI系统的零极点图，对SISO系统而言，pzmap函数可绘出传递函数的零极点；对MIMO系统而言，pzmap可绘制出系统的特征矢量和传递零点。当不带输出变量引用时，pzmap可在当前图形窗口中绘制出系统的零极点图。

◆rlocus

功能： 求系统根轨迹。

格式： [r，k]＝rlocus（num，den）

[r，k]＝rlocus（num，den，k）

[r，k]＝rlocus（a，b，c，d）

[r，k]＝rlocus（a，b，c，d，k）

说明：

❖rolcus函数可计算出SISO系统的根轨迹。

❖在不带输出变量引用函数时，rlocus可在当前图形窗口中绘制出系统的根轨迹图。rlocus函数既适用于连续时间系统，也适用于离散时间系统。

rlocfind

功能：计算给定一组根的根轨迹增益。

格式：

［k，poles］＝rlocfind（a，b，c，d）

［k，poles］＝rlocfind（a，b，c，d，p）

［k，poles］＝rlocfind（num，den）

［k，poles］＝rlocfind（num，den，p）

说明：rlocfind函数可计算出与根轨迹上极点对应的根轨迹增益。rlocfind既适用于连续系统，也适用于离散时间系统。

sgrid

功能： 在连续系统根轨迹图和零极点图中绘制出阻尼系数和自然频率栅格。

格式：

sgrid

sgrid（’new’）

sgrid（z，Wn）

sgrid（z，Wn，’new’）

说明： sgrid函数可在连续系统的根轨迹或零极点图上绘制出栅格线，栅格线由等阻尼系数和等自然频率线构成，阻尼系数线以步长0.1从ξ＝0到ξ＝1绘出。

sgrid(’new’)函数先清除图形屏幕，然后绘制出栅格线，并设置成hold on，使后续绘图命令能绘制在栅格上。典型用法如sgrid（’new’）;rlocus(num,den);或pzmap(num,den)。

◆zgrid

功能： 在离散系统根轨迹和零极点图中绘制出阻尼系数和自然频率栅格线。

格式：

zgrid

zgrid（’new’）

zgrid（z，Wn）

zgrid（z，Wn，’new’）

说明：

zgrid函数可在离散系统的根轨迹图或零极点图上绘制出栅格线，栅格线由等阻尼系数和自然频率线构成，阻尼系数线以步长0．1从ξ=0到ξ=1绘出，自然频率线以步长π/10从0到π绘出。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.3.1 设开环系统

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

绘制出通过单位负反馈构成的闭环系统的根轨迹。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.3.2 设开环系统

$$H(s)=\frac{k(s+5)}{s(s+2)(s+3)}$$

绘制出闭环系统的根轨迹，并确定交点处的增益 k。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.3.3 已知开环传递函数为

$$H(s)=\frac{k}{s^4+16s^3+36s^2+80s}$$

绘制出闭环系统的根轨迹。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题

例 3.2.3.4 已知开环传递函数为

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

绘制出闭环系统的根轨迹，并分析其稳定性。

第三章控制系统数字仿真

第二节 控制系统分析与设计

◆例题　系统如图所示，为获得系统的单位阶跃响应，写出对应程序。

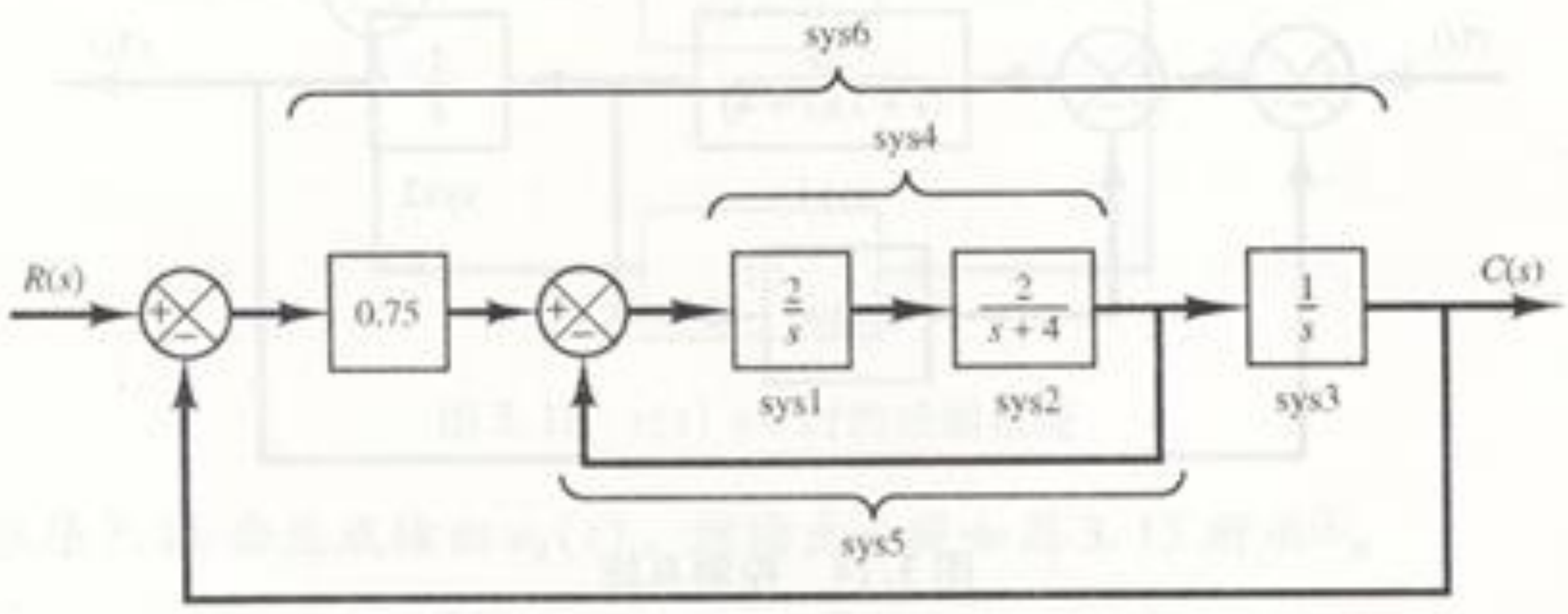
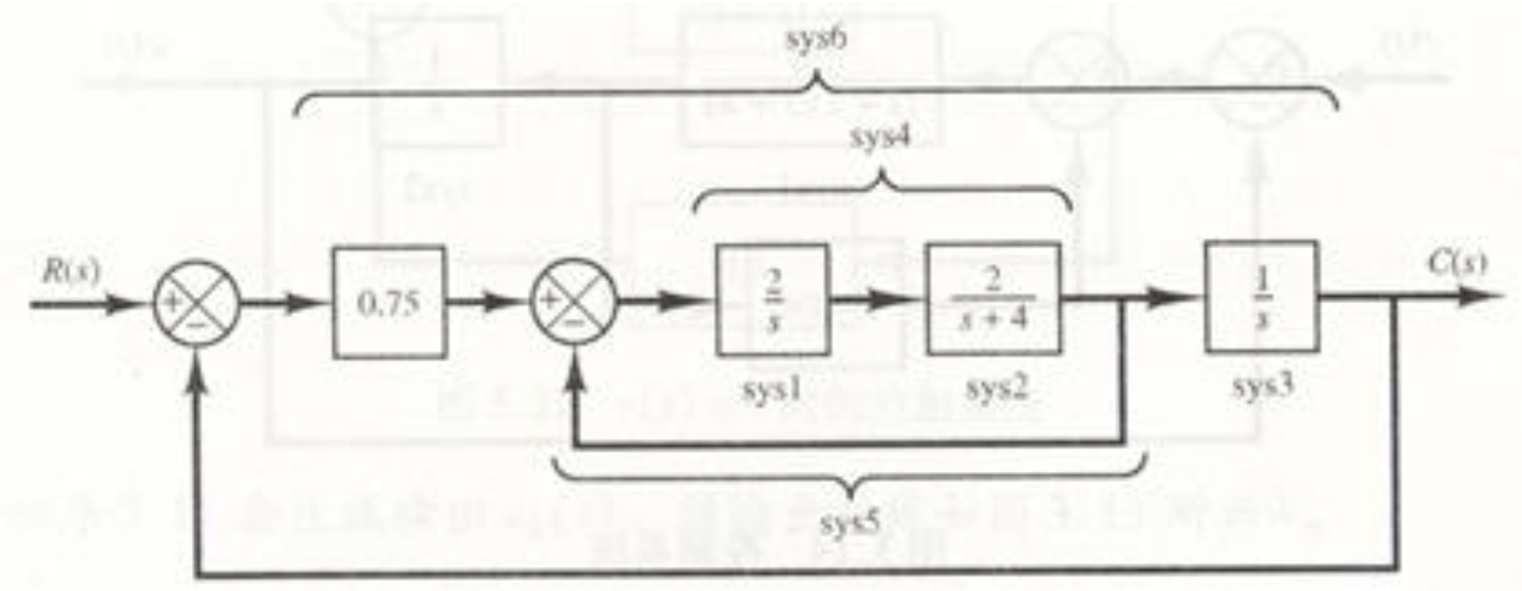


