第4章 计算机控制系统的基本控制策略

4.1 计算机控制系统数学基础

4.2 离散系统的模拟化设计方法

4.3 数字PID控制算法

4.4 直接数字设计方法

4.5复杂计算机控制系统设计方法

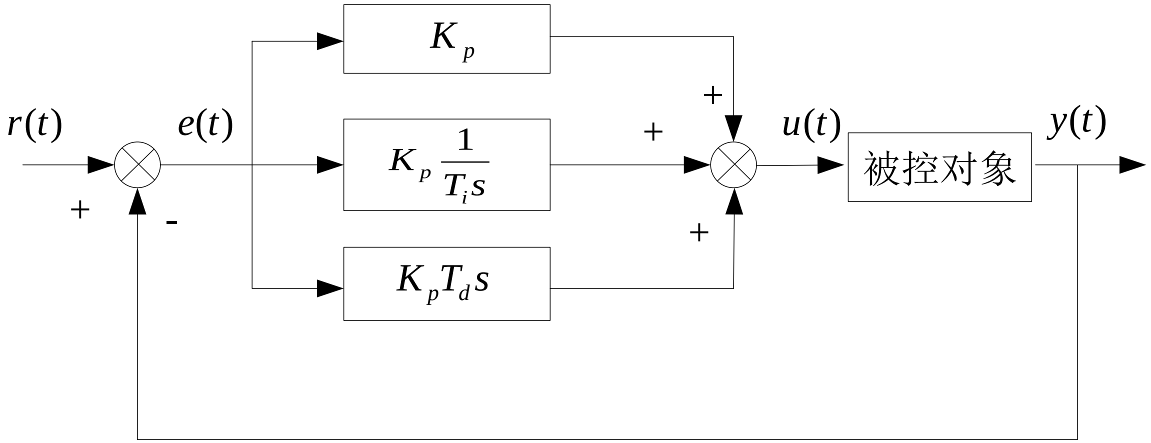
4.6先进PID控制系统设计方法

主要学习内容

数字PID控制算法

- PID控制算法及其作用
- 模拟PID控制器离散化
- PID算法的改进
- 数字PID控制器的实现
- 数字PID控制参数的整定

PID控制算法及其作用



PID控制器方框图

- **PID控制器是一种线性控制器，它将给定值与实际输出值的偏差*e*(*t*)的比例、积分和微分进行线性组合，形成控制量*u*(*t*)输出。**

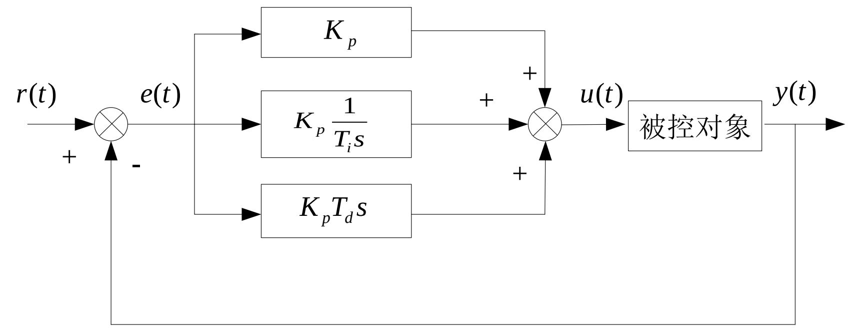
- **应用最广泛，简便，易于实现，鲁棒性强，适应面广。适用于线性、定常、单变量系统，对高阶对象需加入状态反馈调节。**

- **在过程计算机控制系统中，PID算法仍是应用最广泛、最成功的控制算法。**

PID控制算法及其作用

➢续系统中PID控制器的传递函数：

$$D(s)=\frac{U(s)}{E(s)}=K_p(1+\frac{1}{T_i s}+T_d s)$$



➢PID控制规律：

$$u(t)=K_p[e(t)+\frac{1}{T_i}\int_0^t e(t)\mathrm{d}t+T_d\frac{\mathrm{d}e(t)}{\mathrm{d}t}]$$

K_p：比例系数　　T_i：积分时间常数　　T_d：微分时间常数

$e(t)$：PID控制器的输入　　$u(t)$：PID控制器的输出。

➢PID控制器的输出是由比例控制、积分控制和微分控制三项组成，三项在控制器中所起的控制作用相互独立。

➢根据被控对象的特性和控制要求，可以选择不同形式的控制器，如比例（P）、比例积分（PI）、比例微分（PD）、比例积分微分（PID）控制器。

PID控制算法及其作用

➢ **PID控制器中三个环节的作用**

（1）**比例环节**的作用：能迅速反映偏差，从而减小偏差，但不能消除静差，**K_P的加大会引起系统的不稳定**。

（2）**积分环节**的作用：只要系统存在偏差，积分环节就会产生控制作用**减小偏差**，直到最终消除偏差，但**积分作用太强**会使系统超调加大，甚至使系统出现振荡。

（3）**微分环节**的作用：有助于系统**减小超调，克服振荡**，加快系统的响应速度，减小调节时间，从而改善了系统的动态性能，但**T_d过大会使系统出现不稳定**。

