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

第八章 PID控制器设计方法

PID设计的经典方法

Classical Design of PID Controllers

主讲：薛定宇教授

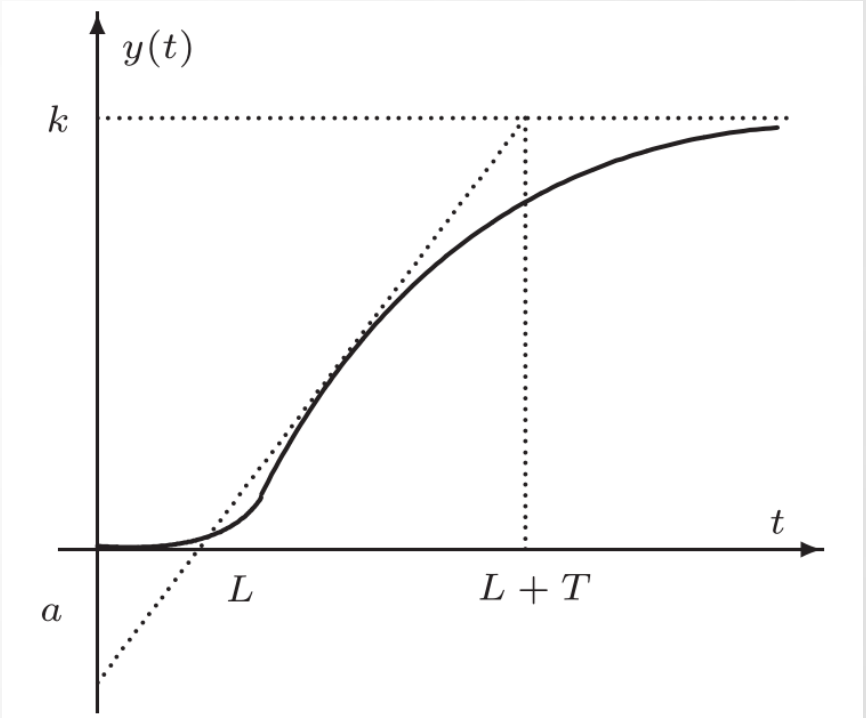
PID控制器设计经典方法

- 常用过程对象一阶延迟模型(FOPDT)的获取
- 基于FOPDT的设计方法
  - Ziegler-Nichols经验公式
  - MATLAB实现
- 其他模型类型的直接设计
  - MATLAB直接求解
  - 存在的问题——设计方法；非线性

过程对象的一阶延迟近似

- 常见近似模型 FOPDT $G(s)=\dfrac{k}{Ts+1}\mathrm{e}^{-Ls}$
- 问题，如何求出 k, L, T
  - 读图方法
  - 最优降阶方法——0/1
  - 最小二乘参数识别
- 基于MATLAB的求解

$[K,L,T,G_c]=\texttt{getfopdt(key,}G\texttt{)}$



例8-2 参数识别

- 受控对象 $G(s)=\dfrac{1}{(s+1)^5}$
- 获取FOPDT模型的方法

```
>> s=tf('s'); G=1/(s+1)^5;
   [K1,L1,T1,G1]=getfopdt(1,G),
   [K2,L2,T2,G2]=getfopdt(2,G)
   [K3,L3,T3,G3]=getfopdt(3,G)
   [K4,L4,T4,G4]=getfopdt(4,G)
   step(G,'-',G1,':',G2,'*',G3,'--',G4,'-.',15)
```

Ziegler-Nichols经验公式

- 1942年Ziegler与Nichols的一篇经典文章
- 已知：k, L, T 参数　　$a = KL/T$
- 求解PID类控制器参数
- 查表方法（直接计算、编MATLAB程序）

