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

控制系统计算机辅助设计

第7章: 控制系统的经典设计方法

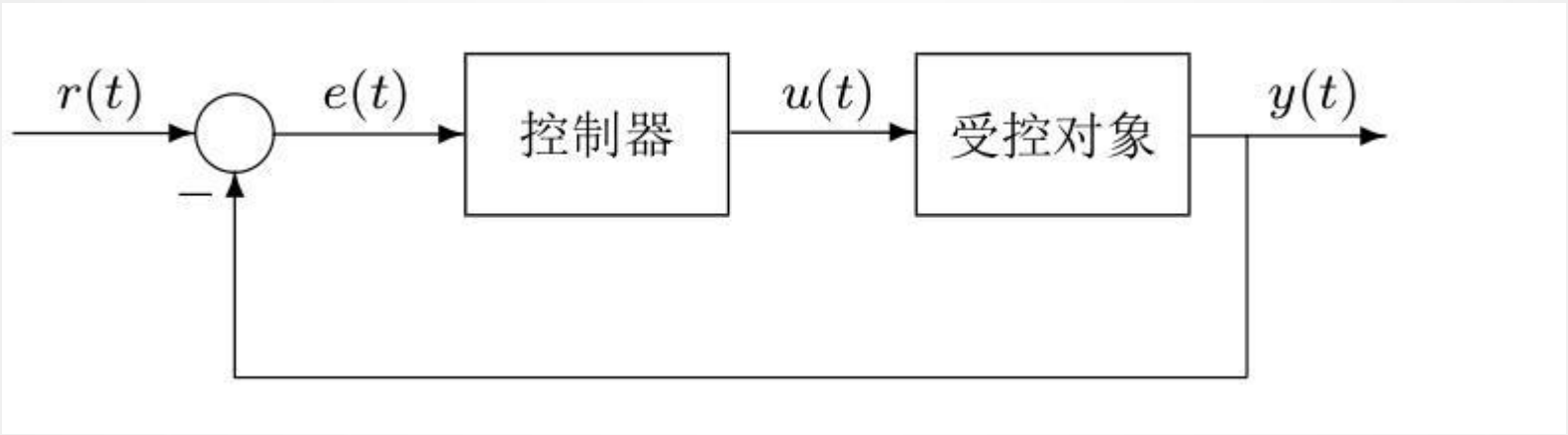
主讲：修贤超

控制系统的经典设计方法

- 7.1 超前滞后校正器设计方法
- 7.2 基于状态空间模型的控制器设计方法
- 7.3 最优控制器设计
- 7.5 多变量系统的频域设计方法
- 7.6 多变量系统的解耦控制

超前滞后控制器

- 典型反馈控制系统结构



- 超前滞后校正器的数学模型

$$G_c(s) = K\frac{(\alpha T_1 s + 1)(T_2 s + 1)}{(T_1 s + 1)(\beta T_2 s + 1)}$$

控制器设计的目标

- 一般伺服控制系统的要求
  - 超调量小，调节时间短，无静态误差
- 如何实现？
  - 如何实现超调量小的目标？—— 可以选择合适的相位裕度
  - 如何实现响应的快速性？—— 提高剪切频率的值
  - 如何实现无静态误差或小静态误差？
    - 引入积分器
    - 增加控制器的增益

超前滞后校正器的设计方法

- 基于剪切频率 ω_c 和相位裕度 γ 的设计

$$G_c(s) = \frac{K_c(s+z_{c_1})(s+z_{c_2})}{(s+p_{c_1})(s+p_{c_2})}$$

- 其中，$z_{c_1} \leqslant p_{c_1}$, $z_{c_2} \geqslant p_{c_2}$, K_c 为校正器的增益
- 设计规则：
  - 若 $\phi_c(\omega_c) > 0$，需要引入超前的校正器

$$\alpha = \frac{z_{c_1}}{p_{c_1}} = \frac{1-\sin\phi_c(\omega_c)}{1+\sin\phi_c(\omega_c)}$$

$$z_{c_1} = \sqrt{\alpha}\omega_c, \quad p_{c_1} = \frac{z_{c_1}}{p_{c_1}} = \frac{\omega_c}{\sqrt{\alpha}}, \quad K_c = \frac{\sqrt{\omega_c^2 + p_{c_1}^2}}{\sqrt{\omega_c^2 + z_{c_1}^2}\,A(\omega_c)}$$

超前滞后校正器的设计方法

- 系统静态误差系数为

$$K_1 = \lim_{s\to 0} s^v G_{\mathrm{o}}(s) = \frac{b_m}{a_{n-v}} \frac{K_{\mathrm{c}} z_{\mathrm{c}_1}}{p_{\mathrm{c}_1}}$$

- 超前滞后校正器进一步设计为

$$z_{\mathrm{c}_2} = \frac{\omega_{\mathrm{c}}}{10}, \quad p_{\mathrm{c}_2} = \frac{K_1 z_{\mathrm{c}_2}}{K_{\mathrm{v}}}$$

- 若 $\phi_{\mathrm{c}}(\omega_{\mathrm{c}}) < 0$，则需要滞后校正器

$$K_1 = b_m \frac{K_{\mathrm{c}}}{a_{n-v}}, \; K_{\mathrm{c}} = \frac{1}{A(\omega_{\mathrm{c}})}, \; z_{\mathrm{c}_2} = \frac{\omega_{\mathrm{c}}}{10}, \; p_{\mathrm{c}_2} = \frac{K_1 z_{\mathrm{c}_2}}{K_{\mathrm{v}}}$$

超前滞后校正器设计函数

- 调用格式 G_{c}=leadlagc(G,ω_{c},γ,K_{v},key)
- 清单

```
function Gc=leadlagc(G,Wc,Gam_c,Kv,key)
G=tf(G); [Gai,Pha]=bode(G,Wc);
Phi_c=sin((Gam_c-Pha-180)*pi/180);
den=G.den{1}; a=den(length(den):-1:1);
ii=find(abs(a)<=0); num=G.num{1}; G_n=num(end);
if length(ii)>0, a=a(ii(1)+1); else, a=a(1); end;
alpha=sqrt((1-Phi_c)/(1+Phi_c)); Zc=alpha*Wc;
Pc=Wc/alpha; Kc=sqrt((Wc*Wc+Pc*Pc)/(Wc*Wc+Zc*Zc))/Gai;
K1=G_n*Kc*alpha/a;
```

超前滞后校正器设计函数

```
if nargin==4, key=1;
   if Phi_c<0, key=2; else, if K1<Kv, key=3; end, end
end
switch key
case 1, Gc=tf([1 Zc]*Kc,[1 Pc]);
case 2
   Kc=1/Gai; K1=G_n*Kc/a; Gc=tf([1 0.1*Wc],[1 K1*Gcn(2)/Kv]);
case 3
   Zc2=Wc*0.1; Pc2=K1*Zc2/Kv; Gcn=Kc*conv([1 Zc],[1,Zc2]);
   Gcd=conv([1 Pc],[1,Pc2]); Gc=tf(Gcn,Gcd);
end
```

例7-1 超前滞后校正器设计

- 受控对象模型 $G(s)=\dfrac{10(s+1)(s+0.5)}{s(s+0.1)(s+2)(s+10)(s+20)}$
- 选择 ω_c =20 rad/sec，相位裕度选择不同的值

```
>> s=tf('s'); G=4*(s+1)*(s+0.5)/s/(s+0.1)/(s+2)/(s+10)/(s+20)
   wc=20; f1=figure; f2=figure;
   for gam=20:10:90
      Gc=leadlagc(G,wc,gam,1000,3);
      figure(f1); step(feedback(G*Gc,1),1); hold on
      figure(f2); bode(Gc*G); hold on;
   end
```

控制器设计结果

- 选择 $\omega_c = 20\text{rad/s},\ \ \gamma = 60°$

```
>> Gc1=zpk(leadlagc(G,20,60,1000,3))
   Gc2=zpk(leadlagc(G,20,60,1000,1))
   step(feedback(G*Gc1,1),feedback(G*Gc2,1),':',1)
```

- 超前滞后校正器

$$G_{c_1}(s) = \frac{27283.5668(s+2.326)(s+2)}{(s+172)(s+0.3173)}$$

- 超前校正器

$$G_{c_2}(s) = \frac{27283.5668(s+2.326)}{(s+172)}$$

不同剪切频率的设计

- 选择相位裕度为 60，不同剪切频率

```
>> gam=60; f1=figure; f2=figure;
   for wc=5:5:30
      Gc=leadlagc(G,wc,gam,1000,3); [a,b,c,d]=margin(Gc*G);
      figure(f1); step(feedback(G*Gc,1),3);
      figure(f2); bode(Gc*G); hold on;
   end
```

- 很多组合由于要求过高，达不到
  - 设计之后应该检验

控制信号曲线绘制

- 设计时未考虑控制信号大小
  - $\omega_c = 30\text{rad/s}$

```
>> Gc=leadlagc(G,30,60,1000,3);
   y=step(feedback(Gc,G)); max(y)
```

  - $\gamma = 60^\circ, \omega_c = 100\text{rad/s}$

```
>> gam=60; wc=100; Gc=leadlagc(G,wc,gam,1000,3);
   step(feedback(G*Gc,1),0.5);
```

  - $\omega_c = 1000\text{rad/s}$

```
>> Gc1=leadlagc(G,10*wc,gam,1000,3);
   bode(Gc*G,Gc1*G)
```

控制系统的经典设计方法

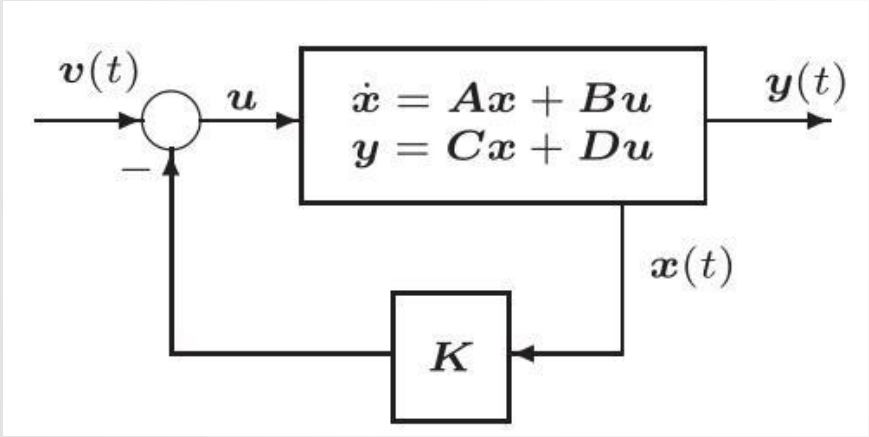
- 7.1 超前滞后校正器设计方法
- 7.2 基于状态空间模型的控制器设计方法
- 7.3 最优控制器设计
- 7.5 多变量系统的频域设计方法
- 7.6 多变量系统的解耦控制

