

# 检测与控制技术

## 第六章

# 数字PID及其改进算法



- ◆ 课时授课计划
- ◆ 提 纲
- ◆ 课 程 内 容



# 第六章 数字PID及其改进算法

课题:

## 第6章 数字PID及其改进算法

- 6.1 PID控制及作用
- 6.2 离散化的方法
- 6.3 数字PID及其算法
- 6.4 标准PID算法的改进
- 6.5 PID调节器参数的整定
- 6.6 纯滞后补偿控制技术
- 6.7 串级控制技术
- 6.8 其它常用的控制方法

目的与要求:

了解和掌握数字PID 控制算法、PID 调节器参数的整定。



## 第六章 数字PID及其改进算法

重点与难点：

重点：数字PID 控制算法、PID 调节器参数的整定。

难点：标准PID 算法的改进。

课堂讨论：

为什么要对标准PID 算法进行改进？

现代教学方法与手段：

微型计算机测控技术网络课程

PowerPoint

复习（提问）：

数字PID 控制算法两种数学模型是什么？



# 第六章 数字PID及其改进算法

## 6.1 PID控制及作用

- 6.1.1 比例调节器
- 6.1.2 比例积分调节器
- 6.1.3 比例积分微分调节器 (PID)

## 6.2 离散化的方法

- 6.2.1 差分变化法
- 6.2.2 零阶保持器法
- 6.2.3 双线性变化法

## 6.3 数字PID及其算法

- 6.3.1 位置式PID控制算法
- 6.3.2 PID增量式控制算法

## 6.4 标准PID算法的改进

- 6.4.1 “饱和”作用的抑制



# 第六章 数字PID及其改进算法

6.4.2 不完全微分PID算法

6.4.3 微分先行PID算法

6.4.4 带死区的PID控制

6.4.5 给定值突变时对控制量进行阻尼的算法

## 6.5 PID调节器参数的整定

6.5.1 凑试法确定PID调节参数

6.5.2 实验经验法确定PID调节参数

## 6.6 纯滞后补偿控制技术

6.6.1 纯滞后补偿算法

6.6.2 纯滞后补偿环节的数字控制器



# 第六章 数字PID及其改进算法

6.8 其它常用的控制方法

6.8.1 比值控制

6.8.2 选择性控制

思考题与习题：P229



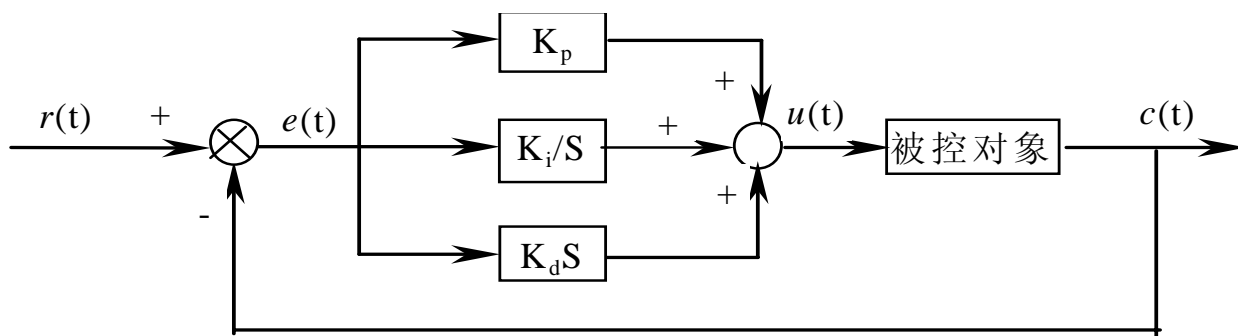
# 第六章 数字PID及其改进算法

## 6.1 PID控制及作用

PID调节器是一种线性调节器，这种调节器是将设定值 $r(t)$ 与输出值 $c(t)$ 进行比较构成控制偏差

$$e(t) = r(t) - c(t)$$

将其按比例、积分、微分运算后，并通过线性组合构成控制量，如图所示，所以简称为P（比例）、I（积分）、D（微分）调节器。



模拟PID调节器控制系统框图



# 第六章 数字PID及其改进算法

## 6.1.1 比例调节器

### 1. 比例调节器的表达式

$$u = K_p e + u_0$$

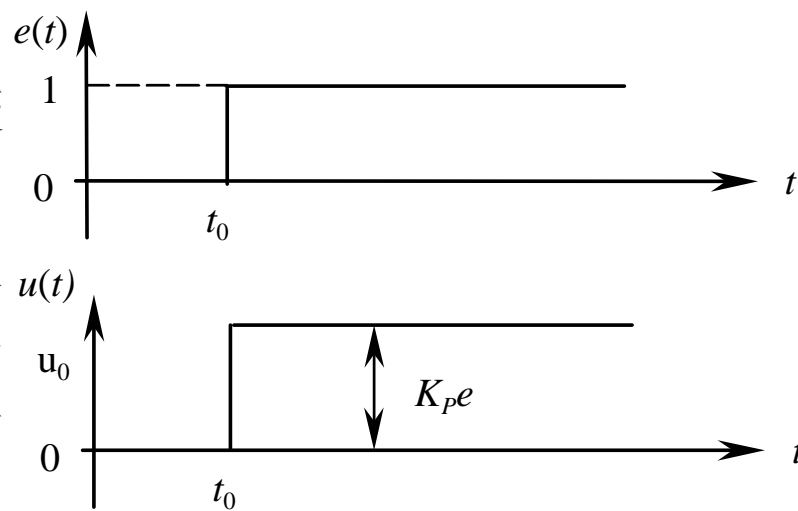
$K_p$ ——比例系数；

$u_0$ ——是控制量的基准，也就是 $e=0$ 时的一种控制作用

### 2. 比例调节器的响应

比例调节器对于偏差阶跃变化的时间响应如图所示。

比例调节器对于偏差 $e$ 是即时反应的，偏差一旦产生，调节器立即产生控制作用使被控量朝着偏差减小的方向变化，控制作用的强弱取决于比例系数 $K_p$ 的大小。



P调节器的阶跃时间响应





# 第六章 数字PID及其改进算法

## 2. 比例调节器的特点

简单、快速，有静差。

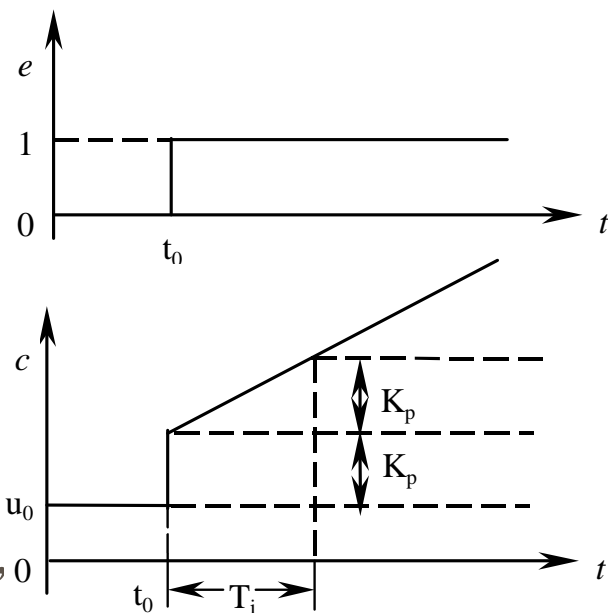
### 6.1.2 比例积分调节器

#### 1. 比例积分调节器的表达式

$$u = K_p \left( e + \frac{1}{T_I} \int_0^t e dt \right) + u_0 \quad T_I \text{——积分时间常数}$$

#### 2. 比例积分调节器的响应

PI调节器对于偏差的阶跃响应如图所示。可看出除按比例变化的成分外，还带有累计的成分。只要偏差 $e$ 不为零，它将通过累计作用影响控制量 $u$ ，并减小偏差，直至偏差为零，控制作用不在变化，使系统达到稳态。



PI调节器的阶跃响应