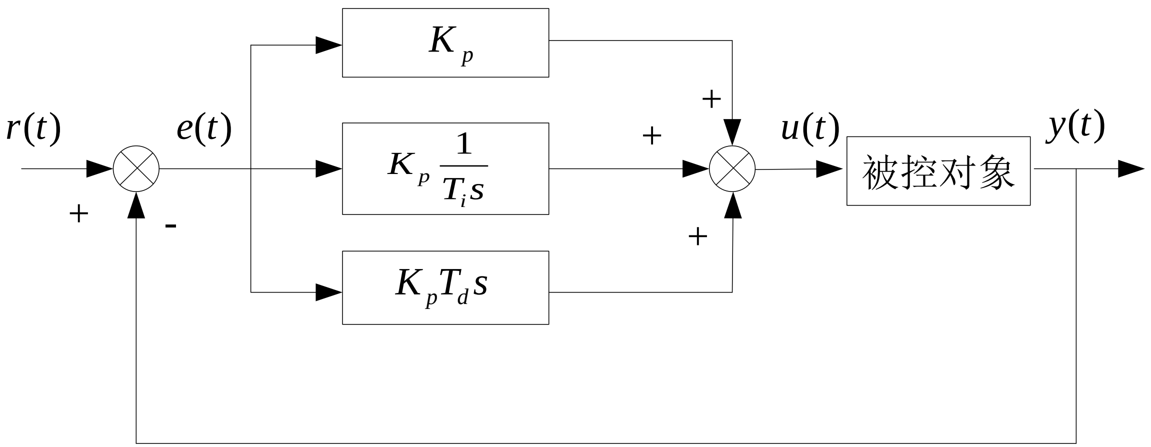
主要学习内容

数字PID控制算法

- ◆PID控制算法及其作用
- ◆模拟PID控制器离散化
- ◆PID算法的改进
- ◆数字PID控制器的实现
- ◆数字PID控制参数的整定

模拟PID控制器离散化

➢ 离散PID算法---模拟PID离散化



K_p：比例增益

T_i：积分时间常数

T_d：微分时间常数

PID控制器： $$u(t) = K_p \cdot e(t) + \frac{K_p}{T_i} \cdot \int_0^t e(t)dt + K_p \cdot T_d \cdot \frac{de(t)}{dt}$$

拉式变换： $$U(s) = K_p \cdot E(s) + \frac{K_p}{T_i} \cdot \frac{E(s)}{s} + K_p \cdot T_d \cdot s \cdot E(s)$$

传递函数： $$D(s) = \frac{u(s)}{E(s)} = K_p + \frac{K_p}{T_i} \cdot \frac{1}{s} + K_p \cdot T_d \cdot s$$

模拟PID控制器离散化

➢ 离散PID算法---模拟PID离散化

- **传递函数:** $$D(s)=\frac{U(s)}{E(s)}=K_p+\frac{K_p}{T_i}\cdot\frac{1}{s}+K_p\cdot T_d\cdot s$$
- **后向差分离散化得脉冲传递函数:**

$$D(z)=\frac{U(z)}{E(z)}=D(s)\Big|_{s=\frac{1-z^{-1}}{T}}$$

$$=K_p+K_p\frac{T}{T_i}\cdot\frac{1}{1-z^{-1}}+\frac{K_pT_d}{T}(1-z^{-1})$$

$$=K_p+K_I\frac{1}{1-z^{-1}}+K_D(1-z^{-1})$$

T：采样周期

$K_I=K_pT/T_i$：积分系数

$K_D=K_pT_d/T$：微分系数

- **将脉冲传递函数写成差分方程:**

$$(1-z^{-1})U(z)=[K_P(1-z^{-1})+K_I+K_D(1-z^{-1})^2]E(z)$$

模拟PID控制器离散化

➢ 离散PID算法---模拟PID离散化

T：采样周期

$K_I=K_pT/T_i$：积分系数

$K_D=K_pT_d/T$：微分系数

- **将脉冲传递函数写成差分方程：**

$$(1-z^{-1})U(z)=[K_P(1-z^{-1})+K_I+K_D(1-z^{-1})^2]E(z)$$

$$\begin{aligned}U(z)-z^{-1}U(z)=K_P[E(z)-z^{-1}E(z)]\\ +K_IE(z)\\ +K_D[E(z)-2z^{-1}E(z)+z^{-2}E(z)]\end{aligned}$$

