

最新工业炉窑

设计、改造及节能环保 治理新工艺新技术

实用手册

ZUIXIN GONGYE YAOLU SHEJI GAIZAO JI JIENENG
HUANBAO ZHILI XINGONGYI XINJISHU SHIYONG SHOUCHE



安徽文化音像出版社

第四章 过程控制算法

第一节 基本控制算法——PID 算法

DDC 级工业控制器的主要任务是执行控制算法。控制规律是反映 DDC 性能的核心。随着控制理论和计算技术的发展,控制算法也在不断地发展,从单变量定值调节到多变量相关调节,直至自适应、最优控制技术的应用。近几年,在过程控制中,模糊智能控制技术也得到重视和发展。过程控制器的功能早已超出了早期单一 PID 控制的范围,不断使计算机速度快、精度高,存储信息量大和逻辑判断能力强的特点得到充分发挥。

目前数字调节器算法的设计、分析有多种方法,它们基本上可归纳为两种:一种是对连续控制系统进行模拟的方法;另一种是用数字控制理论进行直接设计和分析的方法(如用 Z 变换的方法)。在被控对象的时间常数远大于采样周期时,对调节器的设计采用第一种途径是可行的。这种方法可以延用连续控制系统的成熟的经验,掌握和运用相对要简单。在对象的时间常数与采样周期处于或接近同一数量级时,简单地进行模拟连续系统的设计将得不到预期的效果。这时就需要用数字控制理论进行分析和设计。这种方法理论上奠定了数字系统的基础,因此是计算机控制技术控制理论研究的主要工具。

在炉窑和化工行业,仪表过程控制已有相当长的历史。PID 调节器及其它各种配套的控制技术比较成熟。控制对象的过程一般较慢,所以目前在过程计算机控制中主要控制算法还是仿真模拟仪表的基本方法。

PID 控制算法是在模拟仪表控制中发展、并沿用至今的一种古典的控制调节算法。这种算法实际上也是一种优化算法。它的算式简单,特性直观,调试使用方便,尤其是它不需要准确的对象数学模型。对不同的对象均可通过系数调整

进行闭环控制。因此,即使是计算机控制技术发展的今天,在国内外过程控制中仍做为一种主要的调节器算法被应用着。随着 PID 技术在计算机控制系统中的应用,根据计算机特点改进的各种形式的 PID 调节器,使 PID 调节技术在性能指标、适应范围等各方面都有了仪表系统中无法比拟的提高。

一、PID 基本算式

用单参数的 PID 调节器构成的控制系统是一个单变量小闭环控制系统(如图 7-4-1 所示)。

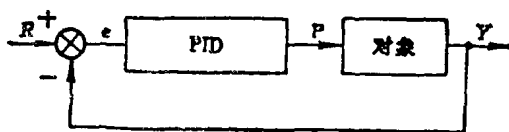


图 7-4-1 单闭环控制系统框图

常用模拟调节器的理论算式为

$$P = K_P \left(e + \frac{1}{T_I} \int e dt + T_D \frac{de}{dt} \right) \quad (7-4-1)$$

式中 $e = R - Y$ 为偏差值。

R ——给定值。

Y ——被控变量。

K_P, T_I, T_D 分别为控制算式中的比例、积分、微分系数。

式(7-4-1)调节方案的特点是每次输出量 P 都对应于阀门的实际位置,故又称为 PID 算法的位置式。这种表达式的另一个显著的特点是控制作用参数 (K_P, T_I, T_D) 有相对独立性,即改变一个控制作用参数时,只影响一个调节作用,不影响其它两个调节作用。

为了用计算机实现 PID 规律,当采样时间 T 很小时,可以通过离散化,将这一方程直接化为差分方程。为此用一阶差分代替一阶微分,用累加代替积分。这时可用矩形或梯形积分来求连续积分的近似值。用矩形积分时,差分方程的表达形式如下:

$$P(k) = K_P \left\{ e(k) + \frac{\Delta T}{T_I} \sum_{i=0}^k e(i) + \frac{T_D}{\Delta T} [e(k) - e(k-1)] \right\} \quad (7-4-2)$$

$$K_P \left[e(k) + \frac{1}{T_I} \sum_{i=0}^k e(i) \Delta T + T_D \frac{\Delta e(k)}{\Delta T} \right]$$

$P(k)$ ——第 k 次采样时控制机的输出量;

$e(k)$ ——第 k 次采样时的偏差值, $e(k) = R(k) - Y(k)$;

$R(k)$ ——第 k 次采样时的给定值;

$Y(k)$ ——第 k 次测量采样值;

K_p, T_I, T_D ——比例、积分、微分系数;

k ——采样序号, $k = 1, 2, 3 \cdots n$;

ΔT ——采样间隔周期(s);