## 第六章 数字PID及其改进算法

### 6.1.3 比例积分微分调节器 (PID)

#### 1. 比例积分微分调节器 (PID) 表达式

$$u = K_p \left( e + \frac{1}{T_i} \int_0^t e dt + T_D \frac{de}{dt} \right) + u_0 \quad T_D \text{——微分时间} \quad (6-3)$$

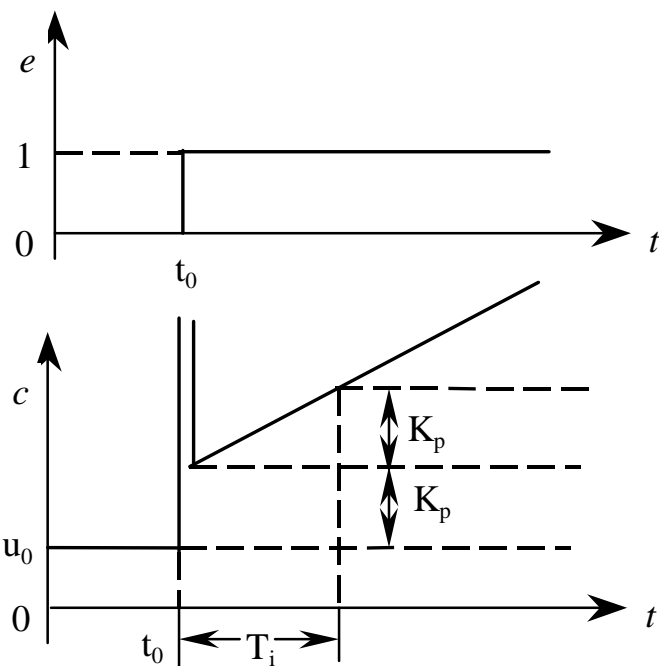
#### 2. 比例积分微分调节器 (PID) 的响应

理想的PID调节器对偏差阶跃变化的响应如图所示。

它在偏差  $e$  阶跃变化的瞬间  $t = t_0$  处有一冲击式瞬时响应，这是由附加的微分环节引起的。

#### 3. 微分环节的作用 其控制规律为

$$u_d = K_p T_D \frac{de}{dt}$$



## 第六章 数字PID及其改进算法

它对偏差的任何变化都产生——控制作用 $u_d$ 以调整系统输出，阻止偏差的变化。偏差变化越快， $u_d$ 越大，反馈校正量则越大。故微分作用的加入将有助于减小超调，克服振荡，使系统趋于稳定。

### 4. PID调节器与PI调节器的区别

PI调节器虽然可以消除静差，但却降低了响应速度。PID调节器加快了系统的动作速度，减小调整时间，从而改善了系统的动态性能。



# 第六章 数字PID及其改进算法

## 6.2 离散化的方法

微机测控系统是采用数字控制方式，所以应对模拟调节器进行离散化处理，以便微机能够通过软件实现其控制算法。

### 6.2.1 差分变化法

离散化处理的方法之一，是将模拟调节器的微分方程表达式的导数可用差分近似代替。

#### 1. 变换的基本方法

把原连续校正装置传递函数 $D(s)$ 转换成微分方程，再用差分方程近似该微分方程。

差分近似法有两种：后项差分和前项差分。微机测控离散化只采用后项差分。

#### 2. 后项差分法



## 第六章 数字PID及其改进算法

(1) 一阶后项差分

一阶导数采用近似式：
$$\frac{du(t)}{dt} \approx \frac{u(k) - u(k-1)}{T}$$

(2) 二阶后项差分

二阶导数采用近似式：

$$\frac{d^2u(t)}{dt^2} \approx \frac{u(k) - u(k-1)}{T} - \frac{u(k-1) - u(k-2)}{T} = \frac{u(k) - 2u(k-1) + u(k-2)}{T^2}$$

例题：求惯性环节  $D(s) = \frac{1}{T_1s + 1}$  的差分方程

解：由  $D(s) = \frac{U(s)}{E(s)} = \frac{1}{T_1s + 1}$  有  $(T_1s + 1)U(s) = E(s)$

化成微分方程：
$$T_1 \frac{du(t)}{dt} + u(t) = e(t)$$



## 第六章 数字PID及其改进算法

以采样周期离散上述微分方程得

$$T_1 \dot{u}(kT) + u(kT) = e(kT) \quad \text{即} \quad T_1 \dot{u}(k) + u(k) = e(k)$$

用一阶后项差分近似代替得：

$$\dot{u}(k) \approx \frac{u(k) - u(k-1)}{T} \quad \text{代入上式得} \quad \frac{T_1}{T} [u(k) - u(k-1)] + u(k) = e(k)$$

$$\text{整理得} \quad u(k) = \frac{T_1}{T+T_1} u(k-1) + \frac{T}{T+T_1} e(k)$$

### 6.2.2 零阶保持器法

1. 基本思想：离散近似后的数字控制器的阶跃响应序列，必须与模拟调节器的阶跃响应的采样值相等，即

$$D(z) = (1 - z^{-1}) Z \left[ \frac{D(s)}{s} \right]$$

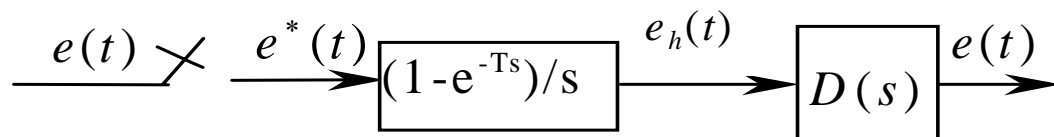


## 第六章 数字PID及其改进算法

因为  $z^{-1} = e^{-Ts}$  上式变为  $D(z) = Z\left[\frac{1-e^{-Ts}}{s}D(s)\right] = Z[H(s)D(s)]$

