关于 PID

经常有人问有关 PID 的用法，看一些有关单片及应用的书上都有关于 PID 的应用原理，但是面对具体的问题就不知道如何应用了，主要的问题是里面所用到的参数以及计算结果需要进行什么处理，通过什么样的换算才能具体的应用于实际，另外在计算方法上也存在着数值计算的算法问题，今天我在这里例举温度控制中的 PID 部分，希望能够把 PID 的具体应用说明白。

一般书上提供的计算公式中的几个名词：

1. 直接计算法和增量算法，这里的所谓增量算法就是相对于标准算法的相邻两次运算之差，得到的结果是增量，也就是说，在上一次的控制量的基础上需要增加（负值意味着减少）控制量，例如对于可控硅电机调速系统，就是可控硅的触发相位还需要提前（或迟后）的量，对于温度控制就是需要增加（或减少）加热比例，根据具体的应用适当选择采用哪一种算法，但基本的控制方法、原理是完全一样的，直接计算法得到的是当前需要的控制量，相邻两次控制量的差就是增量；
2. 基本偏差 e(t). 表示当前测量值与设定目标间的差，设定目标是被减数，结果可以是正或负，正数表示还没有达到，负数表示已经超过了设定值。这是面向比例项用的变动数据。
3. 累计偏差 $\Sigma e(t) = e(t) + e(t-1) + e(t-2) + \ldots + e(1)$，这是我们每一次测量到的偏差值的总和，这是代数和，考虑到他的正负符号的运算的，这是面向积分项用的一个变动数据。
4. 基本偏差的相对偏差 e(t)-e(t-1)，用本次的基本偏差减去上一次的基本偏差，用于考察当前控制的对象的趋势，作为快速反应的重要依据，这是面向微分项的一个变动数据。
5. 三个基本参数：Kp, Ki, Kd. 这是做好一个控制器的关键常数，分别称为比例常数、积分常数和微分常数，不同的控制对象他们需要选择不同的数值，还需要经过现场调试才能获得较好的效果。
6. 标准的直接计算法公式：

   $Pout(t) = Kp * e(t) + Ki * \Sigma e(t) + Kd * (e(t) - e(t-1));$

   上一次的计算值：

   $Pout(t-1) = Kp * e(t-1) + Ki * \Sigma e(t-1) + Kd * (e(t-1) - e(t-2));$

   两式相减得到增量法计算公式：

   $Pdlt = Kp * (e(t) - e(t-1) + Ki * e(t) + Kd * (e(t) - 2*e(t-1) + e(t-2));$

   *这里我们对 Σ 项的表示应该是对 e(i) 从 1 到 t 全部总和，但为了打字的简便就记作 $\Sigma e(t)$.

三个基本参数 Kp, Ki, Kd.在实际控制中的作用：

比例调节作用：是按比例反应系统的偏差，系统一旦出现了偏差，比例调节立即产生调节作用用以减少偏差。比例作用大，可以加快调节，减少误差，但是过大的比例，使系统的稳定性下降，甚至造成系统的不稳定。

积分调节作用：是使系统消除稳态误差，提高无差度。因为有误差，积分调节就进行，直至无差，积分调节停止，积分调节输出一常值。积分作用的强弱取决与积分时间常数 Ti，Ti 越小，积分作用就越强。反之 Ti 大则积分作用弱，加入积分调节可使系统稳定性下降，动态响应变慢。积分作用常与另两种调节规律结合，组成 PI 调节器或 PID 调节器。

微分调节作用：微分作用反映系统偏差信号的变化率，具有预见性，能预见偏差变化的趋势，因此能产生超前的控制作用，在偏差还没有形成之前，已被微分调节作用消除。因此，可以改善系统的动态性能。在微分时间选择合适情况下，可以减少超调，减少调节时间。微分作用对噪声干扰有放大作用，因此过强的加微分调节，对系统抗干扰不利。此外，微分反应的是变化率，而当输入没有变化时，微分作用输出为零。微分作用不能单独使用，需要与另外两种调节规律相结合，组成 PD 或 PID 控制器。