$\Delta e(k)$ ——本次和上次偏差值之差, $\Delta e(k) = e(k) - e(k-1)$ 。

上面的控制算式是位置式, 它的输出是全量形式, 每次的 $P(k)$ 值与控制阀的阀位一一对应, 本次输出与过去的所有状态有关。这种非递推的公式, 不仅需要本次与上次采样的偏差 $e(k)$ 和 $e(k-1)$, 而且还需要用到 $e(0) \sim e(k)$ 的所有值, 当 k 很大时要占用很多内存, 并要花费计算机大量的时间去计算。因此, 直接使用上式计算是很不方便的。为此, 可把它化成递推公式。

根据式(7-4-2)可写出 $k-1$ 采样时刻的输出为:

$$P(k-1) = K_p \left[e(k-1) + \frac{1}{T_I} \sum_{i=0}^{k-1} e(i) \Delta T + T_D \frac{\Delta e(k-1)}{\Delta T} \right] \quad (7-4-3)$$

用式(2-4-2)减去式(2-4-3)得:

$$\begin{aligned} \Delta P(k) = P(k) - P(k-1) &= K_p \{ [e(k) - e(k-1)] \\ &+ \frac{1}{T_I} e(k) \Delta T + T_D \frac{\Delta e(k) - \Delta e(k-1)}{\Delta T} \} \end{aligned} \quad (7-4-4)$$

因为 $\Delta e(k) = e(k) - e(k-1)$

$$\Delta e(k-1) = e(k-1) - e(k-2)$$

所以

$$\begin{aligned} \Delta P(k) &= K_p \{ [e(k) - e(k-1)] \\ &+ \frac{1}{T_I} e(k) \Delta T + \frac{T_D}{\Delta T} [e(k) - 2e(k-1) + e(k-2)] \} \\ &= K_p \Delta e(k) + K_I e(k) + K_D [\Delta e(k) - \Delta e(k-1)] \\ &= K_p \Delta e(k) + K_I e(k) + K_D \Delta^2 e(k) \end{aligned} \quad (7-4-5)$$

其中 $K_I = K_p \cdot \frac{\Delta T}{T_I}$ ——积分系数;

$K_D = K_p \cdot \frac{T_D}{\Delta T}$ ——微分系数;

$$\Delta^2 e(k) = \Delta e(k) - \Delta e(k-1)。$$

式(2-4-5)计算出的 $\Delta P(k)$ 是输出量的增量,因此此式又称为增量式。

增量式一般用于控制有积分能力的执行机构(如步进电机)。这时执行机构相当于累加器。对计算机的输出增量 ΔP 进行累加,以实现阀门的位置调整(即 $P = \int \Delta P$ 的积分作用)。位置式和增量式构成的闭环控制的方框图如图7-4-2所示。从图上可以看出,增量式与位置式在闭环控制中并无本质的区别,只是增量式中将原来积分运算分到其它部件完成,步进电机作为输出部件起着积分器的作用。

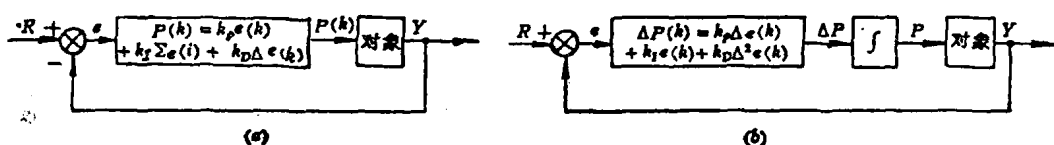


图7-4-2 增量式PID与位置式PID方框图

(a)位置式; (b)增量式。

使用增量式有它独有的优点:

1. 计算机只输出控制增量,即阀门的变化部分的量。因此,误动作影响小,必要时通过逻辑判断限制或禁止本次输出,不会严重影响系统的状态。

2. 手/自动切换冲击小,故障时易于保持原状态。不论是位置式还是增量式,在系统开始启动或手/自动切换时总要先设定一个与手动输出相对应的 $P(k-1)$ 值,然后才能变动,这样才能做到无冲击切换。在增量式中输出与现行执行器位置无关,所以处理起来比较简单。

3. 增量算式的计算结果,只与前后三次的测量偏差有关。因此,调试和修改以获得较好的控制效果比较容易。尤其是它本身无量的累积。故算式中无积分超调的问题。

对于外部执行机构无积分能力的系统(如用DDZ型电动执行机构),也可使用增量式算法,只是积分的运算在另一式中以递推形式给出。

在增量算式运算完成后,接着运算递推式,如下所示:

$$\Delta P(k) = K_p \Delta e(k) + K_I e(k) + K_D \Delta^2 e(k)$$

$$P(k) = P(k-1) + \Delta P(k) \quad (7-4-6)$$

从上两式看出,它实际上是把位置算式分开,第一式计算增量,第二式递推积分。这们的位置输出方式,仍保留了增量式的一些优点。

在分析控制系统特性时,有时要用到 PID 的脉冲传递函数形式。写出 PID 的脉冲传递函数形式,可以从它的差分方程入手:

$$\left. \frac{df(t)}{dt} \right|_{i=k\Delta T} \approx \frac{1}{\Delta T} [f(k\Delta T) - f((k-1)\Delta T)]$$