式中  $H(s)$  ——零阶保持器,  $H(s) = \frac{1-e^{-Ts}}{s}$   
——采样周期

2. 物理模型



微机控制就是用软件实现  $D(s)$  算式, 这样输入的信号必须经过 A/D 转换器对  $e(t)$  进行采样得到  $e^*(t)$ , 然后经过保持器  $H(s)$  将此离散信号变换成近似  $e(t)$  的信号  $e_h(t)$ , 才能加到  $D(s)$  上去。为此, 用  $D(z)$  近似  $D(s)$  求  $Z$  变换表达式时, 不能简单地只将  $D(s)$  进行变换, 而应包括  $H(s)$  在内



## 第六章 数字PID及其改进算法

例题：用零阶保持器法求惯性环节  $D(s) = \frac{1}{T_1 s + 1}$  的差分方程

解：由式  $D(z) = Z\left[\frac{1-e^{-Ts}}{s} \frac{1}{T_1 s + 1}\right] = (1-z^{-1})Z\left[\frac{1}{s(T_1 s + 1)}\right] = (1-z^{-1})Z\left[\frac{1}{s} - \frac{1}{s + \frac{1}{T_1}}\right]$

$$= (1-z^{-1})\left[\frac{1}{1-z^{-1}} - \frac{1}{1-e^{-T/T_1}z^{-1}}\right] = (1-z^{-1})\left[\frac{(1-e^{-T/T_1})z^{-1}}{(1-z^{-1})(1-e^{-T/T_1}z^{-1})}\right]$$
$$= \frac{(1-e^{-T/T_1})z^{-1}}{1-e^{-T/T_1}z^{-1}}$$

所以  $D(k) = \frac{u(k)}{e(k)} = \frac{(1-e^{-T/T_1})z^{-1}}{1-e^{-T/T_1}z^{-1}}$

整理得：  $u(k) = e^{-T/T_1}u(k-1) + (1-e^{-T/T_1})e(k-1)$





## 第六章 数字PID及其改进算法

### 6.2.3 双线性变化法

1. 基本思想：就是将  $s$  域函数与  $z$  域函数进行转换的一种近似方法。由  $z$  变换的定义有

$$z = e^{Ts} = \frac{e^{\frac{T}{2}s}}{e^{-\frac{T}{2}s}} \quad (6-11)$$

用泰勒级数展开为  $e^{\frac{T}{2}s} = 1 + \frac{T}{2}s + \frac{T^2}{8}s^2 + \Lambda$  和  $e^{-\frac{T}{2}s} = 1 - \frac{T}{2}s + \frac{T^2}{8}s^2 + \Lambda$

若近似只取前两项代入式 (6-11)，则得

$$z \approx \frac{1 + \frac{T}{2}s}{1 - \frac{T}{2}s} = \frac{\frac{2}{T}s + 1}{\frac{2}{T}s - 1} = \frac{2 + Ts}{2 - Ts} \quad \text{即 } s \text{ 近似为 } s = \frac{2}{T} \frac{z - 1}{z + 1} = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (6-15)$$



## 第六章 数字PID及其改进算法

例题：已知某连续控制器的传递函数  $D(s) = \frac{s+0.5}{(s+1)^2}$ ，试用双线性变换法求出相应的数字控制器的脉冲传递函数  $D(z)$ ，其中  $T=1\text{s}$ 。

解：由式 (6-15)，有

$$\begin{aligned} D(z) &= \frac{s+0.5}{(s+1)^2} \bigg|_{\frac{21-z^{-1}}{T1+z^{-1}}} = \frac{\frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}} + 0.5}{\left(\frac{2}{T} \frac{1-z^{-1}}{1+z^{-1}} + 1\right)^2} \stackrel{T=1}{=} \frac{2(1-z^{-1})(1+z^{-1}) + 0.5(1+z^{-1})^2}{[2(1-z^{-1}) + (1+z^{-1})]^2} \\ &= \frac{2.5 + z^{-1} - 1.5z^{-2}}{9 - 6z^{-1} + z^{-2}} = \frac{0.278(1 + 0.4z^{-1} - 0.6z^{-2})}{1 - 0.667z^{-1} + 0.111z^{-2}} \end{aligned}$$



# 第六章 数字PID及其改进算法

## 6.3 数字PID及其算法

因为微机是通过软件实现其控制算法。必须对模拟调节器进行离散化处理，这样它只能根据采样时刻的偏差值计算控制量。因此，

不能对积分和微分项直接准确计算，只能用数值计算的方法逼近。用离散的差分方程来代替连续的微分方程。根据输出量 $u(k)$ 的形式可分为位置式PID控制算法和增量式PID控制算法。

### 6.3.1 位置式PID控制算法

当采样时间很短时，可用一阶差分代替一阶微分，用累加代替积分。

$$t = KT$$

连续时间的离散化，即

1. 离散的PID表达式

积分用累加求和近似得

$$\int_0^t e(t)dt = \sum_{i=0}^{(K=0, 1, 2, \dots, n)} e(i)T = T \sum_{i=0} e(i) \quad (6-16)$$



## 第六章 数字PID及其改进算法

微分用一阶差分近似得

$$\frac{de(t)}{dt} \approx \frac{e(k) - e(k-1)}{T} \quad (6-17)$$

$T$  ——为采样周期;

$k$  ——为采样序号,  $k=0, 1, 2, \dots$ ;

$e(k)$  ——系统在第 $k$ 次采样时刻的偏差值;

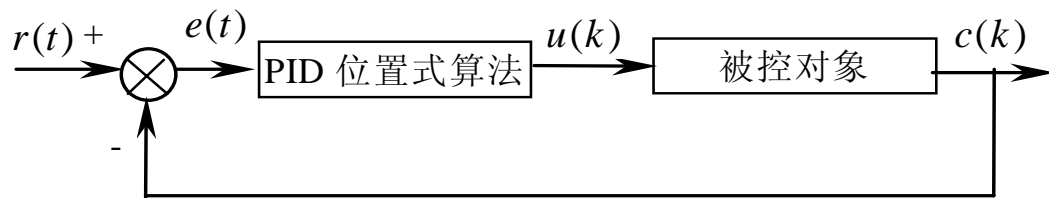
$e(k-1)$  ——为系统在第 $k-1$ 次采样时刻的偏差值。

将式(6-16)和式(6-17)代入式(6-3), 则可得到离散的PID表达式

$$u(k) = K_p \left\{ e(k) + \frac{T}{T_i} \sum_{i=0}^k e(i) + \frac{T_D}{T} [e(k) - e(k-1)] \right\} + u_0 \quad (6-18)$$

