

基本控制算法与仪表

【内容提示】 控制仪表像人的大脑一样,根据预定的目标和测取的信息,按一定的控制算法,形成下一步的行动指令,并提供给执行器实施。本章对过程控制系统中常用的控制算法——PID 算法进行讨论。实现这种算法的具体方法有模拟和数字之分,其表现形式迥异,有人认为是“同工异曲”,有人认为是“大相径庭”,其中的差异与优劣只有学完之后,才有深刻的体会与合理的判断。

粗略地说,控制仪表就是控制器的仪表化,它是过程控制系统的判断和指挥中心,在系统中占有举足轻重的地位。控制器将被控量的期望值与实际值进行比较,根据两者的差值(偏差),按一定的算法计算下一步需要的控制量(操作量),然后传输给执行器实施,以实现预期的控制目标。

伴随着过程控制技术的发展与进步,控制仪表作为过程仪表的一部分,大体上也经历了3个阶段:一是基地式控制仪表,二是单元组合式控制仪表,三是以微处理器为核心的控制仪表。

基地式控制仪表是早期的控制器产品,它与检测装置、显示仪表组合在一个简单的控制系统中,完成比较简单的开关控制和比例-积分-微分(Proportional -Integral -Derivative, PID)控制,实现生产设备的基本操作与调节;单元组合式控制仪表是基地式控制仪表的升级产品,控制算法逐步成熟、定型,并发展了适合多种工艺要求的专用控制算法,如比值控制、选择控制、前馈控制、解耦控制等,控制仪表与测量仪表和执行器之间,按标准的电动信号或气动信号进行联系,电动Ⅱ型仪表的标准信号为 $0\sim 10\text{mA}$ 直流电流,Ⅲ型仪表信号为 $4\sim 20\text{mA}$ 直流电流,气动仪表的标准信号为 $20\sim 100\text{kPa}$;第三阶段的控制仪表利用了微处理器技术,实现了传统控制器的数字化,开发了可编程序逻辑控制器(PLC)和多种智能化控制算法,提高了过程控制质量和水平。

时至今日,人们发明或开发的控制算法众多,施用的对象、控制目的和效果也是千差万别,在这里难以逐一陈述,本节仅就最基本、最常用的PID控制器进行讨论,其他专用或复杂的控制算法将在后续章节中加以叙述。

PID控制算法最早是在1922年,由Nicholas Minorsky(1885—1970)在研究船舶自动驾驶机构时首先提出的^[7]。时过九旬,尽管各种先进的控制算法层出不穷,但这种经典的控制算法仍然经久不衰,它以结构简单、调整方便、适应性强而著称,并被广泛采用。据统计,在当今世界所用的控制算法中,PID算法仍占百分之七十以上的份额,甚至在高级控制算法中,很多也都是与PID相结合的复合型算法。由此可见,PID算法强大的生命力。

生产中的被控过程是形形色色的,生产工艺对控制的要求也是各种各样的,根据实际情况,恰当地为生产过程设计控制模式、选择控制算法(Control Algorithm)是控制工程师的一项基本任务。为此,工程师首先必须了解基本控制算法的特点与适用场合,然后结合具体过程和工艺要求,作出正确选择。PID算法作为一类最基本的、最常用的控制算法,在实践中

有很高的使用率,这类算法主要有比例(P)、比例积分(PI)、比例微分(PD)和比例积分微分(PID)等几种。

3.1 基本控制算法及其特点

3.1.1 比例控制

比例控制算法是最简单的一种,它是控制器的输出 u 与控制器的输入偏差信号 e 成比例关系,用数学表达式表示为

$$u(t) = K_c e(t) + u_0$$

其中, K_c 为比例增益(Proportional Gain),也称放大倍数,其大小决定比例控制作用的强与弱; u_0 为输出信号初始值; $e(t) = r(t) - z(t)$ 为控制器的输入,这里 $r(t)$ 是系统的给定值, $z(t)$ 为系统的输出测量值(可参见图 1-1)。控制器输出增量形式为

$$\Delta u = u(t) - u_0 = K_c e(t) = \frac{1}{\delta} e(t)$$

其中, δ 为比例带(Proportional Band)或称比例度,且

$$\delta = \frac{1}{K_c} \times 100\% = \frac{e}{\Delta u} = \frac{\frac{e}{z_{\max} - z_{\min}}}{\frac{\Delta u}{u_{\max} - u_{\min}}} \times 100\%$$

其中, $z_{\max} - z_{\min}$ 为测量输入的最大变化量,即控制器输入量程, $u_{\max} - u_{\min}$ 为控制器输出量程。采用Ⅲ型仪表,则量程为 4~20mA 电流。

显然,比例带与比例增益成反比关系。比例带有其重要的物理意义,具体是:使控制器的输出变化满量程时(即调节阀从全关到全开或相反),相应的输入测量值变化占仪表输入量程的百分比。换一种说法,如果 Δu 表示调节阀开度的变化, e 表示被控量(测量值)的变化,那么, δ 表示调节阀开度改变 100% 时,所需要的系统被控量的变化范围。当被控量处在该范围时,调节阀的开度才与偏差成比例。否则,调节阀已处于全开或全关状态,控制器暂时不起作用(失效)。

比例控制具有控制及时、响应快的优点,同时,还具备下述特点:

① 比例控制是一种有差控制。当系统采用比例控制时,由于通常的生产过程多为一阶惯性环节(或惯性加纯滞后),对于定值控制(Constant Value Control)会存在稳态误差(Steady State Error),即余差。这是因为只有偏差 e 不为零时,控制器才有输出,调节阀才工作;一旦偏差为零,控制器输出也为零,无法驱动调节阀,控制作用也就消失了。

② 比例控制产生的稳态误差随着比例带的减小而减小,即随着比例增益的增大而减小。比例带的减小意味着比例增益的加大,控制作用加强,被控量向设定值接近。但是,随着比例带的减小,系统振荡会兴起、加剧,甚至使系统不稳定,所以减小比例带也是有限的,不能为了减小稳态误差而牺牲系统的稳定性,两者需综合考虑。

以上主要是比例控制针对定值控制而谈的,对于随动系统,由于给定值会随时间变化,如果选用比例控制,被控量是难以跟踪给定的。所以,随动控制系统通常不单独选用比例控制。

3.1.2 比例积分控制

前面介绍了比例控制,下面介绍比例积分控制。为此,先介绍积分控制,然后过渡到比例积分控制。

当控制器输入为偏差 e 时,采用积分控制算法,于是控制器输出为

$$u(t) = \frac{1}{T_i} \int e(\tau) d\tau + u_0 \quad (3-1)$$

其中, u_0 为 u 的初始状态, T_i 为积分时间, e 为控制器输入。当输入 $e(t)$ 为单位阶跃信号时,积分控制器输出

$$u(t) = t/T_i + u_0$$

即以 $1/T_i$ 为斜率,直线上升。假如从 $t=t_1$ 开始,让 $e(t)=0$,则控制器输出 $u(t)$ 停止上升,此后一直保持 $u(t_1)$ 值不变,即

$$u(t) = t_1/T_i + u_0$$

更为重要的是,如果 $u(t)$ 被用来驱动调节阀,这将使调节阀保持在某一开度不变。

将 I 控制器置于图 1-1 中控制器所在的位置,与其他几部分构成闭环负反馈控制系统。当设定 $r(t)$ 为阶跃信号时, $e(t)=r(t)-z(t)$ 也为阶跃信号($z(t)$ 为零初始状态),随着积分的进行,控制器输出 $u(t)$ 不断上升,驱动阀门开启,被控量 $y(t)$ 开始上升,其测量值 $z(t)$ 也上升。当 $t=t_1$ 时,测量值达到设定值,即 $z(t_1)=r(t_1)$, $e(t_1)=0$,此时的积分控制器输出将维持常值不变,即 $u(t)=u(t_1)$,阀门保持开度,测量量 $z(t)=z(t_1)$ 也保持不变,实现了系统无稳态误差(即系统被控制量与给定量的一致)。所以积分控制可以使系统消除稳态误差,这是积分控制一个非常宝贵而重要的特点。

从式(3-1)来看,控制器输出的大小,与输入,即偏差 e 成正比,与积分时间常数 T_i 成反比。显然,减小积分时间常数,可以加快响应速度。但是这一行为同样受到系统稳定性的制约。与比例控制相比,积分控制的动态过程比较缓慢。这是因为积分(即累加)行为是需要时间的,即控制增强需要有一个积累过程,而不是像比例控制一样,立竿见影。这就使人们开始考虑:是否可以将比例控制与积分控制结合起来,既利用比例控制的快速性,又发挥积分控制消除稳态误差的长处? 是的,这就是下面将要介绍的比例积分控制。

当控制器采用比例积分控制时,其输出 u 与输入偏差 e 之间有关系式:

$$u(t) = K_c e(t) + \frac{K_c}{T_i} \int e(\tau) d\tau + u_0$$

其增量形式为

$$\Delta u = u(t) - u_0 = K_c e(t) + \frac{K_c}{T_i} \int e(\tau) d\tau = \frac{1}{\delta} \left(e(t) + \frac{1}{T_i} \int e(\tau) d\tau \right)$$

其中, δ 为比例带, K_c 为比例增益,且 $K_c=1/\delta$, T_i 为积分时间常数,其数值越小,积分作用越强,同时,可能引起被控量的振荡,甚至不稳定;其数值越大,积分作用越弱,为无穷大时,积分作用消失,PI 控制器就变为 P 控制器。

PI 控制器的传递函数形式为

$$G_c(s) = \frac{U(s)}{E(s)} = K_c \left(1 + \frac{1}{T_i s} \right) = \frac{1}{\delta} \left(1 + \frac{1}{T_i s} \right)$$

给控制器输入 $e(t)$ 以下列信号

$$e(t) = \begin{cases} \Delta e & 0 \leq t \leq 1.5T_i \\ 0 & 1.5T_i < t < \infty \end{cases}$$

在零初始条件下,则输出为

$$u(t) = \begin{cases} \frac{\Delta e}{\delta} + \frac{\Delta e}{\delta T_i} t & 0 \leq t \leq 1.5T_i \\ \frac{2.5\Delta e}{\delta} & t > 1.5T_i \end{cases} \quad (3-2)$$

相关波形见图 3-1。

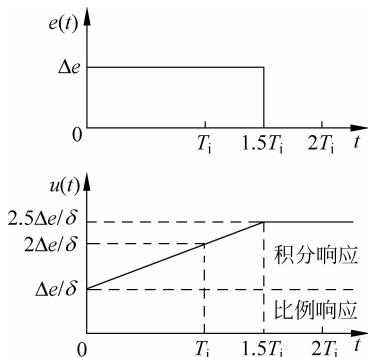


图 3-1 PI 控制器的输入及其响应

由图可见,尽管控制器自 $t=1.5T_i$ 开始输入为零,但控制器输出却一直保持该时刻的数值不变,这是积分产生的效果。该现象表明,如果将控制器接入系统(详见图 1-1),当被控量(这里用其测量值 z 表示)达到预期的给定值 r 时,虽然控制器的输入信号 $e=r-z=0$,控制器输出却维持某常量,调节阀维持某开度,进而保证被控量与给定量之间的误差为零。这就是为什么用了含有积分的控制算法后,系统可消除稳态误差的原因。

由式(3-2)可知,当 $t=T_i$ 时,有 $u(T_i)=2\Delta e/\delta$ 。所以,由图 3-1 可以确定积分时间常数 T_i ,即在控制器输出为 2 倍于比例作用输出时所对应的时间就是积分时间

常数,见图 3-1 中的 $u(t)$ 。

可以进一步推导,如果控制器输入在 $t=1.5T_i$ 时刻不撤除(不为零),控制器输出将一直增长下去,直至饱和限幅,控制器输出失去进一步增长作用。如果输出没有限幅,工程实践中会有麻烦。因为这是一种发散行为,从理论上说,系统将不稳定。

PI 控制器的特点是:

① PI 控制器具有快速反应和消除稳态误差的双重作用。这是因为控制器中的比例部分能迅速响应输入的变化,积分部分能够最终消除稳态误差。两者的结合能满足大多数生产过程对控制的要求,所以 PI 控制器在过程控制中被大量采用。