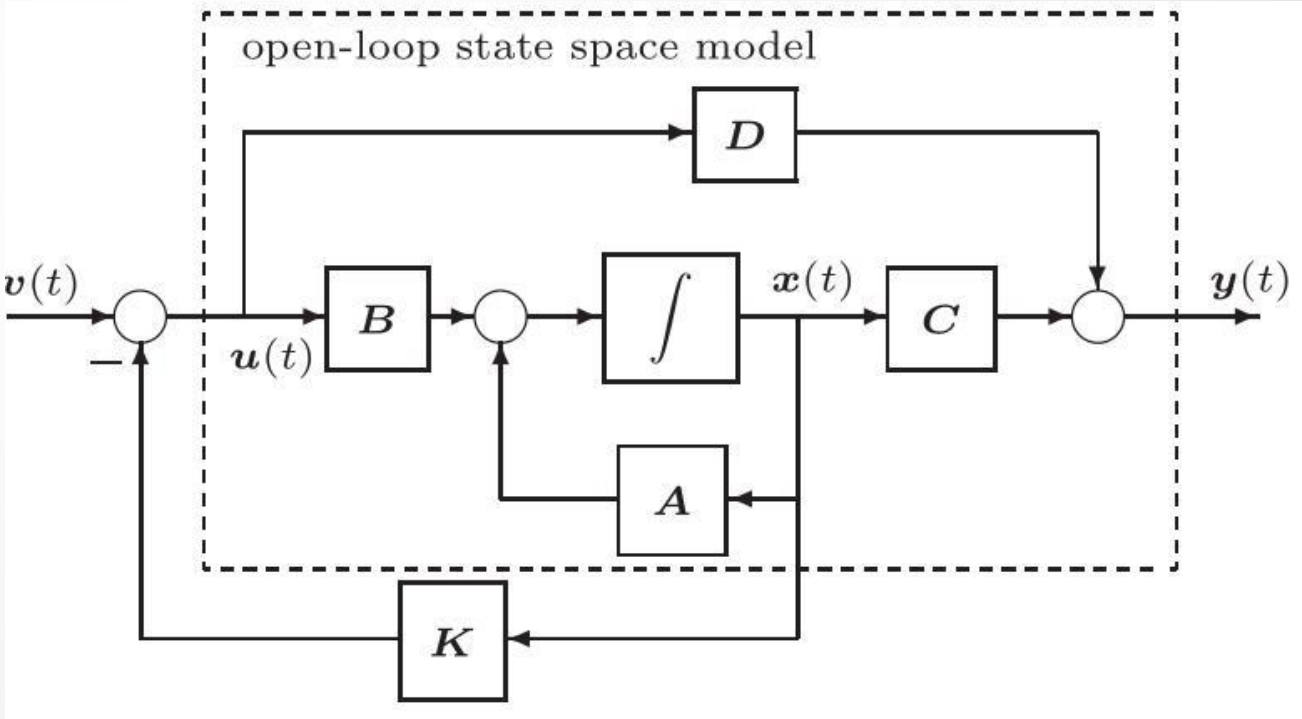
基于状态空间模型的控制器设计方法

- 现代控制理论——状态空间方法
- 本节主要内容
  - 状态反馈系统内部和外部结构
  - 线性二次型最优调节器设计
  - 极点配置控制器设计
  - 观测器及观测器设计
  - 基于观测器的调节器与控制器设计

状态反馈控制结构

- 状态反馈的外部模型与内部模型
- 状态方程 $\boldsymbol{u}(t) = \boldsymbol{v}(t) - \boldsymbol{K}\boldsymbol{x}(t)$





状态反馈理论基础

- 状态反馈与输入信号
- 状态反馈的闭环模型 $\boldsymbol{u}(t)=\boldsymbol{v}(t)-\boldsymbol{K}\boldsymbol{x}(t)$

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

- 如果系统完全可控，则可以将闭环模型 $(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{K})$ 的极点配置到任意指定的位置
  - 当然需要满足：实数或共轭复数

线性二次型指标最优调节器

- 受控对象——状态方程模型

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

- 设计目标 $\boldsymbol{u}(t)$
- 找出最优输入
  - 使得目标函数最小化

$$J = \frac{1}{2}\boldsymbol{x}^T(t_\mathrm{f})\boldsymbol{S}\boldsymbol{x}(t_\mathrm{f}) + \frac{1}{2}\int_{t_0}^{t_\mathrm{f}} \left[\boldsymbol{x}^T(t)\boldsymbol{Q}(t)\boldsymbol{x}(t) + \boldsymbol{u}^T(t)\boldsymbol{R}(t)\boldsymbol{u}(t)\right]\mathrm{d}t$$

线性二次型最优调节器

- 最优控制信号 $\boldsymbol{u}^*(t) = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}(t)\boldsymbol{x}(t)$
- Riccati 微分方程 $\boldsymbol{P}(t_\mathrm{f}) = \boldsymbol{S}$

$$\dot{\boldsymbol{P}}(t) = -\boldsymbol{P}(t)\boldsymbol{A} - \boldsymbol{A}^\mathrm{T}\boldsymbol{P}(t) + \boldsymbol{P}(t)\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\mathrm{T}\boldsymbol{P}(t) - \boldsymbol{Q}$$

- 微分方程不能求解，简化为代数方程

$$\boldsymbol{P}\boldsymbol{A} + \boldsymbol{A}^\mathrm{T}\boldsymbol{P} - \boldsymbol{P}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^\mathrm{T}\boldsymbol{P} + \boldsymbol{Q} = \boldsymbol{0}$$

- MATLAB求解 $\boldsymbol{K} = -\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{P}$

$$[\boldsymbol{K},\boldsymbol{P}]=\mathtt{lqr}(\boldsymbol{A},\boldsymbol{B},\boldsymbol{Q},\boldsymbol{R})$$

离散系统的二次型最优调节器

- 二次型性能指标

$$J=\frac{1}{2}\sum_{k=0}^{N}\left[\boldsymbol{x}^{T}(k)\boldsymbol{Q}\boldsymbol{x}(k)+\boldsymbol{u}^{T}(k)\boldsymbol{R}\boldsymbol{u}(k)\right]$$

- 离散Riccati代数方程

$$\boldsymbol{S}=\boldsymbol{F}^{T}\left[\boldsymbol{S}-\boldsymbol{S}\boldsymbol{G}\boldsymbol{R}^{-1}\boldsymbol{G}^{T}\boldsymbol{S}\right]\boldsymbol{F}+\boldsymbol{Q}$$

- 控制律 $\boldsymbol{K}=\left[\boldsymbol{R}+\boldsymbol{G}^{T}\boldsymbol{S}\boldsymbol{G}\right]^{-1}\boldsymbol{B}^{T}\boldsymbol{S}\boldsymbol{F}$

- MATLAB求解 $[\boldsymbol{K},\boldsymbol{S}]=\mathtt{dlqr}(\boldsymbol{F},\boldsymbol{G},\boldsymbol{Q},\boldsymbol{R})$

例7-2 状态方程模型

- 状态方程受控对象

$$A=\begin{bmatrix}2&0&4&1&2\\1&-2&-4&0&1\\1&4&3&0&2\\2&-2&2&3&3\\1&4&6&2&1\end{bmatrix},\quad B=\begin{bmatrix}1&2\\0&1\\0&0\\0&0\\0&0\end{bmatrix}$$

$$C=[1,0,0,0,0]$$

- 加权矩阵 $Q=\mathrm{diag}(1000,0,1000,500,500)$, $R=I_2$
- 最优调节器设计

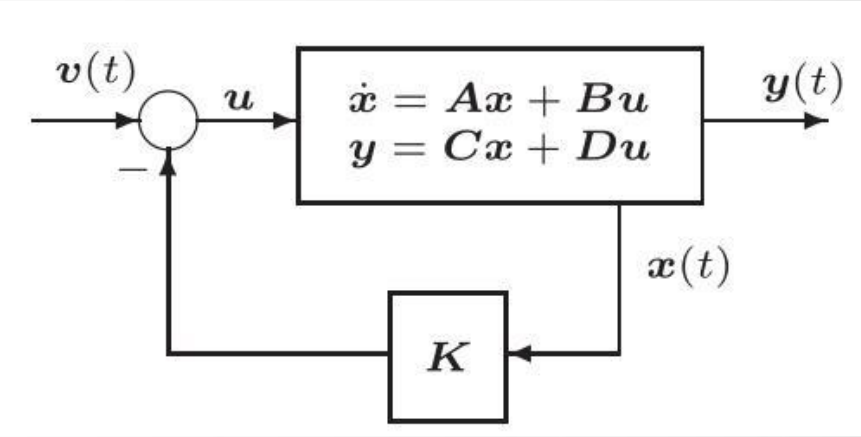
```
>> A=[2,0,4,1,2; 1,-2,-4,0,1; 1,4,3,0,2; 2,-2,2,3,3; 1,4,6,2,1];
   B=[1,2; 0,1; 0,0; 0,0; 0,0]; C=[1,0,0,0,0];
   Q=diag([1000 0 1000 500 500]); R=eye(2); [K,S]=lqr(A,B,Q,R)
```

```
>> G=ss(A-B*K,B,C,0); step(G)
```

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　　K=bass_pp(A,B,p)
  - Ackermann 算法　　K=acker(A,B,p)
  - 鲁棒配置算法　　K=place(A,B,p)

Bass-Gura 算法

- 开环特征方程

$$a(s)=\det(s\boldsymbol{I}-\boldsymbol{A})=s^n+a_1s^{n-1}+a_2s^{n-2}+\cdots+a_{n-1}s+a_n$$