2. 物理模型

如图所示。



PID位置式算法控制原理图



## 第六章 数字PID及其改进算法

式（6-18）表示的控制算法提供了执行机构的位置 $u(k)$ （如阀门开度），即输出值与阀门开度一一对应，所以称为位置式PID控制算法。

### 3. 离散化的PID位置式控制算法的编程表达式

对于式（6-18），令  $K_I = \frac{K_P T}{T_I}$ ，  $K_D = \frac{K_D T_D}{T}$   $K_I$  称为积分系数  
 $K_D$  称为微分系数

则离散化的PID位置式控制算法的编程表达式为

$$u(k) = K_P e(k) + K_I \sum_{i=0}^k e(i) + K_D [e(k) - e(k-1)] + u_0$$

由式可以看出，每次输出与过去的状态有关，要想计算 $u(k)$ ，

不仅涉及到 $e(k-1)$ 和 $e(k-2)$ ，且须将历次相加。



## 第六章 数字PID及其改进算法

上式计算复杂，浪费内存。考虑到第次采样时有

$$u(k-1) = K_p \left\{ e(k-1) + \frac{T}{T_I} \sum_{i=0}^{k-1} e(i) + \frac{T_D}{T} [e(k-1) - e(k-2)] \right\} + u_0$$

两式相减，得

$$u(k) - u(k-1) = K_p \left\{ e(k) - e(k-1) + \frac{T}{T_I} e(k) + \frac{T_D}{T} [e(k) - 2e(k-1) + e(k-2)] \right\}$$

$$u(k) = u(k-1) + K_p \left( 1 + \frac{T}{T_I} + \frac{T_D}{T} \right) e(k) - K_p \left( 1 + \frac{2T_D}{T} \right) e(k-1) + K_p \frac{T_D}{T} e(k-2)$$

$$= u(k-1) + \alpha_0 e(k) - \alpha_1 e(k-1) + \alpha_2 e(k-2) \quad (6-21)$$

$$\text{其中 } \alpha_0 = K_p \left( 1 + \frac{T}{T_I} + \frac{T_D}{T} \right), \quad \alpha_1 = K_p \left( 1 + \frac{2T_D}{T} \right), \quad \alpha_2 = K_p \frac{T_D}{T}$$