两边取 Z 变换:

$$\mathcal{Z} \left[\left. \frac{df(t)}{dt} \right|_{i=k\Delta T} \right] = \frac{1}{\Delta T} (1 - Z^{-1}) F(Z)$$

因此数字微分器的 Z 传递函数为

$$G_D(Z) = \frac{T_D}{\Delta T} (1 - Z^{-1})$$

对于图 7-4-3 的矩形数字积分器的 Z 传递函数为

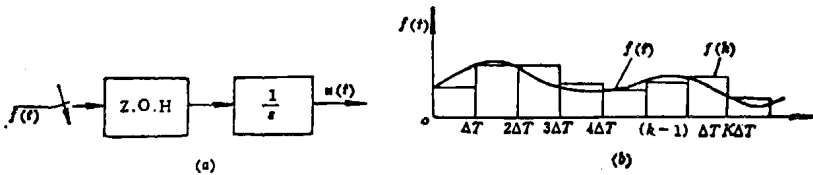


图 7-4-3 具有采样保持器的等价矩形积分

(a)积分器框图; (b)积分曲线。

$$G_I(Z) = T_I Z \left[\frac{1 - e^{-\Delta T}}{S} \cdot \frac{1}{S} \right] = \frac{T_I \Delta T}{Z - 1} = T_I \Delta T \frac{Z^{-1}}{1 - Z^{-1}}$$

PID 数字控制器方框图如图 7-4-4 所示。

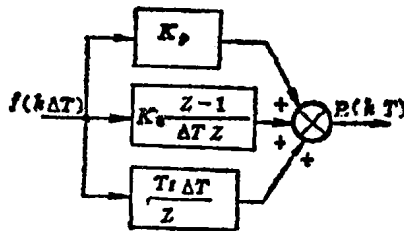


图 7-4-4 数字 PID 控制器方框图(一)

数字控制器的传递函数确定后,控制器可直接由计算机来实现。算子 Z^{-1} 可以理解

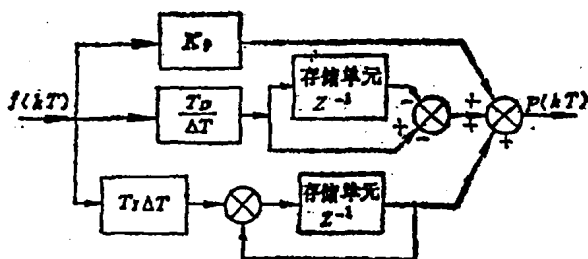


图 7-4-5 数字 PID 控制器方框图(二)

为延迟时间为采样周期 ΔT 秒的延迟环节。在机内是这样实现的。将一个变量储存在机内一个合适的单元,然后经过 ΔT 秒后将它取出来。图 7-4-5 是 PID 数字控制器的方框图之二。

二、标准 PID 算法实际应用的几个问题

在实际生产应用中,在不同的运行条件下,标准 PID 算法控制器会遇到不同的问题,这些问题会影响算法的控制效果。在计算机控制器中,由于计算机性能的优越性,为改进 PID 标准算法的不足创造了条件。这里介绍几个围绕标准 PID 算式的改进方法,对 PID 算法非线性限制条件的应用和一些变形 PID 算法的应用将在下一节讨论。

(一) 定点机运算积分丢失问题

控制器中引入积分作用,与模拟调节器一样,是为了消除系统的静态偏差。但在不同的控制算法中积分作用的形式也不一样。在位置式 PID 算法中积分作用如下式:

$$P_I(k) = \frac{K_P \Delta T}{T_I} \sum_{i=1}^k e(i) = K_I \sum_{i=0}^k e(i) \quad (7-4-7)$$

在增量式控制算法中积分作用由下式决定:

$$\Delta P_I(k) = \frac{K_P \Delta T}{T_I} e(k) = K_I e(k) \quad (7-4-8)$$

由于工业控制机有一个字长的精度限制问题,故当运算结果超过机器字长精度表示的范围时,机器就作为“零”将数丢掉。下面举例说明定点机对积分项运算结果的影响。

例如,当控制某炉出口温度,给定值为 1200°C ,测量值为 1205°C ,机器字长为十进制四位,定点设在最高位。此时机器用十进制 4 位字长定点表示温度偏差时,有

$$e(k) = R - Y(k) = 0.1200 - 0.1205 = -0.0005$$

假设

$$K_P = 1, \Delta T = 3.0\text{s}, T_I = 3.00\text{s}$$

则

$$K_I = \frac{K_P \Delta T}{T_I} = \frac{1}{10} = 0.1000$$

于是 PID 增量算式的积分项输出为:

$$\Delta P_I(k) = \frac{K_P \Delta T}{T_I} e(k) = \frac{1}{10}(-0.0005) = -0.0005$$