具体应用中的数值量化处理：

上面只是控制算法的数学方法，似乎有点抽象，在具体的控制项目中怎样对应呢？也就是具体的量化问题。下面举一个在温度控制中的处理方法。

对于加温的温度控制可以采用调节供电电压或在一定的时间循环周期内的供电时间比例来调节加温控制温度，一般以调节加温时间比例比较简单，也是控制上比较常用的方法。调压法控制的原理是通过调节可控硅的触发相位的相位角达到对电压的调节，这个电压是指有效电压，直观上就是对一个正弦波形的前边切掉一块，用不同的切割位置以保留剩余的面积。为了叙述方便，我们还是采用控制时间比例的办法：我们设定一个标准的加温周期，例如 2 分钟，我们就在这个两分钟周期内对输出进行控制，也就是说在这个 2 分钟内加温多少时间，全速加温就是连续整个周期（2 分钟）都加温，当然停止加温就是完全不输出，根据我们的计算可以让加温时间在 0-2 分钟内变化，比如计算所得我们在这一个周期内应该加温 1 分 30 秒，经过两分钟以后再测量被加温物体的温度，通过计算我们应该加温 1 分 28 秒，等等等等，这里除了加温以外的时间就是不加温，等待下一个周期到来，再进行实际测量计算下一周期我们的输出量，周而复始，不断地修正我们的输出量，以达到对温度的有效控制。

为了对应我们的程序处理上的方便，我们在程序内部一般并不是用时分秒来计算的，通常我们会使用系统的一个定时器用于系统全部时钟，例如显示刷新、键盘扫描等，相对于计算来说，我们的控制周期比较长，所以我们可以对 2 分钟进行细分，例如我们用每分钟进行 100 等分，则两分钟就是 200 等分，用于我们的温度控制，这样的输出比例的变化已经足够细了，我们可以有 200 个输出等级了。取 200 的另一个好处是，对应于我们的 8 位单片机刚好可以在一个字节内进行运算，程序简单运算速度快。当需要改变我们的定时周期时，有些不同的加热对象，例如对较大热惯性的加热对象时，可能 2 分钟周期太短了，我们可以通过修改基本定时常数的办法来实现，而保持我们的 200 等分不变。我们对 2 分钟进行 200 等分，算一下他的每一个基本单位的具体时间？

To = 60 * 2 / 200 = 0.6s = 600 ms

这对于单片机来说太长了，因为如果我让我的定时器做到这么慢的定时周期就干不了别的事了，为了显示、键盘等的处理一般我们定时在 5-10ms，所以就需要另外设定一个变量 tTemp1 在每一个定时中断发生时对 tTemp1 计数。例如我的系统定时器的定时常数对应于 10ms,则设定 tTemp1 在达到 60 的时候才确认是达到 600ms 了，才作为一个基本的输出时间单位。对应于总周期的修改，我们的 200 等分可以不用修改，而只要修改我们的变量 tTemp1 的判断边界就可以了，例如对应于 2 分钟

时是60，则在3分钟为周期时边界改为90就行了，定下了我们的基本控制时间分辨率以后，我们的计算就可以不用改变了。当然，根据您的具体对象也可以修改这个等分数，我这里只是作为一个举例：例如200等分。