- 反馈向量

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

$$\boldsymbol{\Gamma}=\begin{bmatrix} a_{n-1} & a_{n-2} & \cdots & a_1 & 1 \\ a_{n-2} & a_{n-3} & \cdots & 1 & \\ \vdots & \vdots & \ddots & & \\ a_1 & 1 & & & \\ 1 & & & & \end{bmatrix}$$

$$\boldsymbol{K}=\boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{T}_{\mathrm{c}}^{-1}$$

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

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

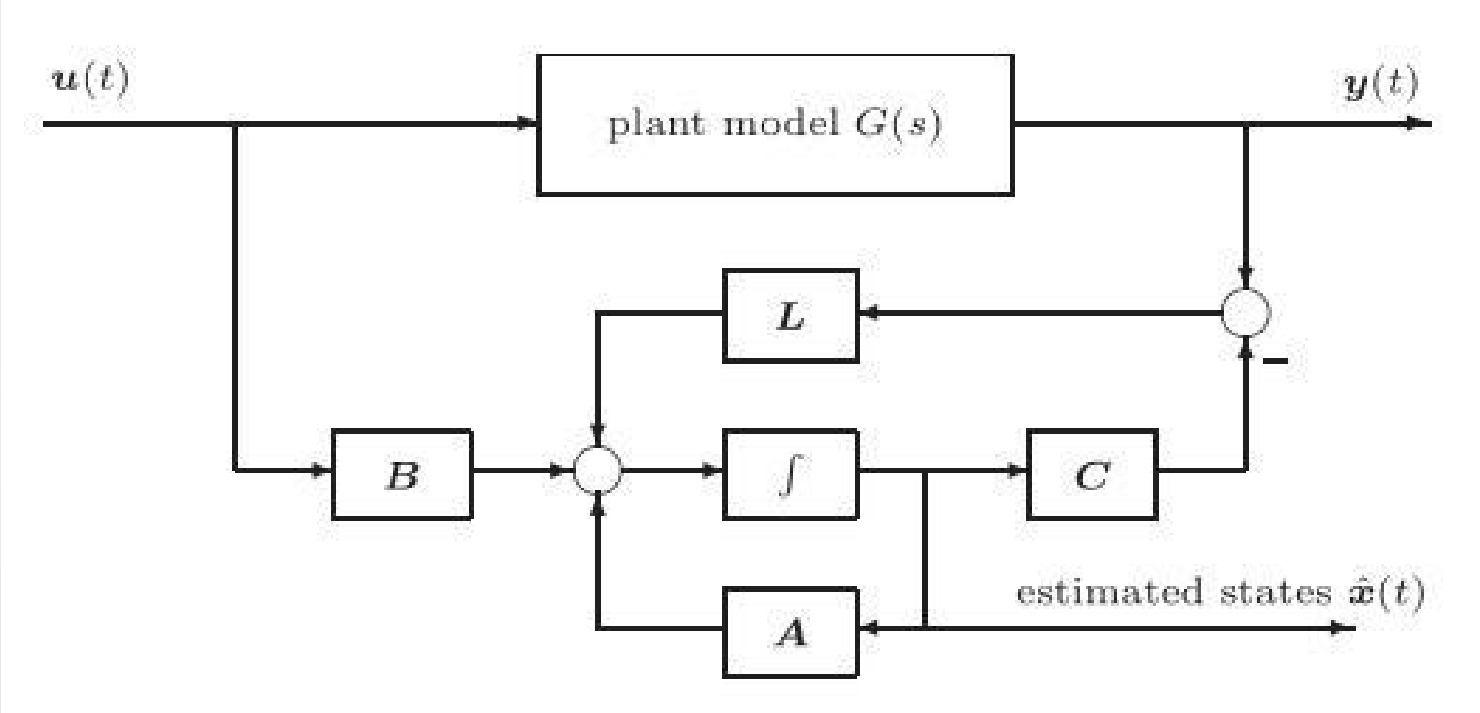
- 期望极点 $-0.1, -0.2, -0.5 \pm 0.2\mathrm{j}$
- 控制器设计 (不能配置)

```
>> 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 $\lim\limits_{t \to \infty}[\hat{\boldsymbol{x}}(t) - \boldsymbol{x}(t)] = \boldsymbol{0}$

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

$[\hat{x},x,t]$=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

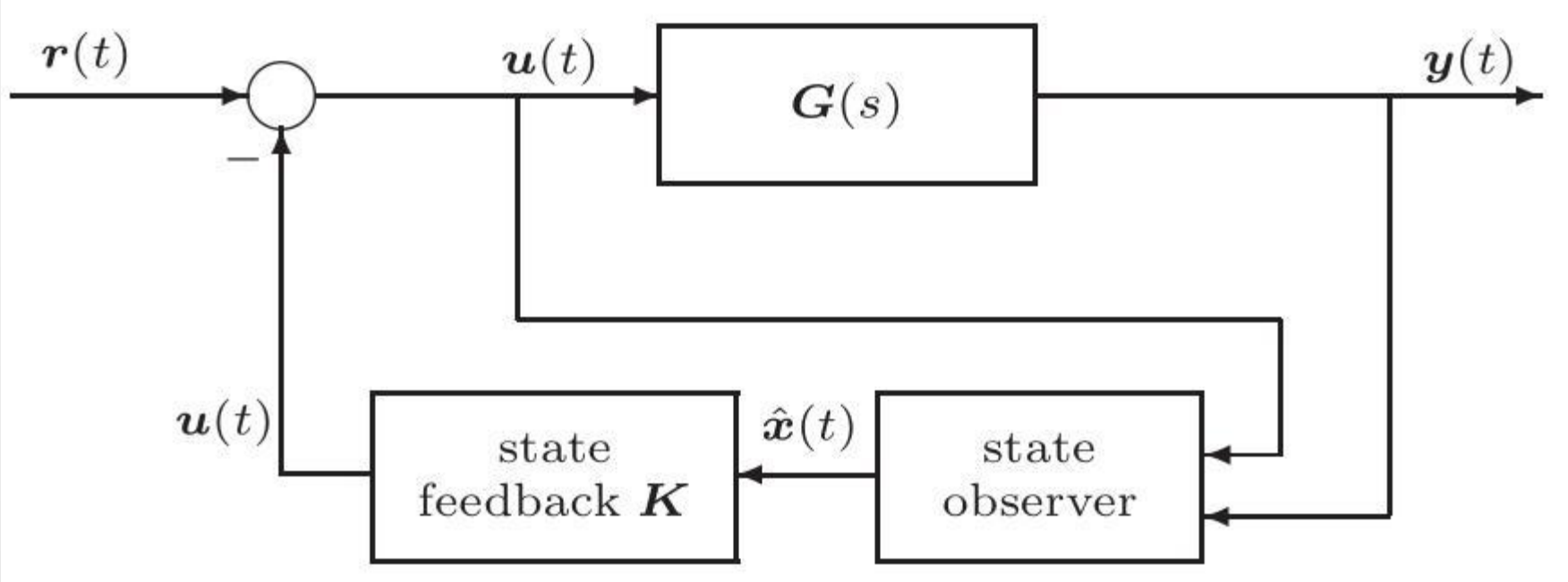
```
>> A=[0,2,0,0; 0,-0.1,8,0; 0,0,-10,16; 0,0,0,-20];
   B=[0;0;0;0.3953]; C=[0.09882,0.1976,0,0]; D=0;
   P=[-1; -2; -3; -4];
   L=place(A',C',P)'; [xh,x,t]=simobsv(ss(A,B,C,D),L);
   plot(t,x,t,xh,':'); axis([0,15,-0.5,4])
```

  - 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])
```