图 3.16　控制系统

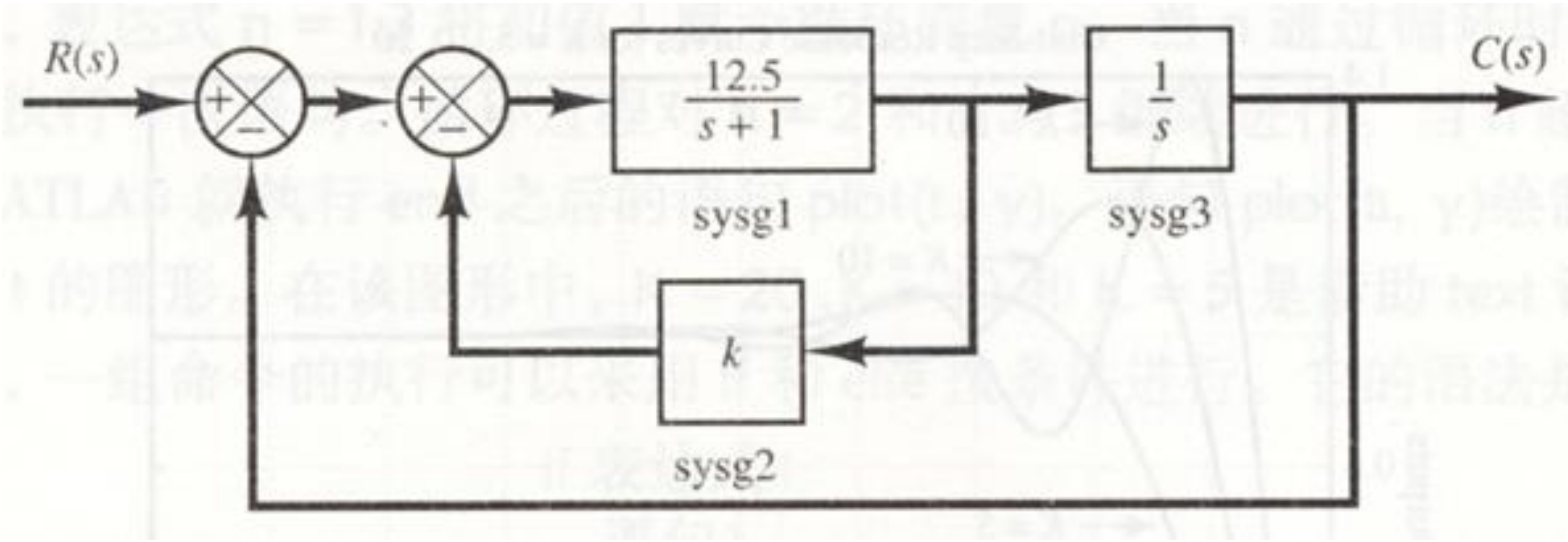


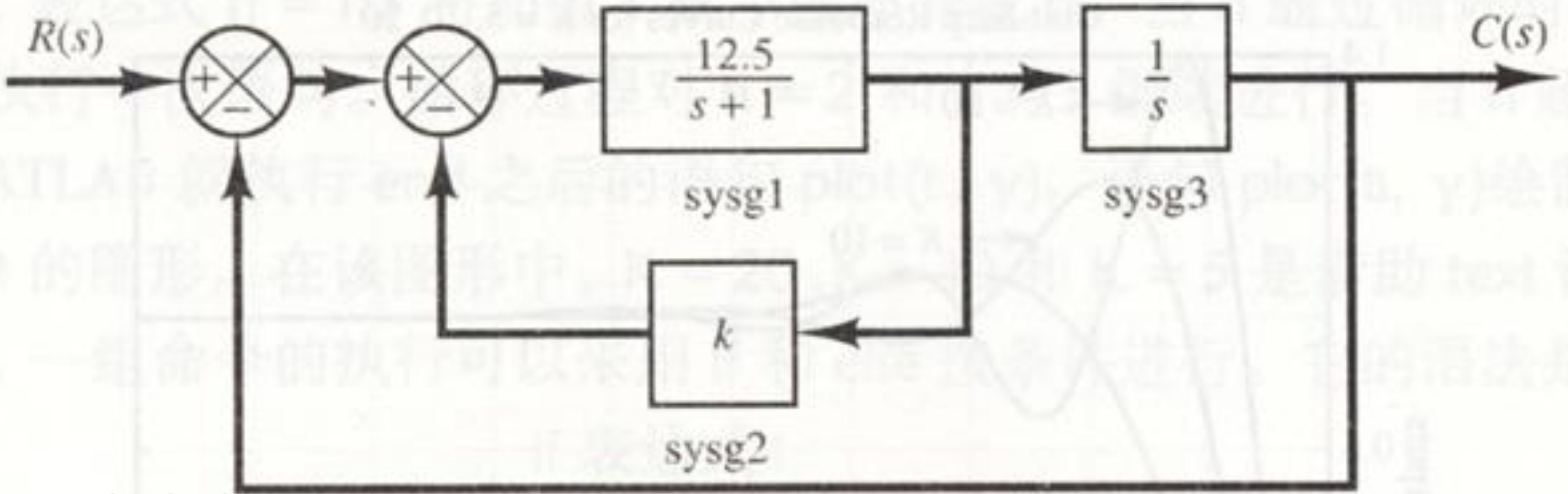

```
num1=[2];den1 =[1 0];sys1=tf(num1,den1);
num2=[2];den2 =[1 4];sys2=tf(num2,den2);
num3=[1];den3 =[1 0];sys3=tf(num3,den3);
sys4  =series(sys1,sys2);
sys5  =feedback(sys4,1);
sys6  =series(0.75*sys5,sys3);
sys  = feedback(sys6,1);
t= 0:0.01:20;
[c,t] =step(sys,t);
plot(t,c), grid,
title('单位阶跃响应')
xlabel('t sec');ylabel('Output c(t)')
```

第三章控制系统数字仿真

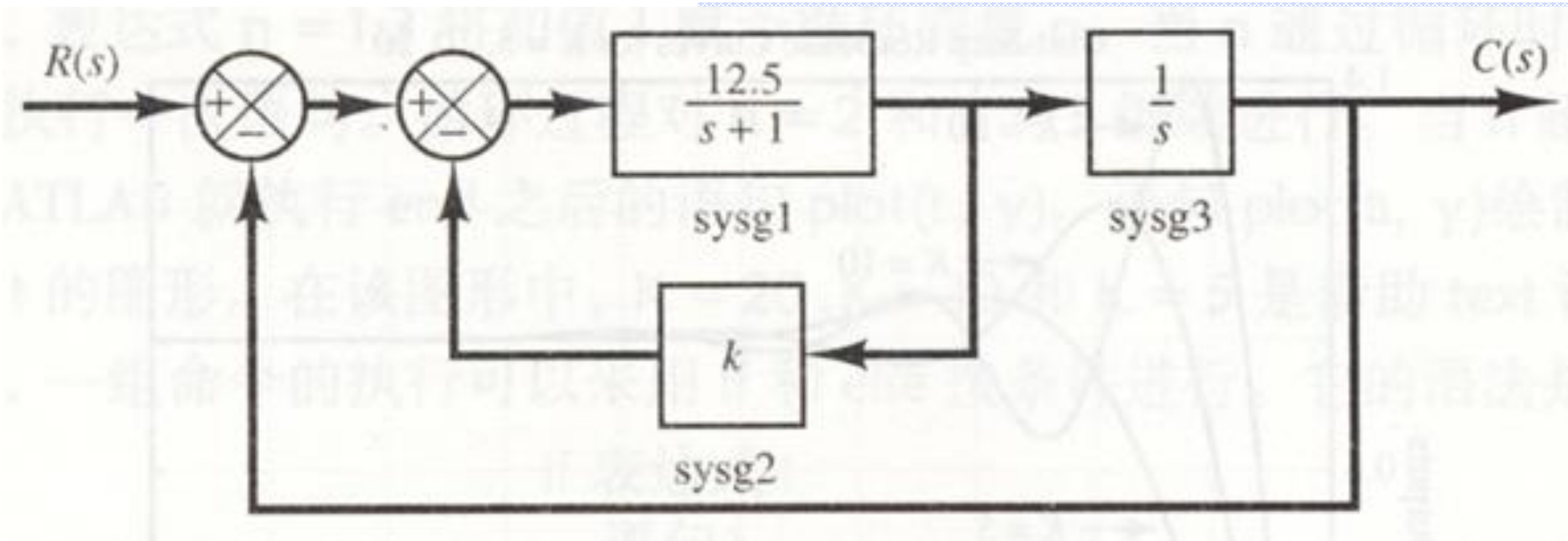
第二节 控制系统分析与设计

◆例题　测速发电机反馈控制系统如图所示，设反馈变量k分别取以下值：0.1,0.2,0.3,0.4,0.5，求系统的单位阶跃响应，写出对应程序。






```
t=0:0.01:0.05;
K=[0.1 0.2,0.3,0.4,0.5];
for n=1:5
    numg1=[12.5];deng1 =[1 1];sys1=tf(numg1,deng1);
    sysg2=[k(n)];
    numg3 = [1];deng3=[1 0];sysg3=tf(numg3,deng3);
    sys1  =feedback(sysg1,sysg2);
    sys2  =series(sys1,sysg3);
    sys   = feedback(sys2,1);
    y(:,n) = step(sys,t);
end
plot(t,y), grid,
```




```
title('单位阶跃响应')
xlabel('t sec');ylabel('Output c(t)')
text(1.5,1.28,'\zeta=0.1')
text(1.5,1.16,'\zeta=0.2')
text(1.5,0.75,'\zeta=0.3')
text(1.5,0.67,'\zeta=0.4')
text(1.5,0.53,'\zeta=0.5')
```

