国家精品课程/ 国家精品资源共享课程/ 国家级精品教材
国家级十一(二)五规划教材/ 教育部自动化专业教学指导委员会牵头规划系列教材

控制系统仿真与CAD

第七章 控制器设计的经典方法

状态空间设计方法（中）

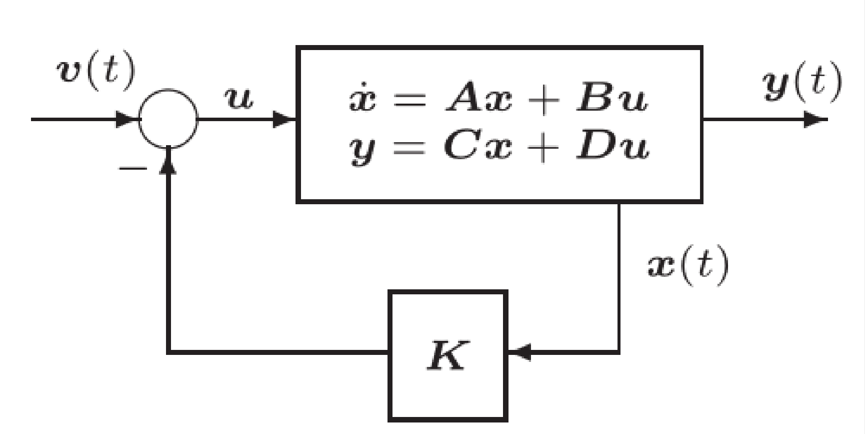
State Space Design Methods (II)

主讲：薛定宇教授

极点配置控制器设计

- 受控对象——状态方程

$$\begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{u}(t) \end{cases}$$



- 控制信号 $\boldsymbol{u}(t) = \boldsymbol{v}(t) - \boldsymbol{K}\boldsymbol{x}(t)$
- 闭环状态方程模型

$$\begin{cases} \dot{\boldsymbol{x}}(t) = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K})\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{v}(t) \\ \boldsymbol{y}(t) = (\boldsymbol{C} - \boldsymbol{D}\boldsymbol{K})\boldsymbol{x}(t) + \boldsymbol{D}\boldsymbol{v}(t) \end{cases}$$

- 指定闭环极点 $\boldsymbol{p}$，如何设计 $\boldsymbol{K}$？

极点配置算法

- 假设闭环系统期望的极点位置为 $\mu_i, i = 1, 2, \cdots, n$

则闭环系统的特征方程 $\alpha(s)$ 可以表示成

$$\alpha(s) = \prod_{i=1}^{n}(s - \mu_i) = s^n + \alpha_1 s^{n-1} + \alpha_2 s^{n-2} + \cdots + \alpha_{n-1} s + \alpha_n$$

- 不同的极点配置算法
  - Bass-Gura 算法　　$\boldsymbol{K}$=bass_pp($\boldsymbol{A}$,$\boldsymbol{B}$,$\boldsymbol{p}$)
  - Ackermann 算法　　$\boldsymbol{K}$=acker($\boldsymbol{A}$,$\boldsymbol{B}$,$\boldsymbol{p}$)
  - 鲁棒配置算法　　$\boldsymbol{K}$=place($\boldsymbol{A}$,$\boldsymbol{B}$,$\boldsymbol{p}$)

Bass-Gura 算法

- 开环特征方程

$$a(s) = \det(s\boldsymbol{I} - \boldsymbol{A}) = s^n + a_1 s^{n-1} + a_2 s^{n-2} + \cdots + a_{n-1}s + a_n$$

- 反馈向量

$$\boldsymbol{\gamma}^{\mathrm{T}} = [(a_n - \alpha_n), \cdots, (a_1 - \alpha_1)], \quad \boldsymbol{T}_{\mathrm{c}} = [\boldsymbol{B}, \boldsymbol{A}\boldsymbol{B}, \cdots, \boldsymbol{A}^{n-1}\boldsymbol{B}]$$

$$\boldsymbol{\Gamma} = \begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \cdots & 1 & \\ \vdots & \vdots & \ddots & & \\ a_1 & 1 & & & \\ 1 & & & & \end{bmatrix} \qquad \boldsymbol{K} = \boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{T}_{\mathrm{c}}^{-1}$$

Bass-Gura算法的MATLAB实现

- Bass-Gura算法编程

```
function K=bass_pp(A,B,p)
a1=poly(p); a=poly(A); L=hankel(a(end-1:-1:1)); C=ctrb(A,B);
K=(a1(end:-1:2)-a(end:-1:2))*inv(L)*inv(C);
```