② 积分时间常数 T_i 的大小表明了积分的强弱和消除稳态误差的快慢。一般来说, T_i 越小,积分作用越强,消除余差所需的时间越短。但是过小的积分时间常数会引起系统振荡,甚至不稳定。 T_i 过大,表明积分作用弱, $T_i \rightarrow \infty$ 时,表明积分作用逐渐消失,PI 控制器演变为 P 控制器。

3.1.3 比例微分控制

为了讨论比例微分(PD)控制,先来谈谈微分控制。采用微分控制规律的控制器输出与输入之间的关系为

$$u(t) = T_d \frac{de(t)}{dt} \quad (3-3)$$

即微分控制的输出与系统偏差的变化率成正比。上式中 T_d 为微分时间,输出越大,说明微分作用越强。

从式(3-3)来看,控制器的输出反映其输入 $e(t)$ 的变化率(大小和方向),而 $e(t)$ 直接与

被控量相关联,所以,微分控制器的输出主导着被控量的变化率。将这样的控制量施加于执行器,无疑将引导被控量的变化趋势和方向,也就是有人说的超前控制作用。这相当于根据其变化趋势而提前行动,这种行动的结果对于减小动态偏差、缩短过渡过程时间将是十分有利的。尤其是对于一些惯性较大的被控过程,能较为显著地提高系统的快速性,并减小超调量。

当输入偏差为阶跃信号时,其输出为冲激信号,由于持续时间太短,不能驱动阀门。实践中,微分控制是不单独使用的。这是因为当系统的被控量出现缓慢变化时,控制器的输入信号偏差 e 需要等到这种缓慢变化积累到一定程度(或者说要突破不灵敏区),才能有校正行为出现。这种不敏感、不作为的表现,在控制中是不能容忍的。所以,微分控制往往与其他控制规律结合在一起,既保持其他控制的功能,又发挥微分控制特有的作用,如 PD 控制,或者 PID 控制等。

控制器采用比例-微分控制算法后,其输出 $u(t)$ 与输入偏差信号 $e(t)$ 之间的关系式为

$$\begin{aligned} u(t) &= K_c e(t) + K_c T_d \frac{de(t)}{dt} + u_0 = K_c \left(e(t) + T_d \frac{de(t)}{dt} \right) + u_0 \\ &= \frac{1}{\delta} \left(e(t) + T_d \frac{de(t)}{dt} \right) + u_0 \end{aligned}$$

式中的 K_c 、 δ 和 T_d 的含义同前, u_0 为控制输出的初始值。

控制器输出增量形式和传递函数形式分别为

$$\begin{aligned} \Delta u(t) &= u(t) - u_0 = K_c e(t) + K_c T_d \frac{de(t)}{dt} = K_c \left(e(t) + T_d \frac{de(t)}{dt} \right) \\ &= \frac{1}{\delta} \left(e(t) + T_d \frac{de(t)}{dt} \right) \\ G_c(s) &= \frac{U(s)}{E(s)} = K_c (1 + T_d s) = \frac{1}{\delta} (1 + T_d s) \end{aligned} \quad (3-4)$$

式(3-4)表示的 PD 控制器在实践中是难以实现的。于是,工程中大都采用近似的 PD 控制器,其传递函数为

$$G(s) = \frac{1}{\delta} \frac{T_d s + 1}{\frac{T_d}{K_d} s + 1} \quad (3-5)$$

其中, K_d 为微分增益,它的取值一般在 5~10 之间。假如输入单位阶跃信号,则响应为

$$u(t) = \frac{1}{\delta} + \frac{1}{\delta} (K_d - 1) e^{-\frac{K_d}{T_d} t}$$

相应的波形图见图 3-2。由式(3-5)可知,近似的 PD 控制器中,追加的惯性环节时间常数比原一阶微分的时间常数小很多,除了使控制器实现起来方便之外,该极点对系统的影响有限。所以,为方便起见,通常仍按式(3-4)考虑问题。

PD 控制的特点如下:

① PD 控制是一种有差调节。在稳态时,由 $de/dt=0$ 可知,它并不像积分控制可以继续发挥作用,而是变成了比例控制,这一点从图 3-2 看得比较清楚。

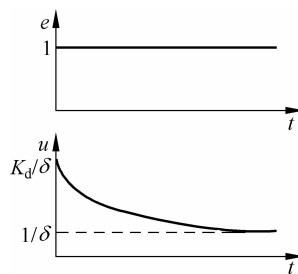


图 3-2 PD 控制器的单位阶跃响应

② PD 控制器中,微分部分对被控量的振荡有较好的抑制作用,这对于稳定系统、减小超调量和缩短响应时间是有益的。

③ 随着微分时间常数 T_d 的增大,系统稳定性提高,但是过大反倒会引起系统振荡,引起调节阀频繁向开和关两端来回运动。所以, T_d 应适当,PD 控制应以比例为主,以微分为辅。

④ 从工程的角度来说,PD 控制较为适合时间常数较大的过程,对变化较快的流量和压力过程不大适合,抗干扰能力也较差,对纯滞后过程无效果。

3.1.4 比例积分微分控制

PID 的控制算式为

$$u(t) = \frac{1}{\delta} \left(e(t) + \frac{1}{T_i} \int e(\tau) d\tau + T_d \frac{de(t)}{dt} \right) + u_0$$

其中, δ 、 T_i 、 T_d 和 u_0 的含义同前。其传递函数形式为

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

传递函数中关于 s 算子的分子阶次高于分母阶次,实践中难以实现。所以,实际中往往采用近似的表达式,如下列两种形式

$$G_c(s) = FK_c \frac{1 + \frac{1}{FT_i s} + \frac{T_d s}{F}}{1 + \frac{1}{K_i T_i s} + \frac{T_d s}{K_d}} \quad (3-6)$$

其中, F 为相互干扰系数(通常大于 1), K_i 为积分增益, K_d 为微分增益。也有的近似表达式采用与式(3-6)相近的形式,但分母中无 $1/(K_i T_i s)$ 项,即相当于原 PID 表达式串接了一个惯性环节

$$G_c(s) = FK_c \frac{1 + \frac{1}{FT_i s} + \frac{T_d s}{F}}{1 + \frac{T_d s}{K_d}}$$

对于式(3-6),施加单位阶跃输入,其响应如图 3-3 所示。

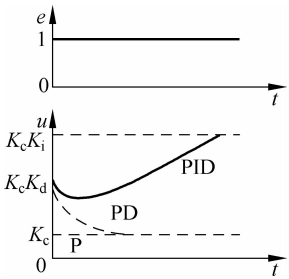


图 3-3 采用 PID 控制的阶跃输入及响应

总的来说,PID 控制综合了几种简单控制的长处,即可改善系统的稳定性,又能消除静差,适用于负荷变化大、容量滞后较严重、控制性能要求高的过程。其特点是 PID 控制兼顾了各方面的长处,控制效果较为理想。但是,实践中并不是任何情况,都选用 PID 控制器的。这是因为,PID 控制器有三个参数要确定,而这些参数对响应的性能影响较为复杂,有的性能指标之间相互制约,调整起来比较费事。如果参数调整不合理、配合不默契,不仅不能发挥该控制算法的优势,反而贻害系统。因此,恰当选择,删繁就简,是我们选择控制算法的基本原则。能用 P 控制器解决问题的,尽量不用 PI 控制器,更不要用 PID 控制器。

3.1.5 基本控制算法的选择要点

前面叙述了 P、PI、PD 和 PID 四种常用的控制算法,对它们的表达形式、控制特点和适用对象进行了阐述,下面将结合过程控制的实际,归纳设计系统时选择这些基本控制算法的要点,以作设计控制器时参考。一般来说控制算法的选择与被控过程特性、工艺要求、负荷变化以及干扰幅度与频率等都有直接的关系。通常,基本控制器需要有比例控制,然后在此基础上,根据要求和实际情况,加入其他控制规律。

① 当广义过程的时间常数较大时,如温度、成分、pH 值等控制,控制器应引入微分,形成 PD 控制器,如果工艺上还要求无余差,则还应加入积分控制,形成 PID 控制。

② 当广义过程的时间常数较小,负荷变化也不大,且工艺要求不高时,可选择 P 控制,如储罐压力控制、液位控制等;如果此时其他条件不变,仅工艺要求变为:无余差,则应选择 PI 控制,如流量控制、管道压力控制等。

③ 当广义过程的时间常数较大,如容积迟延较大,且负荷变化剧烈,或者有较大纯迟延($\tau/T > 1$, τ 为纯迟延时间, T 为过程时间常数),采用 PID 控制难以满足要求时,需考虑其他控制算法。

3.1.6 PID 控制器的工程应用

前面讨论了 PID 控制算法及其特点,如果将其用到具体的控制系统中来,还会遇到一些具体问题,如控制器的正/反作用方式问题, PID 控制器的积分饱和问题等。

1. 控制器的正/反作用方式

对于一个控制回路来说,控制系统是由控制器、调节阀、被控对象和检测器等组成的,由于各组成部分存在正/反作用方式的问题,只有最终构成的闭环为负反馈,系统才能稳定运行。所以,有必要对控制器的正/反作用方式进行讨论。至于其他几部分的正反作用方式的规定之类的问题,等到第 6 章还会具体谈到。

现实中,的确有被控对象存在正反差异的不同,所以控制器也应有正反作用的区别。例如,控制房间内的温度,夏天外面气温高,室内需冷气降温,增大操作量,室内温度会进一步下降;冬天外面气温低,室内需暖气增温,增大操作量,室内温度会进一步上升。由于要达到的温度是以人体感觉舒适温度为准,但是被控量温度与两操作量之间存在不同的变化方向,即被控对象呈现的正/反极性不同(有人解释为稳态增益正/负不同),从而影响到控制器的作用方式。

为了方便实现闭环负反馈控制,过程控制有一套定义闭环控制系统中各部分正反作用方式的规定,其中控制器的正反作用方式是这样表述的:

控制器正作用(Direct Action)方式:随着被控过程输出测量信号的增加,控制器输出量也增加,可用“+”号表示;控制器反作用(Reverse Action)方式:随着被控过程输出测量信号的增加,控制器输出量减小,可用“-”号表示。如果相关量与前表述的方向相反,也是一样的含义。例如,随着被控过程输出测量信号的减小,控制器输出量也减小,则控制器为正作用方式。

实际中,控制器正/反作用方式的选择,是由设在控制仪表上的选择开关来实现的。通常仪表上都有明显的“正”或“反”的字样,设定时拨向对应位置即可。

在一个闭环控制回路中,控制器的正反作用方式不仅与被控对象有关,而且还与调节阀的气开/气闭形式有关(由于检测器一般被视为正作用,通常不考虑)。具体的方法是,画出控制系统方块图,先确定除控制器以外部分的作用方向(即正或负),最后确定控制器的正或反作用方向。其原则是保证各部分(正或负号)的乘积为负即可,6.1.5 节将继续讨论这个问题。

2. PID 控制器的积分饱和及其抵抗方法

当控制器中有积分作用时,如果输入偏差 e 持续存在,并单方向积累,将会导致控制器输出达到、并一直维持在满量程。如果偏差依然存在,虽然控制器输出饱和,且输出无变化,但积分分量将使输出进入深度饱和,同时调节阀维持在最大开度,一直不变。当偏差改变极性后,由于输出的深度饱和,导致输出不能及时退出饱和区,往往需等一段时间才退出饱和,使得调节阀不能及时动作,从而导致系统出现较大的超调量,以及过长的过渡过程时间,严重时还会引起振荡,诱发事故。

当控制器的输出达到一定幅度之后,执行器不接受进一步的增长,而进入执行器的饱和和非线性区。积分饱和现象是由于执行机构的饱和和非线性引起的。造成积分器进入深度饱和的原因,一是偏差信号长期存在,二是控制器输出达到饱和值后,累加继续进行。如果控制器输出达到限幅值后,限制积分进一步累加,比如撤掉积分作用(或采取其他方法),使控制器输出不进入深度饱和,将来退出时会很快。