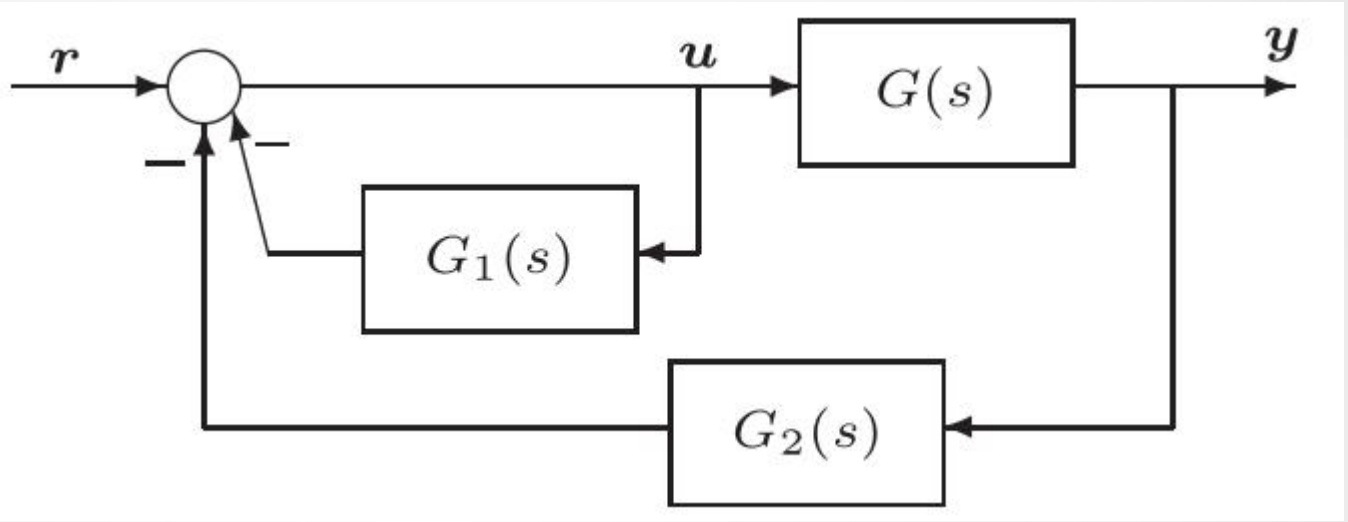
基于观测器的调节器与控制器

- 带有观测器的控制系统框图
  - 状态未知，需要观测器重建



基于状态观测器的调节器

- 模型等效变换



- $G_1(s)$

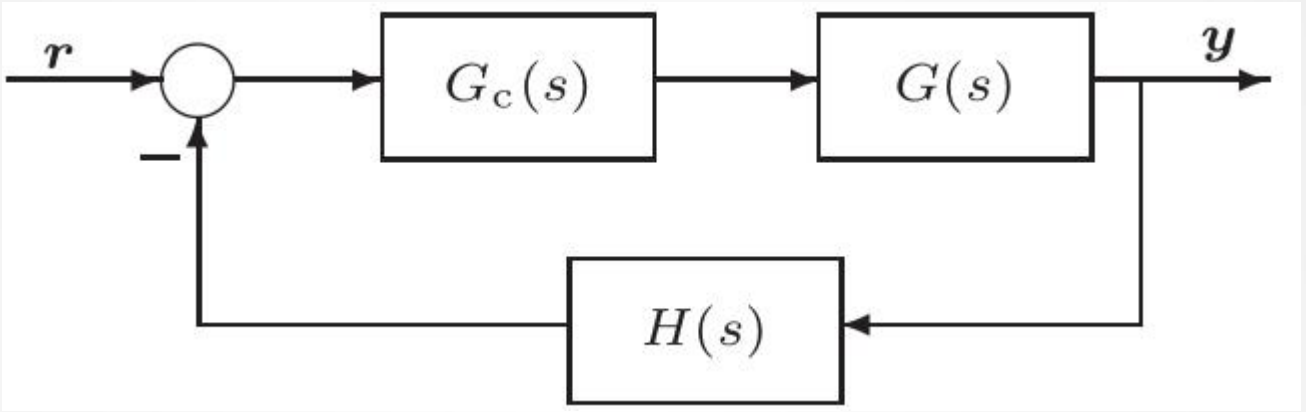
$$\begin{cases} \dot{\hat{x}}_1(t) = (\boldsymbol{A} - \boldsymbol{LC})\hat{x}_1(t) + (\boldsymbol{B} - \boldsymbol{LD})\boldsymbol{u}(t) \\ \boldsymbol{y}_1(t) = \boldsymbol{K}\hat{x}_1(t) \end{cases}$$

- $G_2(s)$

$$\begin{cases} \dot{\hat{x}}_2(t) = (\boldsymbol{A} - \boldsymbol{LC})\hat{x}_2(t) + \boldsymbol{Ly}(t) \\ \boldsymbol{y}_2(t) = \boldsymbol{K}\hat{x}_2(t) \end{cases}$$

结构图化成典型反馈系统模型

- 等效框图



- 等效关系
  - $G_c(s)$

$$\boldsymbol{G_c(s) = I - K(sI - A + BK + LC - LDK)^{-1}B}$$

  - 状态方程

$$\begin{cases} \dot{\boldsymbol{x}}(t) = (\boldsymbol{A - BK - LC + LDK})\boldsymbol{x}(t) + \boldsymbol{Bu}(t) \\ \boldsymbol{y}(t) = -\boldsymbol{Kx}(t) + \boldsymbol{u}(t) \end{cases}$$

基于观测器的控制器等效变换

- 等效关系

$$\begin{cases} \dot{\boldsymbol{x}}(t) = (\boldsymbol{A} - \boldsymbol{BK} - \boldsymbol{LC} + \boldsymbol{LDK})\boldsymbol{x}(t) + \boldsymbol{Bu}(t) \\ \boldsymbol{y}(t) = -\boldsymbol{K}\boldsymbol{x}(t) + \boldsymbol{u}(t) \end{cases}$$

$$\begin{cases} \dot{\hat{\boldsymbol{x}}}_2(t) = (\boldsymbol{A} - \boldsymbol{LC})\hat{\boldsymbol{x}}_2(t) + \boldsymbol{Ly}(t) \\ \boldsymbol{y}_2(t) = \boldsymbol{K}\hat{\boldsymbol{x}}_2(t) \end{cases}$$

- 根据等效关系可以编写出MATLAB函数

```
function [Gc,H]=obsvsf(G,K,L)
H=ss(G.a-L*G.c,L,K,0);
Gc=ss(G.a-G.b*K-L*G.c+L*G.d*K,G.b,-K,1);
```

- 若参考输入为0（调节问题） G_c=reg(G,K,L)

例7-6 系统的等效实现

- 受控对象模型

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

$$y(t)=0.09882x_1(t)+0.1976x_2(t)$$

- 二次型性能指标的加权矩阵

$$R=1,\ \ \boldsymbol{Q}=\mathrm{diag}(0.01,0.01,2,3)$$

状态反馈控制系统的仿真

- 直接设计控制器

```
>> 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;
   Q=diag([0.01,0.01,2,3]); R=1;
   K=lqr(A,B,Q,R), step(ss(A-B*K,B,C,D))
```

- 假设状态不可测，需要构造观测器

```
>> P=[-5;-5;-5;-5]; G=ss(A,B,C,D);
   L=acker(A',C', P)'; [Gc,H]=obsvsf(G,K,L);
   step(ss(A-B*K,B,C,D),feedback(G*Gc,H))
```

- 最小实现

```
>> zpk(minreal(feedback(G*Gc,H)))
   zpk(minreal(ss(A-B*K,B,C,D)))
```

超前滞后校正器设计小结

- 串联控制的一般结构
- 超前、滞后、超前滞后校正器简介
- 超前滞后校正器的设计
  - 给出期望剪切频率和相位裕度
  - 利用MATLAB函数直接设计 leadlagc()
  - 设计出的控制器应该检验一下，是不是预期的指标能够达到，如果不能达到应该修正预期的指标重新设计

状态空间设计方法小结

- 状态反馈系统的结构
- 状态空间设计方法
  - 线性二次型性能指标的最优设计 lqr()、dlqr()
  - 极点配置的： bass_pp()、place()、acker()函数
- 系统状态不可测时，引入观测器重建状态
  - 状态观测器仿真：simobsv() 函数
  - 基于观测器的控制器：obsvf() 函数
  - 基于观测器的调节器：reg() 函数

Q & A

感谢您的聆听和反馈

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

控制系统计算机辅助设计

第7章: 控制系统的经典设计方法

主讲：修贤超

控制系统的经典设计方法

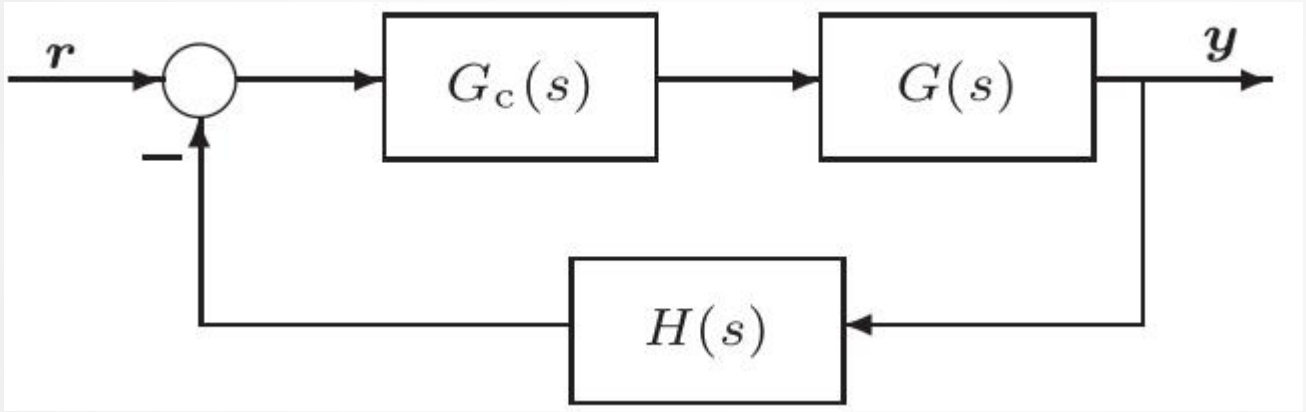
- 7.1 超前滞后校正器设计方法
- 7.2 基于状态空间模型的控制器设计方法
- 7.3 最优控制器设计
- 7.5 多变量系统的频域设计方法
- 7.6 多变量系统的解耦控制

最优控制器设计

- LQR等最优控制存在的问题
  - Q, R 选择认为因素很大，缺乏客观描述
  - LQR 类控制的目标在于解析求解，如果某个问题不可求解，则试图简化问题。LQR问题已经远远偏离实际的要求
- 本节内容
  - 伺服控制选择什么样的性能指标合适
  - 如何利用数值最优化技术设计最优控制器

伺服控制系统的要求

- 伺服控制框图



- 假如 $H(s)=1$，则控制目标是让输出尽快跟踪输入
- 超调量小，调节时间短，跟踪的总体误差小
- 系统中如果有非线性，传统设计方法难以应用

最优控制的概念

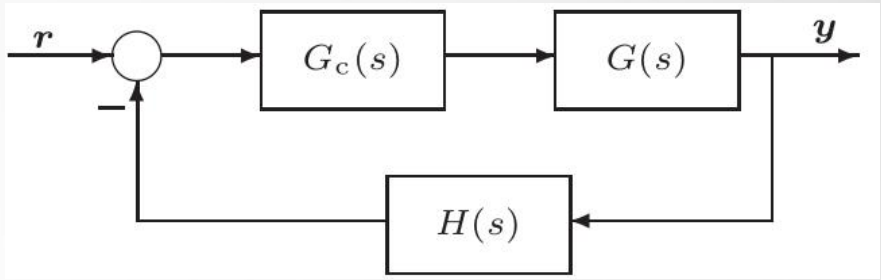
- 所谓“最优控制”，就是在一定的具体条件下，要完成某个控制任务，使得选定指标最小或最大的控制，这里所谓指标就是目标函数。
- 有意义的性能指标
  - 动态误差信号的积分指标

$$J_{\mathrm{ISE}}=\int_0^\infty e^2(t)\mathrm{d}t,\ J_{\mathrm{IAE}}=\int_0^\infty |e(t)|\mathrm{d}t,\ J_{\mathrm{ITAE}}=\int_0^\infty t|e(t)|\mathrm{d}t$$

  - 速度最快、能量最省等

例7-7 系统的最优PID控制器设计

- 复杂的受控对象 $G(s)=\dfrac{1+\dfrac{3\mathrm{e}^{-s}}{s+1}}{s+1}$



- 误差传递函数的计算

$$E(s)=\frac{1}{1+G(s)G_c(s)}R(s)$$

- PID控制器的描述

$$G_c(s)=K_p+K_i\frac{1}{s}+K_ds \qquad \boldsymbol{x}=[K_p,K_i,K_d]$$

控制器的最优设计

- 目标函数的编写 $J_{\mathrm{ISE}}=\int_0^{\infty}e^2(t)\mathrm{d}t$

```
function y=c7fopt(x,s,G,t)
Gc=x(1)+x(2)/s+x(3)*s; E=1/(1+G*Gc);
y0=step(E,t); y=sum(y0.^2)*(t(2)-t(1));
```