- Ackermann 算法

$$\boldsymbol{K} = -[0, 0, \cdots, 0, 1]\, \boldsymbol{T}_{\mathrm{c}}^{-1} \alpha(\boldsymbol{A})$$

- 函数的区别
  - place() 函数能处理多变量系统
  - 另外两种方法可以处理多重极点配置问题

例7-3 极点配置

- 状态方程

$$\dot{\boldsymbol{x}}(t)=\begin{bmatrix}0&2&0&0&-2&0\\1&0&0&0&0&-1\\0&1&0&0&0&0\\0&0&0&3&0&0\\2&0&0&1&0&0\\0&0&-1&0&1&0\end{bmatrix}\boldsymbol{x}(t)+\begin{bmatrix}1&2\\0&0\\0&1\\0&-1\\0&1\\0&0\end{bmatrix}\boldsymbol{u}(t)$$

- 期望极点位置 $-1,-2,-3,-4,-1\pm\mathrm{j}$

```
>> A=[0,2,0,0,-2,0; 1,0,0,0,0,-1; 0,1,0,0,0,0;
      0,0,0,3,0,0;  2,0,0,1,0,0; 0,0,-1,0,1,0];
   B=[1,2; 0,0; 0,1; 0,-1; 0,1; 0,0];
   p=[-1 -2 -3 -4 -1+1i -1-1i];
   K=place(A,B,p), p1=eig(A-B*K)
```

例7-4 极点配置失败

- 状态方程

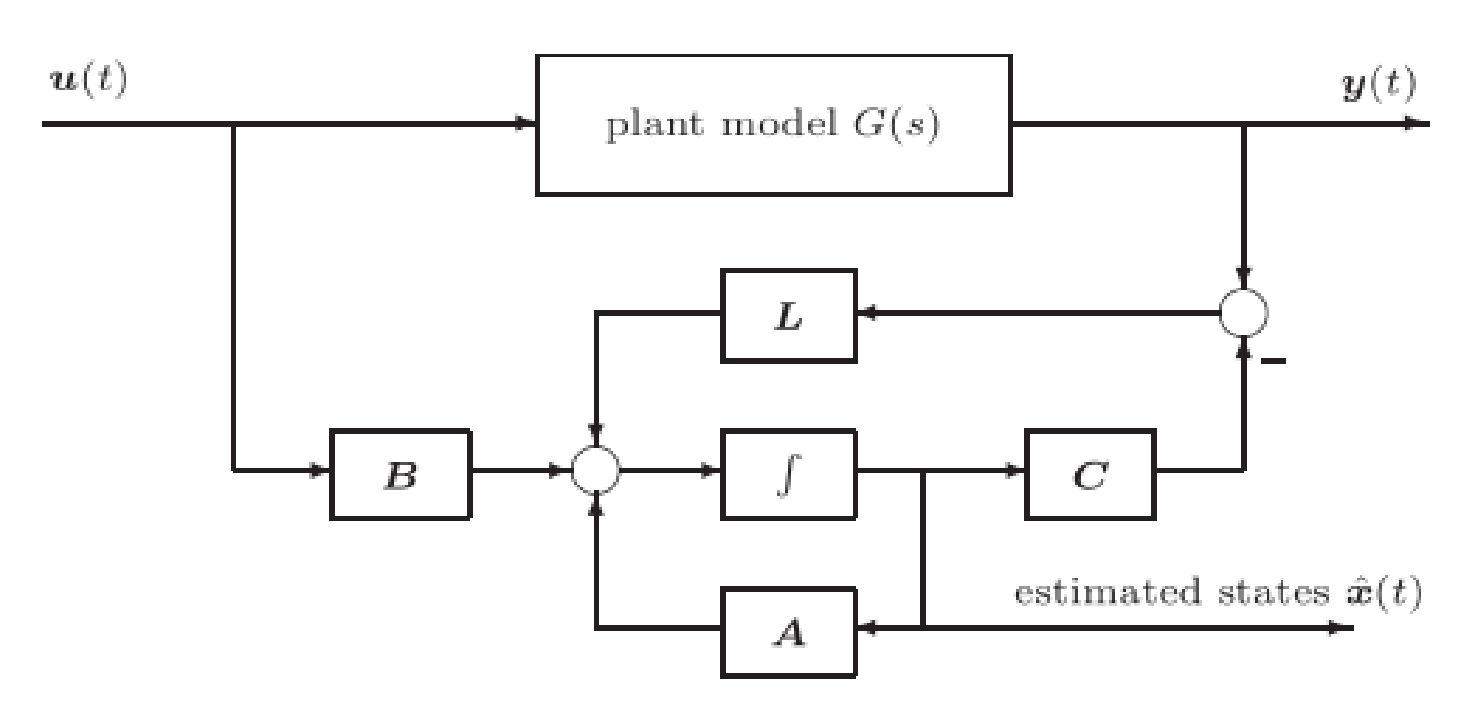
$$\boldsymbol{x}[(k+1)T]=\begin{bmatrix}0&1&0&0\\0&0&-1&0\\0&0&0&1\\0&0&5&0\end{bmatrix}\boldsymbol{x}(kT)+\begin{bmatrix}0&1\\0&-1\\0&0\\0&0\end{bmatrix}\boldsymbol{u}(kT)$$

