

单片机系统中的率表算法

清华 Motorola 单片机应用开发研究中心

邵贝贝

一、问题的提出

近年来,国内许多单位用 Motorola 68HC05C8A、68HC05C9A、68HC05L5、68HC05L16 等单片机开发复费率电表。电力部门也在为开发中的复费率电表制定一些规范。复费率电表中有一项功能要求,能给出所谓“最大需量”。这里最大需量的概念是:过去任何一段 t 时间内的平均功率最大值。 t 的值由电力部门设定。 t 值的典型值为 5 分钟、10 分钟、15 分钟等,直至一小时。由于设备的启动电流往往很大,以电机为例,启动电流通常能达到正常运行时的六、七倍。一台 1 千瓦的电机,满负荷运行时需量就是 1 千瓦。显然不能将启动时的瞬时功率算入最大需量。单片机系统往往以记录脉冲数来累计用电量,以脉冲计数率折算瞬时功率和平均功率,从原理上讲,这和以 A/D 变换采样瞬态电流乘以电压值输入没有什么不同。都是求的曲线下的面积。以瞬时功率为纵坐标,时间为横坐标。对某一时间段求平均值,即瞬时功率曲线下的面积。也可以把这个时间段看成一个“窗口”,通过这个窗口找到整个记录期内该窗口下包含面积的最大值,这个最大值就是最大需量,而窗口的大小是可调节的,电力部门中有些技术人员称之为滑差算法。并要求在复费率电表中采用滑差算法。

二、滑差算法

滑差算法指在复费率电表的单片机系统内存中,开一个循环缓冲区,缓冲区的大小即“窗口”的大小,可以用命令设置。例如,循环缓冲区内每个变量元素是每一分钟电表测到的用电量,窗口的大小从一分钟到六十分钟可调。单片机每到一分钟舍弃缓冲区内最早那一分钟的值,填入最近一分钟的值,然后将整个缓冲区累加,再以此方法求出该窗口下的平均功率。和以前系统中保存的最大值比较,如大于以前的值,就认为是最大需量,并保存下来。满足电力部门定义的“窗口”宽度可调的所谓最大需量的算法。

这种所谓滑差算法,在当前单片机如此功能强大且如此便宜的时代,无疑是可以采用的。但这种算法存在以下几个缺点:

1、内存消耗量大

由于单片机系统内 RAM 的量很有限,一般只有 176 字节至 352 字节。以上面提到的例子,若记录每分钟用电量用两个字节的的话,最大“窗口”宽度取一小时,缓冲区至少需要 120 个字节。这几乎占掉了整个系统 RAM 的一大部分。增加内存无疑将增加成本。

2、不能连续显示当前需量

由于滑差算法中单片机每累计一分钟才运算一次当前窗口下的平均功率,如将该值显示出来,这个值每分钟跳变一次,给人的感觉是,整个系统响应太慢,每分钟才能测到一个值。单片机系统一般都采用数字显示,LCD 也好 LED 也好,合理的显示刷新周期一般认为可接受的范围是两秒至六秒。如果系统显示每分钟刷新一次,将被认为慢得不可接受。

由于滑差算法存在以上严重不足。我们建议以率表算法代替滑差算法。虽然我们并不打算将我们的算法强加于人,各开发厂商都可以使用自己的算法,但我们希望电力部门不要把滑差算法强加在复费率电表规范中。那将会增加千百万只电表的成本,造成不必要的浪费。

三、率表算法

率表算法是我们多年来采用单片机制作测量仪表时,将离散量变成连续量时采用的算法。这种算法的优点是占用内存少,运算量小,可连续显示,时间常数(相当于滑差算法中的窗口)可动态任意调节。这种算法在微机技术一出现,智能化仪表用来代替模拟量输入的指针仪表时就出现了。

在模拟量输入的仪表中,为了减小指针的晃动,使指针稳定地指向一个易读取的值,为

仪表增加了阻尼。在数字式仪表中，为模拟指针式仪表的阻尼，给数字式仪表增加了所谓“伪时间常数”。伪时间常数中的伪，是指这个时间常数是算出来的。作为测量仪表，单片机采样频率实际上可以做得很高。例如 A/D 变换的采样频率，几十微秒，一百微秒采样一次并不困难。对于脉冲计数输入，计算两个脉冲间的时间间隔也不困难。而数字显示的率表，刷新周期不应太快，否则人眼很难跟得上。在率表算法上人为地在单片机中加入了伪时间常数，也叫数字滤波。这就是率表算法。用率表算法，每两秒至六秒运算并显示出一次当前功率值，并将该值与过去发生过的最大值不停地比较，就可以找出与滑差算法相同的最大需量值。伪时间常数是人为设定并可以动态修改的，调整时间常数相当于调整滑差算法的窗口。

我们用以下公式得出率表值：

$$\left(\text{当前瞬时值}\right) \frac{1}{n} + \left(\text{过去平均值}\right) \left(1 - \frac{1}{n}\right) \dots\dots\dots(1)$$