运算结果超过字长而作为“零”丢掉了,即

$$\Delta P_I(k) = 0$$

可见失去了积分的作用,使偏差 5℃ 始终存在,造成余差无法予以消除。只有当偏差达到 10℃ 时,才有积分式的输出和积分作用。这样,势必造成控制过程的余差。虽然可通过增强积分作相,即减小 T_I ,能使余差减小,但是从控制理论可知,积分作用的增强往往使系统的稳定裕度减小而振荡倾向增强,有时又不允许。

不难看出,当采用位置式算式时,其积分项的输出为式(2-4-7)。由于先进行累加再乘以 K_I ,这样虽然一次的 $K_I e(i)$ 运算结果为机器零,但经过若干次的累加之后, $P_I(k)$ 的输出总会大于机器零,也就具备了消除余差的功能。在具体作法上。是在程序编制时,设计一个累加单元,存放 $\sum_{i=0}^k e(i)$,每次计算的偏差 $e(i)$ 都与这个单元相当,然后乘以 K_I ,再与比例、微分作用一起进行相加后输出。

当采用增量式时,为了克服余差,可以借用位置式累加的概念,也在程序编制时,设计一个累加单元存放 $\sum_{i=0}^k e(i)$ 。当 $K_I \sum_{i=0}^k e(i)$ 是机器零时, $\Delta P_I(k)$ 作为零不作输出,当 $K_I \sum_{i=0}^k e(i)$ 大于机器零时, $\Delta P_I(k)$ 就以该数值输出,同时把累加单元清零。这样便通过程序编制在增量式运算中保存了积分作用,解决了由于定点运算丢掉积分作用的问题。

(二)“饱和”作用的影响

在实际生产应用中,控制器输出的控制变量受到执行元件机械和物理性能的限制,或者受到人为给定的输出范围的限制,使其只能在有限范围内作用,即

$$P_{\min} \leq P \leq P_{\max}$$

如果控制算法给出的输出控制量 P 在上述范围内变化,那么控制可以达到预期的效果。如果控制算法输出的控制量超出上述范围,实际执行机构的动作范围将达不到算法要求的位置值;从而被控参数不能按算法计算要求值变化,这样控制过程就可能偏离预定规律,影响控制特性。这种效应通常叫做饱和效应。这种现象在给定值发生大的阶跃变化时最容易发生,所以有时也称为起动效应。克服饱和效应对控制特性的影响,在 PID 算法的实际应用中是一个很重要的问题。

1. 积分饱和对 PID 位置算法的影响及克服方法

在控制过程中,如果给定值产生大的阶跃(如图 7-4-6 中,给定值从 0 ~

R^*), 形成一个较大的偏差 $e(e = R^* - Y)$ 。为迅速克服这个偏差, 标准位置 PID 算式, 会产生一个较大的输出值 P , 如果此时 $P > P_{\max}$, 那么实际执行机构将只取 P_{\max} 值做为最大输出值。此时, 被控参量 Y 的变化速度和变化量受到限制, 克服偏差的速度会比正常状态减慢。由于 e 存在时间增长, 积分项 $K_I \sum_{i=0}^k e(i)$ 会产生较大的累积值。当被控参量到达给定值, 并产生反向偏差时, 由于积分项大累积值的作用, 执行机构还保持在 P_{\max} 位置。包括大累积值积分项的控制输出值要靠反向偏差 $(-e)$ 衰减, 因此它将不能马上脱离饱和状态, 而且累积值越大, 衰减积分累积值的时间就越长。直到经过时间 τ 后, 控制输出量 P 才能脱离饱和区, 产生消除反向偏差的作用。而这时系统往往已经产生了较大的反向偏差(超调), 使系统控制特性变坏。可见, “饱和”产生的作用主要由积分项产生, 因此把它叫做“积分饱和”。

克服积分饱和影响的方法很多, 下面介绍其中的几种。

(1) 判限削弱积分法

这种方法的基本思想是: 一旦控制算法输出量进入饱和区, 就停止增大积分项的运算。具体方法如图 7-4-7 所示。在计算 $P(k)$ 时, 先判断上一时刻的输出控制量 $P(k-1)$ 的值是否已超出限制范围。如果已超出, 根据偏差的符号, 再判断系统输出是否在超调区域, 由此决定积分项是否对偏差 $e(k)$ 进行累积。