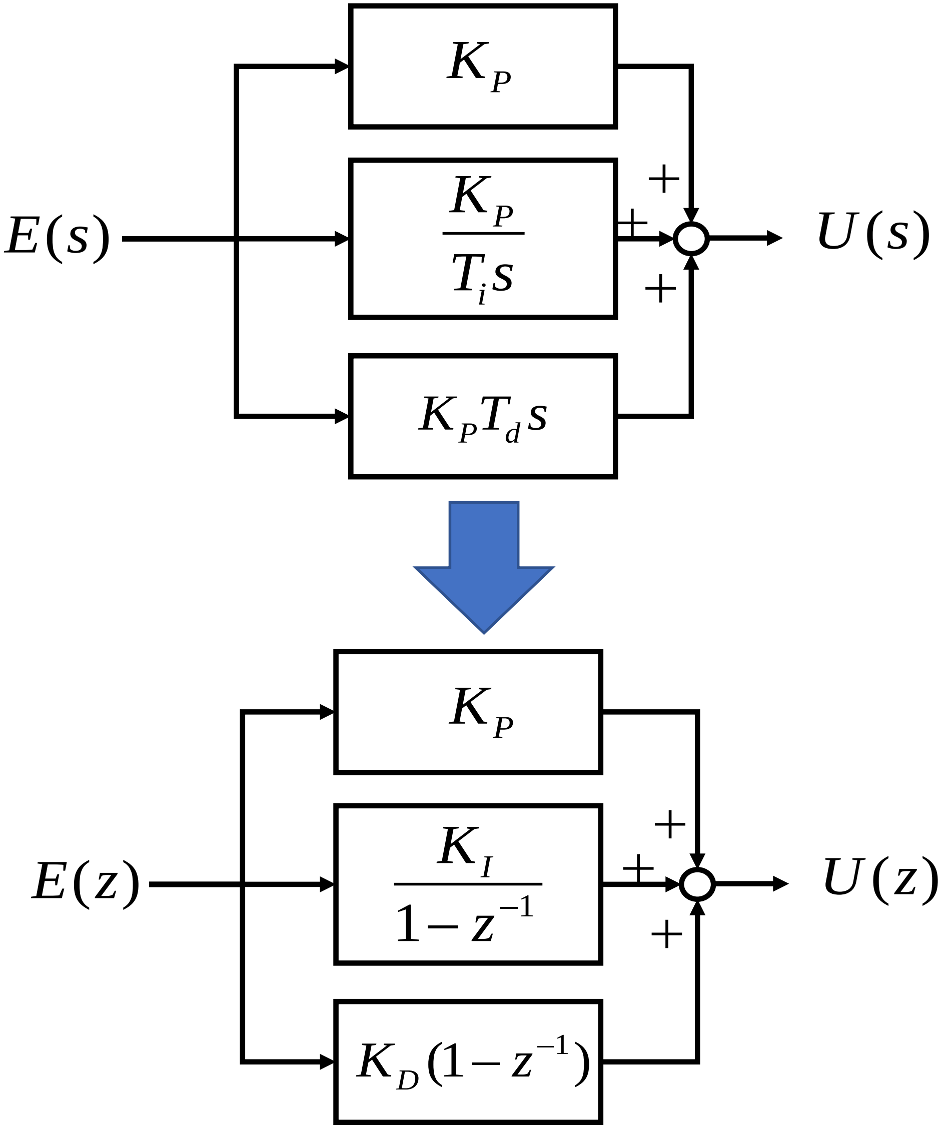
- **对上式两端进行z反变换：**

$$\boxed{\begin{aligned}u(k)=u(k-1)\\ +K_p\{e(k)-e(k-1)\}\\ +K_Ie(k)\\ +K_D\{e(k)-2e(k-1)+e(k-2)\}\end{aligned}}$$

通用数字式 PID控制算式

模拟PID控制器离散化

➢ 离散PID算法



$$D(s)=\frac{U(s)}{E(s)}$$

$$=K_p+\frac{K_p}{T_i}\cdot\frac{1}{s}+K_p\cdot T_d\cdot s$$

$$D(z)=\frac{U(z)}{E(z)}$$

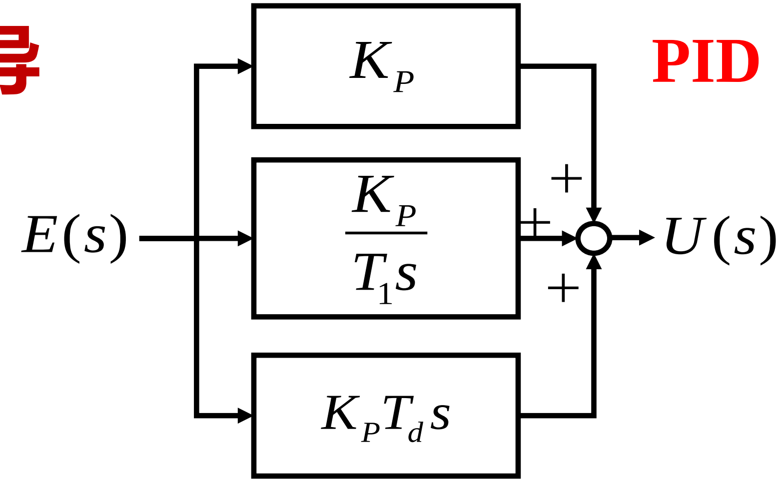
$$=K_p+\frac{K_I}{1-z^{-1}}+K_D(1-z^{-1})$$