抗积分饱和的方法有好多种,既有硬件的方法,也有软件的方法,下面介绍的是硬件法中的一种。

图 3-4 是一种具有抗积分饱和的 PI 控制器电路,其中 A_1 为 PI 运算放大器, R_i 和 C_1 起积分作用, C_1 和 C_2 起比例作用。比较器 A_2 用来比较 u_o 和 u_h ,控制场效应管开关 S 的通和断。

开关 S 断开时,电路为 PI 控制器,其表达式为

$$U(s) = -\frac{C_1}{C_2} \left(\frac{1}{R_i C_1 s} + 1 \right) E(s)$$

当给输入施加 $-e_0$ 后,输出的变化如图 3-5 所示。当 u 上升到 $u = u_h$ 时,比较器 A_2 的输出使 S 闭合,于是 R_1 与 R_i 和 C_1 并联, R_2 与 C_2 并联。设计电路时,考虑 $R_1 = R_2$,且取值较小。随着 S 的闭合,输出 u 的上升停止, C_2 储能通过 R_2 释放,电压开始下降。一旦 $u < u_h$,则 S 断开,PI 控制器重新开始工作, u 开始上升,上升到 $u = u_h$ 时,重复前述过程。从宏观上看, u 维持在 u_h 上,不继续上升,也就不存在深度饱和问题。

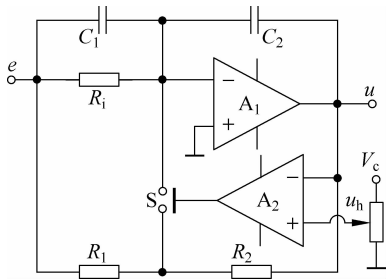


图 3-4 具有抗积分饱和的 PI 控制器电路

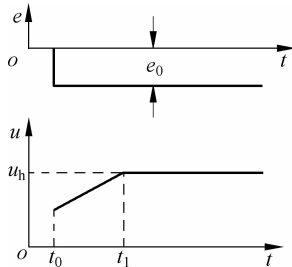


图 3-5 输入信号及其响应

3.2 模拟控制仪表

前面都是从原理上对常规 PID 控制算法进行讨论,实际中控制算法是由控制仪表来实现的。模拟控制器(Analog Controller)是一类基础的、应用极为广泛的控制仪表,有人称它为电动调节器(Electric Regulator/Controller)。该类仪表是用模拟电路实现的,早期控制仪表是用电子管制作的,为Ⅰ型控制仪表,中期控制仪表是用晶体管制作的,为Ⅱ型控制仪表,后期控制仪表是用集成电路制作的,为Ⅲ型控制仪表。下面谈的是电动单元组合Ⅲ控制仪表,即 DDZ-Ⅲ控制仪表。

DDZ-Ⅲ控制器仪表是Ⅲ型电动单元组合仪表中重要的单元之一,该类控制器除了有基本型外,还有专用型控制器,如前馈控制器、非线性控制器、自整定控制器、通断控制器等。基本型为 PID 控制器,它包括全刻度指示控制器和偏差指示控制器两种,两者的区别主要体现在指示电路上,我们不妨以全刻度指示控制器为例,介绍其结构与工作原理。

3.2.1 概况

图 3-6 为 DDZ-Ⅲ型 PID 控制器的结构图,它由输入电路、PD 电路、PI 电路、输出电路、手动电路和指示(显示)电路等组成。

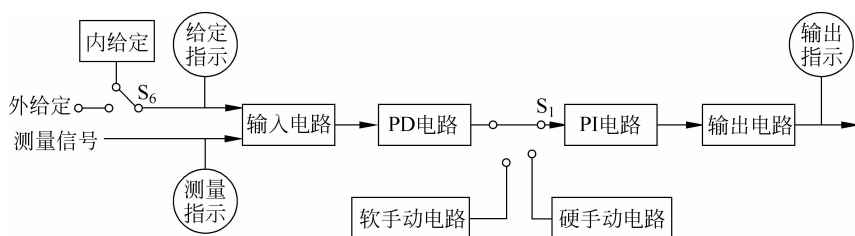


图 3-6 DDZ-Ⅲ型 PID 控制器的结构图

控制器接收来自检测变送器送来的测量值: $1\sim 5\text{V}$ 直流电压,与给定值(有内给定和外给定两种供选择)进行比较,将偏差送入控制器,经 PD 与 PI 组成的 PID 控制运算,获得相应的输出信号,即直流 $4\sim 20\text{mA}$ 电流。然后送往执行器执行。指示电路分别用来显示给定量、测量量和输出量的数值,手动操作用来调试与检查。

与图 3-6 相应的原理电路见图 3-7。它涉及很多具体的东西,我们将在下述内容中逐步讨论,先看一下控制器面板。

图 3-8 是全刻度指示控制器面板图,它汇集了给定、反馈、阀门位置、输出指示和手动操作于一体。图中 1 所指处为自动-软手动-硬手动切换开关,控制系统投入运行时,先手动,待其稳定后切换至自动。如果出现问题,则将自动切回为手动。这里手动又分软手动和硬手动。进行软手动操作时,按软手动操作板件 6,控制器的输出电流随时间按一定速度增加或减小,松开操作板键,则当时的信号值被保持。这是因为控制器输出电流与手动输入电压是积分关系。当切换开关处在硬手动状态时,控制器输出电流与手动输入电压成比例关系,控制器输出的大小取决于硬手动操作杆 4 的位置。通常情况下,用软手操作键进行手动操作,仅在要求给出恒定不变的操作信号,或者紧急情况下,才用硬手动操作。

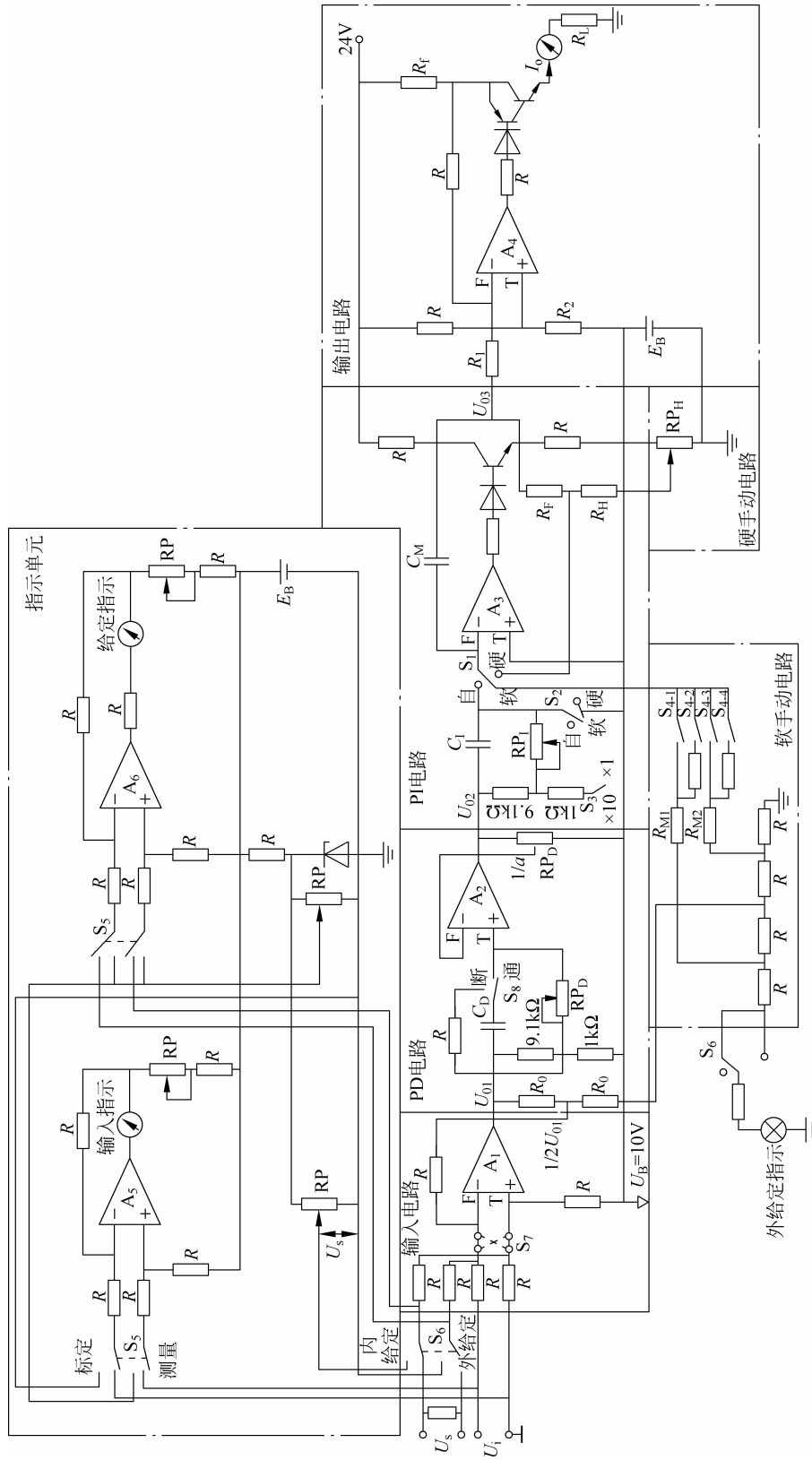


图 3-7 DDZ-III 型 PID 控制器原理图

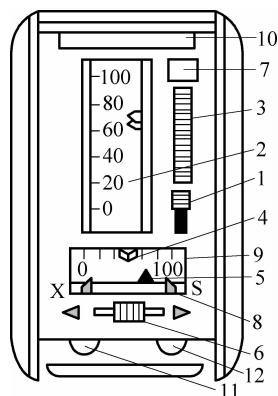


图 3-8 全刻度指示控制器面板图

1—自动/软手动/硬手动切换开关 2—给定、测量双针指示器 3—内给定设定轮 4—硬手动操作杆
5—输出指示器 6—软手动操作键 7—外给定指示灯 8—阀位指示器 9—输出指示器 10—仪表标
牌 11—输入检测插孔 12—手操器插孔

在表盘正中装有双针指示器,黑针代表给定信号,红针代表测量信号,两针之差即为偏差。图中 5 为输出指示器表明控制器输出信号大小,8 为阀位指示器,显示阀门开度,S 表示全开,X 表示关闭。11 为检测输入的插孔,12 为手动输出插孔,当控制器发生故障时,将便携式操作器的输入输出插头插入这两孔中,用手动操作器替代控制器进行控制。

另外,控制器的比例度、积分时间旋钮、微分时间旋钮、内外给定选择开关和控制器正反作用选择开关等,均设置在控制器机芯的右侧面(图中不可见),抽出机芯可见到这些旋钮和开关。下面分别谈谈各组成部分的工作原理。

3.2.2 输入电路

输入电路在图 3-7 中位于左下部,这里将它单独画出,详见图 3-9。它由运算放大器 A_1 、内外给定与选择、正反作用选择开关等组成,其作用是实现与给定信号 u_s 和测量信号 u_i 之差(或者测量信号与给定信号之差,具体需根据回路为负来确定,在单回路反馈控制系统中将详细陈述)成比例的偏差电压信号,为下一步控制算法提供输入。

由于偏差信号可能为正,也可能为负,而 DDZ-III 型仪表提供的电源是直流 +24V,难以实现正负电压,所以,该电路将原零伏电平上移到 +10V,于是原来的正偏差电压将大于 10V,原负偏差电压将小于 10V。这样的改变也为后续的 PID 运算提供了方便。

从图 3-9 来看,输入电路是一种差动输入方式,给定电压 u_s 与测量电压 u_i 分别接入运算放大器 A_1 的正负输入端,输出端为 u_{01} 。下面具体看看输入端与输出端之间的关系。

分别对图中 F 点和 T 点列基尔霍夫电流方程(Kirchhoff's Current Equation)(图中箭头所指方向为参考电流方向):

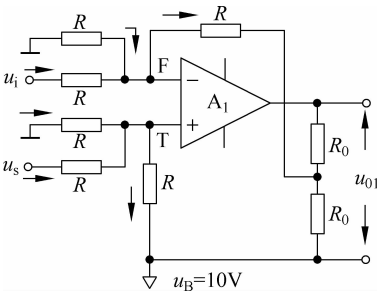


图 3-9 输入电路

$$\frac{u_s - u_T}{R} + \frac{-u_T}{R} - \frac{u_T - u_B}{R} = 0 \quad (3-7)$$

$$\frac{u_i - u_F}{R} - \frac{u_F}{R} - \frac{u_F - \frac{1}{2}u_{01} - u_B}{R} = 0 \quad (3-8)$$