第三章控制系统数字仿真

第二节 控制系统分析与设计

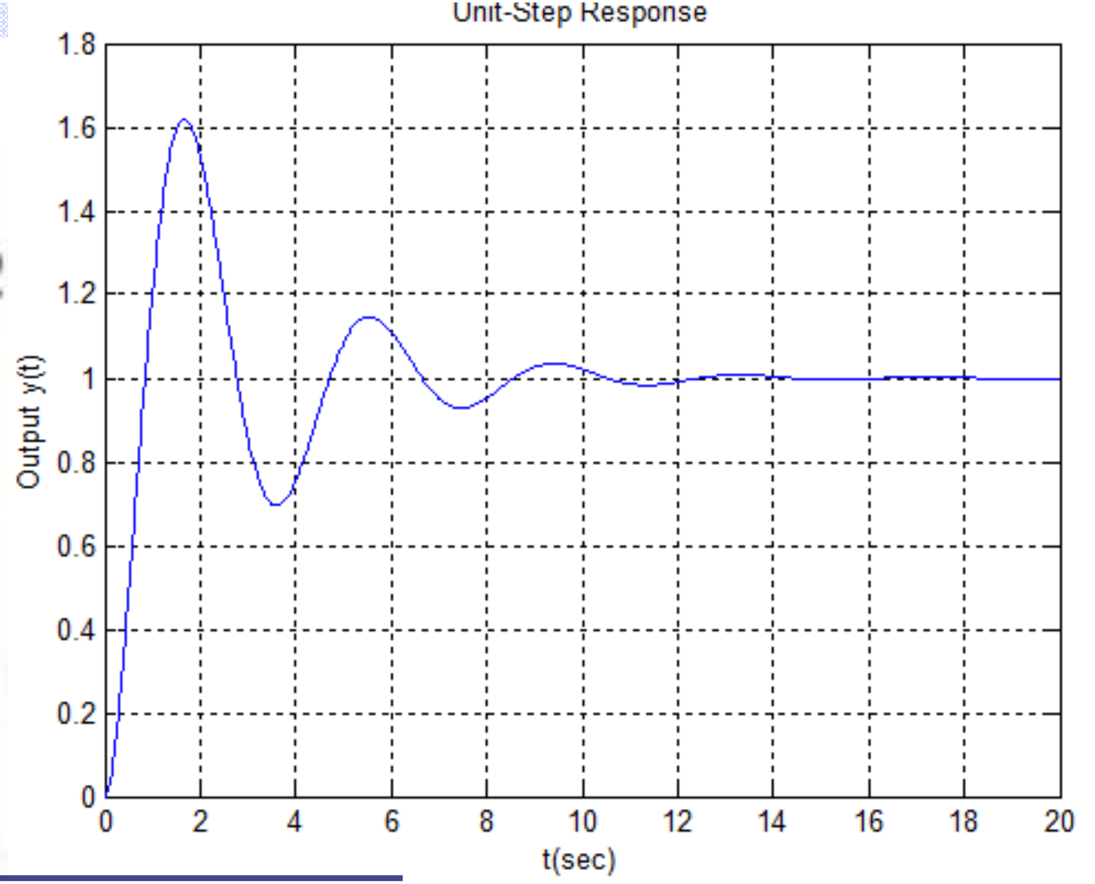
◆例题

已知某高阶系统的传递函数如下：

$$\frac{C(s)}{R(s)}=\frac{6.3223s^2+18s+12.811}{s^4+6s^3+11.3223s^2+18s+12.811}$$

试用Matlab绘制这个系统的单位阶跃响应曲线，并求上升时间，峰值时间，最大超调量和调节时间。

```
num = [6.3223   18   12.811];
den = [1   6   11.3223   18   12
t = 0:0.02:20;
[y,x,t] = step(num,den,t);
plot(t,y)
grid
title('Unit-Step Response')
xlabel('t (sec)')
ylabel('Output y(t)')
```




```
r1 = 1; while y(r1) < 0.1, r1 = r1+1; end;
r2 = 1; while y(r2) < 0.9, r2 = r2+1; end;
rise_time = (r2 – r1)*0.02
```

```
[ymax,tp] = max(y);
peak_time = (tp – 1)*0.02
```

```
max_overshoot = ymax – 1
```

```
s = 1001; while y(s) > 0.98 & y(s) < 1.02; s = s – 1; end;
settling_time = (s – 1)*0.02
```

第三章控制系统数字仿真

第三节常微分方程以及最优问题的求解

- 常微分方程的MATLAB求解
- 最优问题的MATLAB求解

第三章控制系统数字仿真

第三节常微分方程以及最优问题的求解

- 常微分方程的MATLAB求解
  - 求解特点：Matlab可以对一类常微分方程组求取数值解；其他类型的微分方程可以通过适合的算法转换成可解的一阶微分方程（组）求解。
  - 求解器：ode是专门用于解微分方程的功能函数，他有ode23,ode45,ode23s等等，采用的是Runge-Kutta（龙格库塔）算法。ode45表示采用四阶，五阶runge-kutta单步算法,截断误差为(Δx)^3。

◆常微分方程的MATLAB求解

用法:

$$[t,y] = ode45('odefun', tspan, y0)$$

$$[t,y] = ode45('odefun', tspan, y0, options)$$

$$[t,y,TE,YE,IE] = ode45('odefun', tspan, y0, options)$$

$$sol = ode45('odefun', [t0\ tf], y0 \ldots)$$

◆常微分方程的MATLAB求解

[t,y] = ode45('odefun',tspan,y0,options)

- odefun 是函数句柄,可以是函数文件名,
- tspan 是数值解时的初始值和终止值，用区间表示 [t0 tf]；
- y0 是状态变量的初始值向量；
- Options：求解微分方程的一些控制参数；可以用odeset在计算前设定误差，输出参数，事件等。
- t 返回列向量的时间点
- y 返回对应t的求解列向量，解数组 y 中的每一行都与列向量t中返回的值相对应。

◆常微分方程的MATLAB求解

eg：考虑微分方程组：

$$\begin{cases} \dot{x_1}(t) = -x_2(t) - x_3(t) \\ \dot{x_2}(t) = x_1(t) + ax_2(t) \\ \dot{x_3}(t) = b + [x_1(t) - c]x_3(t) \end{cases}$$

选定：

$$\begin{cases} a = b = 0.2 \\ c = 5.7 \end{cases}$$

且 $x_1(0) = x_2(0) = x_3(0)$，求解该微分方程组.

◆常微分方程的MATLAB求解

第一步：若想求解这个微分方程，需要用户自己去编写一个 Matlab 函数来描述这个常微分方程.

$$\begin{cases} \dot{x}_1(t) = -x_2(t) - x_3(t) \\ \dot{x}_2(t) = x_1(t) + a x_2(t) \\ \dot{x}_3(t) = b + [x_1(t) - c] x_3(t) \end{cases}$$

```
function dx=ressler( t , x)
      dx=[-x(2)-x(3);
          x(1)+0.2* x(2);
          0.2+(x(1)-5.7)*x(3)]
```

% 对比此函数和给出的数学方程，编写这样的函数是很直观的.