- 最优化求解

```
>> s=tf('s'); G1=3/(s+1); G1.ioDelay=3; G=(1+G1)/(s+1);
   t=0:0.02:30; x=fminunc(@c7fopt,[1,1,1],optimset,s,G,t)
```

```
>> Gc=x(1)+x(2)/s+x(3)*s;
   step(feedback(G*Gc,1),30)
```

ITAE 性能指标下的最优控制器设计

- ITAE 性能指标的目标函数描述 $J_{\mathrm{ITAE}} = \int_0^{\infty} t|e(t)|\mathrm{d}t$

```
function y=c7fopt2(x,s,G,t)
Gc=x(1)+x(2)/s+x(3)*s; E=1/(1+G*Gc);
y0=step(E,t); y=t*abs(y0)*(t(2)-t(1));
```

- 最优控制器设计

```
>> x=fminunc(@c7fopt2,[1,1,1],optimset,s,G,t)
```

```
>> Gc=x(1)+x(2)/s+x(3)*s;
   step(feedback(G*Gc,1),30)
```

哪种性能指标更合理？

- 不同性能指标得出的控制效果不同，但都是最优解
  - 哪种性能指标是最好的呢？
  - 哪种是最客观的呢？
- ISE同等处理各个时刻的误差
- ITAE对时间加权，时间 t 大，会迫使误差降下来
  - 相比之下，更适合于伺服控制
  - ISE 性能指标可以通过范数计算，可以不经过仿真
  - 现在有了强大的仿真工具，没有必要再依赖ISE指标

这里设计方法的局限性

- 受控对象和控制器必须是线性的
  - 否则不能用step函数进行仿真
  - 实际应用中，可能PID控制器输出过大，需要接饱和非线性
- 应该如何实现有意义的设计
- 充分利用MATLAB的强大功能
  - 用数值最优化语句设计控制器参数
  - 用Simulink的强大仿真功能去处理任意复杂系统的仿真

基于Simulink的最优控制器设计步骤

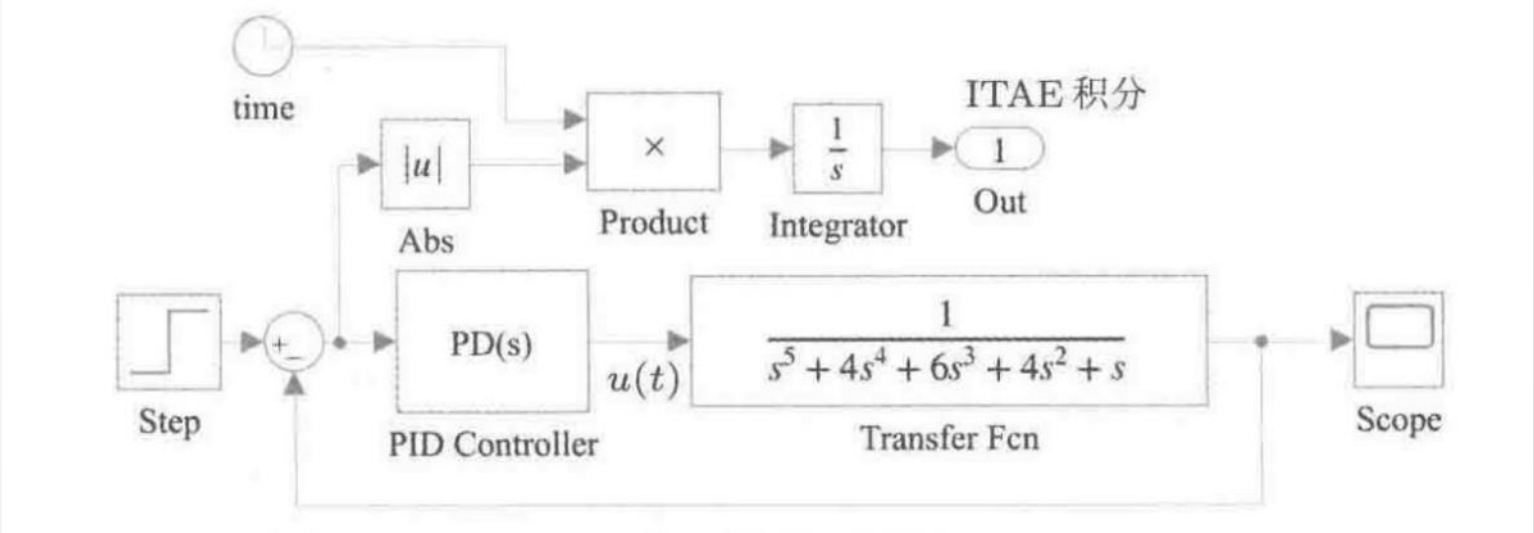
- 将需要设计的控制系统模型
  - 用Simulink绘制出来，并将决策变量用变量表示
  - 在Simulink模型中定义目标函数
  - 用MATLAB写出目标函数文件，用assignin()函数将Simulink中的控制器变量与目标函数建立起来联系
- 调用最优化问题求解程序直接设计控制器
  - 无约束最优化求解：fminsearch()、fminunc()
  - 有约束最优化，定义约束：fmincon()

例7-8 用基于Simulink的方法设计

- 受控对象模型

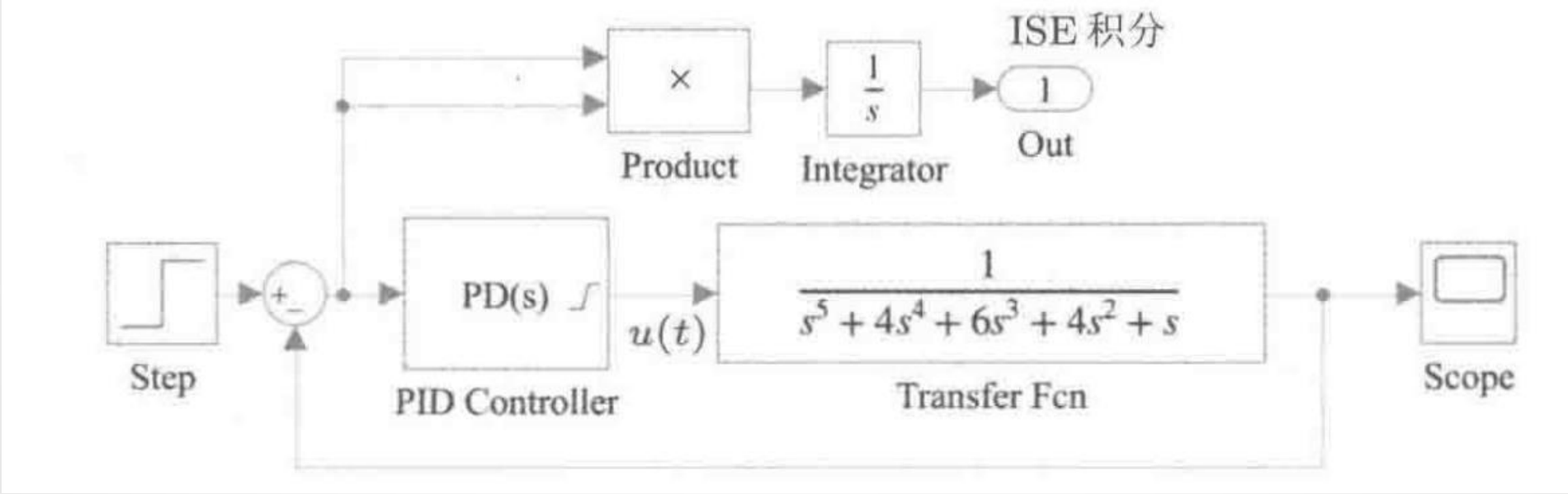
$$G(s) = 1/[s(s+1)^4]$$

- 绘制Simulink框图
  - 模型 c7moptm1
  - 工作空间变量
    - Kp, Kd
  - 快速重启



例7-9 另一个控制器设计问题

- 性能指标ISE



控制系统的经典设计方法

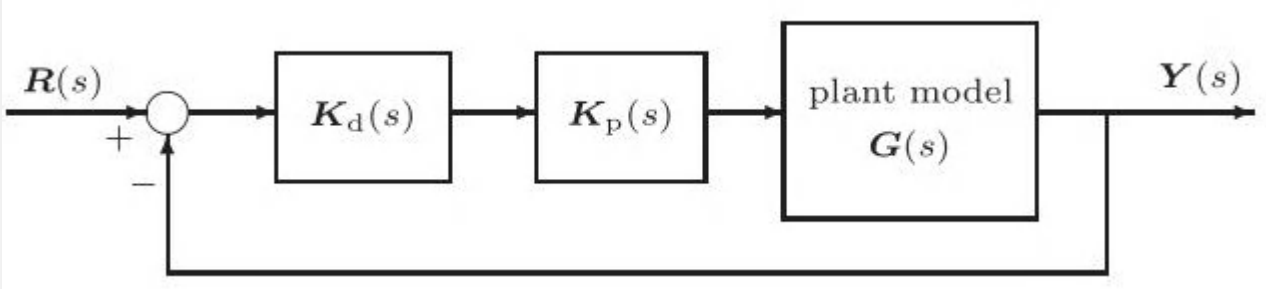
- 7.1 超前滞后校正器设计方法
- 7.2 基于状态空间模型的控制器设计方法
- 7.3 最优控制器设计
- 7.5 多变量系统的频域设计方法
- 7.6 多变量系统的解耦控制

多变量系统的频域设计方法

- 多变量系统比单变量系统设计难很多
  - 主要难点：系统不同输入输出对之间的耦合，难点在如何解耦？
  - 解耦后可以单独回路设计，沿用单变量的设计方式，给每回路单独设计控制器，不影响其他
- 本节主要内容
  - 对角占优化方法与伪对角化方法
  - 参数最优化方法