由式(3-7)和式(3-8)有

$$u_T = \frac{1}{3}(u_s + u_B) \quad (3-9)$$

$$u_F = \frac{1}{3}\left(u_i + \frac{1}{2}u_{01} + u_B\right) \quad (3-10)$$

考虑理想运算放大器,有等式: $u_F = u_T$, 于是由式(3-9)和式(3-10),有

$$u_{01} = 2(u_s - u_i) = -2(u_i - u_s) \quad (3-11)$$

由此可知,输入电路的输出实现了两输入量的差,即偏差,并且放大了 2 倍。当然,式(3-11)中的输入量可改变符号,这可由图 3-5 中开关 S_7 来实施。另外,该电路也实现了以零伏为基准的输入信号,到以 10V 为基准的偏差信号转变。如果考虑运放 A_1 输出端对地的电压,则为

$$u_{10} + u_B = -2(u_i - u_s) + 10$$

这对后续电路来说,是满足其输入要求的。

3.2.3 PID 电路

1. 比例微分电路

紧接着输入电路,是全刻度指示控制器的 PID 运算电路,它由 PD 和 PI 两部分串联而成。先看 PD 电路部分,如图 3-10 所示。该电路可分两部分:无源比例微分电路和比例运算放大器。从图左边到运算放大器 A_2 正输入端为无源比例微分电路,运算放大器 A_2 及其右边为比例运算放大器。图中 R_d 、 C_d 和 R_p 分别为微分电位器、微分电容和比例电位器,调节 R_d 和 R_p 可改变 PD 控制器的微分时间和比例度。

下面分析 S_8 开关接通时, u_{01} 与 u_{02} 的关系。在图 3-10 电路中,考虑 A_2 为理想运算放大器,输入端口 1K 电阻两端的电压为 u_{01}/n (根据图中标出的数据, n 约为 10)。由基尔霍夫电压定律,有

$$u_T(s) = i_d R_d + \frac{1}{n} u_{01} \quad (3-12)$$

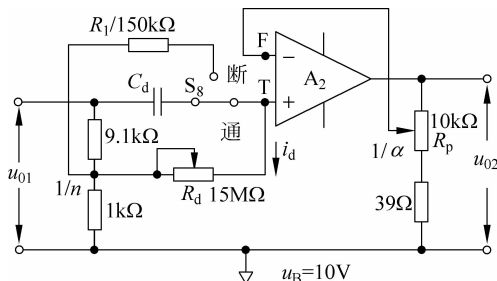


图 3-10 比例微分电路

鉴于流入理想运算放大器输入端口内部的电流很小,可忽略。所以,流过 R_d 的电流 i_d 也流过 C_d ,于是有

$$i_d(s) = \frac{(1 - 1/n)u_{01}}{\frac{1}{C_d s} + R_d} = \frac{n-1}{n} \frac{C_d s}{1 + R_d C_d s} u_{01}$$

将其代入式(3-12),得到

$$u_T(s) = \frac{1}{n} \frac{1 + nR_d C_d s}{1 + R_d C_d s} u_{01}$$

对于图 3-10 运算放大器 A_2 的输出输入端口,有 $u_T = u_F = \frac{1}{\alpha} u_{02}$,所以

$$u_{02}(s) = \alpha u_T = \frac{\alpha}{n} \frac{1 + nR_d C_d s}{1 + R_d C_d s} u_{01}(s) = \frac{\alpha}{n} \frac{1 + T_d s}{1 + \frac{T_d}{n} s} u_{01}(s) \quad (3-13)$$

其中, $T_d = nR_d C_d$ 为微分时间常数。当 $u_{01}(s)$ 为单位阶跃输入时, $u_{02}(s)$ 的时域响应为

$$u_{02}(t) = \frac{\alpha}{n} \left(1 + (n-1)e^{-\frac{n}{T_d}t} \right)$$

相应的波形图与图 3-2 相似。

在图 3-10 中,如果 S_8 开关投向断开,则微分作用取消,电路仅有比例作用,此时

$$u_{02} = \frac{\alpha}{n} u_{01}$$

可以分析,当开关从“断”投向“通”时,电路不会使输出 u_{02} 产生突变,也不会对后续电路带来冲击。

2. 比例积分电路

比例积分电路如图 3-11 所示。图中, S_3 为积分时间换挡开关,当其转向“ $\times 1$ ”位置时, $1k\Omega$ 的电阻将被废掉, C_i 的充电电压为 u_{02} ; 当其转向“ $\times 10$ ”位置时, $1k\Omega$ 电阻接入电路,后面将会看到, C_i 的积分时间将增大 m 倍(即 10 倍)。 S_1 、 S_2 为自动、软手动和硬手动切换开关; 运算放大器 A_3 的输出端接 $3.9k\Omega$ 电阻、二极管 VD_1 和三极管 VT_1 , 它们组成射极跟随器, 主要作用是进行功率放大。下面讨论该电路输入与输出的关系。

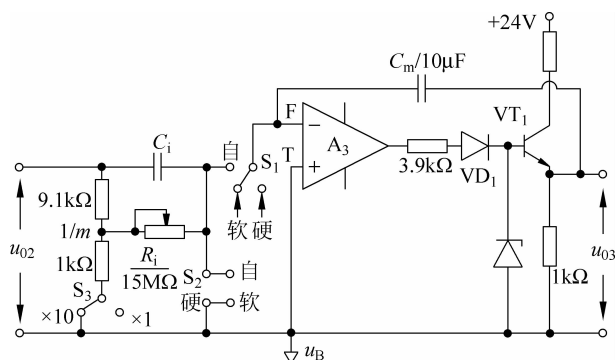


图 3-11 比例积分电路

将积分时间换挡开关 S_3 接在“ $\times 1$ ”的位置, S_1 开关打到自动位置。此时,对 F 点列基尔霍夫电流方程,并考虑由于 $9.1k\Omega$ 电阻远小于 $15M\Omega$ 电阻,可忽略其影响,并假设所有的

电流流入节点

$$\frac{u_{02}}{R_i} + \frac{u_{02}}{1/sC_i} + \frac{u_{03}}{1/sC_m} = 0$$

整理得到

$$u_{03}(s) = -\frac{C_i}{C_m} \left(1 + \frac{1}{R_i C_i s}\right) u_{02} = -\frac{C_i}{C_m} \left(1 + \frac{1}{T_i s}\right) u_{02} \quad (3-14)$$

其中, $T_i = R_i C_i$, 称作积分时间常数。

当 S_3 开关转向“ $\times 10$ ”时, 同样由基尔霍夫电流定律, 有

$$\frac{1}{m} \frac{u_{02}}{R_i} + \frac{u_{02}}{1/sC_i} + \frac{u_{03}}{1/sC_m} = 0$$

整理得到

$$u_{03}(s) = -\frac{C_i}{C_m} \left(1 + \frac{1}{m R_i C_i s}\right) u_{02} = -\frac{C_i}{C_m} \left(1 + \frac{1}{T_i s}\right) u_{02}$$

形式上与式(3-14)相似, 但是此时的积分时间常数 $T_i = m R_i C_i$, 即此时的积分时间常数比 S_3 接在“ $\times 1$ ”的位置时扩大了 m 倍(这里 $m=10$), 所以 S_3 也称为积分时间倍乘开关。

当输入 u_{02} 为阶跃信号时, 其输出响应为

$$u_{03}(t) = -\frac{C_i}{C_m} \left(1 + \frac{t}{T_i}\right) u_{02}$$

相应波形如图 3-12 所示。

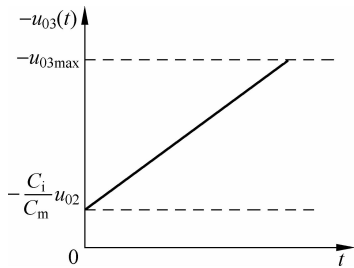


图 3-12 PI 阶跃响应

3. PID 传递函数

从输入电路, 经 PD 到 PI 电路, 我们已经获得了各部分的传递函数表达形式。如果考虑整个电路的传递函数, 将各部分串联起来, 即可获得整个传递函数的表达形式。如果给出阶跃输入信号, 也可以获得输出的表达式及其波

形。下面具体讨论。

由式(3-14)、式(3-13)和式(3-11), 可画出相应几部分的串联控制方块图, 如图 3-13 所示。输出表达式可写为

$$\begin{aligned} u_{03}(s) &= -\frac{C_i}{C_m} \left(1 + \frac{1}{T_i s}\right) \frac{\alpha}{n} \frac{1 + \frac{T_d s}{n}}{1 + \frac{T_d s}{n}} (-2)(u_i - u_s) \\ &= \frac{2\alpha C_i}{nC_m} \frac{1 + \frac{T_d}{T_i} + \frac{1}{T_i s} + T_d s}{1 + \frac{T_d s}{n}} (u_i - u_s) \end{aligned}$$

设 $F = 1 + \frac{T_d}{T_i}$ 为干扰系数, $\delta = \frac{nC_m}{2\alpha C_i}$ 为比例度, $K_d = n$ 为微分增益, 则上面的输出表达式可写为

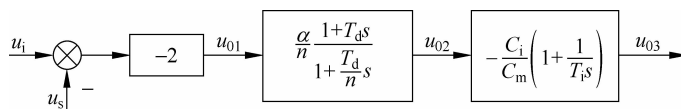


图 3-13 PID 控制器框图

$$u_{03}(s) = \frac{F}{\delta} \frac{1 + \frac{1}{FT_i s} + \frac{T_d}{F} s}{1 + \frac{T_d}{K_d} s} (u_i - u_s)$$

相应的传递函数为

$$\frac{u_{03}(s)}{u_i(s) - u_s(s)} = \frac{F}{\delta} \frac{1 + \frac{1}{FT_i s} + \frac{T_d}{F} s}{1 + \frac{T_d}{K_d} s}$$

若给定单位阶跃输入,即 $u_i(s) - u_s(s) = -1/s$, 则相应的响应波形趋势图如图 3-14 所示。

通常,在 DDZ-III 型控制器中有: $C_i = C_m = 10\mu\text{F}$, $C_d = 4\mu\text{F}$, $\alpha = 1 \sim 250$, $R_d = 62\text{k}\Omega \sim 15\text{M}\Omega$, $R_i = 62\text{k}\Omega \sim 15\text{M}\Omega$, $n = 10$, $m = 10$ 。

对于一般工业过程而言,相关 PID 控制器参数取值范围为: 比例度 $\delta 2\% \sim 500\%$; 微分时间常数 $T_d 0.04 \sim 10\text{min}$; 积分时间常数 T_i , 对“ $\times 1$ ”, 为 $0.01 \sim 2.5\text{min}$, 对“ $\times 10$ ”, 为 $0.1 \sim 25\text{min}$ 。

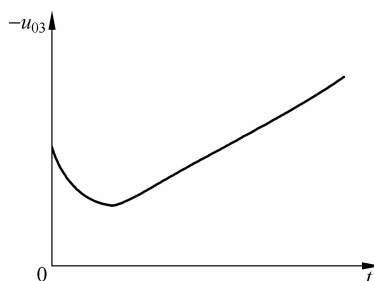


图 3-14 阶跃响应波形

3.2.4 输出电路

输出电路的主要任务是将电路输出的 $1 \sim 5\text{V}$ 电压信号转换为 $4 \sim 20\text{mA}$ 的电流信号, 以驱动执行器工作。同时, 由于前面的电压信号是以 10V 为起点, 在其上下波动, 所以需要将其转变为以 0V 为基点的信号。总之, 该电路的作用相当于一个比例环节。

输出电路见图 3-15。它由运算放大器 A_4 和复合管 VT_1 、 VT_2 等组成, A_4 进行电压放大, 复合管进行电流放大, 进而提高总的放大倍数, 增强恒流性能, 提高转换精度。