控制器类型	由阶跃响应整定			由频域响应整定		
	K_p	T_i	T_d	K_p	T_i	T_d
P	$1/a$			$0.5K_\mathrm{c}$		
PI	$0.9/a$	$3L$		$0.4K_\mathrm{c}$	$0.8T_\mathrm{c}$	
PID	$1.2/a$	$2L$	$L/2$	$0.6K_\mathrm{c}$	$0.5T_\mathrm{c}$	$0.12T_\mathrm{c}$

基于MATLAB的设计程序

- 直接设计

$$[G_c, K_p, T_i, T_d] = \texttt{ziegler(key,vars)}$$

  - 时域公式直接设计　$\texttt{vars}=[K, L, T, N]$
  - 频域公式直接设计

$$\texttt{vars}=[K_c, T_c, N],\ K_c,\ T_c = 2\pi/\omega_c$$

  - 改进方法 —— 可调参数 r_b, ϕ_b

$$[G_c, K_p, T_i, T_d] = \texttt{ziegler(key,[}K_c, T_c, r_b, \phi_b, N\texttt{])}$$

例8-3 直接设计

K_p	T_i	T_d
$1/a$		
$0.9/a$	$3L$	
$1.2/a$	$2L$	$L/2$

- 受控对象 $G(s)=\dfrac{0.1}{(s+1)^6}$
- 直接设计 $k=0.1, T=2.883, L=3.37$
- 控制器设计与效果比较

```
>> s=tf('s'); G=0.1/(s+1)^6; N=10;
   K=0.1; T=2.883; L=3.37; a=K*L/T;
   Kp=0.9/a; Ti=3*L; G1=Kp*(1+tf(1,[Ti 0]));
   Kp=1.2/a; Ti=2*L; Td=0.5*L; p=[Kp,Ti,Td]
>> G2=Kp*(1+tf(1,[Ti,0])+tf([Td 0],[Td/N 1]));
   step(feedback(G*G1,1),'-',feedback(G*G2,1),'--')
```

不同设计方法设计的控制器

K_p	T_i	T_d
$0.5K_c$		
$0.4K_c$	$0.8T_c$	
$0.6K_c$	$0.5T_c$	$0.12T_c$

- 频率响应数据设计

```
>> [Kc,b,wc,d]=margin(G); Tc=2*pi/wc;
   Kp=0.4*Kc; Ti=0.8*Tc; G1=Kp*(1+tf(1,[Ti 0]));
   Kp=0.6*Kc; Ti=0.5*Tc; Td=0.12*Tc;
   G2=Kp*(1+tf(1,[Ti,0])+tf([Td 0],1));
   step(feedback(G*G1,1),'-',feedback(G*G2,1),'--')
```

- 改进设计方法

```
>> s=tf('s'); G=0.1/(s+1)^6;
   G1=pidtune(G,'pi'), G2=pidtune(G,'pid')
   step(feedback(G*G1,1),feedback(G*G2,1))
```

其他基于模型的设计方法

- PI/PID设计手册
  - Aidan O'Dwyer. Handbook of PI and PID controller tuning rules, Imperial College Press, 2001, 2006, 2009