温控仪离不开测温器件，无论用什么测温器件（传感器），对于控制上来说，首先需要将测到的值换算为温度数据，一般我们国内都采用摄氏度℃，工业上使用的测温器件一般都是非线性的器件，经过放大、A/D转换所得到的电压数据与温度呈非线性关系，存在着微小的差异，一般采用电压值查表的办法获得实际温度，这个表格是以每一个温度点上的电压值来表述的，由于我们的单片机rom的大小限制，这个表格也不可能做得很细，基本上以度作为间隔，也就是说直接查表只能获得度为单位的温度值，而实际测量的温度可能是介于T与T+1度之间，在PID控制计算上，这样的分辨率是不够的，所以我们还需要进一步获得具体的温度精确数据，一般采用将T与T+1之间的电压差和AD实际值(mv)进行定比分点的办法(更精确的是采用二次插值计算法)获得温度的精确数值，也就是获得小数部分。如果能够做到1/10度的温度分辨率精度就可以基本满足控制运算要求了，所以我们可以用定点数的办法处理。不采用浮点数是因为单片机的运算速度不适合用浮点数，定点数处理，就是将温度的内部运算单位放大10倍，在用于显示的时候再除以10也就是固定显示一个小数点位置。如果想让我们的控制做得更好，还可以再提高温度的内部精度，例如精确到1/100度，这也是现在高级温控仪常采用的精度，但在通常情况下这个精度似乎有点过剩。

PID的三个基本参数Kp、Ki、Kd，一般由试验确定，根据我们的实际工作对象去初步确定，然后在实际运行过程中进行调节，以达到相对理想的效果，为了达到比较好的控制效果，这三个参数一般不采用整数，但同时为了减轻单片机的运算量，通常采用2的整倍数放大的办法确定这些参数，在运算结果中再除以2的整倍数，因为单片机运算中可以用移位来完成，速度比较快，常用的是8倍或16倍放大，注意这三个参数采用相同的放大比例。编程的过程中自己从头到尾要清楚我的参数是经过放大了的，就不会忘记对运算结果还原。

通过怎样的运算来获得0-200加温比例数据呢？很简单，为了说明这个问题，我们先假定只考虑最简单的比例控制算法，假定我们的控制范围是在200度，则设定温度与实际温度的差的最大值就是200（度），我们就用他去输出，这时的参数Kp=1,当我们为了提高加热速度，而使受控制的区域缩小，例如只控制50度范围，如目标温度设定为230，我们控制的范围就在180-230范围内，这时的差值不够200，我们就把计算得到的数字乘以4就得到0-200的数据了，假定当前实测温度为222，则230 - 222 = 8再乘以4算得32，这就作为我们的输出比例数据。当然我们这里还没有考虑超温的情况，计算产生了负数。这一部分将作为我们控制输出的基本量，上面的计算是对应于Kp=4的，再加入微分和积分项，这时我们的Kp可以基本保持不变。Kp在这里基本确立了我们的起始控制点到目标值之间的控制范围，微分和积分项在这里只是作为附加部分，基本不影响控制范围。当温度突然下降一度时，我们希望补上多少比例区进行下一轮的加热呢？这就是微分系数；早我经过这么久的控制，目标温度还是低了一点，我希望用多少的比例去弥补这个长期欠温呢？这就是积分常数，我想你已经大概已经确定了这些数据了。一句话，比例常数决定我们参与在目标点以前真

正控制的范围，Kp = 基本时间总周期/控制范围。Ki,Kd 是您希望的反应对策速度，看你有没有耐心去逐步达到稳定点，过激了会起反作用的，过于胆小怕事是达不到预期目标的。

最后，在计算结果交付于输出之前，还需要进行一些修正，例如当计算结果大于 200 时按 200 输出，计算结果小于零时按零输出。

处理上的一些常用方法

为了提高测温的准确性，往往需要进行多次测温，然后剔除测得结果中的最大值和最小值，把剩余的数值相加在计算平均值，这样比较有利于抗干扰，或者由于其他原因引起的测量值波动，初期的处理是很必要的。测温工作和 PID 计算一般安排在上一个输出周期的最后阶段就提前进行，的最后阶段就提前进行，测温次数一般在 7-13 次，去掉最大最下数以后保留 5-11 次用于运算，均值运算的累加部分将安排在每一次测温以后逐步进行，不会过多占用系统时间。只要系统的时间允许，尽可能的测多几次，所得到的结果也会相对精确点。