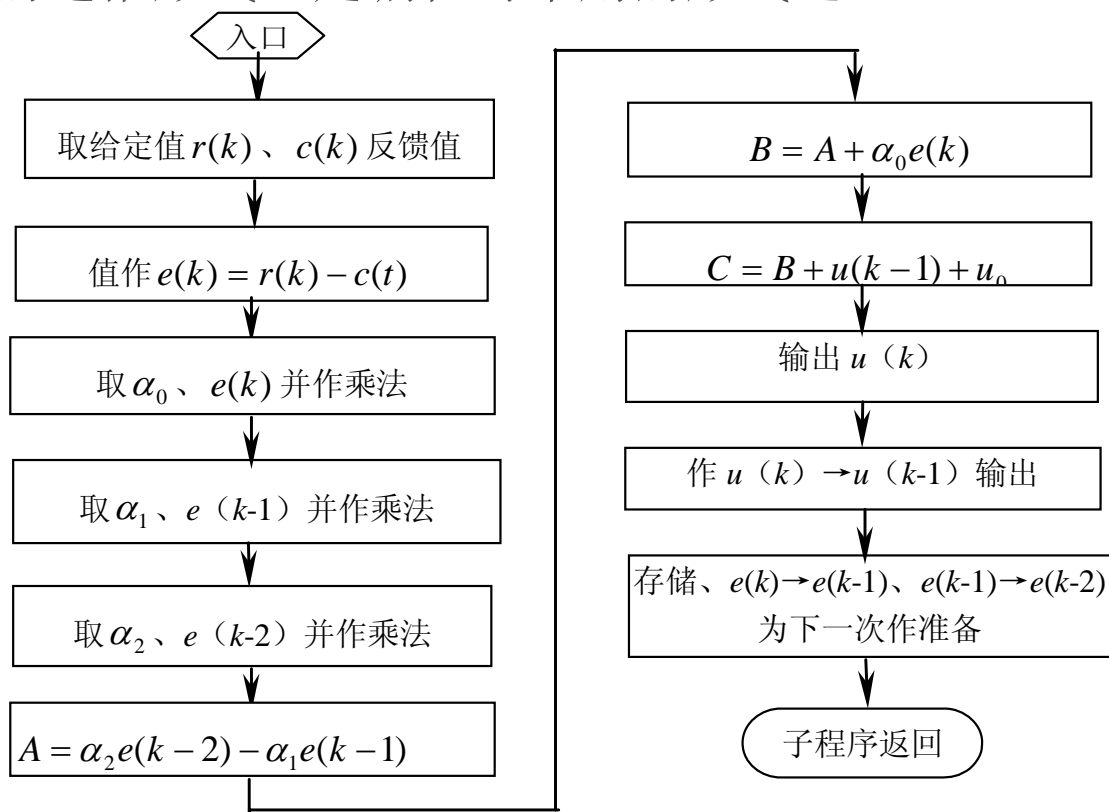
## 第六章 数字PID及其改进算法

式(6-21)是PID的递推形式,是编程时常用的形式之一。

### 4. 软件算法流程图

流程图如图所示。

其中系数  $\alpha_0$ 、 $\alpha_1$ 、 $\alpha_2$  可以预先算出。



位置式PID算法程序框图

# 第六章 数字PID及其改进算法

## 6.3.2 PID增量式控制算法

当执行机构需要的不是控制量的绝对数值，而是其增量（例如去驱动步进电机）时，要采用PID增量式控制算法。

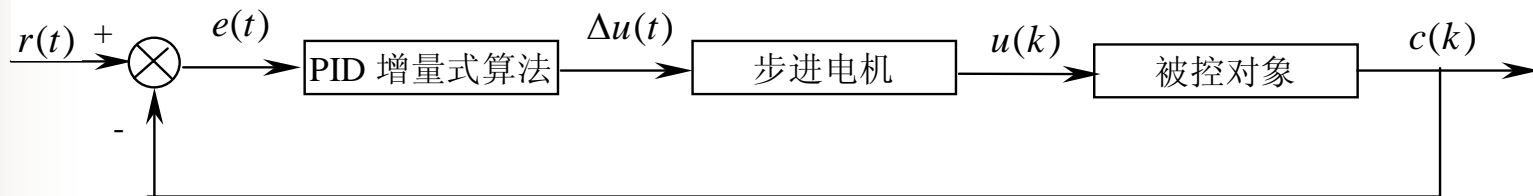
1. 采用PID增量式控制算法表达式

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

则  $\Delta u(k) = \alpha_0 e(k) - \alpha_1 e(k-1) + \alpha_2 e(k-2)$

式中  $\alpha_0$ 、 $\alpha_1$ 、 $\alpha_2$ 与式（6-21）中一样。

2. 物理模型 如图所示。



PID增量式算法控制原理图





## 第六章 数字PID及其改进算法

### 3. 软件算法流程图

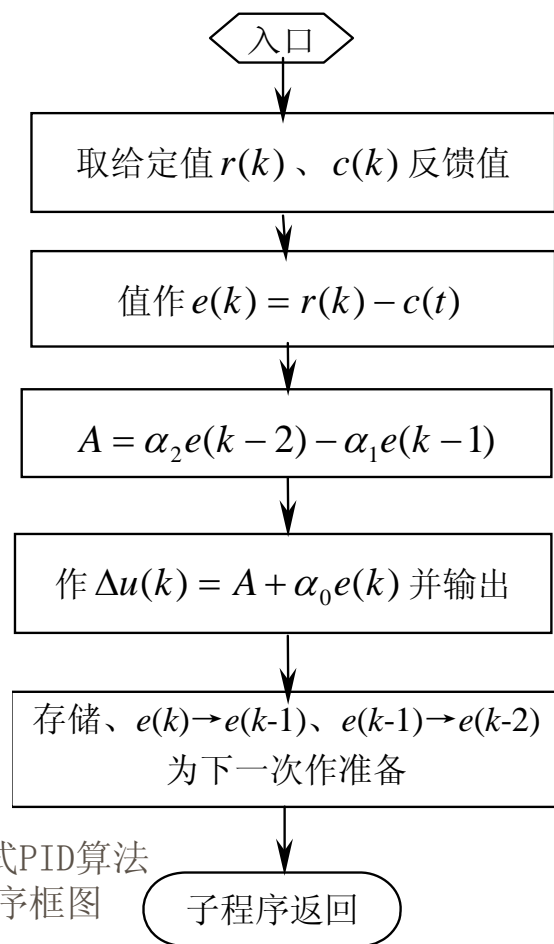
流程图如图所示。在实际编程时

$\alpha_0$ 、

$\alpha_1$ 、 $\alpha_2$ 可预先算出，存入预先固定的单元，设初值 $e(k-1)$ 、 $e(k-2)$ 为0。

### 4. 增量式PID算法的优点

(1) 位置式算法每次输出与整个过去状态有关，计算式中要用到过去偏差的累加值，容易产生较大的积累误差。而增量式只需计算增量，当存在计算误差或精度不足时，对控制量计算的影响较小。



增量式PID算法  
程序框图

## 第六章 数字PID及其改进算法

(2) 对于位置式算法，控制从手动切换到自动时，必须先将计算机的输出值设置为原始阀门开度 $u_0$ ，才能保证无冲击切换。如果采用增量算法，则由于算式中不出现 $u_0$ ，易于实现手动到自动的无冲击切换。此外，在计算机发生故障时，由于执行装置本身有寄存作用，故可仍然保持在原位。

### 5. 应用

若执行部件不带积分部件，其位置与计算机输出的数字量是一一对应的（如电液伺服阀），就要采用位置式算法。若执行部件带积分部件（如步进电机、步进电机带动阀门或带动多圈电位器）时，就可选用增量式算法。