从这个方法的基本思想上分析可以看到, PID 的增量算法没有 $\sum e(i)$ 的运算, 累积作用由执行机构的机械位置记忆完成, 当机械位置到达极限时, 不会产生超出极限的累积, 因此自动完成了判限和削弱积分的作用。这也就消除了积分饱和的影响, 所以增量算法具有防积分饱和的功能, 这也是 PID 增量算法的一个优点。

(2) 有效偏差方法

由 PID 位置式计算出的输出控制量超出限制范围时, 控制量实际只能取边界值, 即

$$P \geq P_{\max} \text{ 时, } P = P_{\max}$$

$$P \leq P_{\min} \text{ 时, } P = P_{\min}$$

有效偏差法间将相应的控制输出边界值的偏差值作为有效偏差值计入积分累计, 而不按实际偏差进行积分运算。

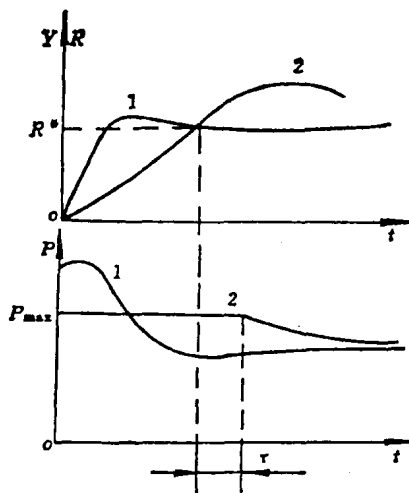


图 7-4-6 PID 位置算法的积分饱和现象

1—理想情况控制；
2—有积分饱和的情况

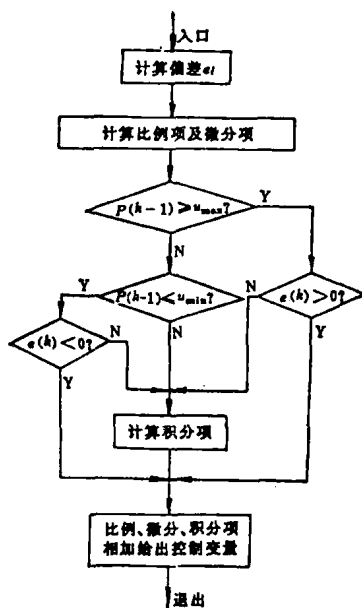


图 7-4-7 判限削弱积分法框图

当实际输出控制量达到极限值时,即 $P = P^*$ (P^* 为上限值或下限值),有效偏差可按式(7-4-2)解出,即

$$e(k) = \frac{\frac{1}{K_P} P^* - \frac{\Delta T}{T_I} \sum_{i=0}^{k-1} e(i) + \frac{T_D}{\Delta T} e(k-1)}{1 + \frac{\Delta T}{T_I} + \frac{T_D}{\Delta T}}$$

算法的运算框图如图 7-4-8 所示。

2. 饱和作用对 PID 增量式的影响和解决方法

PID 增量算法可以避免积分饱和效应的影响,但却有可能出现比例及微分饱和现象。在增量算法中,特别在给定值发生跃变时,由算法的比例部分和微分部分计算出的控制增量可能比较大。如果该值超过了执行机构的极限位置,实际计算增量受到限制,丢失了一部分信息,这部分遗失的信息只能通过积分部分来补偿。因此,与没有限制时比较,系统的动态过程将会减慢(图 7-4-9)。