运算将会占用很长的时间，这是相对于我们的一个基本定时周期来讲的，一个系统定时周期内我们的系统还要处理很多事情，例如显示的刷新、键盘的扫描、键盘码的处理执行等等，如果我们的运算占用很多时间的话那么就会出现一个系统定时周期内完不成的情况。所以如果您的系统仅仅是一台温控仪的话，就可以将运算部分放到主程序中进行，而常规的刷新扫描之类的才放在定时中断里处理，这样编制的主程序就非常清晰明了，这是比较理想的情况。如果温度控制仅仅作为您的系统中的一部分小插曲，那就需要认真考虑程序的布局了，主程序有更重要的任务要处理，温度控制运算部分就只能委托系统定时器来承担了，时间不够用怎么办？假定我们的定时周期是 10ms，显示刷新部分要用 1ms,键盘扫描处理要 1ms,PID 运算要 2ms（假定），留给主程序的时间可能不够了，我们就需要对我们的 PID 运算进行任务的分割，把运算分成几个部分进行，每一次进入运算程序只计算其中的一小部分，经过 N 次的调用才完成一个完整的 PID 运算，这样就不会影响整个系统的运作了，这里面需要细细的推敲、分割。这是面对一个较大系统的一般处理方法，对其他任务也可以采用分割的办法进行细化，例如对显示刷新，现在液晶屏用得比较多，环境的干扰，数据的变化都会引起花屏或反应迟钝，而刷新需要占用很长的时间，我们也可以采用分片刷新的办法处理，把每次刷新所占用的时间减到最短。

PID 的输出部分可以放到系统定时中断里处理，这部分占用的时间不长，可以附带完成。

前面我们讲到将时间作 200 等分记作 Tset，这就是将一个约定时间作 200 级不同的输出时间比例，通过我们的计算得到一个介于 0-200 之间的数 Tout，然后每次在我们的 PID 基本定时时间中对这个 Tout 减 1，如果 Tout 不为零则输出 1，否则输出 0，每次在我们的 PID 基本定时时间中对 Tset 减 1，当等分计数器 Tset 也达到零时进行下一轮控制循环，重新计算 Tout.这样我们就完成了一个控制比例 Tout/200，这个 Tout 是计算获得的原始数据。

运算中往往会出现数据溢出的情况，所以一般在运算中都要求对数据强制转换成 int 型，注意需要考虑符号，另外，对有些参数如积分项的累加数过大时会起反作用，使调节失灵，或者引起大幅震荡，为此我们对这一项引入一个数值最大界限，当结果超出约定界限时，不再增加（或减少）。

加温的整个过程没有必要全程 PID 控制，一般可以在设定目标值前一个温度区域才进行 PID 控制，例如，设定目标温度为 300 度，则我们可以在 250 度以前全速加温，当达到 250 以后才开始计算 PID 并予以控制，这样可以加快加温速度又不影响温度控制。在不产生过大的过冲的情况下，尽可能把起控点抬高，有利于后面控制部分的进一步细化。在进入控制之前我们的积分项记录数据为零。

对于用调压法控制输出时，由于正弦波相对于延时导通的相位角输出的电压有效值是非线性的，而且三角函数计算也很费时，所以建议用查表法处理，同样可以采用查表加插值获得移相数据，其他的计算方法和用继电器比例输出法是一样的。硬件上还需要做一个比较精确的过零负脉冲电路，让脉冲发生于过零点以前，这样我们就可以在过零点以前先关闭输出，脉冲宽度越窄越好，至少这个脉冲宽度不能大于 1ms.过零脉冲的识别也需要一点小小的处理，防止假脉冲混入，防止电网干扰，这是软件上的问题。

