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

第十一章 分数阶控制基础

分数阶PID控制器

Fractional-order PID Controllers

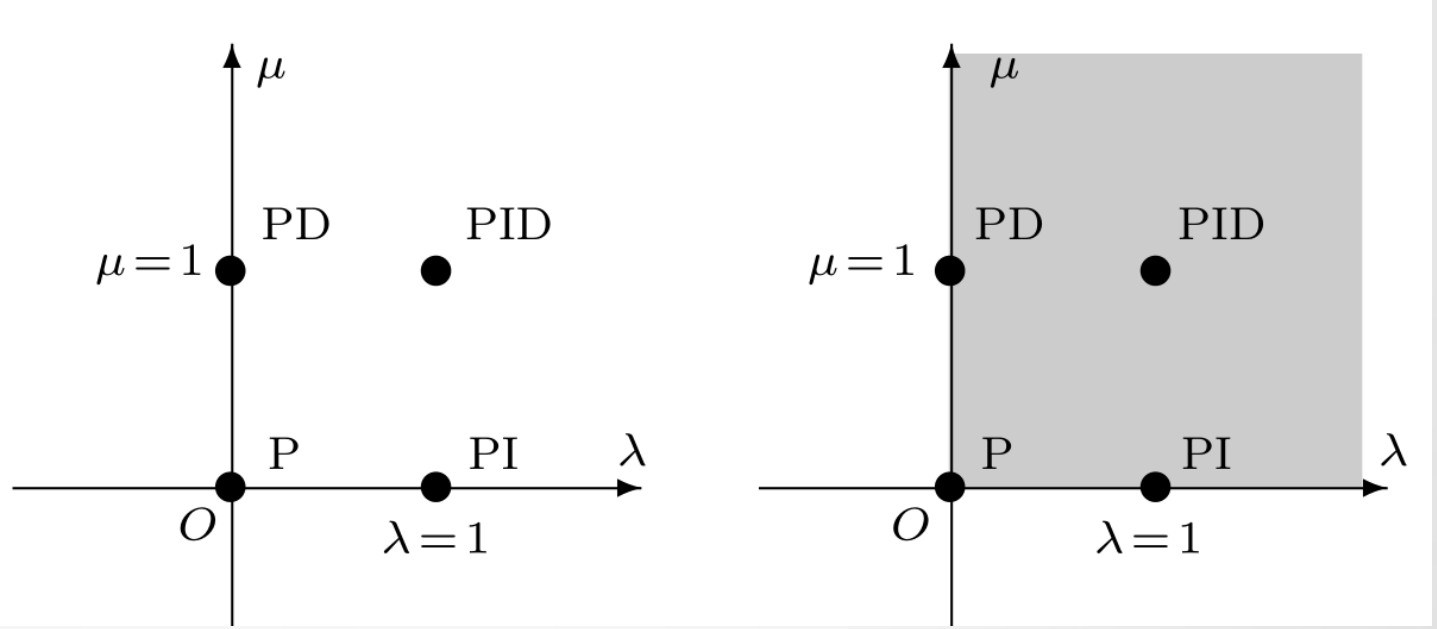
主讲：薛定宇教授

分数阶PID控制器设计

- 分数阶PID控制器的结构
- 分数阶PID控制器的设计方法
  - 基于频域响应的设计方法
  - 基于数值最优化的设计方法
  - 最优分数阶PID控制器设计界面

分数阶PID控制器

- 数学形式 $G_c(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu$
- Igor Podlubny教授



分数阶PID控制器输入

- 编写函数 $G_{\mathrm{c}}(s)=K_{\mathrm{p}}+\dfrac{K_{\mathrm{i}}}{s^{\lambda}}+K_{\mathrm{d}}s^{\mu}$

```
function Gc=fopid(Kp,Ki,Kd,lam,mu0)
s=fotf('s');
if length(Kp)==1,
   Gc=Kp+Ki*s^(-lam)+Kd*s^mu0;
else,
   x=Kp, Gc=x(1)+x(2)*s^(-x(4))+x(3)*s^(x(5));
end
```

- 调用格式 G_c=fopid(K_p,K_i,K_d,λ,μ)

例11-10 最优分数阶PID控制器设计

- 受控对象 $G(s)=\dfrac{1}{s^{2.6}+2.2s^{1.5}+2.9s^{1.3}+3.32s^{0.9}+1}$
- 目标函数

```
function fy=fpidfun(x,G,t,key)
C=fopid(x); dt=t(2)-t(1); e=step(feedback(1,G*C),t);
if key==1, fy=dt*sum(t.*abs(e)); else, fy=dt*sum(e.^2); end
disp([x(:).', fy])
```

- 直接寻优设计与仿真

```
>> s=fotf('s'); G=1/(s^2.6+3.3*s^1.5+2.9*s^1.3+3.32*s^0.9+1);
   xm=zeros(5,1); xM=[30; 30; 30; 2; 2]; x0=rand(5,1)';
   t=0:0.01:8; x=fminsearchbnd(@fpidfun,x0,xm,xM,[],G,t,1)
   Gc1=fopid(x); step(feedback(G*Gc1,1),t);
```

通用的设计框架

- 目标函数

```
function [fy,C]=fpidfuns(x)
global G t key1 key2; s=fotf('s'); t=t(:);
switch key1
   case {'fpid',1}, C=x(1)+x(2)*s^(-x(4))+x(3)*s^(x(5));
   case {'fpi',2}, C=x(1)+x(2)*s^(-x(3));
   case {'fpd',3}, C=x(1)+x(2)*s^x(3);
   case {'fpidx',4}, C=x(1)+x(2)/s+x(3)*s^x(4);
   case {'pid','PID',5}, C=x(1)+x(2)/s+x(3)*s;
end
dt=t(2)-t(1); e=step(feedback(1,G*C),t);
switch key2
   case {'itae','ITAE',1}, fy=dt*abs(e)*t;
   case {'ise','ISE',2}, fy=dt*sum(e.^2);
   case {'iae','IAE',3}, fy=dt*sum(abs(e));
   case {'itse','ITSE',4}, fy=dt*e.^2*t;
   otherwise,
      error(['Available criteria are itae, ise, iae, itse.'])
end, disp([x(:).' fy])
```