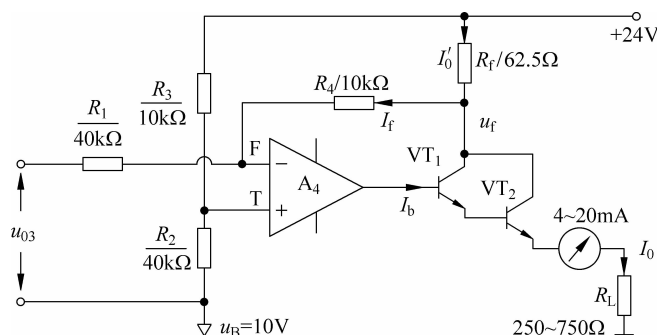


图 3-15 输出电路

对运算放大器 A_4 的输入端点 T 和 F, 分别列基尔霍夫电流方程

$$\frac{u_T - u_B}{R_2} = \frac{24 - u_T}{R_3}$$

$$\frac{u_{03} + u_B - u_F}{R_1} + \frac{u_F - u_T}{R_4} = 0$$

整理并代数, 有

$$u_T = \frac{R_3}{R_2 + R_3} u_B + \frac{R_2}{R_2 + R_3} \times 24 = \frac{1}{5} u_B + \frac{4}{5} \times 24$$

$$u_F = \frac{R_1}{R_1 + R_4} u_f + \frac{R_4}{R_1 + R_4} (u_{03} + u_B) = \frac{4}{5} u_f + \frac{1}{5} (u_{03} + u_B)$$

由于 $u_T = u_F$, 所以有

$$u_f = -\frac{1}{4} u_{03} + 24$$

又由图 3-15 有

$$u_f = -R_f I'_0 + 24$$

比较上两式, 可有

$$I'_0 = \frac{1}{4R_f} u_{03} \quad (3-15)$$

考虑 VT_1 的基极电流 I_b 和运算放大器 A_4 的反馈电流 I_f 都比较小, 如果忽略的话, 则有

$$I_0 = I'_0 - I_f + I_b \approx I'_0$$

于是, 式(3-15)可变为

$$I_0 = \frac{1}{4R_f} u_{03} \quad (3-16)$$

若取 $R_f = 62.5\Omega$, 则当 $u_{03} = 1 \sim 5V$ 时, $I_0 = 4 \sim 20mA$ 。即完成了规范电压到规范电流的转换。

式(3-16)是在忽略 I_b 和 I_f 的基础上得到的。实际上, I_f 比 I_b 大, 如果不忽略 I_f , I_0 的精度会提高。可以证明, 当 $R_1 = 4(R_3 + R_f) = 40.25k\Omega$ 时, 式(3-16)可以比较准确地实现电压—电流的转换。

3.2.5 手动操作电路及无扰切换

手动操作(Manual Operation)是一种不同于自动控制的另一类调节方式, 它往往用于投运前的设定与调试, 或者检查与维护等情况。这里的手动操作包括软手动操作和硬手动操作两种。所谓软手动操作是指控制器的输出(电流)与手动输入(电压)之间呈积分关系, 而硬手动操作是控制器的输出(电流)与手动输入(电压)之间呈比例关系。由于积分具有慢变的特点, 所以行动具有“软”性; 而比例具有迅速反应的个性, 所以行动具有“硬”性。手动操作电路如图 3-16 所示, 它是前述的 PI 电路的基础上, 附加软手动操作和硬手动操作电路而成的。图中, S_1 、 S_2 为自动/软手动/硬手动联动切换开关, $S_{41} \sim S_{44}$ 为软手动操作开关, RP_H 为硬手动操作电位器。

1. 硬手动操作电路

当联动切换开关 S_1 和 S_2 转至硬手动状态时, 其电路如图 3-17 所示。此时电容 C_i 右端接至 u_B , 它两端的电压最后与 u_{02} 一致, 这为电路再从硬手动切回至自动状态时, 电路无扰动创造了条件。

由图 3-17, 对运放 A_3 负输入节点列电流方程, 并整理后, 有

$$\frac{u_{03}(s)}{u_h(s)} = -\frac{R_f}{R_h} \frac{1}{1 + R_f C_m s} = -\frac{1}{1 + 0.3s}$$

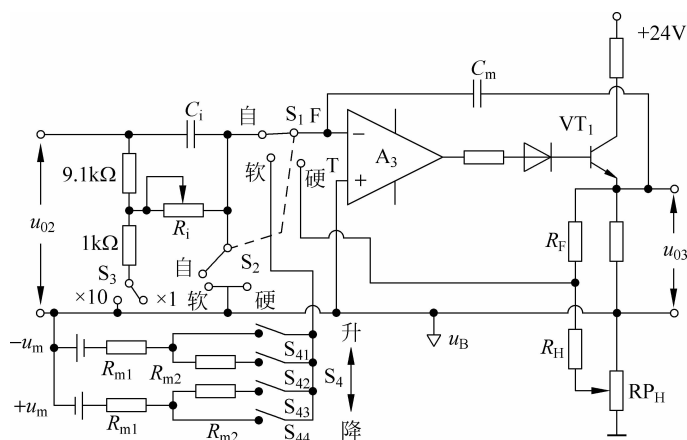


图 3-16 手动操作电路

由于惯性环节的时间常数较小,该环节可近似看作是一个比例环节。控制器的输出可以由硬手动操作电位器 RP_H 来调整。

值得指出的是,当电路由自动切换至硬手动时,由于硬手动操作电位器活动臂位置的不同,控制器输出 u_{03} 一般会有突变,即为有扰切换。如果要消除切换扰动,需事先调节好 RP_H 的位置,使其与切换前的控制器输出一致,即先平衡,才无扰。而当电路由硬手动转向自动状态时,由于电容 C_i 的作用,切换是无扰动的。

2. 软手动操作电路

当联动开关 S_1 、 S_2 投向软手动后,如果软手动操作扳键开关 $S_{41} \sim S_{44}$ 无一接通,则此时由于 A_3 的负输入端悬空,其输出 u_{03} 将在 C_m 的作用下,维持原输出值不变,处在一种“保持”状态。即此时不会对输出有扰动。另外,此时由于 S_2 与 u_B 的接通, C_i 两端的电压与 u_{02} 相等,当 S_1 和 S_2 再由软手动操作投回自动时, A_3 的输出电压也不会有突变(S_4 中有开关合上的情况除外)。

当处在软手动状态后,如果按下 S_4 的任一扳键开关,电路将成为一个反相输入的积分运算器,如图 3-18 所示。相应的输出表达式为

$$u_{03} = -\frac{\pm u_m}{R_m C_m} t \quad (3-17)$$

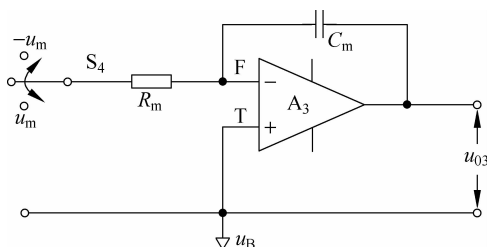


图 3-18 软手动操作等效电路

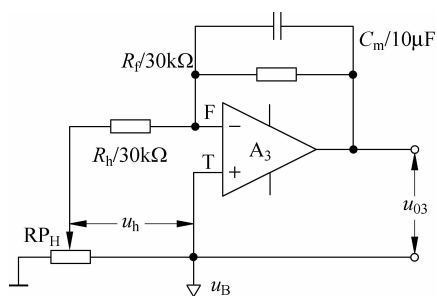


图 3-17 硬手动操作等效

正负号取决于开关的投向,而且开关闭合的数量不同, R_m 也会有变化,总之,这四个开关决定着输出 u_{03} 的极性和变化速率。

具体来说,当操作者向“升”的方向轻轻推压时,首先是 S_{42} 接通, $-u_m$ 经 R_{m1} 、 R_{m2} 向电容 C_m 充电,此时 $R_m = R_{m1} + R_{m2}$,它乘以 C_m 即为积分时间常数, u_{03} 按 $u_m/R_m C_m$ 斜率上升;当用力推压时,除了 S_{42} 接通之外, S_{41} 也被合上,此时 R_{m2} 被短接, $R_m = R_{m1}$,显然,积分时间常数减小, u_{03} 以比前者更大的斜率上升,即上升速度更快。当输出达到预期的数值时,松开按键, S_{42} 和 S_{41} 回复断开,从而 A_3 的负输入端因断开而悬空,输出 u_{03} 停止上升,其值被电容“保持”不变。当然,随着时间的延长,由于 C_m 存在漏电,输出值也会有所减小。同理,当 S_{43} 和 S_{44} 接通时,情形与前述相似,仅仅是输出的极性不同而已。

3. 软硬手动操作之间的切换

当控制器由软手动向硬手动切换时,其输出值将会由原来的某值跃变为硬手动操作电位器 RP_H 确定是数值,这将会产生扰动。为消除其扰动,需要在切换之前调整好硬手动操作器的位置,使其控制器输出与切换前的控制器输出一致。

反过来说,当控制器由硬手动操作转为软手动操作时,由于切换后电路呈保持状态,使得切换前后的输出值不变化,即为无扰切换。

综合前面的讨论,现将自动-软手动-硬手动之间切换有无扰动情况进行小结,详见图 3-19。

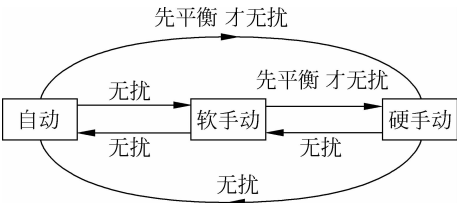


图 3-19 自动/软手动/硬手动切换时扰动情况

3.2.6 指示电路

从图 3-8 全刻度指示控制器面板图中可知,序号 2 所指为测量、给定双针指示器,它们的功能就是指示/显示系统被测量的测量值和给定量的设定值,这两个值的绝对值大小可以由指针所指刻度反映出来,两指针相距越远,表明数值相差越大,两针重叠,表明测量值与设定值数值相等。

由于 DDZ-Ⅲ 仪表使用的指示表是 5mA 的满偏转电流表,它需将以零为基础的 1~5V 直流电压信号,转换为以 u_B 为基础的 1~5mA 直流电流。具体电路如图 3-20 所示。

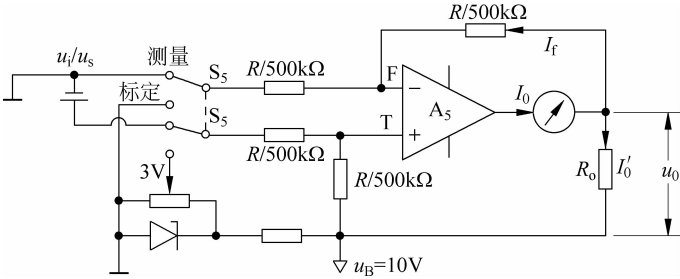


图 3-20 输入指示电路

图 3-20 中,分别对 F 点和 T 点列基尔霍夫电流方程

$$\frac{u_F}{R} = \frac{u_0 + u_B - u_F}{R}$$

$$\frac{u_T - u_B}{R} = \frac{u_i - u_T}{R}$$

整理为

$$u_F = \frac{1}{2}(u_0 + u_B)$$

$$u_T = \frac{1}{2}(u_i + u_B)$$

由于 $u_F = u_T$, 所以有

$$u_0 = u_i$$

如果忽略流过反馈回路的电流 I_f , 则有

$$I_0 \approx I'_0 = \frac{u_0}{R_0} = \frac{u_i}{R_0}$$

在取 $R_0 = 1\text{k}\Omega$ 的情况下, 当 u_i 为 $1\sim 5\text{V}$ 时, I_0 为 $1\sim 5\text{mA}$ 。

另外, 为了方便对指示电路进行校验, 图 3-20 中除了设有测量功能之外, 还设有标定功能, 两者的选择由双键联动开关 S_5 来实现。当它置于“标定”位置时, 应有 3V 的电压输入电路, 同时, 输出电流表应有 3mA 的电流。如果指示不准确, 可进行表头的零点调整。