参数的设定与调整

这是 PID 最困难的部分，编程时只设定他们的大概数值，然后通过反复的调试才能找到相对比较理想的参数值。面向不同的控制对象参数都不同，所以我们无法提供参考数值，但是我们可以根据这些参数在整个 PID 过程中的作用原理，来讨论我们的对策。

1. 加温很迅速就达到目标值，但是温度过冲很大：
   a) 比例系数太大，致使在未达到设定温度前加温比例过高；
   b) 微分系数过小，致使对对象反应不敏感；
2. 加温经常达不到目标值，小于目标值的时间较多：
   a) 比例系数过小，加温比例不够；
   b) 积分系数过小，对恒偏差补偿不足；
3. 基本上能够在控制目标上，但上下偏差偏大，经常波动：
   a) 微分系数过小，对即时变化反应不够快，反映措施不力；
   b) 积分系数过大，使微分反应被淹没钝化；
   c) 设定的基本定时周期过短，加热没有来得及传到测温点；
4. 受工作环境影响较大，在稍有变动时就会引起温度的波动：
   a) 微分系数过小，对即时变化反应不够快，不能及时反映；
   b) 设定的基本定时周期过长，不能及时得到修正；

选择一个合适的时间常数很重要，要根据我们的输出单元采用什么器件来确定，如果是采用可控硅的，则可设定时间常数的范围就很自由，如果采用继电器的则过于频繁的开关会影响继电器的使用寿命，所以就不太适合采用较短周期。一般的周期设定范围为 1-10 分钟较为合适。

为了调试方便，起码在调试阶段您必须编制一个可以对参数进行随时修改和记忆的接口，否则你会很辛苦，老是在现场与办公室之间来回跑。

关于自整定问题：

在通用仪表行业用得比较多，因为他们的工作对象是不确定的，而不同的对象所使用的参数是千变万化的，所以无法为用户设定参数。这就引入了自整定的概念，用户在首次使用的时候，启动自整定功能专门为新的工作对象寻找一套参数，并把他们记忆下来，作为今后工作的依据。其实自整定也就是作 N 次的测定，根据实际变化反应反复对参数进行修正，当达到一定要求后宣告自整定完成。这是一个多元素的优化问题，可以采用很多种优化方法，例如 0.618 黄金分割法就是一种很不错的算法，可以参考有关优化算法的书籍，我们这里就不做讨论了，需要说明的是，我们是对全部参数进行同时优化的，根据出现的不同情况去修正不同参数，我们的修正量将逐步缩小，当我们的修正量达到一定的范围以后，或者实测温度变化波动已经达到一定范围以后就可以认为自整定过程结束，记录最后结果返回正常控制状态。由于优化过程需要进行大量的试探，所以一次自整定过程往往需要花费很长时间，一般在半小时到两个小时左右。

修正的对策基本上就是上一节我们讨论几种，对具体每一项参数的修正量，我们还可以赋以不同的比例参数，根据您认为的影响重要程度来确定，在 0-1 之间选取。

编后话

控制的方法不是只有 PID 一种，在很多场合他也未必就是最佳的控制算法，随着大量的应用研究，有很多种新的理念新的方法可以完全替代它，但是 PID 作为一种基本的控制算法已经比较成熟，基于 PID 为基础的变种算法或者叫做加强算法也派生了很多，他们的控制效果往往会比 PID 更有效，最常见的是对这里的 3 个参数以控制目标为基准，参数相应成比例的变化，PID 上的 PID。面向每一个的工作对象，您可以摸索出一种最适合您的对象的控制方法，不拘一格，开创自己的路。

本人对 PID 的认识还很肤浅，具体作的不多，难免会对有些地方叙述有错误或不恰当，敬请您批评指正，我非常乐意接受您的指正，我更希望能够修改这篇文稿，对我们大量从事工控的同行能有一点参考价值，这就是我写这篇文章的愿望。

晓 奇

http://www.xiao-qi.com/

info@xiao-qi.com