模拟PID控制器离散化

➢ 位置式PID算法-时域近似推导

$$u(t)=K_p\{e(t)+\frac{1}{T_i}\int_0^t e(t)dt+T_d\frac{de(t)}{dt}\}$$

PID



采样周期T足够小：

$$\begin{cases} u(t)\approx u(k) \\ e(t)\approx e(k) \\ \int_0^t e(t)dt \approx \sum_{j=0}^{k} e(j)\Delta t = T\sum_{j=0}^{k} e(j) \\ \frac{de(t)}{dt}\approx \frac{e(k)-e(k-1)}{\Delta t}=\frac{e(k)-e(k-1)}{T} \end{cases}$$

$$\boxed{\begin{aligned} u(k) &= K_p\{e(k)+\frac{T}{T_i}\sum_{j=0}^{k}e(j)+\frac{T_d}{T}[e(k)-e(k-1)]]\} \\ &= K_p e(k)+K_I\sum_{j=0}^{k}e(j)+K_D[e(k)-e(k-1)]]\} \end{aligned}}$$

$K_I=K_pT/T_i$：积分系数

$K_D=K_pT_d/T$：微分系数

T：采样周期

模拟PID控制器离散化

➢ 位置式PID算法-时域近似推导

$$u(k)=K_p\{e(k)+\frac{T}{T_i}\sum_{j=0}^{k}e(j)+\frac{T_d}{T}[e(k)-e(k-1)]\}$$
$$=K_pe(k)+K_I\sum_{j=0}^{k}e(j)+K_D[e(k)-e(k-1)]\}$$

$K_I=K_pT/T_i$：积分系数

$K_D=K_pT_d/T$：微分系数

T：采样周期

注意：

(1)公式非递推，要累加偏差；

(2) 当T比对象时间常数Tc小得多时，控制效果与模拟控制器相近；

(3) $u(k)$对应执行机构的实际位置，故称作位置式PID算法。

模拟PID控制器离散化

➢ 增量式PID算法

$$\begin{cases} u(k) = K_p\{e(k) + \frac{T}{T_i}\sum_{j=0}^{k} e(j) + \frac{T_d}{T}[e(k) - e(k-1)]\} & (1) \\ u(k-1) = K_p\{e(k-1) + \frac{T}{T_i}\sum_{j=0}^{k-1} e(j) + \frac{T_d}{T}[e(k-1) - e(k-2)]\} & (2) \end{cases}$$

(1)-(2):

$$\Delta u(k) = u(k) - u(k-1)$$

$K_I = K_p T / T_i$　　$K_D = K_p T_d / T$

$$= K_p[e(k) - e(k-1)] + K_I e(k) + K_D[e(k) - 2e(k-1) + e(k-2)]$$

- **增量式PID合并同类项:**

$$\boxed{\Delta u(k) = Ae(k) + Be(k-1) + Ce(k-2)}$$

其中: $A = K_P(1 + \frac{T}{T_i} + \frac{T_d}{T})$　　$B = -K_P(1 + 2\frac{T_d}{T})$　　$C = K_P \cdot \frac{T_d}{T}$

模拟PID控制器离散化

➢ 增量式PID算法

$$\boxed{\Delta u(k) = Ae(k) + Be(k-1) + Ce(k-2)} \qquad u(k) = u(k-1) + \Delta u(k)$$

其中: $A = K_P(1 + \frac{T}{T_i} + \frac{T_d}{T})$ $\quad B = -K_P(1 + 2\frac{T_d}{T})$ $\quad C = K_P \cdot \frac{T_d}{T}$

- 该公式是递推的，与位置式无本质区别；
- 不同的执行机构，需要不同的控制输出。步进电机有保持器，接受增量信号。
- 增量式便于计算机实现；
- 增量式从手动到自动控制切换时,不需要给定初始值;
- 增量式可有效抵制过调量,避免积分饱和。