以上分六部分对 DDZ-Ⅲ型 PID 控制器的主要内容进行了介绍, 除此之外, 还有一些辅助内容, 如整流、滤波、补偿、保护等, 限于篇幅, 不过多介绍。

3.3 数字控制仪表

模拟控制仪表虽然在一定程度上满足了过程控制的需要, 但是, 它的局限性也是明显的, 主要表现在:

- ◇ 功能单一, 应用范围窄, 灵活性差;
- ◇ 仪表布局分散, 不便于集中监视和操作;
- ◇ 布线多而杂, 维护与改造不方便。

随着计算机技术的发展与进步, 数字式控制仪表于 20 世纪 70 年代逐步进入应用领域。这是一类不同于模拟式的控制仪表: 以微处理器(Microprocessor)为核心, 通过编制应用程序或组态(Configuration), 实现运算及控制功能。它接受多路模拟量和开关量的输入, 按编制的控制算法或组态, 进行控制运算, 输出信号经 D/A 转换, 供执行器实施, 以实现既定的控制目标。

数字式控制器通常是通过编程或者组态来设计 PID 之类的控制规律的, 所以, 工程中称其为可编程控制器或可编程调节器(Programmable Controller)。它是一种将回路控制、数学运算、逻辑运算及通信等多种功能融于一体的智能仪表, 具有如下特点:

① 控制器功能的实现依赖于软件。一般情况下, 将若干完成运算及控制的功能模块(子程序)固化在 ROM(Read Only Memory)中, 当用户完成了编制应用程序或组态工作之后, 才能实现预定的运算控制功能。

② 编程所使用的是一种专用的 POL 语言(Problem Oriented Language)。这种语言简单易学, 但专用性较强。程序编好后, 或用专用的编程器, 或通过控制器本机上的按钮、开关写入 EPROM(Erasable Programmable Read-Only Memory)。

③ 控制器内部功能模块采用软连接, 即所谓的“组态”, 外部采用硬连接。巧妙的组态

是发挥可编程控制器的关键。

④ 控制器的模拟量输入输出采用国际统一标准信号(4~20mA DC,或 1~5V DC),大多具有多个模拟量输入/输出通道和开关量输入/输出通道。其中只有一个 4~20mA DC 输出的称为单回路可编程控制器。它尤其适用于构成复杂控制系统。

⑤ 数字显示与模拟显示混合使用。面板设计尽可能沿袭模拟调节器人机对话的形式,极易实现操作人员在技术上的过渡。

⑥ 具有通信功能。通过适当的接口,可以与操作站或上位机进行通信,从而实现中、大规模的集中监视、操作和管理。

⑦ 在硬件设计上采用了一系列可靠性技术措施。如使用大规模集成电路,尽量减少元件的数量,备用电源保护 RAM(Random Access Memory)中的信息等。

⑧ 具有自诊断功能。这是可靠性在软件上的具体措施,它能随时对自身进行故障监视,一旦出现问题即时采取相应措施,并输出故障状态警报。

⑨ 精度高,性能稳定。可编程调节器的精度主要取决于 A/D、D/A 转换技术和计算机技术,计算过程中由于受环境干扰、温度影响极小,故精度和稳定性很高。

近年来,由于技术进步和生产成本的下降,DCS、PLC、FCS 等控制系统不仅牢牢占据大、中规模生产过程市场,而且还逐渐延伸到小规模生产过程市场中,数字控制器的生存空间被大大挤压。但是,现实情况是,一些小规模生产过程和附属设备根本不需要 DCS、PLC、FCS 这类中大型控制系统,而是青睐规模小、组建灵活、成本低廉的数字控制器。加上数字控制器近年来自身的技术进步,如强化控制能力和联网通信功能等,使得数字控制器市场不仅没有消失,而且发展方兴未艾。目前数字控制器市场的产品繁多,像德国 EMG 公司的 SPC 数字控制器、德国西门子公司的 PAC353 控制器、美国 Honeywell 公司的 XL 系列数字控制器、日本横河公司的 YS1700、日本大仓电气株式会社的 EC 系列高性能数字式控制器,以及国内浙江中控自动化仪表公司推出的 MultiF C3000 控制器等,使用率比较高。早期的产品,如 DK 系列的 KMM 控制器(日本山武-霍尼韦尔公司)、YS80 系列的 SLPC(日本横河公司)、FC 系列的 PMK 控制器(日本富士公司),以及 VI 系列的 VI87MA-E 控制器(日本日立公司)等,都曾经显赫一时。这些产品后来经过改进或升级,也都继续发挥着重要作用。

3.3.1 数字式控制器的基本构成

1. 硬件电路与组成部分

数字式控制器硬件电路如图 3-21 所示。它由主机、过程输入通道、过程输出通道、人机联系设施和通信电路等部分组成。

主机:由中央处理单元(CPU)、只读存储器(ROM 和 EPROM)、随机存储器(RAM)、定时/计数(CTC)器,以及输入输出接口等组成。

输入电路:包括模拟量输入和开关量输入两部分。模拟量输入电路由多路模拟开关、采样/保持器和 A/D 转换器等组成,其作用是将模拟量转化为相应的数字量;开关量经过输入缓冲器,将开关量转变为能被计算机识别的数字信号。

输出电路:包括模拟量输出通道和开关量输出通道两部分。其中模拟量输出通道由 D/A 转换、多路模拟开关、输出保持电路和 V/I 转换等组成,其作用是将数字信号转换为 1~5V 的模拟电压和 4~20mA 的模拟电流。开关量经输出锁存器,以便可靠地控制触点,

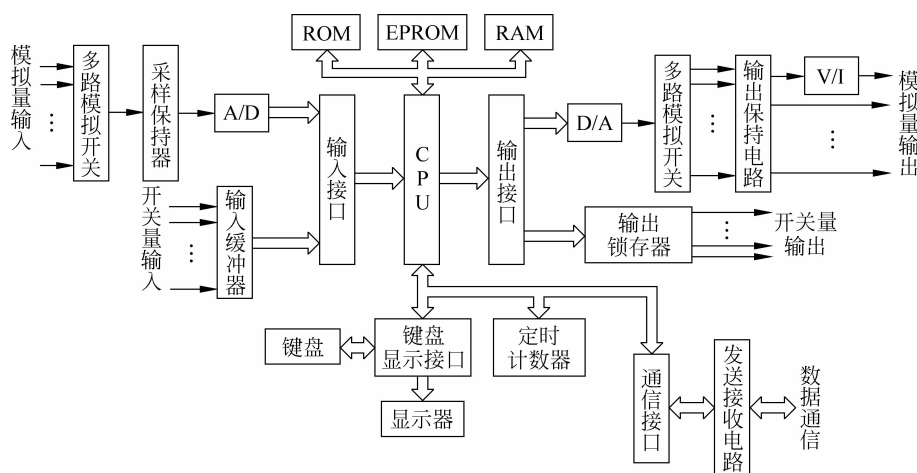


图 3-21 数字式控制器硬件组成框图

或保持电位状态。

人机联系设施：包括显示仪表或者显示器、操作按钮与开关、键盘等。它们有的置于仪表正面，有的置于仪表侧面，有的在控制台上，根据各个生产厂家的设计不同而异。

通信电路：由通信接口和发送接收电路等组成。将发送的数据转换为标准通信格式的数字信号，经发送电路送往外部通信线路，接收电路接到该信号后，转换为计算机能识别的数据。它也接受来自操作站或上位机的操作命令和控制参数。

2. 软件系统

数字控制器的软件包括系统管理软件和用户应用软件两部分。

系统管理软件包括监控程序和中断处理程序两部分。监控程序包括系统初始化、显示与键盘管理、中断管理、运行状态管理和自诊断处理等功能模块；中断处理程序包括定时处理、输入处理、通信处理、断电处理和运算控制等模块。

用户应用软件由用户根据具体情况使用。该应用程序以指令或模块的形式出现，用户用 POL 编程，或将模块按一定规则进行连接（组态），形成控制系统。编程通过专用的编程器进行，程序调试或修改完成后，写入 EPROM 中，即可投入运行。

3.3.2 数字式控制器实例

由于数字式控制器形式众多，我们不妨选取一种，通过对它的认识与了解，达到举一反三的目的。下面以单回路可编程控制器（Single Loop Programmable Controller, SLPC）为例，这是日本横河公司 1980 年推出的 YS-80 系列产品中的一种类型（A 型），1986 年对它进行了改进（E 型）。尽管公司后来还推出了进步型的 YS-170（1991 年）和 YS-1700（2007 年），但是 SLPC 不仅在当时具有典型意义，而且现在仍有参考和指导作用。

1. SLPC 电路及其各组成部分

图 3-22 为 SLPC 控制器的电路结构框图。

主机：CPU 为 8 位微处理器 8085A，时钟频率为 10MHz，控制周期为 0.1s 和 0.2s。使用两片 27256 芯片作为系统 ROM（64KB），用来存放监控程序和各种功能模块；使用 1 片

2716 作为用户 ROM,提供 2KB 存储空间,用来存放用户程序;采用四片 PD4464 低功耗 CMOS 存储器作为 RAM(16KB),用来存放设计参数和计算结果。

模拟量输入/输出通道:模拟量输入有 5 路: $X_1 \sim X_5$,接受 1~5V 直流输入信号,输入模拟量经 RC 滤波、输入多路开关,进行 A/D 转换后,送入 CPU;经 CPU 的控制运算后,输出数字量,再经 D/A 转换回模拟量。输出通道中有两路模拟量输出为 1~5V 直流电压(Y_2 和 Y_3),一路为 4~20mA 直流电流(Y_1),它可驱动现场执行器。由于原则上它只能驱动一个控制回路,所以称其为单回路可编程控制器(SLPC)。其中,用一片型号为 μ PC648D 的 12 位高速 D/A 芯片,将 CPU 输出的数字量转变为模拟量。同时,该芯片借助程序支持,实现逐位比较式 A/D 转变。

开关量输入/输出端口:有 6 个开关量输入输出端口,可输入或输出通/断型,以及高 ($3.5 \pm 1V$)/低 ($0 \pm 1V$)型电平 of 的开关量,所有开关量的输入和输出与内部电路之间均有高频变压器隔离。这六个 DI/DO 端口是通过编程定义输入和输出功能的。

人机联系:在仪表侧面或正面,装有八位十六段笔画显示器,以及十六个键码的键盘,键盘/显示器接口用 8279 芯片。键盘可以设定和变更多种运行参数,有的参数可通过显示器显示出来。例如,PID 控制运算的测量值、给定值、偏差、输出电流、比例度、积分时间、微分时间,以及各种报警设定等。另外,用户程序是用专门的 SPRG 编码器编写的,使用时通过电缆接到编程器插口上,借助 SLPC 的 CPU 工作。程序编好后,可先在 RAM 中试运行,修改无误后,写入 EPROM。如果需要改变 EPROM 中的程序,可用紫外线照射 EPROM,擦除原来固化在其中的程序,然后写入新的用户程序。

通信接口电路:由 8251 型可编程通信接口芯片和光电隔离电路组成。它采用半双工、串行异步通信方式,将发出的信号转换为标准通信格式的数字信号,同时将外部通信信号转换成 CPU 能接受的数据。SLPC 既可在没有上位机的情况下独立工作,也可与上位机系统(YEWPACK 或 CENTUM 集散控制系统)连接,进行数据通信,在集散控制系统操作站的监视与管理下工作,成为集散控制系统的一个基层单元。

2. 数字控制器的功能

(1) 数字运算控制程序与算式

SLPC 的软件有系统软件(管理程序)与应用软件。应用软件包括过程控制软件包与用户程序。ROM 中植入了过程控制需要的几十种运算控制功能,每一种运算控制功能都有一段相应的实现该功能的程序,称作功能模块。这些模块化的标准程序组成了过程控制软件包,它们在制造 SLPC 时已被存放在 ROM 中,供用户调用。由用户编写的、按实际运算控制需要调用过程控制软件包中标准程序的语句段,就是用户程序。SLPC 的用户程序是指令语句式程序,用面向问题的、助记符式的语言编写。这种语言包括许多条指令语句,每条指令语句与过程控制软件包中的一个标准程序相对应。用户用一条指令语句即可从过程控制软件包中调用一个相应的标准程序,执行相应的动作。用户根据过程控制方案中需要进行的运算,选用若干条指令语句,按适当次序组成一个完整的程序段落,通过编程器存入控制器中的 EPROM 中。控制器在每个控制周期(0.1s 或 0.2s)内依次调用过程控制软件包中的若干个标准程序,执行用户程序规定的全部运算,然后输出数字式的控制量,经 D/A 转换为模拟量,传输给执行器实施。