- 求解函数 $[G_{\mathrm{c}},\boldsymbol{x},y]=\mathtt{fpidtune}(\boldsymbol{x}_0,\boldsymbol{x}_{\mathrm{m}},\boldsymbol{x}_{\mathrm{M}},\mathtt{kA})$

最优分数阶PID控制器设计程序

- 需要给全局变量赋值
  - 受控对象 —— FOTF数据结构 G
  - 时间向量 t
  - 开关变量 key1—— ‘fpi’, ‘fpid’,’fpd’,‘fpidx’,’pid’
  - 开关key2 —— ‘itae’, ‘ise’, ‘iae’, ‘itse’
- 直接设计 $[G_{\mathrm{c}},\boldsymbol{x},y]=\mathtt{fpidtune}(\boldsymbol{x}_0,\boldsymbol{x}_{\mathrm{m}},\boldsymbol{x}_{\mathrm{M}},\mathtt{kA})$
  - kA —— 寻优算法，区间 $\lambda,\mu\in(0,2)$

例11-12 最优分数阶PID控制器

- 受控对象

$$G(s)=\frac{1}{s^{2.6}+2.2s^{1.5}+2.9s^{1.3}+3.32s^{0.9}+1}$$

- 最优设计

```
>> global G t key1 key2; s=fotf('s');
   G=1/(s^2.6+3.3*s^1.5+2.9*s^1.3+3.32*s^0.9+1);
   xm=zeros(5,1); xM=[30; 30; 30; 2; 2];
   x0=rand(5,1); t=0:0.01:8;
   key1='fpid'; key2='itae'; [Gc,x]=fpidtune(x0,xm,xM,1)
```

```
>> [Gc,x]=fpidtune(x,xm,xM,1)
```

与整数阶最优控制器比较

- 设计并比较整数阶PID控制器

```
>> xm=zeros(3,1); xM=[30; 30; 30]; x0=[1;1;1].';
   key1='pid'; key2='itae';
   [Gc1,x]=fpidtune(x0,xm,xM,1)
   step(feedback(G*Gc,1),t); hold on;
   step(feedback(G*Gc1,1),t);
```

- 遗留的问题：含有延迟的受控对象怎么办？
  - FOTF不能处理闭环延迟模型
  - 数值Laplace反变换 —— 速度较慢

例11-13 延迟系统的最优分数阶PID设计

- 目标函数描述
  - 采用数值Laplace反变换为核心计算仿真

```
function fy=fun_opts(x)
global G t key1 key2; t0=t(1); t1=t(end); N=length(t);
dt=t(2)-t(1); Gf=get_fpidf(x,G,key1); U='1/s';
[t,y]=INVLAP_new(Gf,t0,t1,N,1,U); e=1-y;
switch key2
    case {'itae','ITAE',1}, fy=dt*sum(t.*abs(e));
    case {'ise','ISE',2}, fy=dt*sum(e.^2);
    case {'iae','IAE',3}, fy=dt*sum(abs(e));
end, disp([x(:).' fy])
```

控制器设计与仿真

- ITAE性能指标

```
>> clear; global G t key1 key2
   G='exp(-s)/(0.8*s^2.2+0.5*s^0.9+1)'; key2='itae';
   key1='fpid'; t=0.01:0.01:20; x0=rand(5,1);
   xm=zeros(5,1); xM=[20;20;20;2;2];
   x=fminsearchbnd(@fun_opts,x0,xm,xM)
>> t0=0.01; tn=20; N=2000; Gf=get_fpidf(x,G,'fpid');
   [t1,y1]=INVLAP_new(Gf,t0,tn,N,1,'1/s'); plot(t1,y1)
```

- 直接设计

```
>> key2='ise'; x=fminsearchbnd(@fun_opts,x0,xm,xM);
   U='1/s'; Gf=get_fpidf(x,G,'fpid');
   [t2,y2]=INVLAP_new(Gf,t0,tn,N,1,U);
```

闭环系统效果比较

- 常规PID控制器设计

```
>> key1='pid'; key2='itae';
   x0=[1 1 1]'; xm=[0 0 0]'; xM=[30,30,30]';
   x=fminsearchbnd(@fun_opts,x0,xm,xM);
   Gf=get_fpidf(x,G,'pid');
   [t3,y3]=INVLAP_new(Gf,t0,tn,N,1,U);
```

- 仿真比较

```
>> plot(t1,y1,t2,y2,t3,y3)
```

例11-14 基于界面的设计

- 受控对象模型

$$G(s)=\frac{1}{0.8s^{2.2}+0.5s^{0.9}+1}$$

- 输入模型

```
>> G=fotf([0.8 0.5 1],[2.2 0.9 0],1,0)
```

- 设计控制器
  - 参数上界为 15， 时间 (0,8)

```
>> optimfopid
```

最优分数阶PID控制器小结

- 基于数值最优化的最优PID控制器设计
- 通过三个例子演示控制器设计
  - 一个简单的例子
  - 通用的无延迟受控对象的最优分数阶PID控制器设计函数 fpidtune
  - 有延迟受控对象，通过数值Laplace反变换求解
- 最优分数阶PID控制器设计界面 optimfopid

薛定宇教授主讲

东北大学

NORTHEASTERN UNIVERSITY 1923

控制系统仿真与CAD