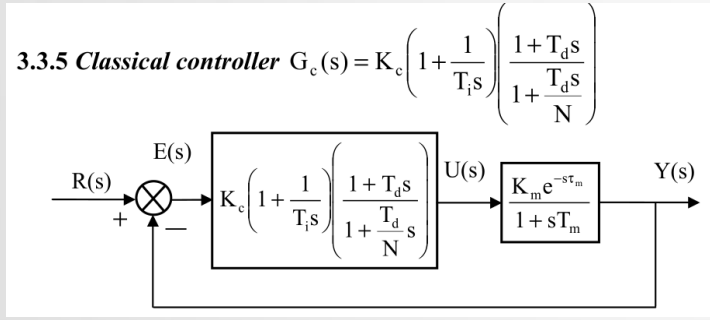
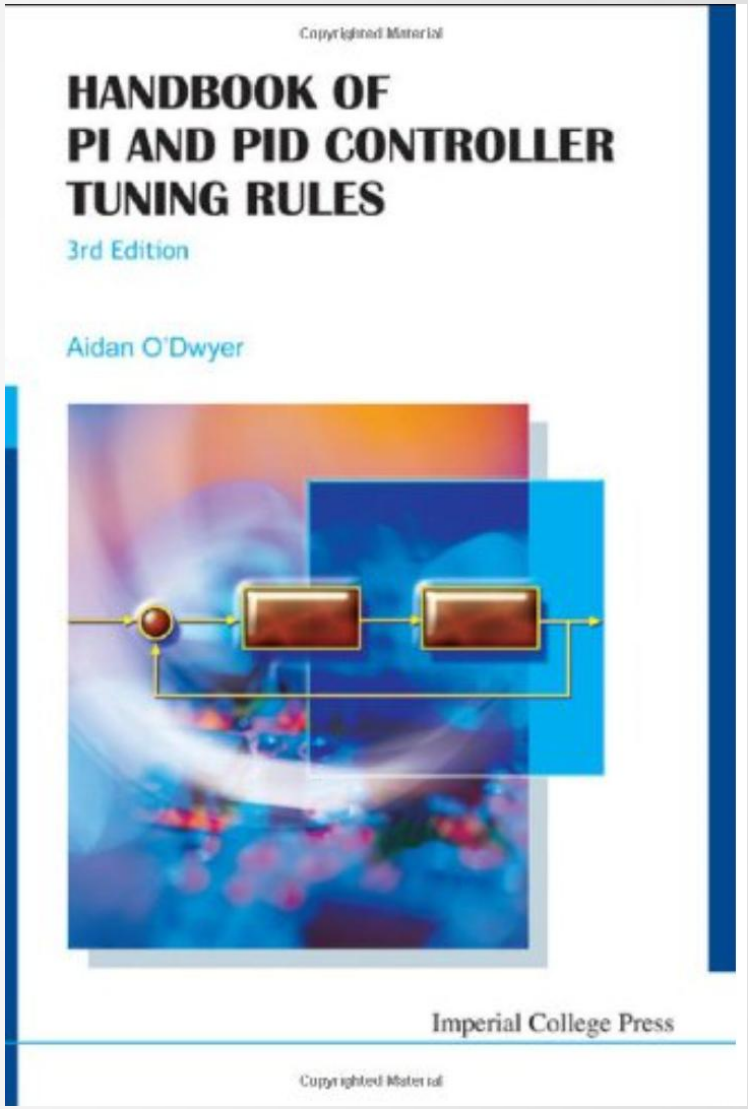


Table 13: PID controller tuning rules – FOLPD model $G_m(s) = \frac{K_m e^{-s\tau_m}}{1+sT_m}$

Rule	K_c	T_i	T_d	Comment
	Process reaction			
Kraus (1986). *Model: Method 23*	$0.833T_m/K_m\tau_m$	$1.5\tau_m$	$0.25\tau_m$	$N = \infty$
	Foxboro EXACT controller pre-tuning. Also given by Hang *et al.* (1993b), *p. 76*.			
Witt and Waggoner (1990).	$x_1 T_m/K_m\tau_m$	τ_m	τ_m	$10 \le N \le 20$; $0.6 \le x_1 \le 1.0$.
	Model: Method 2. 'Equivalent to' Ziegler and Nichols (1942).			
Witt and Waggoner (1990).	[1] K_c	T_i	T_d	$10 \le N \le 20$
	Model: Method 2. 'Preferred' tuning.			