模拟PID控制器离散化

➢ PID算法执行过程

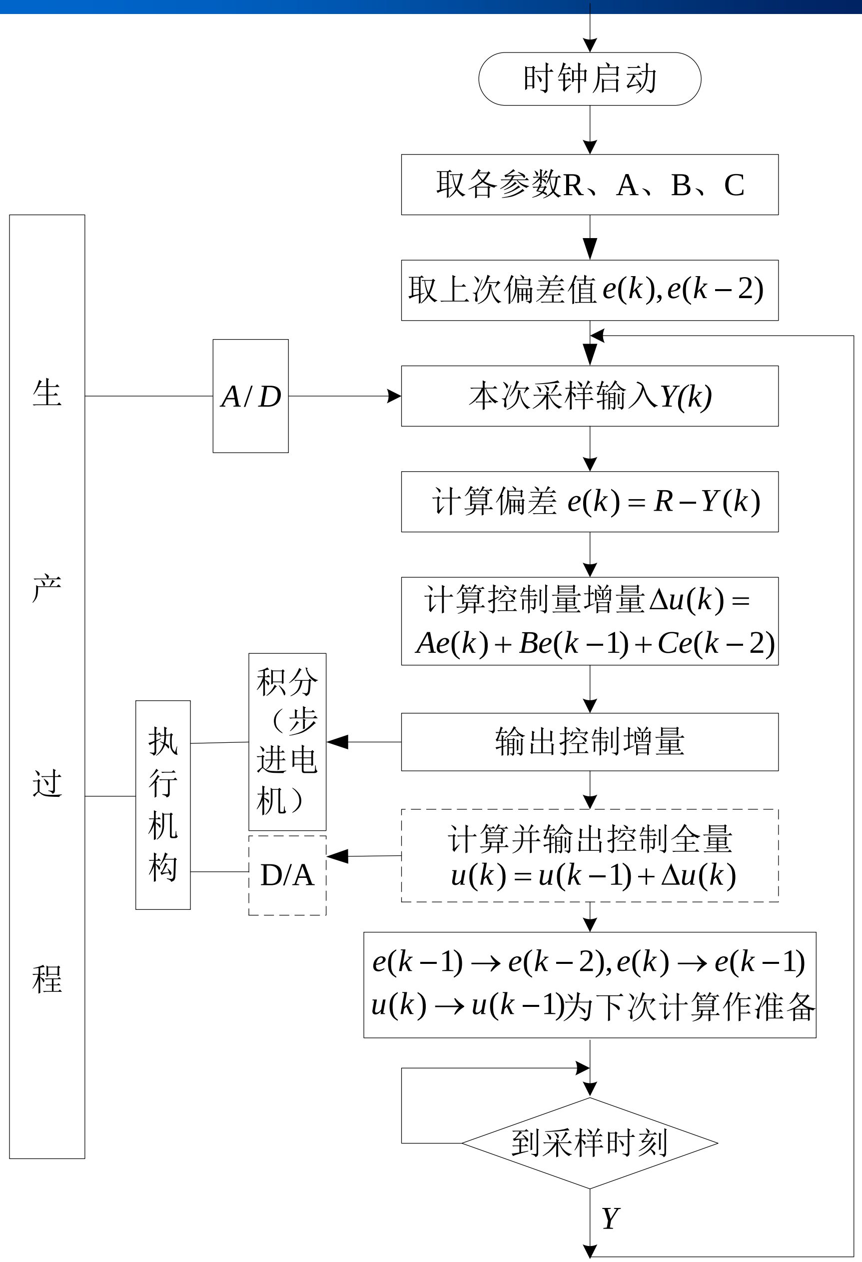
- **增量式算法：**

$$\boxed{\Delta u(k)=Ae(k)+Be(k-1)+Ce(k-2)}$$

其中：

$$\begin{cases} A=K_P\left(1+\dfrac{T}{T_i}+\dfrac{T_d}{T}\right) \\ B=-K_P\left(1+2\dfrac{T_d}{T}\right) \\ C=K_P\cdot\dfrac{T_d}{T} \end{cases}$$

$$\boxed{u(k)=u(k-1)+\Delta u(k)}$$



主要学习内容

数字PID控制算法

- ◆PID控制算法及其作用
- ◆模拟PID控制器离散化
- ◆PID算法的改进
- ◆数字PID控制器的实现
- ◆数字PID控制参数的整定

PID算法的改进

- 为了解决过程计算机应用中的一些实际问题，便于进一步提高控制性能、发挥计算机控制的优点，需要对数字PID控制算法作针对性的改进

◆微分算法的改进

◆积分算法的改进

◆带死区的PID控制

◆可变增益PID控制

◆时间最优PID控制

PID算法的改进-微分算法的改进

- 微分作用的引进改善了系统动态特性，但也容易引进高频干扰
- 在数字PID实际使用时多采用**实际微分**或加上**低通滤波**

1、实际微分PID算法

2、带一阶滤波器的PID控制

3、微分先行PID控制算法

4、四点中心差分算法改进微分作用

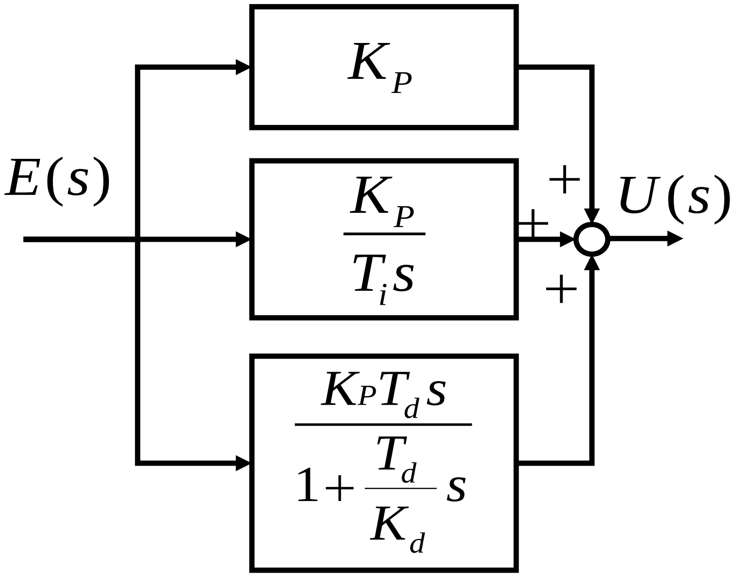
PID算法的改进-微分算法的改进

1、实际微分PID算法

理想微分PID:

$$D(s)=\frac{U(s)}{E(s)}=K_P+\frac{K_P}{T_i}\cdot\frac{1}{s}+K_P\cdot T_d\cdot s$$

$$\begin{aligned}\Delta u(k)=&K_P[e(k)-e(k-1)]+K_I e(k)\\&+K_D[e(k)-2e(k-1)+e(k-2)]\end{aligned}$$