## 第六章 数字PID及其改进算法

### 6.4 标准PID算法的改进

#### 6.4.1 “饱和”作用的抑制

##### (1) 原因

在实际过程中，控制变量 $u$ 因受到执行元件机械和物理性能的约束而控制在有限范围内，即  $u_{\min} \leq u(k) \leq u_{\max}$

其变化率也有一定的限制范围，即  $|\dot{u}| < \dot{u}_{\max}$

##### (2) 饱和效应

如果由计算机给出的控制量 $u$ 在上述范围内，那么控制可以按预期的结果进行。一旦超出上述范围，那么实际执行的控制量就不再是计算值，由此将引起不期望的效应。



## 第六章 数字PID及其改进算法

### 1. PID位置算法的积分饱和和作用及其抑制

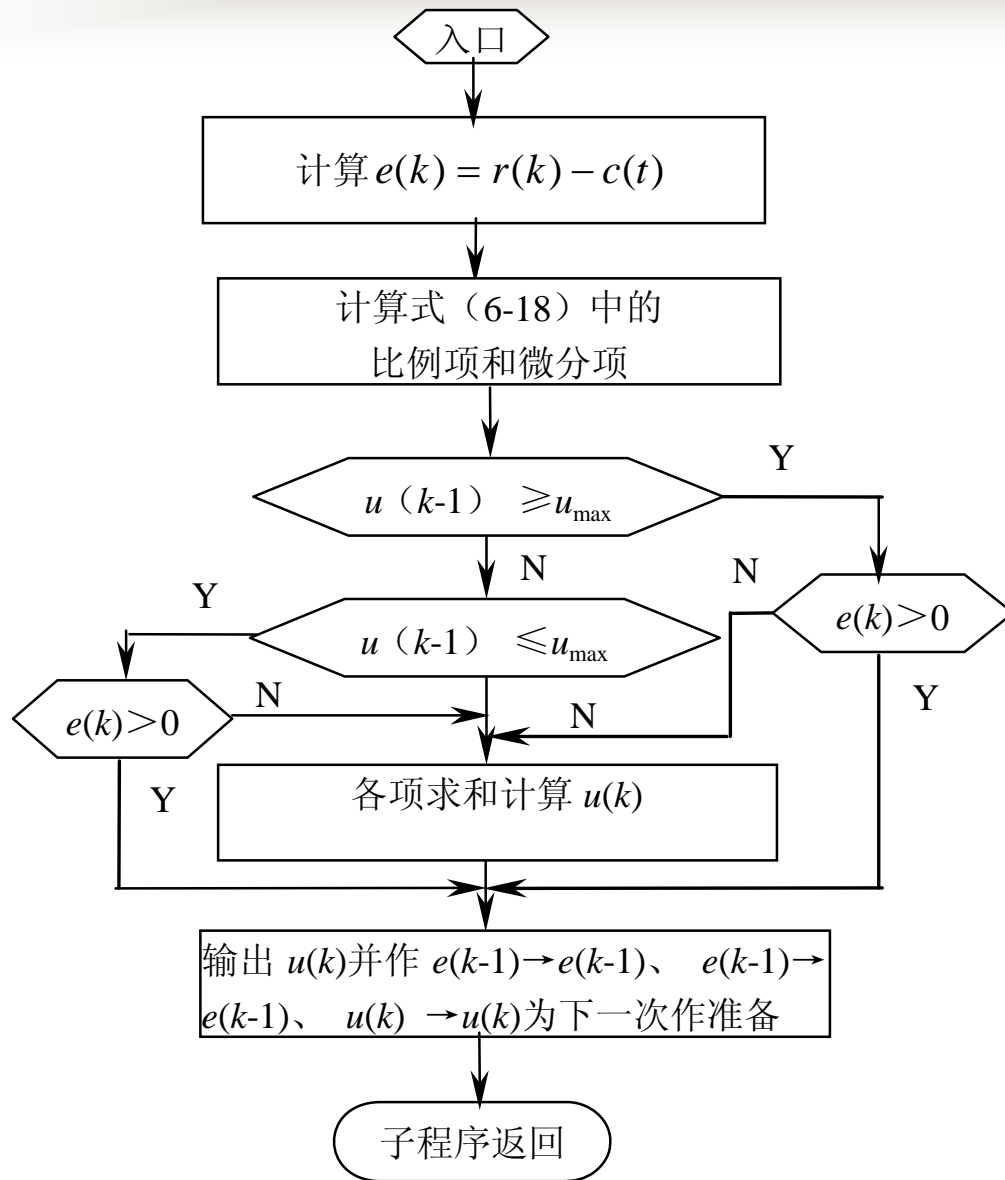
#### (1) 遇限削弱积分法

① 基本思想 一旦控制变量进入饱和区，将只执行削弱积分项的运算而停止进行增大积分项的运算。具体地说，在计算 $u(k)$ 时，将判断上一时刻的控制量 $u(k)$ 是否已超出限制范围，如果已超出，

那么将根据偏差的符号，判断系统输出是否在超调区域，由此决定是否将相应偏差计入积分项。

② 算法流程图如下图所示。





## 第六章 数字PID及其改进算法

(2) 积分分离法 减小积分饱和的关键在于不能使积分项累积过大。上面的修正方法是一开始就积分，但进入限制范围后即停止累积。这里介绍的积分分离法正好与其相反，它在开始时不进行积分，直到偏差达到一定的阈值后才进行积分累计。这样，一方面防止了一开始有过大的控制量，另一方面即使进入饱和后，因积分累积小，也能较快退出，减少了超调。

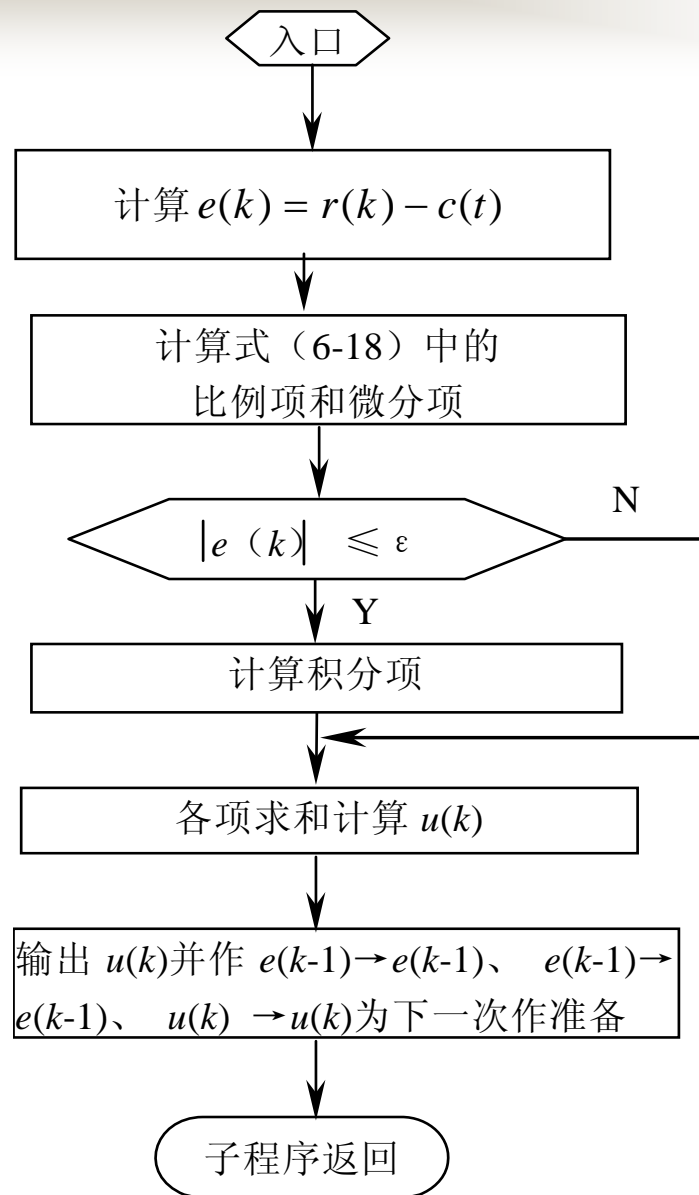
将式(6-19)改写为下面形式

$$u(k) = K_P e(k) + K_L K_I \sum_{i=0}^k e(i) + K_D [e(k) - e(k-1)]$$

式中  $K_L = \begin{cases} 1, & \text{当 } e(k) \leq \varepsilon \text{ 时} \\ 0, & \text{当 } e(k) > \varepsilon \text{ 时} \end{cases}$  其中  $\varepsilon$  为预定门限值。



采用积分分离法的PID位置算法流程图如图所示。系统输出在门限外时，该算法相当于PD调节器。只有在门限范围内，积分部分才起作用，以消除系统静差。



## 第六章 数字PID及其改进算法

### 2. PID增量算法饱和和作用及其抑制

在PID增量算法中，由于执行元件本身是机械或物理的积分储存单元，在算法中不出现累加和式，所以不会发生位置算法那样的累积效应，这样就直接避免了导致大幅度超调的积分累积效应。这是增量式算法相对于位置算法的一个优点。但是，在增量算法中，却有可能出现比例及微分饱和现象。

(1)“积累补偿法” 基本思想是将那些因饱和而未能执行的增量信息积累起来，一旦有可能时，再补充执行，这样就没有遗失，动态过程也得到了加速。

(2)方法 在每次计算积分项时，应判断其 $e(kT)$ 的符号，是否将继续增大累加器的积累。如果增大，就将积分项略去，这样，可以使累加器的数值积累不致过大，从而避免了积分饱和现象。