对角占优系统与伪对角化

- 典型多变量反馈系统框图



- $\boldsymbol{K}_{\mathrm{p}}(s)$ 预补偿矩阵，使 $\boldsymbol{G}(s)\,\boldsymbol{K}_{\mathrm{p}}(s)$为对角占优矩阵
- $\boldsymbol{K}_{\mathrm{d}}(s)$ 为动态补偿矩阵
- $\boldsymbol{K}_{\mathrm{p}}(s)$ 可以用试凑方法，可以取 $\boldsymbol{K}_{\mathrm{p}}(s)=\boldsymbol{G}^{-1}(0)$
- 还可以采用其他方法，如伪对角化方法

伪对角化方法

- 在 $\mathrm{j}\omega_0$ 点处的逆Nyquist阵列（INA）为

$$\hat{g}_{ik}(\mathrm{j}\omega_0) = \alpha_{ik} + \mathrm{j}\beta_{ik},\ i,k = 1,2,\cdots,m$$

- 假定输入、输出路数相同。步骤：
  - 选择一个频率点 $\mathrm{j}\omega_0$，求出 INA $\hat{g}_{ik}(\mathrm{j}\omega_0)$
  - 对各个 q，构造

$$a_{il,q} = \sum_{k=1\ \ k\neq q}^{m} \left[\alpha_{ik}\alpha_{lk} + \beta_{ik}\beta_{lk}\right],\quad i,l = 1,2,\cdots,m$$

  - 求 $\boldsymbol{A}_q$ 矩阵的特征向量，最小特征值的向量 $\boldsymbol{k}_q$
  - 由各个 q 对应的最小特征向量构造补偿矩阵

多点伪对角化

- 上述算法的 $\mathrm{j}\omega_0$ 可以扩展到一系列频率
  - 频率点 $\omega_1, \omega_2, \cdots, \omega_N$ ψ_r
  - 对第 r 频率点加权 ψ_{r}，构造

$$A_{il,q} = \sum_{r=1}^{N} \psi_r \left[\sum_{k=1 \text{ and } k \neq q}^{m} (\alpha_{ik,r}\alpha_{lk,r} + \beta_{ik,r}\beta_{lk,r}) \right]$$

- 基本思想：在某几个频率点实现对角化
- 编写MATLAB函数生成补偿矩阵
  - 调用格式 $\boldsymbol{K}_{\mathrm{p}}=\mathtt{psuediag}(\boldsymbol{G},\boldsymbol{R})$

伪对角化算法的实现

- 调用格式 $\boldsymbol{K}_{\mathrm{p}} = \mathrm{psuediag}(\boldsymbol{G}, \boldsymbol{R})$

```
function Kp=psuediag(G1,R)
A=real(G1); B=imag(G1); [n,m]=size(G1); N=n/m; Kp=[];
if nargin==1, R=ones(N,1); end
for q=1:m, L=[1:q-1, q+1:m];
   for i=1:m, for l=1:m, a=0;
      for r=1:N, k=(r-1)*m;
         a=a+R(r)*sum(A(k+i,L).*A(k+l,L)+B(k+i,L).*B(k+l,L));
      end,
      Ap(i,l)=a;
   end, end
   [x,d]=eig(Ap); [xm,ii]=min(diag(d)); Kp=[Kp; x(:, ii)'];
end
```

例7-15 多变量问题伪对角化

- 蒸汽锅炉温度控制模型

$$G(s)=\begin{bmatrix} 1/(1+4s) & 0.7/(1+5s) & 0.3/(1+5s) & 0.2/(1+5s) \\ 0.6/(1+5s) & 1/(1+4s) & 0.4/(1+5s) & 0.35/(1+5s) \\ 0.35/(1+5s) & 0.4/(1+5s) & 1/(1+4s) & 0.6/(1+5s) \\ 0.2/(1+5s) & 0.3/(1+5s) & 0.7/(1+5s) & 1/(1+4s) \end{bmatrix}$$

- 原系统

```
>> s=tf('s'); w=logspace(-1,0);
   G=[1/(1+4*s), 0.7/(1+5*s), 0.3/(1+5*s), 0.2/(1+5*s);
      0.6/(1+5*s), 1/(1+4*s), 0.4/(1+5*s), 0.35/(1+5*s);
      0.35/(1+5*s), 0.4/(1+5*s), 1/(1+4*s), 0.6/(1+5*s);
      0.2/(1+5*s),  0.3/(1+5*s), 0.7/(1+5*s), 1/(1+4*s)];
    H=mfrd(G,w); gershgorin(H)
```

伪对角化

- 选定预校正矩阵 $\boldsymbol{K} = \boldsymbol{G}^{-1}(0)$

```
>> K=inv(mfrd(G,0)); W=mfrd(G*K,w); gershgorin(W)
```

$\omega = 0.9$ rad/s

- 选频率点

```
>> v=0.9; iH=mfrd(inv(G),v); Kp=inv(pseudiag(iH)),
   V=mfrd(G*Kp,w); gershgorin(V)
```

- 选一组频率点 $\omega \in (0.01, 0.4)$

```
>> v=logspace(-2,log10(0.4)); iH=mfrd(inv(G),v);
   Kp=inv(pseudiag(iH)), Q=mfrd(G*Kp,w);
   gershgorin(Q)
```

例7-16 多变量延迟系统

- 系统模型

$$G(s)=\begin{bmatrix}\dfrac{0.1134\mathrm{e}^{-0.72s}}{1.78s^2+4.48s+1} & \dfrac{0.924}{2.07s+1}\\ \dfrac{0.3378\mathrm{e}^{-0.3s}}{0.361s^2+1.09s+1} & \dfrac{-0.318\mathrm{e}^{-1.29s}}{2.93s+1}\end{bmatrix}$$

- 系统不是对角占优系统
- 引入补偿器 $\boldsymbol{K}=\boldsymbol{G}^{-1}(0)$

```
>> G=[tf(0.1134,[1.78 4.48 1]), tf([0.924],[2.07,1]);
     tf(0.3378,[0.361,1.09,1]), tf(-0.318,[2.93 1])];
G1=G; G1.ioDelay=[0.72 0; 0.3 1.29];
w=logspace(0,1); Kp1=inv(mfrd(G,0)); Kp2=inv([1 0; 0.5 1]);
H3=mfrd(G1*Kp1*Kp2,w); gershgorin(H3)
```

其他补偿矩阵

- 引入补偿矩阵

$$\boldsymbol{K}_{\mathrm{d}}(s)=\begin{bmatrix}1 & 0\\ 0 & (0.3s+1)/(0.05s+1)\end{bmatrix}$$

$$\boldsymbol{K}_{\mathrm{d}}^{-1}(s)=\begin{bmatrix}1 & 0\\ 0 & (0.05s+1)/(0.3s+1)\end{bmatrix}$$

- INA绘制

```
>> s=tf('s'); Kd=[1 0; 0 (0.3*s+1)/(0.05*s+1)];
   gershgorin(mfrd(G1*Kp1*Kp2*Kd,w))
```

- 闭环系统的阶跃响应

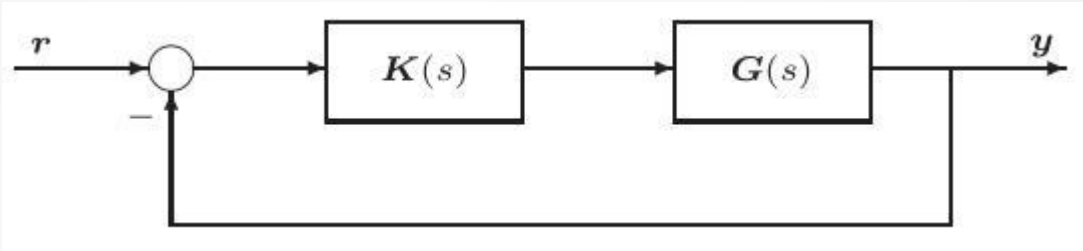
```
>> step(feedback(ss(G1)*Kp1*Kp2*Kd,eye(2)),15)
```

多变量系统的参数最优化设计

- 前面介绍的方法不能完全解耦
- 利用数值最优化技术设计解耦控制器
- 主要参考文献：
  - John Edmunds，Control system design and analysis using closed-loop Nyquist and Bode arrays. International Journal of Control, 1979
- MFD给出直接求解的函数
  - 函数 fedmunds

参数最优化方法的数学基础

- 多变量系统框图



- 闭环系统模型 $\boldsymbol{T}(s)=\boldsymbol{G}(s)\boldsymbol{K}(s)\left[\boldsymbol{I}+\boldsymbol{G}(s)\boldsymbol{K}(s)\right]^{-1}$
- 期望在某频率段内闭环模型趋近于 $\boldsymbol{T}_{\mathrm{t}}(s)$
- 目标控制器 $\boldsymbol{K}_{\mathrm{t}}$ 满足 $\boldsymbol{G}\boldsymbol{K}_{\mathrm{t}}=\boldsymbol{T}_{\mathrm{t}}(\boldsymbol{I}-\boldsymbol{T}_{\mathrm{t}})^{-1}$
- 引入误差 $\boldsymbol{E}=(\boldsymbol{I}-\boldsymbol{T})(\boldsymbol{G}\boldsymbol{K}-\boldsymbol{G}\boldsymbol{K}_{\mathrm{t}})(\boldsymbol{I}-\boldsymbol{T}_{\mathrm{t}})$

数学基础

- 定义最优准则 $$||\boldsymbol{E}||_2^2 = \min_{\boldsymbol{N}(s)} \int_{-\infty}^{\infty} \mathrm{tr}\left[\boldsymbol{E}^{\mathrm{T}}(-\mathrm{j}\omega)\boldsymbol{E}(\mathrm{j}\omega)\right]\mathrm{d}\omega$$
- 选择控制器结构 $$\boldsymbol{K}(s) = \frac{\boldsymbol{N}(s)}{d(s)}$$
- 选择合适的控制器极点，通过寻优方法搜索出控制器的分子多项式矩阵
- 调用MFD现成函数直接设计

$$\boldsymbol{N}=\mathtt{fedmunds}(\boldsymbol{w},\boldsymbol{H},\boldsymbol{H}_{\mathrm{t}},\boldsymbol{N}_0,\boldsymbol{D})$$