◆常微分方程的MATLAB求解

第二步: 求解微分方程的数值解

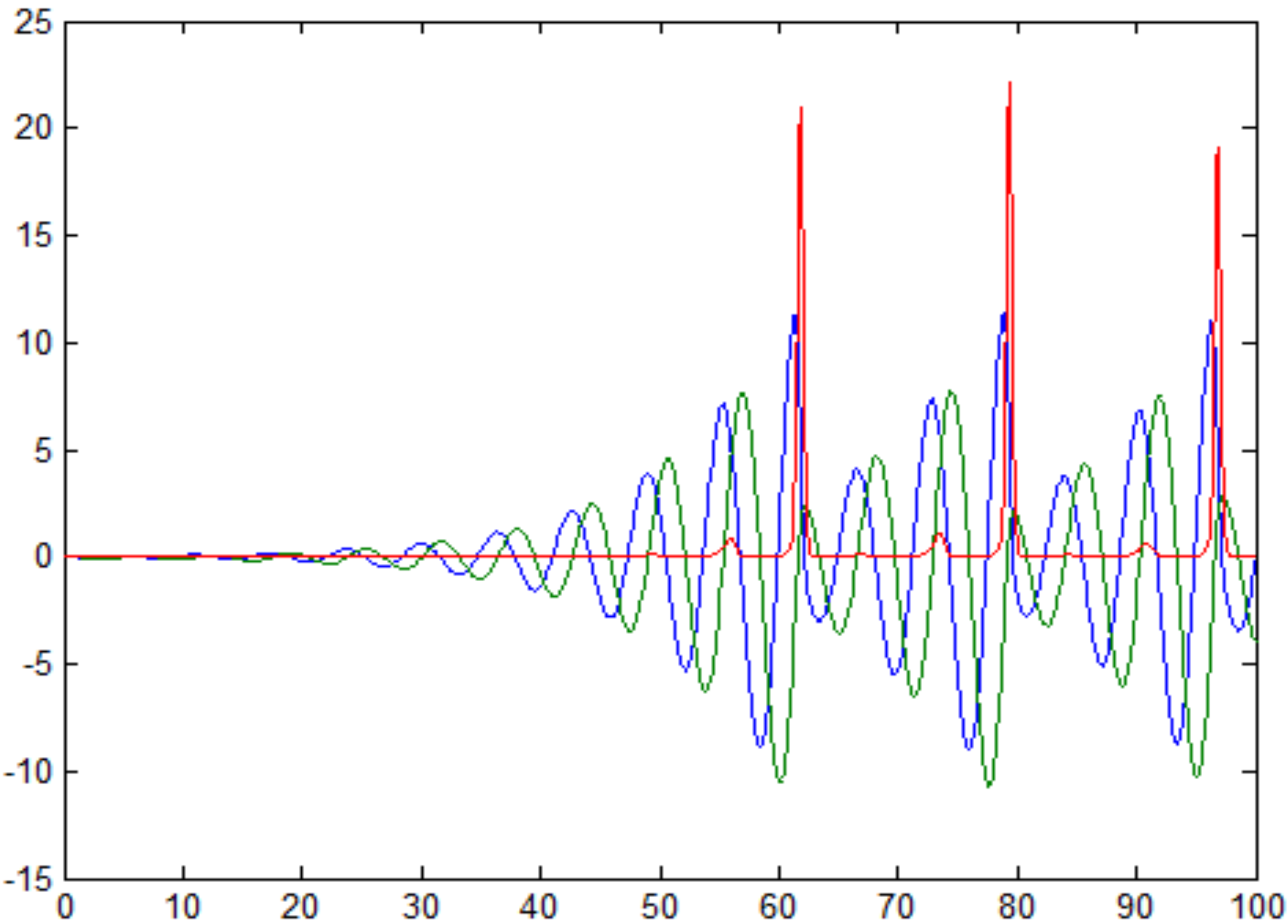
$\gg$ x_0=[0;0;0]　% 微分方程　$x_1(0) = x_2(0) = x_3(0)$

$\gg$ [t,y]=ode45('ressler',[0,100],x_0)

% 求解微分方程. 注意函数名要有‘’号

$\gg$ plot(t,y)　% 绘制各个状态的响应曲线

◆常微分方程的MATLAB求解



求解出来的各个状态的响应曲线

常微分方程的MATLAB求解

例:求解二阶*van der Pol*方程

$$y_1'' - \mu(1 - y_1^2)y_1' + y_1 = 0$$

在时间区间 [0 20] 和初始值 [2 0]条件下的解，其中$\mu > 0$ 为标量参数。

◆常微分方程的MATLAB求解

例:求解二阶 *van der Pol* 方程

$$y_1'' - \mu(1 - y_1^2)y_1' + y_1 = 0$$

在时间区间 [0 20] 和初始值 [2 0]条件下的解，其中$\mu > 0$ 为标量参数。

解：通过执行$y_1' = y_2$ 代换，将此方程重写为一阶 *ODE* 方程组。生成的一阶 *ODE* 方程组为

$$\begin{cases} y_1' = y_2 \\ y_2' = \mu(1 - y_1^2)y_2 - y_1 \end{cases}$$

◆常微分方程的MATLAB求解

◆例:求解二阶 *van der Pol* 方程

$$y_1'' - \mu(1 - y_1^2)y_1' + y_1 = 0$$

在时间区间 [0 20] 和初始值 [2 0]条件下的解，其中$\mu > 0$ 为标量参数。

解： 使用$\mu = 1$的 *van der Pol* 方程。变量y_1 和 y_2 是二元素向量 *dydt* 的项 *y*(1) 和 *y*(2)。

```
function dydt = vdp1(t,y)
%VDP1  Evaluate the van der Pol ODEs for mu = 1
dydt = [y(2); (1-y(1)^2)*y(2)-y(1)];
```