## 第六章 数字PID及其改进算法

### 6.4.2 不完全微分PID算法

在标准的PID算式中，当有阶跃信号输入时，微分项急剧增加，容易引起调节过程的振荡，导致调节品质下降。

1. 不完全微分PID算法基本思想 仿照模拟调节器的实际微分调节，加入惯性环节，以克服完全微分的缺点。

2. 算法的传递函数表达式为

式中  $K_D$ ——微分增益。

$$\frac{U(s)}{E(s)} = K_P \left[ 1 + \frac{1}{T_I s} + \frac{T_D s}{1 + \frac{T_D}{K_D} s} \right]$$



## 第六章 数字PID及其改进算法

将上式分成比例积分和微分两部分，则

$$U(s) = U_{PI}(s) + U_D(s)$$

$$\text{其中 } U_{PI} = K_P \left[ 1 + \frac{1}{T_I s} \right] E(s) \quad U_D(s) = K_P \frac{T_D s}{1 + \frac{T_D}{K_D} s} E(s)$$

将上式化成微分方程，再用一阶向后差分近似代替微分，得到不完全微分的PID增量式算式为

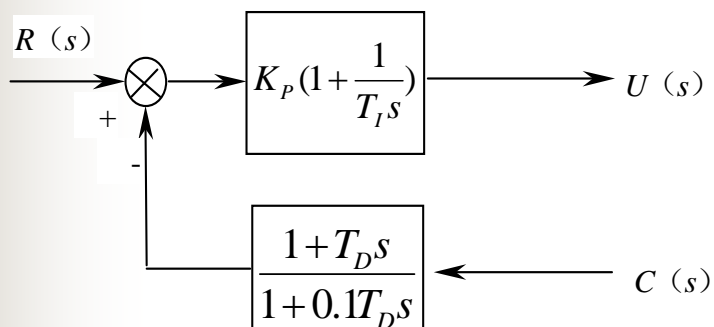
$$\begin{aligned} \Delta u_D(k) = & K_P [e(k) - e(k-1)] + K_P \frac{T}{T_I} e(k) \\ & + K_P \frac{T_D}{T_S} [e(k) - 2e(k-1) + e(k-2)] + \alpha [u_D(k-1) + u_D(k-2)] \end{aligned}$$



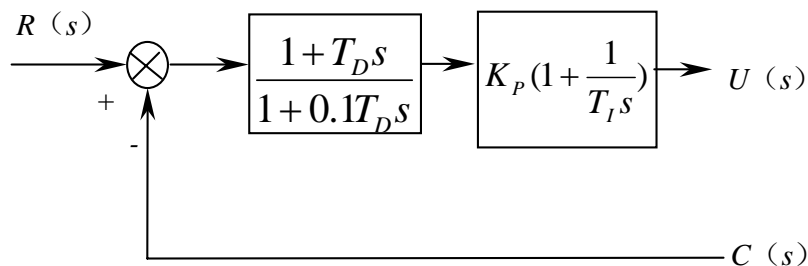
## 第六章 数字PID及其改进算法

### 6.4.3 微分先行PID算法

微分先行PID算法的实质是将微分运算提前进行。有两种结构，一种是对输出量的微分；另一种是对偏差的微分，如图所示。



对输出量先行微分PID算法



对偏差量先行微分PID算法



## 第六章 数字PID及其改进算法

在第一种结构中，只对输出量 $c(t)$ 进行微分，它适用于给定量频繁升降的场合，可以避免升降给定值时所引起的超调量过大，阀门动作过分剧烈振荡。

后一种结构是对偏差值先行微分，它对给定值和偏差值都有微分作用，适用于串级控制的副控制回路。因为副控制回路的给定值是由主控回路给定的，也应对其作微分处理，因此，应该在副控制回路中采用偏差PID控制。

### 6.4.4 带死区的PID控制

在控制精度要求不高、控制过程要求平稳的测控系统中，为了避免控制动作过于频繁，消除由此引起的振荡，可以人为的设置一个不灵敏区 $B$ ，即带死区的PID控制。只有不在死区范围内时，才按PID算式计算控制量。



## 第六章 数字PID及其改进算法

### 6.4.5 给定值突变时对控制量进行阻尼的算法

在给定值发生迅速变化时，例如阶跃变化时，为了防止出现过大的控制量，一般可采用下列办法：

- (1) 前置滤波器
- (2) 修改算法中对给定值变化敏感的项



# 第六章 数字PID及其改进算法

## 6.5 PID调节器参数的整定

### 6.5.1 凑试法确定PID调节参数

凑试法是通过模拟或闭环运行（如果允许的话）观察系统的响应曲线（例如阶跃响应），然后根据各调节参数对系统响应的大致影响，反复凑试参数，

以达到满意的系统响应，从而确定PID调节参数。

#### 1. 1.PID的参数对系统性能的影响

(1) 比例系数 $K_p$ 对系统性能的影响 增大比例系数 $K_p$ 一般将加快系统的响应，在有静差的情况下有利于减小静差。但过大的比例系数会使系统有较大的超调，并产生振荡，使稳定性变坏。

(2) 积分时间 $T_I$ 对系统性能的影响：增大积分时间 $T_I$ 有利于减小超调，减小振荡，使系统更加稳定，但系统静差的消除将随之减慢。

(3) 微分时间 $T_D$ 对系统性能的影响：增大微分时间 $T_D$ ，也有利于加快系统响应，使超调量减小，稳定性增加，但系统对扰动的抑制能力减弱。



## 第六章 数字PID及其改进算法

### 2. 凑试法确定PID调节参数

在凑试时，可参考以上参数对控制过程的影响趋势，对参数实行下述先比例，后积分，再微分的整定步骤。

(1) 首先只整定比例部分。即将比例系数由小变大，并观察相应的系统响应，直至得到反应快、超调小的响应曲线。

(2) 如果在比例调节的基础上系统的静差不能满足设计要求，则须加入积分环节。整定时首先置积分时间  $T_I$  为一较大值，并将经第一步整定得到的比例系数略为缩小（如缩小为原值的0.8倍），然后减小积分时间，在保持系统良好动态性能的情况下，使静差得到消除。