例7-17 多变量系统设计

- 多变量受控对象

$$A=\begin{bmatrix}0 & 0 & 1.1320 & 0 & -1\\ 0 & -0.0538 & -0.1712 & 0 & 0.0705\\ 0 & 0 & 0 & 1 & 0\\ 0 & 0.0485 & 0 & -0.8556 & -1.013\\ 0 & -0.2909 & 0 & 1.0532 & -0.6859\end{bmatrix},\quad B=\begin{bmatrix}0 & 0 & 0\\ -0.120 & 1 & 0\\ 0 & 0 & 0\\ 4.419 & 0 & -1.665\\ 1.575 & 0 & -0.0732\end{bmatrix}$$

- 模型输入 $C=\begin{bmatrix}I_3 & \mathbf{0}_{3\times 2}\end{bmatrix}$

```
>> A=[0,0,1.1320,0,-1; 0,-0.0538,-0.1712,0,0.0705; 0,0,0,1,0;
     0,0.0485,0,-0.8556,-1.013;0,-0.2909,0,1.0532,-0.6859];
   B=[0,0,0; -0.120,1,0; 0,0,0; 4.419,0,-1.665; 1.575,0,-0.0732];
   C=eye(3,5); G=ss(A,B,C,0); w=logspace(-3,2); Hg=mfrd(G,w);
```

目标闭环模型与目标控制器

- 选定目标闭环模型

$$\boldsymbol{T}_{\mathrm{t}}(s)=\mathrm{diag}\left[\frac{3^2}{(s+3)^2},\frac{3^2}{(s+3)^2},\frac{10^2}{(s+10)^2}\right]$$

- 求目标控制器

```
>> s=tf('s'); g=3^2/(s+3)^2;
   T=[g,0,0; 0,g,0; 0,0,10^2/(s+10)^2];
```

```
>> I=eye(3); Kt=inv(G)*T*inv(I-T);
   Hk=mfrd(Kt,w);
   bodemag(Kt,w)
```

选择控制器的极点

- 由目标控制器的Bode图选择控制器结构
  - 第一输入无需积分器，其他两个需要
  - 第一输入选择极点 -6
  - 另外两个输入的控制器选择极点 -6、-30
- 控制器结构

$$\boldsymbol{k}_{i1}(s)=\frac{\boldsymbol{v}_{i1}^{0}s+\boldsymbol{v}_{i1}^{1}}{s+6},\ \boldsymbol{k}_{i2}(s)=\frac{\boldsymbol{v}_{i2}^{0}s^{2}+\boldsymbol{v}_{i2}^{1}s+\boldsymbol{v}_{i2}^{2}}{s(s+6)},\ \boldsymbol{k}_{i3}(s)=\frac{\boldsymbol{v}_{i3}^{0}s^{2}+\boldsymbol{v}_{i3}^{1}s+\boldsymbol{v}_{i3}^{2}}{s(s+30)}$$

矩阵选择

- 扩展成二阶控制器 $\boldsymbol{k}_{i1}(s)=\dfrac{\mathbf{0}s^2+\boldsymbol{v}_{i1}^0 s+\boldsymbol{v}_{i1}^1}{0s^2+s+6}$
- 分子分母矩阵设置

$$\boldsymbol{N}=\begin{bmatrix}0&1&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1&1\\0&1&1&1&1&1&1&1&1\end{bmatrix}$$

$$\boldsymbol{D}=\begin{bmatrix}0&1&6&1&6&0&1&30&0\\0&1&6&1&6&0&1&30&0\\0&1&6&1&6&0&1&30&0\end{bmatrix}$$

控制器的设计

- 输入相关矩阵，直接设计控制器

```
>> d=[0 1 6 1 6 0 1 30 0]; den=[d; d; d];
   num=[zeros(3,1) ones(3,8)]; Ht=mfrd(T,w);
   N=fedmunds(w,Hg,Ht,num,den)
```

- 控制器的数学模型

```
>> d1=[1,6]; d2=[1 6 0]; d3=[1 30 0];
   K11=tf(N(1,2:3),d1); K12=tf(N(1,6),d2);
   K13=tf(N(1,7:9),d3); K21=tf(N(2,2:3),d1);
   K22=tf(N(2,5:6),d2); K23=tf(N(2,7:9),d3);
   K31=tf(N(3,2:3),d1); K32=tf(N(3,6),d2); K33=tf(N(3,7:9),d3);
```

闭环系统的阶跃响应

- 控制器的数学模型

$$\boldsymbol{K}(s)=\begin{bmatrix} \dfrac{-6.5183s-4.1806}{s+6} & \dfrac{1.9101}{s(s+6)} & \dfrac{-5.2977s^2+6.3218s+77.927}{s(s+30)} \\ \dfrac{-0.7822s+0.1328}{s+6} & \dfrac{9s+0.7134}{s(s+6)} & \dfrac{-0.6161s^2+0.6246s+22.991}{s(s+30)} \\ \dfrac{-17.3s-5.6199}{s+6} & \dfrac{5.3316}{s(s+6)} & \dfrac{-99.857s^2-63.275s+104.83}{s(s+30)} \end{bmatrix}$$

- 闭环系统的阶跃响应

```
>> K=[K11 K12 K13; K21 K22 K23; K31 K32 K33];
   Gc=feedback(G*K,I); step(Gc,5)
```

控制系统的经典设计方法

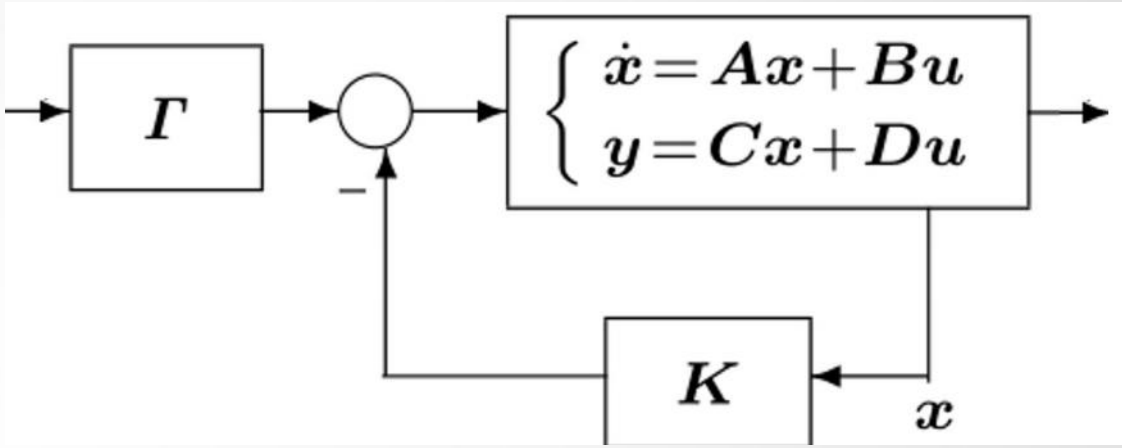
- 7.1 超前滞后校正器设计方法
- 7.2 基于状态空间模型的控制器设计方法
- 7.3 最优控制器设计
- 7.5 多变量系统的频域设计方法
- 7.6 多变量系统的解耦控制

多变量系统的解耦控制

- 解耦的概念与理解，解耦的意义
- 基于状态方程的解耦方法与实现
- 本节主要内容
  - 状态反馈的解耦
  - 标准传递函数
  - 状态反馈的极点配置解耦

状态反馈的解耦

- 开环系统状态方程 $(\boldsymbol{A},\boldsymbol{B},\boldsymbol{C},\boldsymbol{D})$
- 引入状态反馈 $\boldsymbol{u}=\boldsymbol{\Gamma r}-\boldsymbol{Kx}$
- 闭环系统的模型



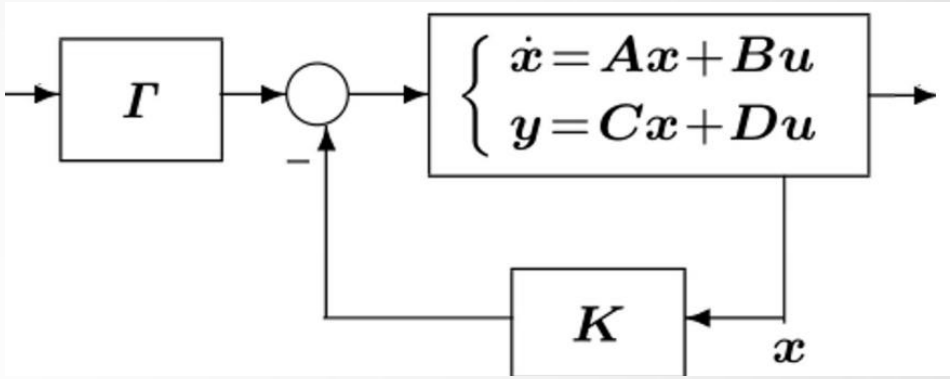
$$\boldsymbol{G}(s)=\left[(\boldsymbol{C}-\boldsymbol{DK})(s\boldsymbol{I}-\boldsymbol{A}+\boldsymbol{BK})^{-1}\boldsymbol{B}+\boldsymbol{D}\right]\boldsymbol{\Gamma}$$

- 定义阶次 d_j，使得 $\boldsymbol{c}_j^{\mathrm{T}}\boldsymbol{A}^{d_j}\boldsymbol{B}\neq\boldsymbol{0}$
- 其中，$\boldsymbol{c}_j^{\mathrm{T}}$ 为 $\boldsymbol{C}$ 的第 j 行

状态反馈解耦

- 如果下面的矩阵非奇异

$$\boldsymbol{F}=\begin{bmatrix}\boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{A}^{d_1}\boldsymbol{B}\\ \vdots\\ \boldsymbol{c}_m^{\mathrm{T}}\boldsymbol{A}^{d_m}\boldsymbol{B}\end{bmatrix}$$



- 则可以设计出状态反馈解耦矩阵

$$\boldsymbol{\Gamma}=\boldsymbol{F}^{-1},\quad \boldsymbol{K}=\boldsymbol{\Gamma}\begin{bmatrix}\boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{A}^{d_1+1}\\ \vdots\\ \boldsymbol{c}_m^{\mathrm{T}}\boldsymbol{A}^{d_m+1}\end{bmatrix}$$

解耦算法的MATLAB实现

$$\boldsymbol{F}=\begin{bmatrix}\boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{A}^{d_1}\boldsymbol{B}\\ \vdots\\ \boldsymbol{c}_m^{\mathrm{T}}\boldsymbol{A}^{d_m}\boldsymbol{B}\end{bmatrix}$$