SLPC 的主要控制功能包括基本控制、串级控制、选择控制等。基本控制算法除了常规

的 PID 外,还有几种特殊的 PID 控制算法,如微分先行的 PID 控制(PI-D)、比例微分先行的 PID 控制(I-PD)、带可变型设定值滤波的 PID 控制、采样 PI 控制、批量 PID 控制等。它们适用于不同的场合与控制要求。下面介绍几种常用的控制算式。

① 基本 PID 算式

PID 的控制算式时域输出表达式为

$$y(t) = \frac{1}{\delta} \left(e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right)$$

其中, $e(t)$ 为输入偏差, δ 、 T_i 和 T_d 分别为比例度、积分时间和微分时间常数。

对上面表达式进行离散化,并考虑积分和微分

$$\int e(t) dt \approx T_s \sum_{i=0}^k e(i) \quad (3-18)$$

$$\frac{de(t)}{dt} \approx \frac{e(k) - e(k-1)}{T_s} \quad (3-19)$$

其中, T_s 为采样周期。于是,可获得第 k 次采样后的 PID 位置算式

$$y(k) = \frac{1}{\delta} \left(e(k) + \frac{T_s}{T_i} \sum_{i=1}^k e(i) + \frac{T_d}{T_s} (e(k) - e(k-1)) \right)$$

其中, $e(i)$ 为偏差信号第 i 次的采样值。相应的增量算式为

$$\begin{aligned} \Delta y(k) &= y(k) - y(k-1) \\ &= \frac{1}{\delta} \left[(e(k) - e(k-1)) + \frac{T_s}{T_i} e(k) \right. \\ &\quad \left. + \frac{T_d}{T_s} (e(k) - 2e(k-1) + e(k-2)) \right] \end{aligned} \quad (3-20)$$

实际中,为了避免微分部分对高频干扰的放大,往往对其进行改进,即将式(3-20)右边中括号中的第三项进行改造,从而获得实用的 PID 控制算式。选用的近似微分传递函数为

$$\frac{Y_d(s)}{E(s)} = \frac{T_d s}{1 + \frac{T_d}{K_d} s}$$

其中, K_d 为微分增益。该近似微分与原微分相比,多了一个一阶惯性环节。上式可写为:

$$Y_d \left(1 + \frac{T_d}{K_d} s \right) = T_d s E(s)$$

相应的微分方程为

$$y_d + \frac{T_d}{K_d} \frac{dy_d}{dt} = T_d \frac{de}{dt}$$

其差分方程为

$$y_d(k) + \frac{T_d}{K_d} \frac{y_d(k) - y_d(k-1)}{T_s} = T_d \frac{e(k) - e(k-1)}{T_s}$$

整理为

$$y_d(k) = \frac{T_d}{T_s + T_d/K_d} [e(k) - e(k-1)] + \frac{T_d/K_d}{T_s + T_d/K_d} y_d(k-1)$$

由于它已经是增量形式,将其直接替换式(3-20)右边括号中微分项,并注意 $y_d(k-1) \rightarrow y(k-1)$,式(3-20)则为

$$\begin{aligned}\Delta y(k) &= y(k) - y(k-1) \\ &= \frac{1}{\delta} \left[(e(k) - e(k-1)) + \frac{T_s}{T_i} e(k) + \frac{T_d}{T_s + T_d/K_d} (e(k) - e(k-1)) \right. \\ &\quad \left. + \frac{T_d}{T_s + T_d/K_d} y(k-1) \right]\end{aligned}$$

② 微分先行的 PID 控制算式(PI-D)

实际操作中,改变给定时,由于微分的作用,往往会而引起输出的跳变,进而影响系统的稳定性。为此,可对这种状况进行改进,具体思路是微分仅对测量值进行,避开对给定值的微分,即微分先行的 PID 控制算式,也称 PI-D 算式。相应的输出表达式和控制框图见图 3-23。

$$Y(s) = \frac{1}{\delta} \left[\left(\frac{1}{T_i s} + 1 \right) E(s) - T_d s Y_m(s) \right]$$

其中 Y_m 为输出测量值, R 为给定。

另外,也可以在给定输入通道(偏差之前),追加一个惯性环节,用以缓解改变给定对输出的冲击。

③ 比例微分先行的 PID 控制算式(I-PD)

对于 B 的情况外,有时为了避免比例部分对突然改变给定的冲击,可将比例部分纳入先行之列,形成比例微分先行的控制算式

$$Y(s) = \frac{1}{\delta} \left[\frac{1}{T_i s} E(s) - (1 + T_d s) Y_m(s) \right]$$

相应的算式框图见图 3-24。

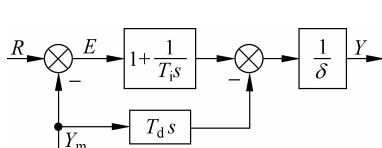


图 3-23 微分先行的 PID 算式框图

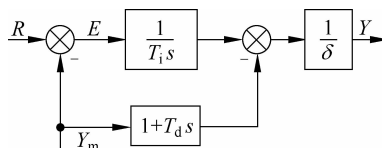


图 3-24 比例微分先行 PID 算式框图

④ 可调参数的滤波器 PID 算法(Setting Value Filter, SVF)

具有可调参数滤波器的 PID 算法框图如图 3-25 所示,输出表达式为

$$Y(s) = \frac{1}{\delta} \left(1 + \frac{1}{T_i s} + T_d s \right) \left(\frac{(1 + \alpha T_i s)(1 + \beta T_d s)}{(1 + T_i s)(1 + T_d s)} R - Y_m \right)$$

其中, α, β 均为滤波器调整参数,且 $\alpha = 0 \sim 1, \beta = 0 \sim 1$ 。通过调整这两个参数,可获得最佳滤波效果。

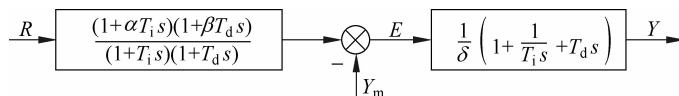


图 3-25 滤波参数可调的 PID 框图

在进行 PID 控制器设计与调整时,需确定控制器的比例度、积分常数和微分常数,以及控制器的正/反作用方式。该控制器的比例度 δ 可调范围是 6.3%~999.9%,积分时间常

数 T_i 的可调范围为 $1\sim 9999s$, 微分时间常数 T_d 的可调范围为 $0\sim 9999s$ 。这些参数的最后确定, 需根据具体的被控过程和对系统性能指标的要求来综合考虑。

(2) 自整定功能与自诊断功能

改进版的 SLPC 配有自整定控制(Self-tuning Control, STC)功能。它利用微处理机技术, 将熟练操作人员、系统工程师的参数整定经验, 整理成多种调整规程, 储存在 ROM 的“知识库”中。在控制运行过程中, 由程序根据各项控制指标反映出来的控制特性及其动态变化, 自动选择与调用知识库中的调整规程, 计算出最佳 PID 的参数(如比例度、积分时间、微分时间等), 向操作人员显示或进行自动变更, 从而达到期望的控制效果。当然, 什么样的控制效果是最佳的, 应根据具体生产过程而定。例如, 有的过程要求被控量的超调量很小, 有的侧重被控量衰减振荡收敛得较快等。不同的生产过程对控制指标要求可能有所不同或侧重, STC 功能针对这些情况设计了几种控制目标形式, 由用户选择设定。实际上, 这种自整定功能就是通常所说的“专家系统”(Expert System)的初级形式。

该控制器能实施周期性自诊断, 当有内部电路的重要器件发生故障, 或运算出现异常, 或过程参数发生变异时, 可通过面板指示灯 FALL 或 ALM, 将异常信息告诉操作人员, 同时自动采取某些应急措施。如果要了解异常情况的具体内容, 可通过侧面板的键盘操作, 使其在显示器中显示出来。SLPC 设有一组自诊断标志寄存器 FL17~FL29, 当发生某种异常情况时, 其中相应的某个寄存器内的状态信号自动由 0 变为 1, 可利用用户程序查出。

3. 用户程序

为了方便用户针对具体问题编写控制程序, SLPC 的设计者建立了面向问题的语言 POL 编程平台(前面提到的用户程序编码器 SPRG)和编制规则, 并将常用的各种运算控制功能编制成相应的标准功能程序模块, 植入 ROM 中。这相当于模拟控制中具有一定功能的组合仪表, 用户使用时, 根据需要, 将从中挑选相应的功能程序模块, 用指令连接起来, 组成一个完整的程序段。运行后, 发出一系列的指令, 完成一系列动作, 满足数字控制的需要。实际上, 这种编程方式就是一种“组态”行为。

SLPC 的用户指令一共有八类 46 条。这八类分别是输入、输出、结束、基本运算、函数运算、条件判断、存储位移和控制功能等。所有指令均以运算寄存器为操作平台进行, 运算寄存器一共有 5 个, 分别标为 $S_1\sim S_5$, 它们是在 RAM 中指定的先进后出的堆栈。

(1) 输入/输出、结束模块

输入指令: LD 数据寄存器名(读入数据寄存器, 即数据寄存器内容 $\rightarrow S_1$)

SLPC 有 16 个数据寄存器, 分别存放各类数据。它们分别是 X_n 模拟量输入数据寄存器、 Y_n 模拟量输出数据寄存器、 P_n 可变常数寄存器、 K_n 固定常数寄存器、 T_n 中间数据暂存器、 A_n 模拟量控制模块功能扩展用寄存器、 FL_n 状态标志寄存器(开关量)、 B_n 控制模块整定参数寄存器、 D_{in} 开关量输入寄存器、 D_{on} 开关量输出寄存器、 E_n 通信接收用模拟量寄存器、 D_n 通信发送用模拟量寄存器、 C_{In} 通信接收用数字量寄存器、 C_{on} 通信发送用数字量寄存器、 KY_n 可编程功能键状态输入寄存器、 LP_n 可编程功能指示灯输入寄存器。

输出指令: ST 数据寄存器名(向数据寄存器输出, 即 $S_1\rightarrow$ 数据寄存器)

结束指令: END(运算终结)

这里, LD 是 Load 的简写, ST 是 Store 的简写。

(2) 基本运算模块

一共有 11 个,它们是 +、-、 $\times$ 、 $\div$ 、 $\sqrt{\quad}$ 、 $\sqrt{E}$ 、ABS、HSL、LSL、HLM、LLM。其中, $\sqrt{\quad}$ 为开方、 $\sqrt{E}$ 为小数点删除型开方,ABS 为取绝对值,HSL 为高值选择,LSL 为低值选择,HLM 和 LLM 分别为高低限幅。

① 四则运算模块 +、-、 $\times$ 、 $\div$

用运算寄存器 S_1 和 S_2 进行,结果存进 S_1 中。作减法和除法运算时, S_2 中存放被减数或被除数, S_1 中存放减数或除数。

② 开方运算模块 $\sqrt{\quad}$ 、 $\sqrt{E}$

一般用于差压信号计算流量值,并且具有微弱信号忽略(删除)功能。这是由于差压信号弱小时,检测误差大,开方后的值误差就更大,此时删除比较合理。其中, $\sqrt{\quad}$ 的小信号删除点是固定的,即小于量程的 1% 时,令开方结果为零;而 $\sqrt{E}$ 的小信号删除点是变化的,运算前,小信号删除值存入 S_1 寄存器,被开方数置于 S_2 寄存器,当开方的结果小于删除值时,输出为零。

③ 其他

ABS 为取绝对值模块,数据置于寄存器 S_1 中,取绝对值后的结果仍置于 S_1 中。HSL 和 LSL 是从寄存器 S_1 、 S_2 中分别选取高值或低值,结果放在 S_1 中。HLM 和 LLM 是将 S_2 中的数据幅值限制在寄存器 S_1 规定的上下限范围之内。