- 期望极点 $-0.1, -0.2, -0.5\pm 0.2\mathrm{j}$
- 控制器设计 (不能配置)

```
>> A=[0 1 0 0 ; 0 0 -1 0; 0 0 0 1; 0 0 5 0];
   B=[0 1 ; 0 -1; 0 0 ; 0 0];
   p=[-0.1; -0.2; -0.5+0.2i; -0.5-0.2i]; K=place(A,B,p)
>> rank(ctrb(A,B))
```

观测器的框图

- 如果状态不能直接测出，则需要观测器重建状态



观测器稳定性和收敛性

- 观测器数学模型

$$\begin{aligned}\dot{\hat{\boldsymbol{x}}}(t) &= \boldsymbol{A}\hat{\boldsymbol{x}}(t) + \boldsymbol{B}\boldsymbol{u}(t) - \boldsymbol{L}(\boldsymbol{C}\hat{\boldsymbol{x}}(t) + \boldsymbol{D}\boldsymbol{u}(t) - \boldsymbol{y}(t)) \\ &= (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\hat{\boldsymbol{x}}(t) + (\boldsymbol{B} - \boldsymbol{L}\boldsymbol{D})\boldsymbol{u}(t) + \boldsymbol{L}\boldsymbol{y}(t)\end{aligned}$$

- 观测器收敛性

$$\begin{aligned}\dot{\hat{\boldsymbol{x}}}(t) - \dot{\boldsymbol{x}}(t) &= (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})\hat{\boldsymbol{x}}(t) + (\boldsymbol{B} - \boldsymbol{L}\boldsymbol{D})\boldsymbol{u}(t) + \boldsymbol{L}\boldsymbol{y}(t) - \boldsymbol{A}\boldsymbol{x}(t) - \boldsymbol{B}\boldsymbol{u}(t) \\ &= (\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})[\hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t)]\end{aligned}$$

- 方程解析解 $\hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t) = \mathrm{e}^{(\boldsymbol{A}-\boldsymbol{L}\boldsymbol{C})(t-t_0)}[\hat{\boldsymbol{x}}(t_0) - \boldsymbol{x}(t_0)]$
- $(\boldsymbol{A} - \boldsymbol{L}\boldsymbol{C})$ 稳定性 $\lim\limits_{t\to\infty}[\hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t)] = \mathbf{0}$

观测器仿真与设计

➢ MATLAB程序

```
function [xh,x,t]=simobsv(G,L)
[y,t,x]=step(G); G=ss(G); A=G.a; B=G.b; C=G.c; D=G.d;
[y1,xh1]=step((A-L*C),(B-L*D),C,D,1,t);
[y2,xh2]=lsim((A-L*C),L,C,D,y,t); xh=xh1+xh2;
```

➢ 调用格式

$$[\hat{x},x,t]=\mathrm{simobsv}(G,L)$$

例7-5 状态观测与仿真

- 原始状态方程模型

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & -0.1 & 8 & 0 \\ 0 & 0 & -10 & 16 \\ 0 & 0 & 0 & -20 \end{bmatrix} \boldsymbol{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.3953 \end{bmatrix} u(t)$$

- 输出方程 $y(t) = 0.09882x_1(t) + 0.1976x_2(t)$
- 如何设计观测器重构系统的状态？

观测器的仿真

- 状态观测器设计
  - 期望极点 -1,-2,-3,-4

```
>> A=[0,2,0,0; 0,-0.1,8,0; 0,0,-10,16; 0,0,0,-20];
   B=[0;0;0;0.3953]; C=[0.09882,0.1976,0,0]; D=0;
   P=[-1; -2; -3; -4];
   L=place(A',C',P)'; [xh,x,t]=simobsv(ss(A,B,C,D),L);
   plot(t,x,t,xh,':'); axis([0,15,-0.5,4])
```

  - 期望极点： -10

```
>> P=[-10;-10;-10;-10]; L=acker(A',C',P)'; L'
   [xh,x,t]=simobsv(ss(A,B,C,D),L);
   plot(t,x,t,xh,':'); axis([0,30,-0.5,4])
```

薛定宇教授主讲

控制系统仿真与CAD