◆常微分方程的MATLAB求解

◆例:求解二阶 *van der Pol* 方程

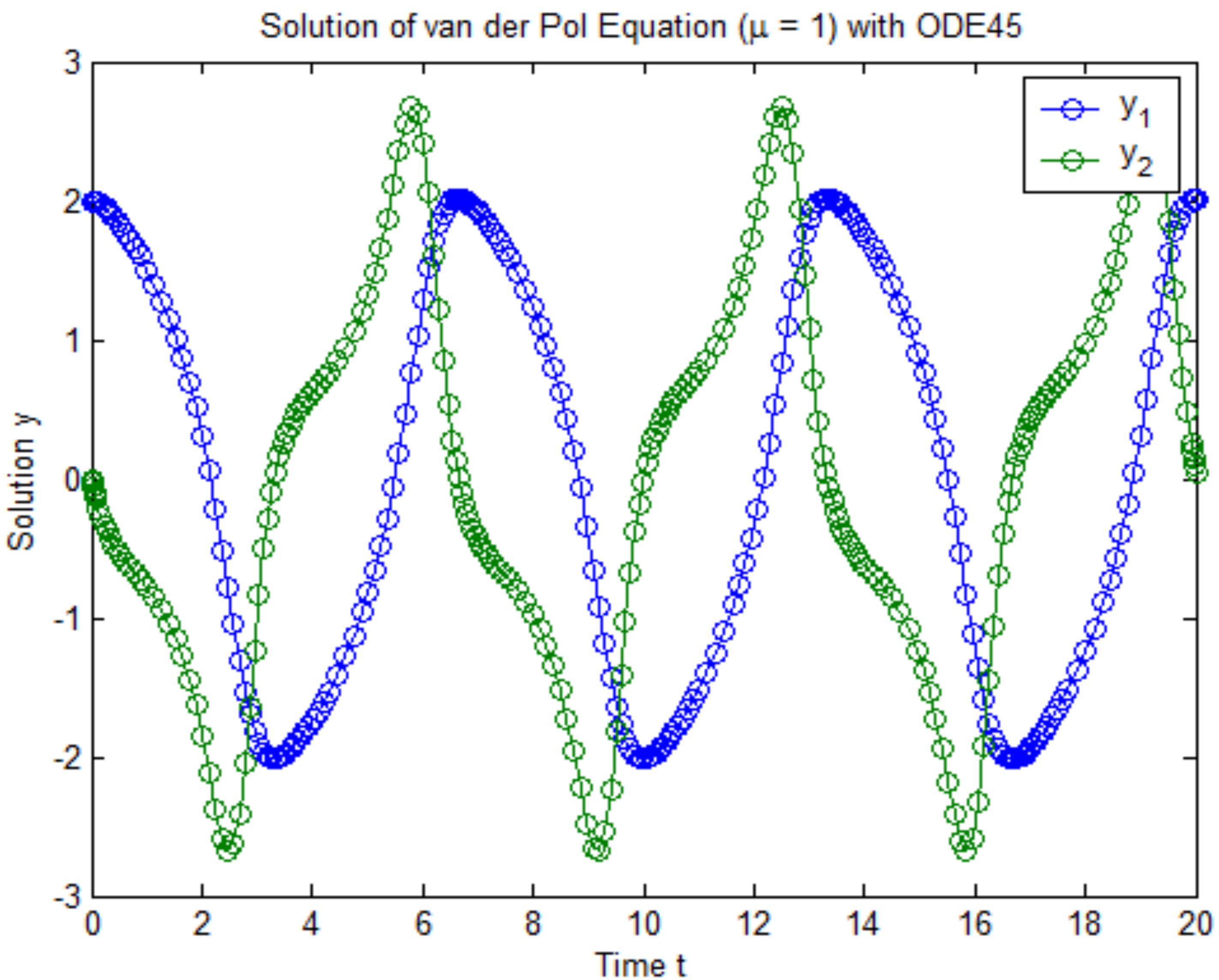
$$y_1'' - \mu(1 - y_1^2)y_1' + y_1 = 0$$

在时间区间 [0 20] 和初始值 [2 0]条件下的解，其中$\mu > 0$ 为标量参数。

解:

```
[t,y] = ode45(vdp1,[0 20],[2; 0]);
plot(t,y(:,1),'-o',t,y(:,2),'-o')
title('Solution of van der Pol Equation
(\mu = 1) with ODE45');
xlabel('Time t');
ylabel('Solution y');
legend('y_1','y_2')
```

◆常微分方程的MATLAB求解



第三章控制系统数字仿真

第三节常微分方程以及最优问题的求解

- 常微分方程的MATLAB求解
- 最优问题的MATLAB求解

◆最优问题的MATLAB求解

- 无约束最优问题

$$\min_{x} \quad F(x)$$

- 求解该优化问题的的 Matlab 函数

$$[\mathrm{x}, f_{opt}, \mathrm{key}, \mathrm{c}] = f\,minsearch(\mathrm{Fun}, \mathrm{x}_0, \mathrm{opt})$$

- Fun 为要求解问题的数学描述；
- x_0 为自变量的起始搜索点；
- opt 为最优化工具箱的选项设定；
- x 为返回的解；
- f_{opt} 是目标函数在 x 点的值；
- key 函数返回的条件
  1—已经求解出方程的解
  0—未搜索到方程的解
- c 为解的附加信息

◆最优问题的MATLAB求解

- 有约束的最优问题

$$
\begin{aligned}
& \min_{x} \quad f(x) \\
\text{s.t.} \quad & \mathrm{Ax} \leq \mathrm{B} \\
& A_{eq}\mathrm{x} = B_{eq} \\
& \mathrm{x}_m \leq \mathrm{x} \leq \mathrm{x}_m \quad \text{\% 按元素给出} \\
& \mathrm{c(x)} \leq 0 \\
& c_{eq}(\mathrm{x}) = 0
\end{aligned}
$$

◆最优问题的MATLAB求解

格式：

$x = \text{fmincon}(fun, x_0, A, B, Aeq, Beq, X_m, X_M, nonlcon，opts)$

- 1. fun为你要求最小值的函数。
- 2. x0, 表示初始的猜测值，大小要与变量数目相同
- 3. A B 为线性不等约束，A*x <= B， A应为n*n阶矩阵
- 4 Aeq Beq为线性相等约束，Aeq*x = Beq。
- 5 Xm ,XM为变量的上下边界， 正负无穷用 -Inf和Inf表示，Xm,XM应为N阶数组
- 6 nonlcon 为非线性约束，可分为两部分，非线性不等约束 c，非线性相等约束ceq。
- 7 opts 可以用OPTIMSET函数设置，具体可见OPTIMSET函数的帮助文件。

◆最优问题的MATLAB求解

◆eg: $$\underset{x}{\text{minimize}} \quad e^{x_1}\left(4x_1^2+2x_2^2+4x_1x_2+2x_2+1\right)$$

subject to

$$x_1+x_2\le 0$$

$$-x_1x_2+x_1+x_2\ge 1.5$$

$$x_1x_2\ge -10$$

$$-10\le x_1,x_2\le 10$$

◆最优问题的MATLAB求解

◆目标函数：

$$\underset{x}{\text{minimize}} \quad e^{x_1}\left(4x_1^2 + 2x_2^2 + 4x_1x_2 + 2x_2 + 1\right)$$

```
function y=myobj(x)

y=exp(x(1))*(4*x(1)^2+2*x(2)^2+4*x(1)*x(2)+2*x(2)+1)
```