实际微分PID:

$$D(s)=\frac{U(s)}{E(s)}=\overset{D_P(s)}{\boxed{K_P}}+\overset{D_I(s)}{\boxed{\frac{K_P}{T_i}\cdot\frac{1}{s}}}+\overset{D_D(s)}{\boxed{K_P\cdot\frac{T_d s}{1+\frac{T_d}{K_d}s}}}$$

$$\begin{cases}\Delta u_P(k)=K_P[e(k)-e(k-1)]\\ \Delta u_I(k)=K_I e(k)\end{cases}$$

◆ 与普通PID算式中的比例和积分环节相同。

PID算法的改进-微分算法的改进

1、实际微分PID算法

实际微分环节： $D_D(s)=\dfrac{U_D(s)}{E(s)}=\dfrac{K_P T_d s}{1+\dfrac{T_d}{K_d}s}$

后向差分离散化： $(s=\dfrac{1-z^{-1}}{T})$

$$u_D(k)=\frac{T_d}{K_d T+T_d}\{u_D(k-1)+K_P K_d[e(k)-e(k-1)]\}$$

令： $\alpha=\dfrac{T_d}{K_d T+T_d},\alpha<1$ **则：** $1-\alpha=\dfrac{K_d T}{K_d T+T_d}$

$$u_D(k)=\alpha\cdot u_D(k-1)+(1-\alpha)\boxed{K_D[e(k)-e(k-1)]}$$

理想微分

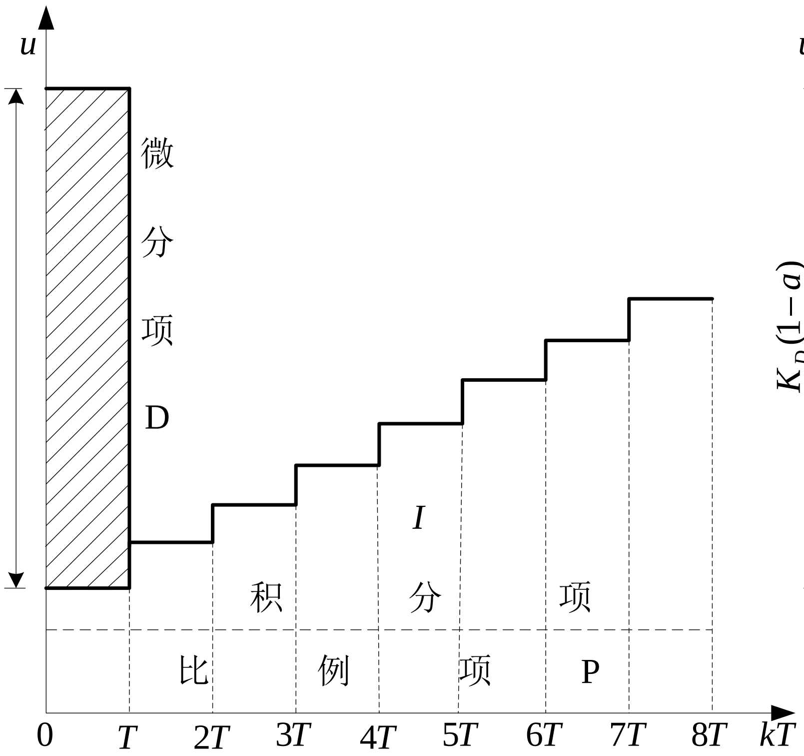
PID算法的改进-微分算法的改进

1、实际微分PID算法

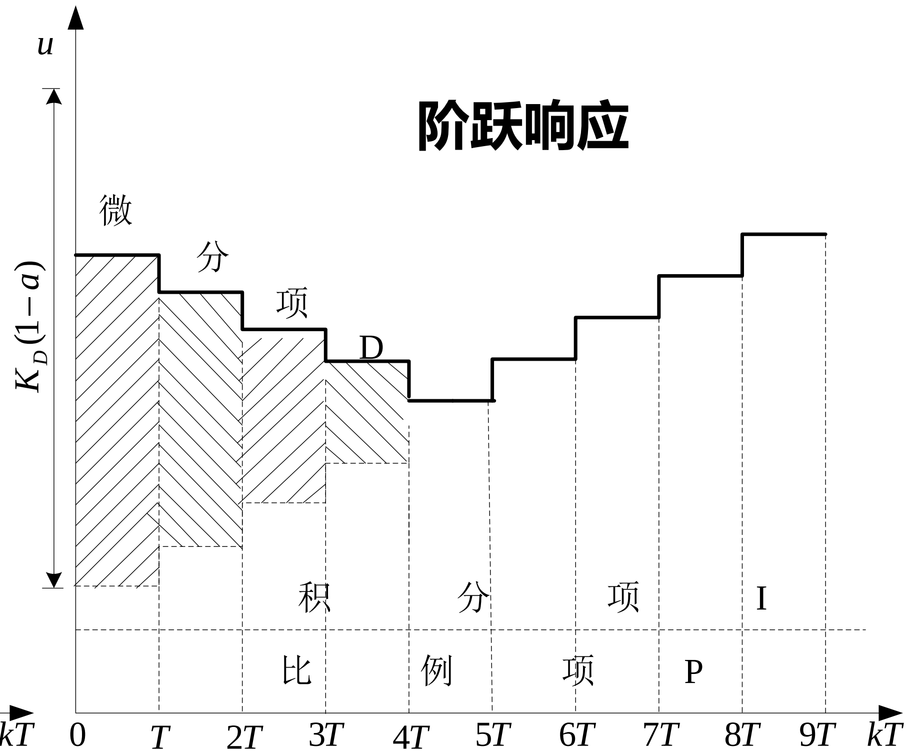
实际微分环节： $u_D(k)=\alpha\cdot u_D(k-1)+(1-\alpha)K_D[e(k)-e(k-1)]$