(3) 函数运算模块

函数运算模块有 13 个,它们是:FX1,2(十折线函数)、FX3,4(任意折线函数)、LAG1~8(一阶惯性)、LED1,2(微分)、DED1~3(纯滞后)、VEL1~3(变化率运算)、VLM1~6(变化率限幅)、MAV1~3(移动平均运算)、CCD1~8(状态变化检出)、TIM1~4(计时运算)、PGM1(程序设定)、PIC1~4(脉冲输入计数)、CPO1,2(积算脉冲输出)。

① 折线函数模块 $FX_1 \sim FX_4$

用 10 段折线逼近非线性函数模块。但是, FX_1 和 FX_2 的折线是在自变量轴上平均分段,而 FX_3 和 FX_4 是自由分段的,它是为了让编程者能根据函数在各区间的不同曲率合理分段,更好地逼近所需曲线。

② 一阶惯性运算模块 LAG_n

该模块对变量有缓冲作用,减缓变化速度,传递函数为

$$\frac{1}{1 + Ts}$$

③ 微分运算模块 LED_n

它是微分增益 K_d 为 1 的实际微分表达式,其传递函数为

$$\frac{T_d s}{1 + T_d s}$$

④ 纯滞后运算模块 DED_n

该模块用于 Smith 补偿控制中,其作用是改善具有纯滞后特征的过程对控制性能的影响,其作用机制在后续内容中将进一步介绍。纯滞后的传递函数为

$$e^{-\tau s}$$

其中 τ 为纯滞后时间。

⑤ 变化率运算模块 VELn 和变化率限幅模块 VLMn

变化率运算模块对过程变量的变化率进行监视,发现系统异常和故障。该模块的输入与输出关系是:输入变量的当前值减去 Δt 之前的值即为变化率

$$y(t) = x(t) - x(t - \Delta t)$$

变化率限幅模块用来限制输出的变化速率,减少因此对过程造成的冲击。如果输入变量呈阶跃式的上下跳变,经过该模块后,输出将按限定的升降速率变化。

⑥ 移动平均运算模块 MAVn

该模块的作用是滤除信号中的干扰,它是将变量的当前值与之前的若干个历史值(最多 19 个)相加后取平均值。

⑦ 状态变化检测模块 CCDn

它用来检测输入到 S_1 寄存器中的信号是否发生了由 0 变为 1,即正跳变。要求其跳变延续时间为一个运算周期。

⑧ 计时模块 TIMn

它用来累计动作或指令执行的时间,多用于顺序控制和批量过程控制。该模块工作时,每个周期先查看寄存器 S_1 的状态,若其中数据为 1,则开始或继续计时;若数据为 0,则对计数器清零,并停止工作。

⑨ 程序设定模块 PGM1

它是一个时间函数发生器,用于有固定时间段不同输出要求的工艺过程,如热处理。程序可分为 10 个时间段,每个时间段按预定的规律输出。

⑩ 脉冲计数模块 PICn

它用来对通、断时间均大于控制周期 20ms 以上的脉冲进行计数。当 S_2 内的数据由 0 变 1 时,计入 1 个脉冲。

⑪ 积算脉冲输出模块 CPOn

它用于对流量等变量的累积,通过 Don 端口向外发宽度为 100ms 的积算脉冲。

(4) 判断模块

判断模块指令共有 14 个,下面具体予以说明。

① 上下限报警模块 HALn 和 LALn

对输入变量的上限值和下限值予以报警。

② 逻辑运算模块 AND、OR、NOT、EOR

分别进行两个量的与、或、非、异或等逻辑运算。

③ 转移指令 GOnn、GIFnn

这里的 GOnn 为无条件转移指令,而 GIFnn 为条件转移指令。后者的含义为,若 $(S_1) = 1$,则转向 nn 步;等于 0,则继续顺序执行。

④ 转子指令 GO SUBnn, GIF SUBnn

GO SUBnn 为无条件转向子程序 nn, GIF SUBnn 为有条件转向子程序,若 $(S_1) = 1$,则转向子程序 nn,若等于 0,则不转移。

⑤ 子程序块 SUBnn 及其返回指令 RTN

SLPC 可以分别编写多至 99 步的主程序和子程序。在 99 步子程序区域中,最多可以分割成 30 个子程序块,每块子程序以 SUBnn 开始,以返回指令 RTN 结束。子程序块可反

复调用。

⑥ 比较指令 CMP

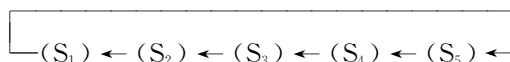
对 S_1 和 S_2 中的内容进行比较,若 $(S_1) > (S_2)$,则置 S_1 为 0,否则,置 1。

⑦ 信号切换模块 SW

相当于一个单刀双掷开关,根据信号进行投掷。运算前,将两个输入信号分别存入 S_2 和 S_3 ,控制切换信号存入 S_1 。工作时,若 $(S_1) = 1$,则 $(S_2) \rightarrow (S_1)$,并向外输出;若 $(S_1) = 0$, $(S_3) \rightarrow (S_1)$,并向外输出。

(5) 运算寄存器位移指令 CHG、ROT

CHG 为交换指令,将运算寄存器 S_1 、 S_2 的内容互换。ROT 为 5 个运算寄存器的循环互换指令,即



(6) 控制模块

SLPC(A 版)有 3 个控制模块,分别为 BSC(基本控制)、CSC(串级控制)和 SSC(选择控制),如图 3-26 所示。用这些模块,可以构成具有不同功能的控制系统。

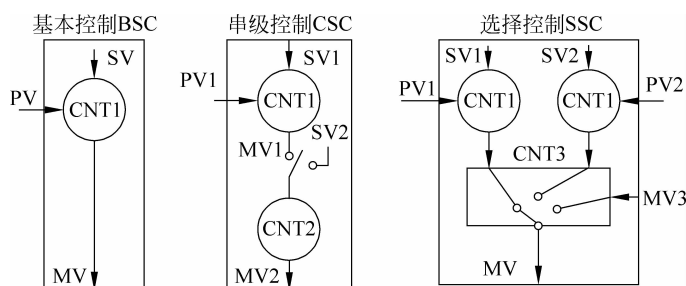


图 3-26 SLPC 的 3 种控制模块

基本控制模块 BSC,内含一个控制单元 CNT1,它相当于模拟仪表中的一个 PID 控制器,能组成各种单回路控制系统;串级控制模块 CSC 内含两个互相串联的控制单元 CNT1 和 CNT2,可构成串级控制系统;选择控制模块 SSC,内含两个并联的控制单元 CNT1、CNT2 和一个选择单元 CNT3(单刀三掷切换开关),可以构成选择控制系统。

图中相关的符号或控制字解释如下: MV 为操作变量(控制量),PV 为被测变量(一般为被控变量),SV 为给定量。CNT1 和 CNT2 均为 PID 控制单元,CNT1=1 时,为连续 PID,CNT1=2 时,为采样 PID,CNT1=3 时为批量 PID,CNT2 为 1 和 2 时,与 CNT1 相同;CNT3 为选择单元,CNT3=0 为选低值,CNT3=1 为选高值。另外,后来的 YS-80(E 型)还增加了控制字 CNT4 和 CNT5,CNT4 为控制周期,CNT4=0 为 0.2s,CNT4=1 为 0.1s,CNT5 为变形 PID 单元,CNT5=0 时为 I-PD,CNT5=1 为 PI-D,CNT5=2 为 SVF(设定值滤波)型 PID。这些控制字在系统设计时,需和编制的程序一道,通过编程器 SPRG 写入用户 ROM。

下面给出编程的例子,以进一步熟悉指令。

例 3-1 对偏差信号编写实现基本的 PID 控制算法的程序。

1. LD X_1 (将 X_1 (数据源)装入(Load)目标寄存器 S_1 。LD 指令目标一定)

2. BSC (基本 PID 运算控制功能)
3. ST Y_1 (将运算结果 S_1 存入(Store)目标寄存器 Y_1 , ST 指令数据源一定)
4. END (用户程序结束标志)

例 3-2 根据下式编写程序

$$y_1 = \frac{x_1(2.5 - 0.3)}{0.9 - x_2} + 1.8$$

1. LD X_1 (读取 X_1)
2. LD $K_1(2.5)$ (读取 K_1)
3. LD $K_2(0.3)$ (读取 K_2)
4. — (减法运算)
5. $\times$ (乘法运算)
6. LD $K_3(0.9)$ (读 K_3)
7. LD X_2 (读取 X_2)
8. — (减法运算)
9. / (除法运算)
10. LD $K_4(1.8)$ (读 K_4)
11. + (加法运算)
12. ST Y_1 (结果输到 Y_1)
13. END (结束)

例 3-3 编写下列实际 PD 算法输出表达式的程序

$$y_1 = K_p \frac{1 + T_d s}{1 + \frac{T_d s}{K_d}} x_1 = K_p \left(\frac{1}{1 + t_d s} x_1 + \frac{T_d}{t_d} \frac{t_d s}{1 + t_d s} x_1 \right)$$

其中, K_p 为比例增益, T_d 为微分时间, K_d 为微分增益, $T_d/K_d = t_d$ 为微分时间常数。

1. LD X_1 (读取输入 X_1)
2. LD P_1 (读取微分时间常数 $t_d \rightarrow P_1$)
3. LAG1 (一阶惯性运算)
4. LD X_1 (读取输入 X_1)
5. LD P_1 (读取微分时间常数 $t_d \rightarrow P_1$)
6. LED1 (微分运算)
7. LD P_2 (读取微分时间 $T_d \rightarrow P_2$)
8. LD P_1 (读取微分时间常数 t_d)
9. $\div$ (T_d/t_d)
10. $\times$ (乘法运算)
11. + (加法运算)
12. LD P_3 (读取比例增益 K_p)
13. $\times$ (乘法运算)
14. ST Y_1 (结果输到 Y_1)
15. END (结束)

以上对 SLPC 的具体情况进行了粗略介绍,如果能够配合做 1~2 个相应的实验,印象和体会将会更深刻,效果也会更好。实践中遇到的数字控制器将是多种多样的,本节叙述的只是其中之一。尽管各公司生产制造的数字控制器相异,学习本节所述的数字控制器却有触类旁通的作用,掌握了这个例子,今后不论遇到什么类型的数字控制器,我们都将有一个好的基础,了解和熟悉起来可以触类旁通,应用起来也容易得心应手。

本章小结

本章对过程控制中最为基本的、使用频率最高的 PID 控制仪表进行了较为全面的讨论,分别涉及基本控制算法、模拟控制器和数字控制器。模拟控制器是基础,可由集成器件来构建,国内主要是 DDZ-III 仪表,它具有原理清晰、使用方便的特点,但是控制器的修改和调整不是很方便,使用的灵活性也不足;数字控制器由编程来实现,具有调整方便、修改灵活和使用范围广等特点,但使用时需增加 A/D 和 D/A 转换器。

在基本控制算法中,P 控制器响应迅速,但有余差,虽然提高比例增益可减小余差,但容易使输出产生波动,甚至出现不稳定;PI 控制器具有比例控制的优点,同时消除了比例控制有差的缺点,因而在实践中被大量采用,一般适用于要求响应快,并且无稳态误差的场合;PID 控制具有 PI 的优点,同时可使超调量得到一定程度的抑制,但它需要调整三个参数,在一定程度上增大了整定工作量和整定难度。

思考题与习题

- 3-1 比例控制有何特点? 为什么会产生稳态误差?
- 3-2 比例积分控制的表达式如何? 它为什么可消除稳态误差? 积分时间大小与消除稳态误差有什么联系?
- 3-3 为什么通常不单独使用微分控制,而是要与比例或比例积分结合使用?
- 3-4 在 PID 控制器中,比例带(度)、积分时间和微分时间分别有什么含义? 当它们增大或减小时,对系统的被控量会产生什么影响?
- 3-5 DDZ-III 中,干扰系数是如何定义的,它的变化是否在 PID 控制中会影响 δ 、 T_i 和 T_d 的实际效果?
- 3-6 DDZ-III 型控制器的硬手动操作与软手动操作有何区别? 分别用于什么情况?
- 3-7 与模拟控制仪表相比,数字控制仪表有哪些显著的优点?
- 3-8 图 3-22 是图 3-21 的具体化,对照两图,找出相对应的部分。