◆最优问题的MATLAB求解

subject to

$$x_1 + x_2 \le 0$$

◆约束变量：

$$-x_1x_2 + x_1 + x_2 \ge 1.5$$

$$x_1x_2 \ge -10$$

$$-10 \le x_1, x_2 \le 10$$

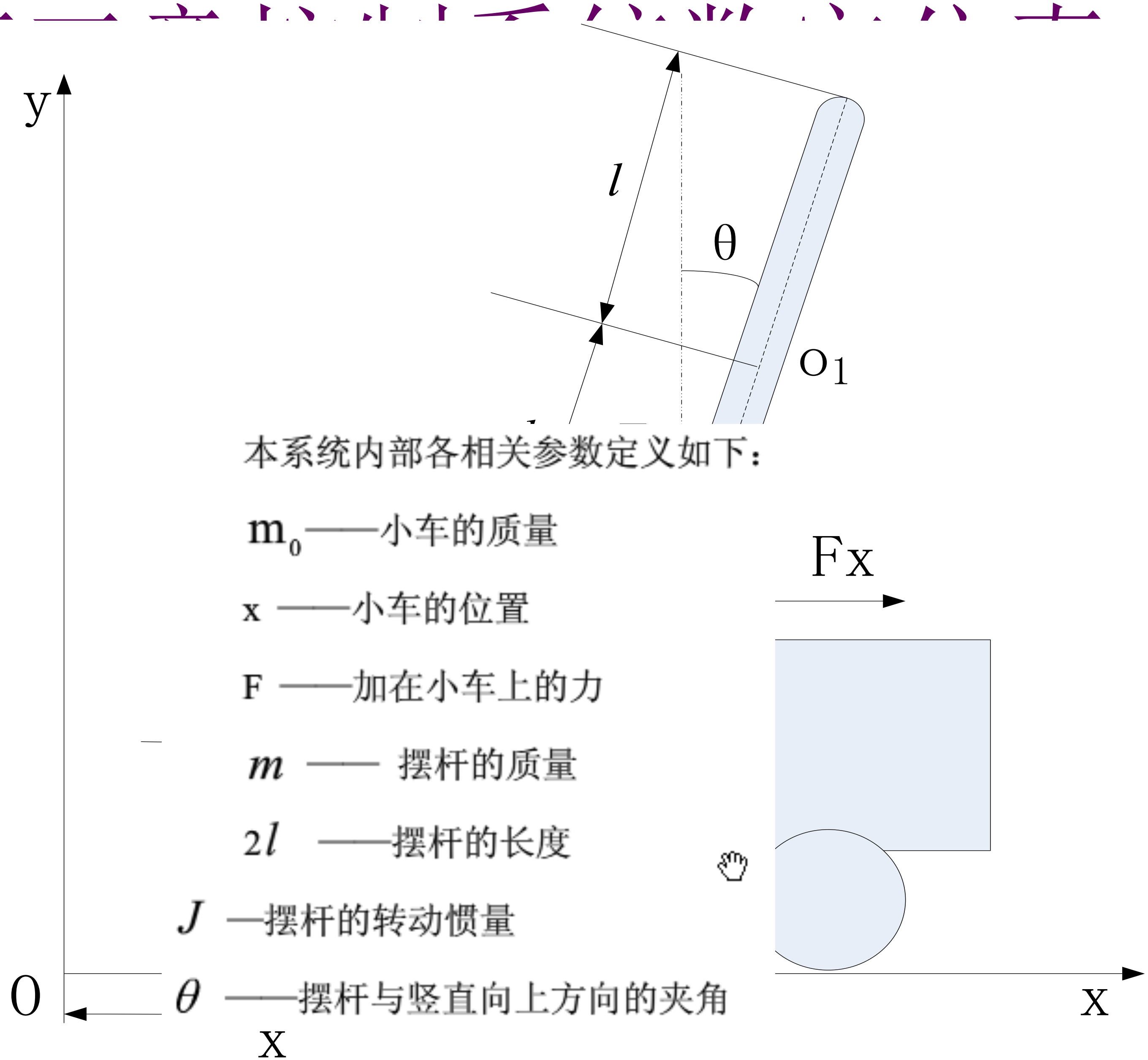
```
function[c,ce]=mycon(x)
    ce=[ ]; % 无等式约束
    c=[ x(1) + x(2);
        x(1) * x(2) – x(1) – x(2) + 1.5;
        –10 – x(1) * x(2)]
```

注： x(1)+x(2) 也可以采用AX ≤ B来描述

最优问题的MATLAB求解

- 主函数：
  - A=[ ]; B=[ ];
  - Aeq=[ ]; Beq=[ ];
  - xm=[-10;-10]; xM=[10;10];
  - x0=[5;5];

[x, fopt , key, c] =
fmincon(myobj , xo,A,B,Aeq,Beq, xm, xM,mycon)



