

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

Chapter 7 Classical Controller Design Methods



Professor Dingyu Xue, xuedingyu@mail.neu.edu.cn
School of Information Science and Engineering,
Northeastern University, Shenyang, CHINA



本章主要内容

- 已知受控对象模型，如何为其设计控制器，得到满意的控制效果？
 - 超前滞后校正器设计方法
 - 基于状态空间模型的控制器设计方法
 - 最优控制器设计——OCD
 - 控制系统工具箱的设计界面
 - 多变量系统的频域设计方法
 - 多变量系统动态解耦

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

超前滞后校正器设计方法

Design of Phase Lead/Lag Compensators

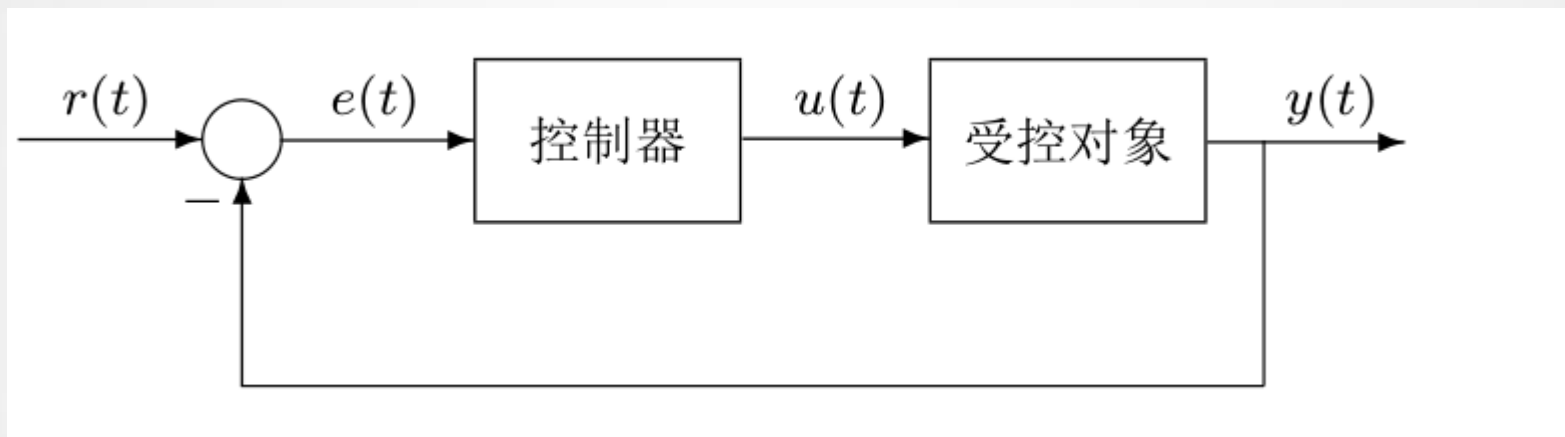


主讲：薛定宇教授



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

$$G_c(s) = \frac{K_c(s + z_{c1})(s + z_{c2})}{(s + p_{c1})(s + p_{c2})}$$

- 其中 , $z_{c1} \leq p_{c1}$, $z_{c2} \geq p_{c2}$, 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

$$\alpha = \frac{z_{c1}}{p_{c1}} = \frac{1 - \sin \phi_c(\omega_c)}{1 + \sin \phi_c(\omega_c)}$$

$$z_{c1} = \sqrt{\alpha} \omega_c, \quad p_{c1} = \frac{z_{c1}}{\sqrt{\alpha}} = \frac{\omega_c}{\sqrt{\alpha}}, \quad K_c = \frac{\sqrt{\omega_c^2 + p_{c1}^2}}{\sqrt{\omega_c^2 + z_{c1}^2} A(\omega_c)}$$



➤ 系统静态误差系数为

$$K_1 = \lim_{s \rightarrow 0} s^v G_o(s) = \frac{b_m}{a_{n-v}} \frac{K_c z_{c1}}{p_{c1}}$$

➤ 超前滞后校正器进一步设计为

$$z_{c2} = \frac{\omega_c}{10}, \quad p_{c2} = \frac{K_1 z_{c2}}{K_v}$$

➤ 若 $\phi_c(\omega_c) < 0$, 则需要滞后校正器

$$K_1 = b_m \frac{K_c}{a_{n-v}}, \quad K_c = \frac{1}{A(\omega_c)}, \quad z_{c2} = \frac{\omega_c}{10}, \quad p_{c2} = \frac{K_1 z_{c2}}{K_v}$$



超前滞后校正器设计函数

➤ 调用格式 $G_c = \text{leadlagc}(G, \omega_c, \gamma, K_v, \text{key})$

➤ 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;
```




```
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 $G(s) = \frac{10(s+1)(s+0.5)}{s(s+0.1)(s+2)(s+10)(s+20)}$

➤ 选择 $\omega_c = 10 \text{ rad/sec}$, 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 $\omega_c = 20\text{rad/s}$, $\gamma = 60^\circ$



```
>> 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)`



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
 - 后面将介绍其他的设计方法



控制系统仿真与 CAD