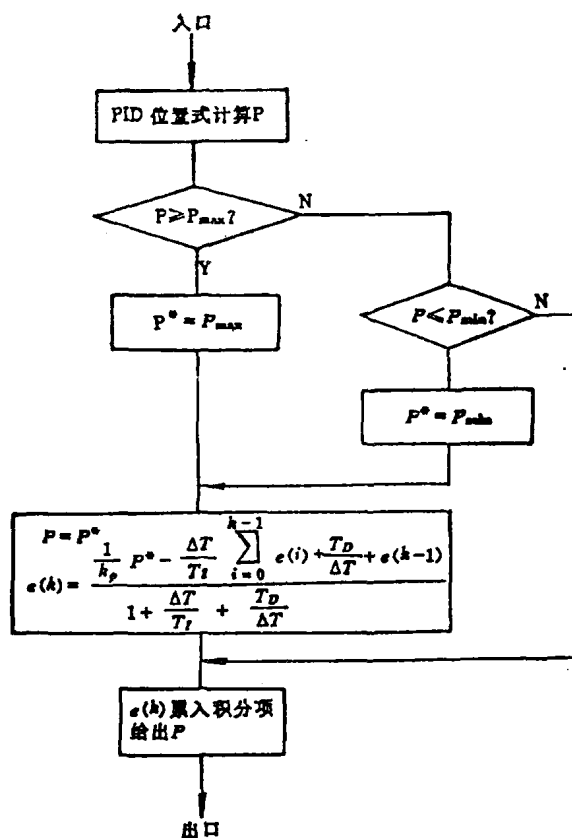


图 7-4-8 有效偏差法的 PID 位置算法

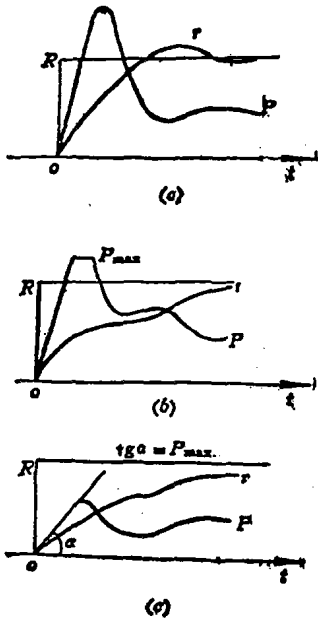


图 7-4-9 PID 增量式的比例与微分饱和

- (a) 无限制时的控制结果;
- (b) 控制输出量受限制时控制结果;
- (c) 输出变化率受限制时的比例及微分饱和。

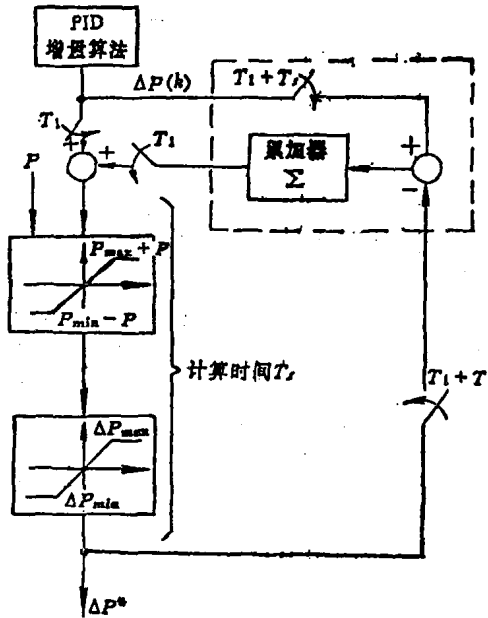


图 7-4-10 用积累法纠正比例及微分饱和流程图

“积累补偿法”是减小这种饱和影响的一种方法。这种方法是把因饱和而未执行的增量信息积累起来,在一定时机,补充执行,加速动态过程。其算法执行流程如图 7-4-10 所示。

从流程图中可以看出,如果计算出来的 ΔP 超限,多余未执行的控制增量将存储在累加器中,一旦控制量脱离了饱和区,累加器中的量全部或部分地加到计算出控制增量上,以补充由于限制而未能执行的控制。

使用“积累补偿法”虽然可以抑制比例和微分饱和,但由于引入累加器,使得增量算法也可能出现积分饱和现象。因此使用这种方法时,应综合考虑累加器累加的条件,丢失补偿完成后应及时削除积分作用。

(三) 提高微分项抗干扰能力的方法

在离散型 PID 算式的微分项中如何减少数据误差和噪声也是应用中经常遇

到的一个问题。这一点可从微分项的表达式看出。在位置式和增量式中微分项的输出可分别表示为

$$P_D(k) = \frac{K_P T_D}{\Delta T} [e(k) - e(k-1)] = K_D [e(k) - e(k-1)]$$

$$\Delta P_D(k) = \frac{K_P T_D}{\Delta T} [\Delta e(k) - \Delta e(k-1)] = K_D [\Delta e(k) - \Delta e(k-1)]$$

由于微分作用是在相邻的采样周期内进行的,因此它的强弱不仅与微分时间 T_D 、放大系数 K_P 有关,而且与采样周期 ΔT 也有明显的关系。当 ΔT 太小时,二次采样之间被控参数的变化不会太大,因而微分作用就很弱。为了在 ΔT 小时增强微分作用,可以增加 K_D 。但增大 K_D 时,会使抗噪声特性恶化。而噪声和数据误差在离散化过程中是不可避免的,微分作用对它们又特别敏感,因此应设法减少噪声和误差在微分项中的影响。改进微分算法是增强微分抗干扰能力的一种方法。

这种方法是在计算微分之前,先进行一定的平滑。常用的方法是内插法,图 7-4-11 所示为四点中值差分法示意图。在这种方法中,微分项不是直接应用现时偏差、 $e(k)$ 而是取相邻 4 个采样时刻的偏差平均值作为基准,设

$$\bar{e}(k) = \frac{1}{4} \sum_{i=0}^3 e(k-i) = [e(k) + e(k-1) + e(k-2) + e(k-3)]/4$$