$$\left(\alpha=\frac{T_d}{K_dT+T_d},\alpha<1\right)$$

理想微分环节： $u_D(k)=K_D[e(k)-e(k-1)]$



理想微分PID



实际微分PID

PID算法的改进-微分算法的改进

1、实际微分PID算法

$$u_D(k) = \alpha \cdot u_D(k-1) + (1-\alpha)K_D[e(k) - e(k-1)]$$

$$\alpha = \frac{T_d}{K_d T + T_d} = \frac{{}^{Td}/_{Kd}}{T + {}^{Td}/_{Kd}}$$

$T_d/K_d \rightarrow 0$ 则 $\alpha=0$

$T_d/K_d \rightarrow \infty$ 则 $\alpha=1$

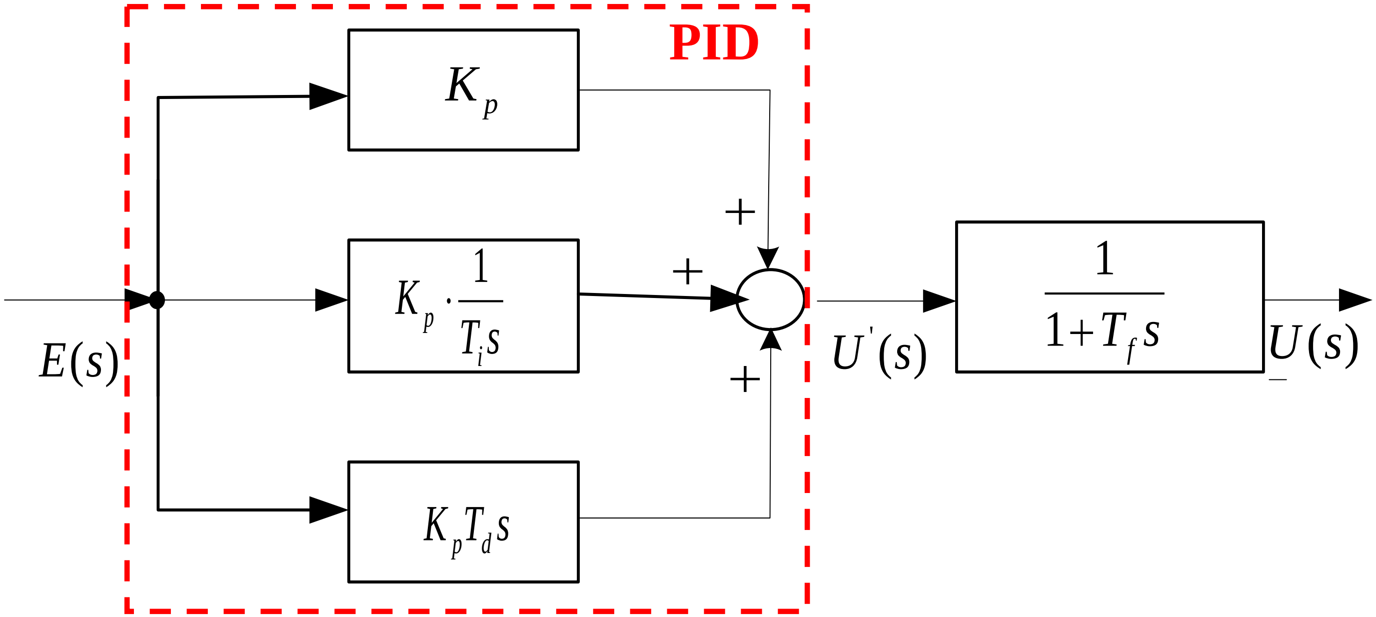
小结：

- **实际微分即第一次微分输出比例可调，下调（1-α）之后，输出按$\alpha U_d(k\text{-}1)$ 逐步下降，微分作用持续多个采样周期，使得一般的工业用执行机构，能比较好地跟踪微分作用输出；**
- **理想微分输出太大，容易引起溢出，造成系统震荡；**
- **理想微分对高频干扰敏感，实际微分具有滤波作用，抑制高频。**