- 调用格式 [G_1,K,d,Γ]=decouple(G)

$$\boldsymbol{\Gamma}=\boldsymbol{F}^{-1},\quad \boldsymbol{K}=\boldsymbol{\Gamma}\begin{bmatrix}\boldsymbol{c}_1^{\mathrm{T}}\boldsymbol{A}^{d_1+1}\\ \vdots\\ \boldsymbol{c}_m^{\mathrm{T}}\boldsymbol{A}^{d_m+1}\end{bmatrix}$$

```
function [G1,K,d,Gam]=decouple(G)
G=ss(G); A=G.a; B=G.b; C=G.c;
[n,m]=size(G.b); F=[]; K0=[];
for j=1:m, for k=0:n-1
   if norm(C(j,:)*A^k*B)>eps, d(j)=k; break; end, end
   F=[F; C(j,:)*A^d(j)*B]; K0=[K0; C(j,:)*A^(d(j)+1)];
end
Gam=inv(F); K=Gam*K0;
G1=minreal(tf(ss(A-B*K,B,C,G.d))*Gam);
```

例7-19 状态反馈解耦

- 受控对象

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

- 解耦

```
>> A=[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];
   B=[4, 6; 2, 4; 2, 2; 0, 2]; C=[0, 0, 0, 1; 0, 2, 0, 2];
   G=ss(A,B,C,0); [G1,K,d,Gam]=decouple(G)
```

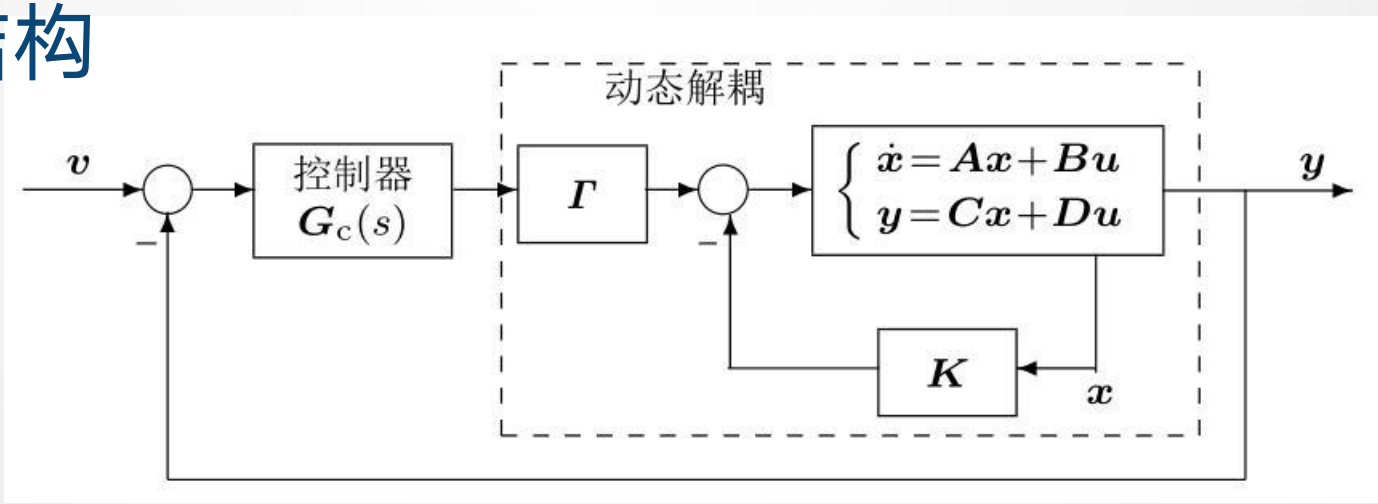
动态解耦的结果

- 解耦的结果

$$\boldsymbol{G}_1(s)=\begin{bmatrix}1/s & 0\\ -8.9\times10^{-16}/s & 1/s\end{bmatrix},$$

$$\boldsymbol{K}=\frac{1}{8}\begin{bmatrix}-1 & -3 & -3 & 5\\ 5 & -7 & -1 & -3\end{bmatrix},\ \boldsymbol{\Gamma}=\begin{bmatrix}-1.5 & 0.25\\ 0.5 & 0\end{bmatrix}$$

- 状态反馈结构



标准传递函数

- n 阶标准传递函数

$$T(s)=\frac{a_n}{s^n+a_1s^{n-1}+a_2s^{n-2}+\cdots+a_{n-1}s+a_n}$$

- ITAE最优系数

n	超调量	$\omega_n t_s$	首一化的分母多项式
1			$s+\omega_n$
2	4.6%	6.0	$s^2+1.41\omega_n s+\omega_n^2$
3	2%	7.6	$s^3+1.75\omega_n s^2+2.15\omega_n^2 s+\omega_n^3$
4	1.9%	5.4	$s^4+2.1\omega_n s^3+3.4\omega_n^2 s^2+2.7\omega_n^3 s+\omega_n^4$
5	2.1%	6.6	$s^5+2.8\omega_n s^4+5.0\omega_n^2 s^3+5.5\omega_n^3 s^2+3.4\omega_n^4 s+\omega_n^5$
6	5%	7.8	$s^6+3.25\omega_n s^5+6.6\omega_n^2 s^4+8.6\omega_n^3 s^3+7.45\omega_n^4 s^2+3.95\omega_n^5 s+\omega_n^6$
7	10.9%	10.0	$s^7+4.475\omega_n s^6+10.42\omega_n^2 s^5+15.08\omega_n^3 s^4+15.54\omega_n^4 s^3+10.64\omega_n^5 s^2+4.58\omega_n^6 s+\omega_n^7$

MATLAB实现

- 调用格式　T=std_tf(ω_n,n)
- 程序清单

```
function G=std_tf(wn,n)
M=[1,1,0,0,0,0,0 0; 1,1.41,1,0,0,0,0 0;
   1,1.75,2.15,1,0,0,0 0; 1,2.1,3.4,2.7,1,0,0 0;
   1,2.8,5.0,5.5,3.4,1,0 0;
   1,3.25,6.6,8.6,7.45,3.95,1,0;
   1,4.475,10.42,15.08,15.54,10.64,4.58,1];
G=tf(wn^n,M(n,1:n+1).*(wn.^[0:n]));
```

极点配置状态反馈解耦

- 定义矩阵 $\boldsymbol{E}$，每一行 $\boldsymbol{e}_i^{\mathrm{T}} = \boldsymbol{c}_i^{\mathrm{T}} \boldsymbol{A}^{d_i} \boldsymbol{B}$
- 另一个矩阵 $\boldsymbol{F}$ 的第 i 行

$$\boldsymbol{f}_i^{\mathrm{T}} = \boldsymbol{c}_i^{\mathrm{T}} \left(\boldsymbol{A}^{d_i+1} + a_{i,1} \boldsymbol{A}^{d_i} + \cdots + a_{i,d_i+1} \boldsymbol{I} \right)$$

- 状态反馈与前置矩阵为 $\boldsymbol{\Gamma} = \boldsymbol{E}^{-1}$，$\boldsymbol{K} = \boldsymbol{\Gamma} \boldsymbol{F}$
- 编写极点配置状态解耦的MATLAB函数

$$[\boldsymbol{G}_1, \boldsymbol{K}, \boldsymbol{d}, \boldsymbol{\Gamma}] = \texttt{decouple\_pp}(\boldsymbol{G}, \omega_{\mathrm{n}})$$

例7-20 前面的系统模型

- 例7-19的系统模型
- 选择期望自然频率 $\omega_n = 5$
- 输入模型并求取解耦控制器

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

```
>> A=[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];
   B=[4, 6; 2, 4; 2, 2; 0, 2];
   C=[0, 0, 0, 1; 0, 2, 0, 2];
   G=ss(A,B,C,0); [G1,K,d,Gam]=decouple_pp(G,5)
```

得出的结果

- 解耦结果

$$\boldsymbol{G}_1(s)=\begin{bmatrix}1/(s+5) & 0\\ \epsilon & 1/(s+5)\end{bmatrix},\ \epsilon\approx 10^{-14}$$

- 其他矩阵

$$\boldsymbol{K}=\frac{1}{8}\begin{bmatrix}-1 & 17 & -3 & -35\\ 5 & -7 & -1 & 17\end{bmatrix},\ \boldsymbol{\Gamma}=\begin{bmatrix}-1.5 & 0.25\\ 0.5 & 0\end{bmatrix}$$

- 解耦模型的阶跃响应

```
>> step(G1,10)
```

例7-21 考虑3×3多变量系统

- 选择 $\omega_n = 3$

```
>> A=[0,0,1.1320,0,-1; 0,-0.0538,-0.1712,0,0.0705;
      0,0,0,1,0; 0,0.0485,0,-0.8556,-1.013;
      0,-0.2909,0,1.0532,-0.6859];
   B=[0,0,0; -0.120,1,0; 0,0,0;
      4.419,0,-1.665; 1.575,0,-0.0732];
   C=eye(3,5); G=ss(A,B,C,0);
   [G1,K,d,Gam]=decouple_pp(G,3)
```

```
>> step(G1,10)
```

- 解耦后模型为 $G_1 = \mathrm{diag}\left(\dfrac{1}{s^2+4.23s+9}, \dfrac{1}{s+3}, \dfrac{1}{s^2+4.23s+9}\right)$

最优控制器设计小结

- 探讨了目标函数的选择问题
  - 演示了ITAE类指标比ISE指标更适合伺服控制
- 结合数值最优化技术和Simulink建模仿真技术
- 给出了对任意复杂系统的最优控制器设计方法
  - assignin()
  - fminsearch()

多变量系统频域设计小结

- 多变量系统的频域响应与对角占优
  - 现成函数频域响应分析：mfrd()
  - 伪对角占优化：pseudiag()
- 多变量系统参数最优化设计
  - John Edmunds 算法：fedmunds() 函数
  - 选择目标闭环传递函数，计算目标控制器
  - 由目标控制器特性选择极点和控制结构直接优化

多变量系统解耦小结

- 给出了状态方程系统解耦的系统结构
- 给出了两种多变量状态方程解耦的方法
  - 解耦成对角积分器模型
  - 解耦成对角标准传递函数模型
- 由解耦模型再设计外环控制器可能得出好的控制效果

Q & A

感谢您的聆听和反馈