然后以加权求和形式近似构成微分项,即

$$\frac{T_D \Delta e(k)}{\Delta T} = \frac{T_D}{4} \left[\frac{e(k) - \bar{e}(k)}{1.5\Delta T} + \frac{e(k-1) - \bar{e}(k)}{0.5\Delta T} - \frac{e(k-2) - \bar{e}(k)}{0.5\Delta T} - \frac{e(k-3) - \bar{e}(k)}{1.5\Delta T} \right]$$

$$\text{符号} = \frac{T_D}{6\Delta T} [e(k) - e(k-3) + 3e(k-1) - 3e(k-2)]$$

这种方法又称辛卡生(Simpson)匀整差分算式。在应用中, $\frac{T_P}{\Delta T}$ 的选取应比理想情况稍小一些。

(四) 给定值大阶跃变化时的阻尼方法

PID 算法按照给定值与被控参量的偏差运算决定控制输出量。当给定值突然大阶跃变化时,给控制系统造成一个比较大的扰动。尤其对像炉窑这样的具有纯

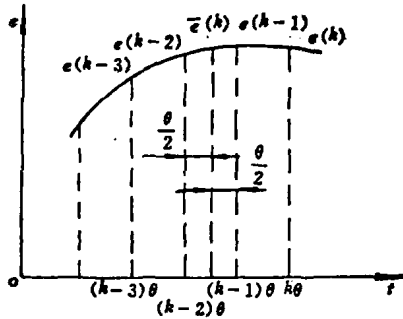


图 7-4-11 四点中值差分法示意图

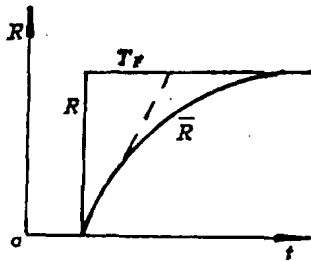


图 7-4-12 给定值
阶跃的前置滤波

滞后,大惯性特性的控制系统,大的突然扰动会使系统很难在短时间内稳定下来,严重时会使生产受到影响。为防止这种情况发生,可用以下方法。

1. 前置滤波器

这种方法是采用一前置滤波器(一般为一阶惯性环节)对给定值 R 进行预处理,使进入控制算法的给定值 \bar{R} 不突变,而是一个惯性延迟缓变量(图 7-4-14)。

其数字化算法可由下式给出:

$$\bar{R}(k) = \beta \bar{R}(k-1) + (1 - \beta) R(k)$$

式中 $\beta \approx \frac{\Delta T}{T_F}$, T_F 为滤波时间常数。

2. 修改算法中对给定值变化的敏感项

为了减小给定扰动对控制过程的影响,在 PID 增量算式(2-4-4)的微分项中,可将给定值因素排除在外,使微分仅由被控参量决定。也就是在微分项的偏差 $e(k) = R(k) - Y(k)$ 中不考虑 $R(k)$ 的变化,而将二阶差分项 $e(k) - 2e(k-1) + e(k-2)$

用 $-[Y(k) - 2Y(k-1) + Y(k-2)]$ 来代替。修改后的算式为

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

如果将比例环节的偏差项也进行修改,可以得到具有更大阴尼的算式,即

$$\Delta P(k) = K_p \left\{ -Y(k) + Y(k-1) + \frac{\Delta T}{T_I} e(k) + \frac{T_D}{\Delta T} (-Y(k) + 2Y(k-1) - Y(k-2)) \right\} \quad (7-4-9)$$

这时,给定值变化的因素只影响积分项,而积分系数一般较小,所以控制输出的增长得到了有效阻尼。

2. 增量算法中动态过程的加速

在 PID 增量算法中,比例项 $e(k) - e(k-1)$ 与积分项 $\frac{\Delta T}{T_I} e(k)$ 的符号有如下关系:如果被控参量继续偏离给定值,则这两项符号相同;而当被控量向给定值方向变化时,则这两项符号相反。例如 $Y < R$ 时,如果 Y 继续减小,则 $e(k) > e(k-1)$, $e(k) > 0$;如果 Y 向 R 接近,则有 $e(k) < e(k-1)$, $e(k) > 0$ 。由于算式的这种规律,反号的比例作用阻碍了积分作用,因而减小了积分超调,增强了系统的稳定性。但如果被控参量远离给定值时,比例项与积分项的这种反作用,会减慢控制过程。

针对以上情况,在一些控制过程中,为了提高系统的动态过程速度,可以人为地选择一个偏差范围 e 。当偏差 $|e(k)| < \varepsilon$ 即被控量已接近给定值时,就按正常规律调节而当 $|e(k)| > e$ 时,则不管比例作用 $e(k) - e(k-1)$ 为正为负,均使它朝着有利于接近给定值的方向调整,即其值取为 $1e(k) - e(k-1)_1$,其符号和积分项取一致。这样就能够加快控制的动态过程。