(3) 若使用比例积分调节器消除了静差，但动态过程经反复调整仍不能满意，则可加入微分环节，构成比例积分微分调节器。



# 第六章 数字PID及其改进算法

## 常见被控对象的PID参数经验选择范围

被调量	特点	$K_P$	$T_I$ (min)	$T_D$ (min)
流量	对象时间常数小, 并有噪声, 故 $K_P$ 较小, $T_I$ 较短, 不用微分	1~2.5	0.1~1	
温度	对象为多容系统, 有较大滞后, 常用微分	1.6~5	3~10	0.5~3
压力	对象为容量系统, 滞后一般不大, 不用微分	1.4~3.5	0.4~3	
液位	再允许有静差时, 不必用积分, 不用微分	1.25~5		





## 第六章 数字PID及其改进算法

### 6.5.2 实验经验法确定PID调节参数

#### 1. 扩充临界比例度法

扩充临界比例度法是以模拟调节器中使用的临界比例度法为基础的一种PID数字控制器参数的整定方法。

#### 2. 阶跃响应曲线法

在上述方法中，不需要预先知道对象的动态性能，而是直接在闭环系统中进行整定的。如果已知系统的动态特性曲线，数字控制器的参数的整定也可采用类似模拟调节器的响应曲线法来进行，称为阶跃响应曲线法，也称扩充响应曲线法。这一方法适用于多容量自平衡系统。



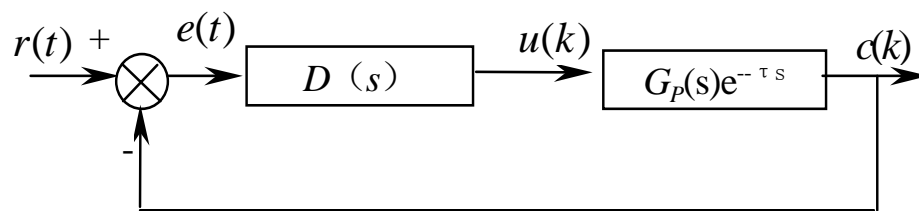
## 第六章 数字PID及其改进算法

### 6.6 纯滞后补偿控制技术

#### 6.6.1 纯滞后补偿算法

##### 1. 纯滞后补偿原理

在工业控制中，不少控制对象往往具有纯滞后的性质。对象的纯滞后性质，会导致控制作用不及时，引起系统超调和振荡。带纯滞后环节的控制系统如图所示。



其传递函数为

$$G(s) = \frac{K_p e^{-\tau s}}{1 + T_1 s} = G_p(s) e^{-\tau s}$$



## 第六章 数字PID及其改进算法

对于这样的对象，用一般的PID调节规律是难以得到好的动态特性的，特别是当滞后时间  $\tau$  较大时会产生持续振荡，使系统的稳定性降低。一个补偿的办法是采用史密斯（Smith）预测器。

### 2. 史密斯纯滞后补偿的原理

并接一补偿环节，用来补偿被控对象中的纯滞后部分，这个环节称为史密斯预测器。整个纯滞后补偿器的传递函数为

$$D'(s) = \frac{D(s)}{1 + D(s)G_P(s)(1 - e^{-\tau s})}$$

经补偿后，系统的闭还传递函数为

$$\Phi(s) = \frac{D'(s)G_P(s)e^{-\tau s}}{1 + D'(s)G_P(s)e^{-\tau s}} = \frac{D(s)G_P(s)}{1 + D(s)G_P(s)} e^{-\tau s}$$

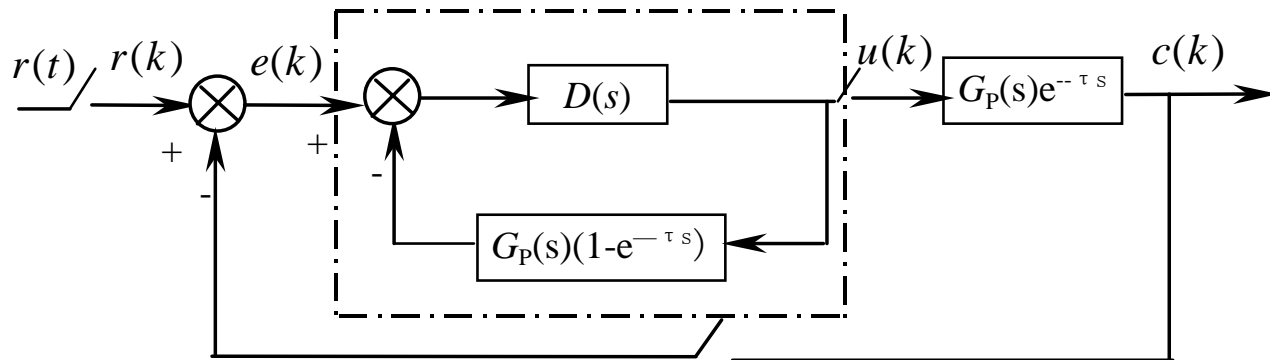


## 第六章 数字PID及其改进算法

上式表明滞后项在闭环控制回路之外，不影响系统的稳定性。由拉氏变换平移定理可知，仅将控制作用在时间上推迟了  $\tau$  时间，控制系统的过度过程及其它的性能指标指标都与对象特性的完全相同。这样就消除了滞后环节对控制系统的性能影响。

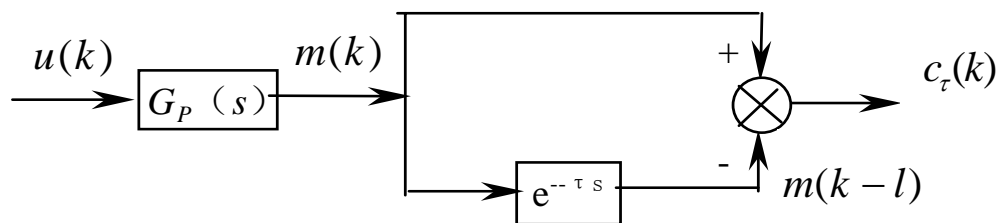
### 6.6.2 纯滞后补偿环节的数字控制器

带纯滞后补偿的离散控制系统如图所示。



## 第六章 数字PID及其改进算法

将纯滞后补偿环节首先分为两个环节，如图所示。先计算PID控制器的输出 $u(k)$ ，再计算预估器 $G_P(s)$ 的输出 $m(k)$ ，其中史密斯预测器的输出 $c_\tau(k)$ 为



$$c_\tau(k) = m(k) - m(k-l)$$

式中， $l = \tau / t$ 是纯滞后折合采样周期 $T$ 的倍数。

在程序中采用下面方法可得到 $m(k-l)$ ：在内存中专门设置 $l$ 个单元用存放历史数据，当前采样值总是存放在第1个单元（具体做法是将历史数据区的各单元依次右移，空出第1个单元），这样，从 $l$ 号单元输出的信号就是滞后了 $l$ 个采样周期的信号 $m(k-l)$ 。