本系统内部各相关参数定义如下：

m_0——小车的质量

x ——小车的位置

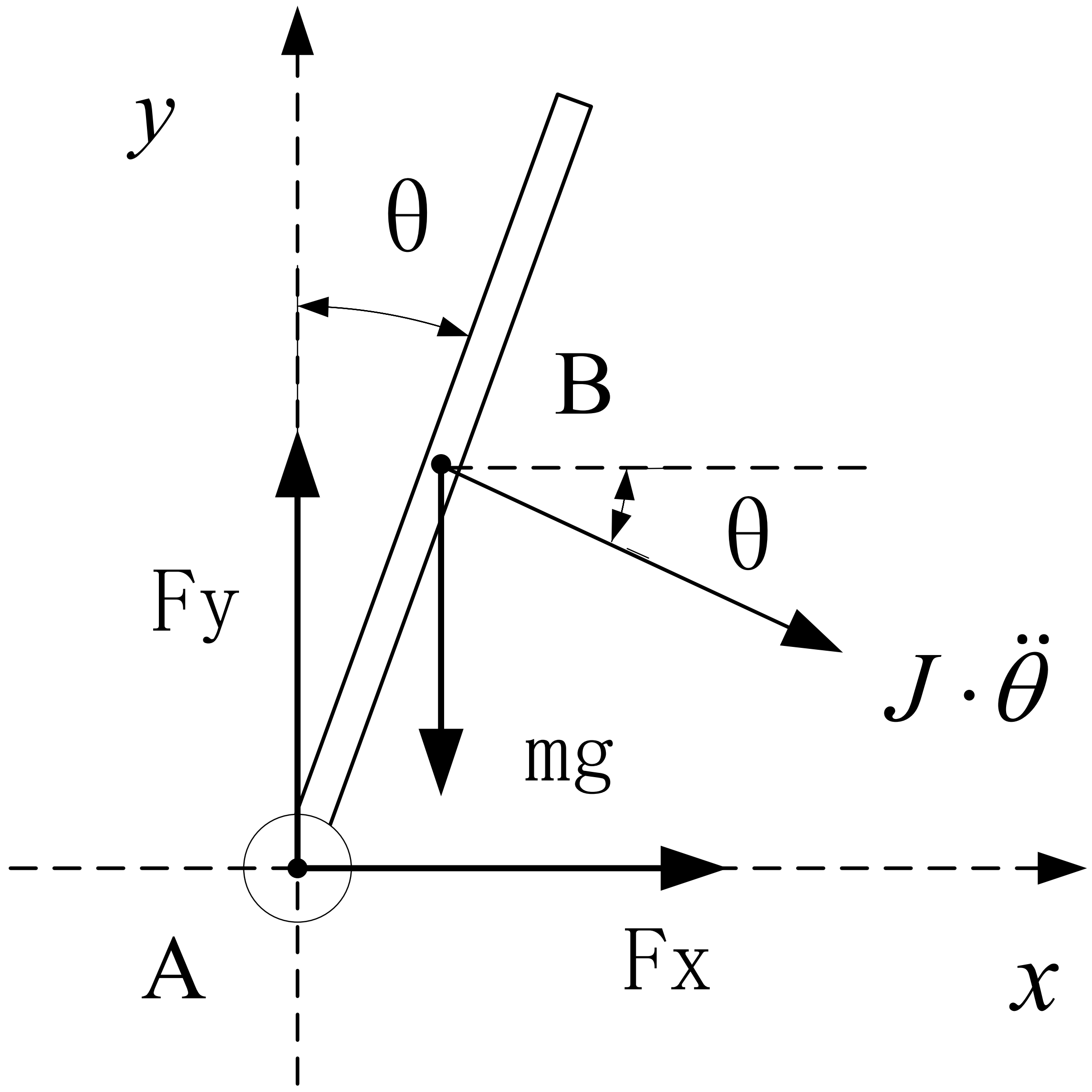
F ——加在小车上的力

m —— 摆杆的质量

$2l$ ——摆杆的长度

J —摆杆的转动惯量

θ ——摆杆与竖直向上方向的夹角



y
θ
B
θ
Fy
$J \cdot \ddot{\theta}$
mg
A
Fx
x

根据刚体绕定轴转动的动力学微分方程，转动惯量与加速度乘积等于刚体主动力对该轴力矩的代数和，则摆杆绕其重心的转动方程为

$$J\ddot{\theta} = F_y l \sin\theta - F_x l \cos\theta \qquad (2\text{-}1)$$

摆杆重心的水平运动可以描述为

$$F_x = m\frac{d^2}{dt^2}(x + l\sin\theta) \qquad (2\text{-}2)$$

摆杆重心在垂直方向上的运动可描述为

$$F_{\mathrm{y}} - mg = m\frac{d^2}{dt^2}(l\cos\theta) \qquad (2\text{-}3)$$

小车水平方向运动可描述为

$$F - F_{\mathrm{x}} = m_0\frac{d^2x}{dt^2} \qquad (2\text{-}4)$$

由式（2-2）和式（2-4）得

$$(\mathrm{m}_0 + m)\ddot{x} + ml(\cos\theta\cdot\ddot{\theta} - \sin\theta\cdot\ddot{\theta}) = F \qquad (2\text{-}5)$$

由式（2-1）和式（2-3）得

$$(J + ml^2)\ddot{\theta} + ml\cos\theta\cdot\ddot{x} = m\mathrm{lg}\cdot\sin\theta \qquad (2\text{-}6)$$

$$\begin{cases} \ddot{x} = \dfrac{(J+ml^2)F + lm(J+ml^2)\sin\theta\cdot\dot{\theta}^2 - m^2l^2 g\sin\theta\cos\theta}{(J+ml^2)(m_0+m) - m^2l^2\cos^2\theta} \\ \ddot{\theta} = \dfrac{ml\cos\theta\cdot F + m^2l^2\sin\theta\cos\theta\cdot\dot{\theta}^2 - (m_0+m)m\lg\sin\theta}{m^2l^2\cos^2\theta - (J+ml^2)(m_0+m)} \end{cases}$$

（2-7）

因为摆杆是均匀细杆，所以可求其对质心的转动惯量。因此设细杆的摆为 $2l$，单位长度的质量为 ρl,取杆上一个微段 dx，其质量为 $\mathrm{m} = \rho l\ dx$，则此杆对于质心的转动惯量有

$$J = \int_{-l}^{l} (\rho_l dx) x^2 = 2\rho l^3 / 3$$

杆的质量为

$$m = 2\rho_l l$$

所以此杆对于质心的转动惯量有

$$J = \frac{ml^2}{3}$$

由式（2-7）可见，一阶直线倒立摆系统的动力学模型为非线性微分方程组。为了便于分析和计算，必须将其简化为线性定常的系统模型。

若只考虑θ在其工作点$\theta_0=0$附近（$-10^0<\theta<10^0$）的细微变化，则可近似认为$\dot{\theta}^2\approx 0$，$\sin\theta\approx\theta$，$\cos\theta\approx 1$。在这一简化思想下，系统的精确模型式（2-7）可简化为

$$\begin{cases}\ddot{x}=\dfrac{(J+ml^2)F-m^2l^2g\theta}{J(m_0+m)+m_0ml^2}\\ \ddot{\theta}=\dfrac{(m_0+m)m\lg\theta-mlF}{J(m_0+m)+m_0ml^2}\end{cases}$$

若给定一阶倒立摆系统的参数为：小车的质量m_0=1kg；倒摆振子的质量m=1.5kg；倒摆长度 2l=0.8m；重力加速度取$g=10m/s^2$，则可以进一步简化模型：

$$\begin{cases} \ddot{x} = \dfrac{-3.6\theta + 0.32F}{0.44} \\ \ddot{\theta} = \dfrac{15\theta - 0.6F}{0.44} \end{cases} \tag{2-8}$$

上式为系统的微分方程，对其进行拉氏变换课的系统的传递函数模型为

$$\begin{cases} G_1(\mathrm{s}) = \dfrac{\Theta(s)}{F(s)} = \dfrac{-0.6}{0.44s^2 - 15} \\ G_2(s) = \dfrac{X(s)}{\Theta(s)} = \dfrac{-0.176s^2 + 3.3}{0.33s^2} \end{cases} \tag{2-9}$$

同理可求系统的状态方程模型如下：

设系统状态为

$$x=\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}=\begin{bmatrix} \theta \\ \dot{\theta} \\ x \\ \dot{x} \end{bmatrix}$$

则有系统状态方程

$$\dot{x}=\begin{bmatrix}\dot{x}_1\\\dot{x}_2\\\dot{x}_3\\\dot{x}_4\end{bmatrix}=\begin{bmatrix}0 & 1 & 0 & 0\\ \dfrac{15}{0.44} & 0 & 0 & 0\\ 0 & 0 & 0 & 1\\ \dfrac{-3.6}{0.44} & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}+\begin{bmatrix}0\\ \dfrac{-0.6}{0.44}\\ 0\\ \dfrac{0.32}{0.44}\end{bmatrix}F=Ax+BF \quad (2\text{-}10)$$

输出方程

$$y=\begin{bmatrix}\theta\\x\end{bmatrix}=\begin{bmatrix}1000\\0010\end{bmatrix}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}=Cx \quad (2\text{-}11)$$