对于采用计数方式的复费率电表，当前瞬时值可以认为是以最近一个采样周期内采到的值计算出的功率值。例如，如采样周期为 6 秒，在刚刚过去的一个 6 秒钟内电表测到用电量，将这个值乘以 10 得到一分钟值，再乘以 60 就是瞬态功率。用 6 秒钟的值乘以 600，得到以每小时计的功率，这个瞬态功率并不能代表电力部门所谓的当前需量。将瞬态功率乘以 1/n，(n 为大于 1 的整数)，这个 1/n 可以叫做置信度或权重因子。也就是说，当前平均功率由两部分组成。由最近一个 6 秒钟测量到的值算出来的瞬态值只是平均值的一部分。它的权重因子是 1/n，而 6 秒钟前系统的真实平均值占有的权重是(1 - 1/n)。

对于 n 取不同的值，可以得出不同的伪时间常数。以下我们以系统输入由一个稳态，阶跃到另一个稳态，率表算法是如何跟踪这种变化的来说明以上公式的合理性。

当系统由稳态 1 阶跃地降至稳态零时，系统每个采样 t 周期内采到零，可是由于过去的平均功率为 1，M 个采样周期后，率表算法得出的值符合以下算法：

$$\text{当前平均功率} = \left(1 - \frac{1}{n}\right)^M$$

这是一条指数衰减曲线，当 M 足够大时，达到稳态零。我们将率表衰减 63%的时间或在由零上升到 1 过程中增长到 63%时所用的时间称之为系统的时间常数。3 倍时间常数以后，系统达到稳定值的 95%以上，5 倍时间常数以后达到稳定值 99%以上。

不难看出，当阶跃变化的系统由稳态零上升到稳态 1，用率表算法得出的值是一条指数上升曲线并符合下式：

$$\text{当前平均功率} = 1 - \left(1 - \frac{1}{n}\right)^M$$

总之，率表算法以指数规律跟踪系统瞬时值的变化，靠调整 n 值达到不同“窗口宽度”的滑差算法，这种算法的优点是几乎没有什么缓冲区。节约了大量的宝贵的内存资源。显示结果是连续的。

图 1 给出当 n 取不同值时，对阶跃变化的输入值用公式(1)算出的 n 与 M 的关系。

四、应用实例

我们在以 Motorola 68HC05C8A 以及 68HC05L5 制作的复费率代电表中采用了率表算法。以下给出一个典型的运算实例。

我们以 6 秒为脉冲累积周期，取 n=16，不难算出，此时时间常数是 93 秒，当阶跃变化由零变为功率 1 以后，到第 15 个 6 秒以后显示由零升至 0.62，第 16 个 6 秒以后为 0.64。故可以认为系统时间常数为 93 秒。三个 93 秒以后达到 95%以上，5 个 93 秒即 7.75 分钟以后达到 99%以上。由于输入由零阶跃为 1 再由 1 阶跃零为 0，率表曲线下面积不变，不会由于算法丢失数据或损失精度。

n 值实际上是可以随时任意输入的，但如果 n 值只取 2、4、8、16、32 等 2 的整倍数。率表算法中的乘除法可以用移位指令进行，15/16 的算法也可以用 $1 - 1/16$ 的算法，即移位，然后做减法的方法进行。这样将使程序大大减化，而 CPU 的运算量也减至最小。而当前平均功率又可以得到实时的显示。这种算法比所谓滑差算法有明显的优越性。

在开发复费率电表的选型上，如果采用率表算法，68HC05C8A 就足够了。而采用滑差算法则需要使用 68HC05C9A。前者 RAM 有 176 字节，后者为 352 字节。68HC05C9A 要比 68HC05C8A 贵 1 美元。对于批量大的产品，算法带来的经济效益是可观的。

五、结论

本文建议在复费率电表中算平均需量，进而找出最大需量时以率表算法代替滑差算法。这样做不损失精度，节省大量有限的内存，又可以连续显示当前率。应引起复费率电表设计厂家注意。尤其编写电力部门相应规范的人员，不应只把滑差算法写入规范，造成不必要的成本增加与浪费。建议电力部门有关人员研究一下率表算法，根据实际情况给出优 n 值、t 值以及优选的时间常数，规范有关算法，对复费率电表的开发商进行技术的宏观指导。

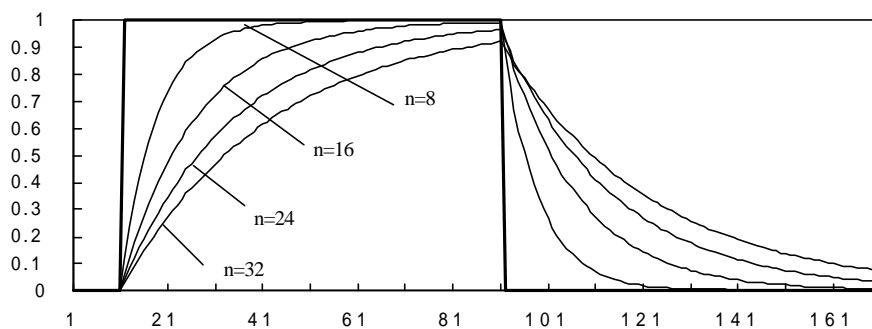


图 1.