PID算法的改进-微分算法的改进

2、带一阶滤波器的PID控制

作用：加入一阶滤波器，滤去高频干扰。

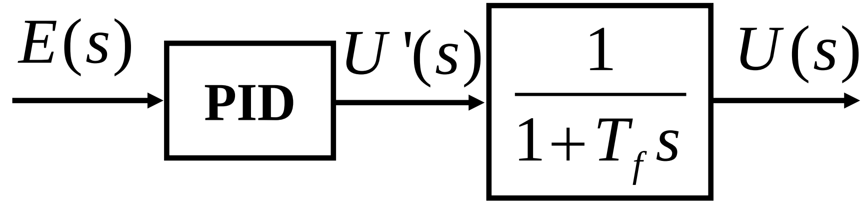


$$D(s)=\frac{U(s)}{E(s)}=\frac{U(s)}{U'(s)}\cdot\frac{U'(s)}{E(s)}$$

$$\begin{cases}\dfrac{U'(s)}{E(s)}=K_p+\dfrac{K_p}{T_i s}+K_P T_d s\\[2ex]\dfrac{U(s)}{U'(s)}=\dfrac{1}{1+T_f s}\end{cases}$$

PID算法的改进-微分算法的改进

2、带一阶滤波器的PID控制



$$\frac{U'(s)}{E(s)} = K_p + \frac{K_p}{T_i}\cdot\frac{1}{s} + K_P \cdot T_d \cdot s$$

$$\begin{aligned} u'(k) = u'(k-1) \\ &+K_p[e(k)-e(k-1)] \\ &+K_I e(k) \\ &+K_D[e(k)-2e(k-1)+e(k-2)] \end{aligned}$$

$$\begin{cases} K_I = K_P T / T_i \\ K_D = K_P T_d / T \end{cases}$$

$$\frac{U(s)}{U'(s)} = \frac{1}{1+T_f s}$$

$$T_f sU(s) + U(s) = U'(s)$$

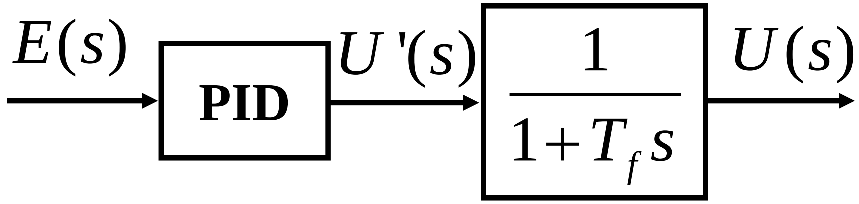
$$T_f \frac{du(t)}{dt} + u(t) = u'(t)$$

$$T_f \frac{u(k)-u(k-1)}{T} + u(k) = u'(k)$$

$$u(k) = \frac{T_f}{T+T_f} u(k-1) + \frac{T}{T+T_f} u'(k)$$

PID算法的改进-微分算法的改进

2、带一阶滤波器的PID控制



$$\frac{U(s)}{E(s)}=\frac{1}{1+T_f s}\left(K_p+\frac{K_p}{T_i s}+K_p T_d s\right)$$

$$D(s)=\frac{U(s)}{E(s)}=\frac{K_1(T_1 s+1)(T_2 s+1)}{T_1 s(\nu T_2 s+1)}=\frac{T_2 s+1}{\nu T_2 s+1}K_1\left(1+\frac{1}{T_1 s}\right)$$

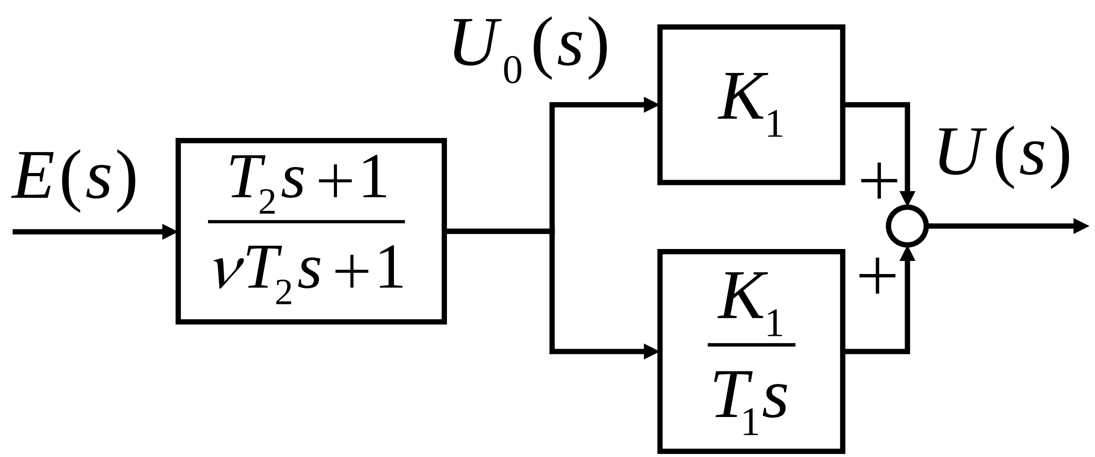
其中：

$$T_1=\frac{1}{2}\left(T_i+\sqrt{T_i^2-4T_iT_d}\right)$$

$$K_1=\frac{K_p}{2}\left(1+\sqrt{1-\frac{4T_d}{T_i}}\right)$$

$$T_2=\frac{1}{2}\left(T_i-\sqrt{T_i^2-4T_iT_d}\right.$$

$$\nu=\frac{T_f}{T_2}=\frac{2T_f}{T_i-\sqrt{T_i^2-4T_iT_d}}$$



微分先行PID控制