MATLAB提供的设计函数

- Ziegler-Nichols 类方法的局限性
  - 只能用于 FOPDT 受控对象模型
  - 需要任意受控对象的PID控制器设计程序
- MATLAB的 pidtune 函数可以用于任意SISO LTI对象
  - 选项 —— p, i, pi, pd, pdf, pid, pidf

G_c=pidtune(G,'pi',ω_c), G_c=pidtune(G,'pid',ω_c)

```
>> s=tf('s'); G=0.1/(s+1)^6;
   G1=pidtune(G,'pi'), G2=pidtune(G,'pidf')
   step(feedback(G*G1,1),feedback(G*G2,1))
```

例8-4 复杂受控对象的

- 受控对象

$$G(s)=\frac{1+\dfrac{3\mathrm{e}^{-s}}{s+1}}{s+1}$$

- 设计命令
  - 用LTI模型表示复杂受控对象模型，直接设计控制器

```
>> s=tf('s'); G1=3/(s+1); G1.ioDelay=1;
   G=(1+G1)/(s+1);
   Gc1=pidtune(G,'pi'), Gc2=pidtune(G,'pidf')
   step(feedback(Gc1*G,1),feedback(Gc2*G,1))
```

不稳定模型的PID控制器设计

- 不稳定受控对象 $G(s)=\dfrac{s+2}{s^4+8s^3+4s^2-s+0.4}$,
- PID控制器

```
>> s=tf('s'); G=(s+2)/(s^4+8*s^3+4*s^2-s+0.4);
   Gc=pidtune(G,'pidf'), step(feedback(Gc*G,1))
```

```
>> Gc=pidtune(G,'pidf',1), step(feedback(Gc*G,1))
```

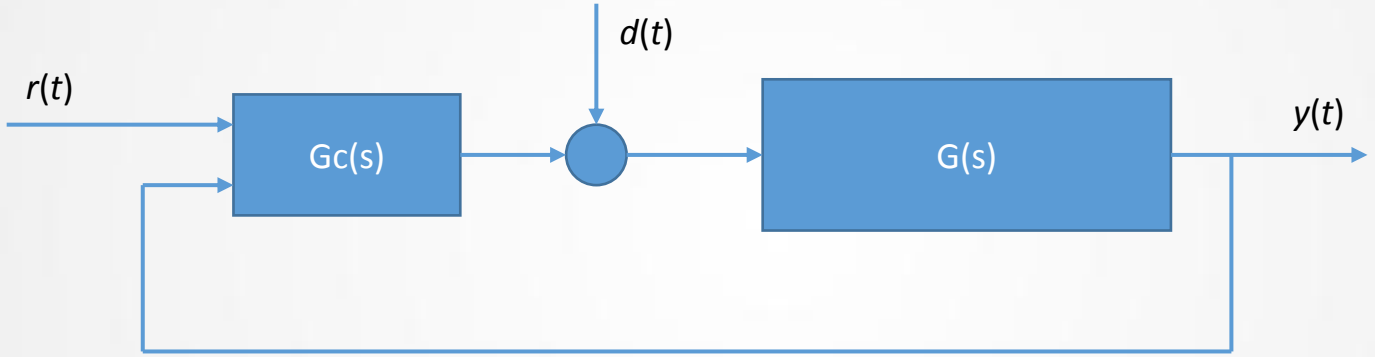
  - 真需要这么大的调节时间吗？

- 控制信号的曲线

```
>> step(feedback(Gc,G))
```

例8-5 二自由度PID控制器

- 二自由度PID控制器结构



- 设计界面 —— pidTuner

$$G(s)=\frac{1+\dfrac{3\mathrm{e}^{-s}}{s+1}}{s+1}$$

```
>> s=tf('s'); G1=3/(s+1); G1.ioDelay=1;
   G=(1+G1)/(s+1); Gc1=pidTuner(G,'pidf2')
```

PID经典设计小结

- 针对一大类过程模型的近似设计方法
  - FOPDT模型的参数识别方法
  - 针对FOPDT模型的设计方法，Ziegler Nichols方法
  - 二自由度PID控制结构与设计
  - 存在的问题 —— 算法繁杂，不宜比较
- MATLAB提供的设计函数
  - 简单使用，效果良好
  - 不能处理时变模型、非线性模型等，效果有待改进

薛定宇教授主讲

东北大学

NORTHEASTERN UNIVERSITY 1923

控制系统仿真与CAD