三、变形 PID 算法的应用

在实际工业生产过程中,为保证控制器对具有非线性、纯滞后、强扰动特性的复杂工业对象的良好控制性能,除采用上述改进方法外,还利用程序编制技巧、或将非线性环节及其它算法与 PID 算式相结合形成变形 PID 等方法,提高 PID 算法

的适应性和控制品质。从这些方法中可以看到,计算机的应用使 PID 这个古典的控制方法有了更广泛的应用前途。

(一)带滤波器的 PID 控制规律的数字算法

为了克服高频干扰,有时将一阶惯性延迟数字滤波器加到 PID 位置算式中,使 PID 调节器成为带滤波功能的调节器。这种 PID 调节器的数学表达式为:

$$\frac{P(s)}{E(s)} = G(s) = \frac{K_P}{1 + T_F S} \left[1 + T_D S + \frac{1}{T_I S} \right] \quad (7-4-10)$$

式中 $\frac{1}{1 + T_F S}$ 是一阶惯性低通滤波器,角频率为 $1/T_F$ 。

引入如下参数:

$$K_P = K_1 \frac{T_1 T_2}{T_1}$$

$$T_I = T_1 + T_2$$

$$T_D = \frac{T_1 T_2}{T_1 + T_2}$$

由式(2-4-10)得到

$$\frac{P(s)}{E(s)} = G(s) = K_1 \frac{T_2 S + 1}{T_F S + 1} \left(1 + \frac{1}{T_I S} \right) \quad (7-4-11)$$

上式控制结构图如图 7-4-13 所示。

利用图 7-4-13 所示独立环节形式将式(7-4-11)变化成差分方程:

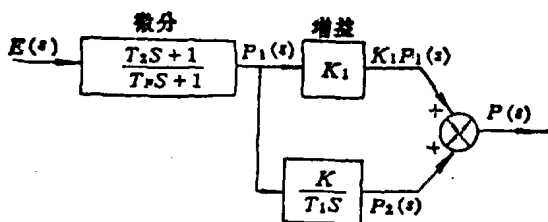


图 7-4-13 带滤波器的 PID 控制器框图

1. 微分部分的差分方程为

$$P_I(k) = \frac{T_F}{T_F + \Delta T} P_I(k-1) + \frac{T_2}{T_F + \Delta T} [e(k) - e(k-1)] + \frac{\Delta T}{T_F + \Delta T} e(k)$$

或

$$\begin{aligned}
P_1(k) &= P_1(k-1) - \left(1 - \frac{T_F}{T_F + \Delta T}\right) P_1(k-1) \\
&+ \frac{T_2}{T_F + \Delta T} [e(k) - e(k-1)] + \frac{\Delta T}{T_F + \Delta T} e(k) \\
&= P_1(k-1) + \frac{T_2}{T_F + \Delta T} [e(k) - e(k-1)] \\
&+ \frac{\Delta T}{T_F + \Delta T} [e(k) - P_1(k-1)] \tag{7-4-12}
\end{aligned}$$

当 $\Delta T \leq T_2$ 时, 式(7-4-12)可近似写成

$$\begin{aligned}
P_1(k) &\approx P_1(k-1) + \frac{T_2}{T_F} [e(k) - e(k-1)] \\
&+ \frac{\Delta T}{T_F + \Delta T} [e(k) - P_1(k-1)] \tag{7-4-13}
\end{aligned}$$

2. 积分部分的差分方程为

$$P_2(k) = P_2(k-1) + \frac{K_1 \Delta T}{T_1} P_1(k) \tag{7-4-14}$$

式(2-4-13)引入比例作用后的差分方程为

$$\begin{aligned}
K_1 P_1(k) &= K_1 \left\{ P_1(k-1) + \frac{T_2}{T_F} [e(k) - e(k-1)] \right. \\
&\left. + \frac{\Delta T}{T_F + \Delta T} [e(k) - P_1(k-1)] \right\} \tag{7-4-15}
\end{aligned}$$

3. 由式(7-4-13)和式(7-4-14), 带滤波 PID 位置式的最终形式为

$$\begin{aligned}
P(k) &= K_1 P_1(k) + P_2(k) \\
&= P_2(k-1) + \frac{K_1(T_1 + \Delta T)}{T_1} \{ P_1(k-1) \\
&+ \frac{T_2}{T_F} [e(k) - e(k-1)] + \frac{\Delta T}{T_F + \Delta T} [e(k) - P_1(k-1)] \} \tag{7-4-16}
\end{aligned}$$

应用中, 常设 $T_F = rF_2$, 这样在整个算式中可以定义各系数为

T_1 ——实际积分时间;

T_2 ——实际微分时间;

r ——微分放大系数;