

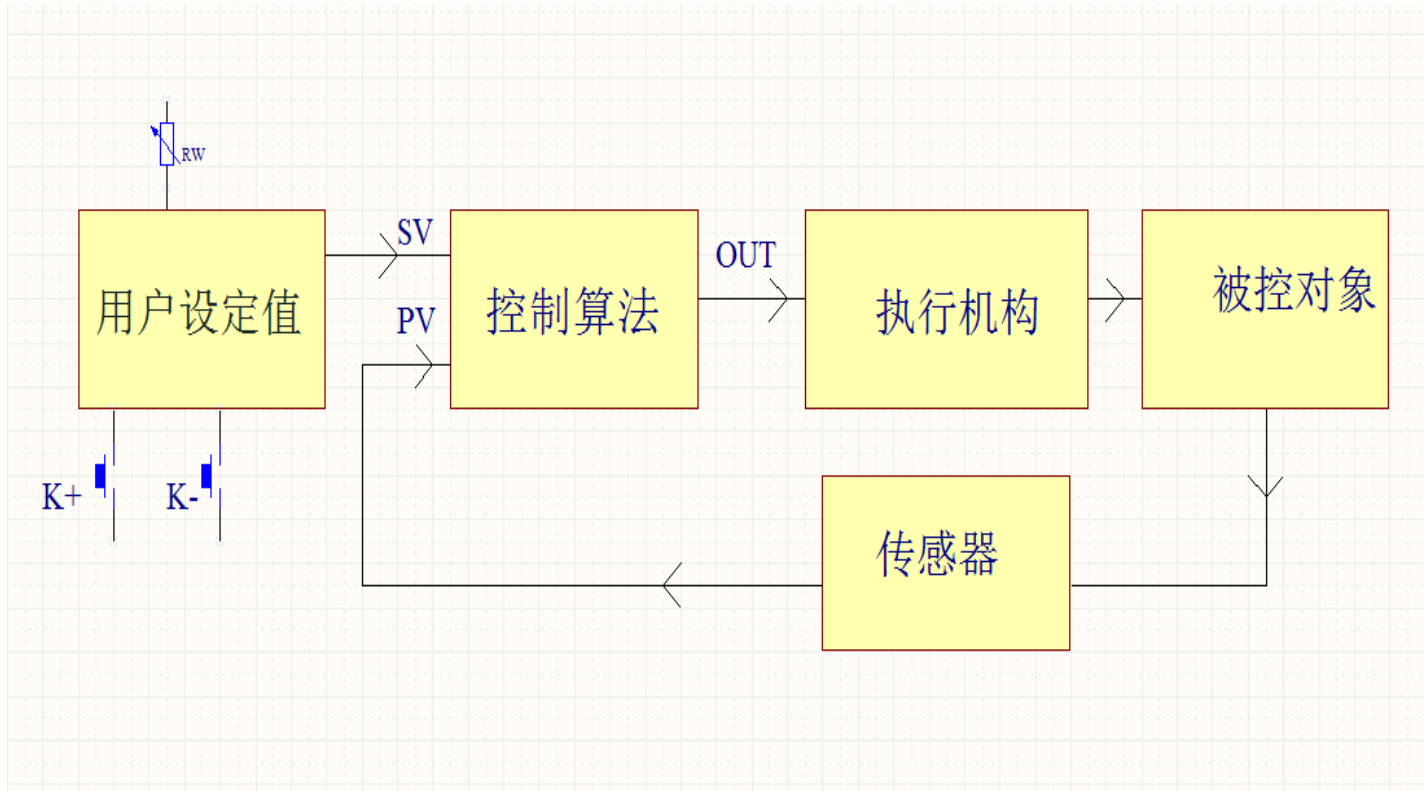
PID 控制专题

主要内容：

- 1.常用的控制算法与 PID 控制算法的异同点；
- 2.PID 控制算法的理论分析
- 3.基于单片机的 PID 算法实现
- 4.PID 算法的工程应用的一些注意事项
- 5.演示板电路分析
- 6.PID 算法 C 语言实现---基于 ARM-CortexM3(STM32)的增量式 PID 温度控制

一、常用的控制算法：

1. 控制系统的基本结构：



控制目的：

控制的根本目的就是要使控制对象当前的状态值与用户的设定值相同（最大限度的接近）。

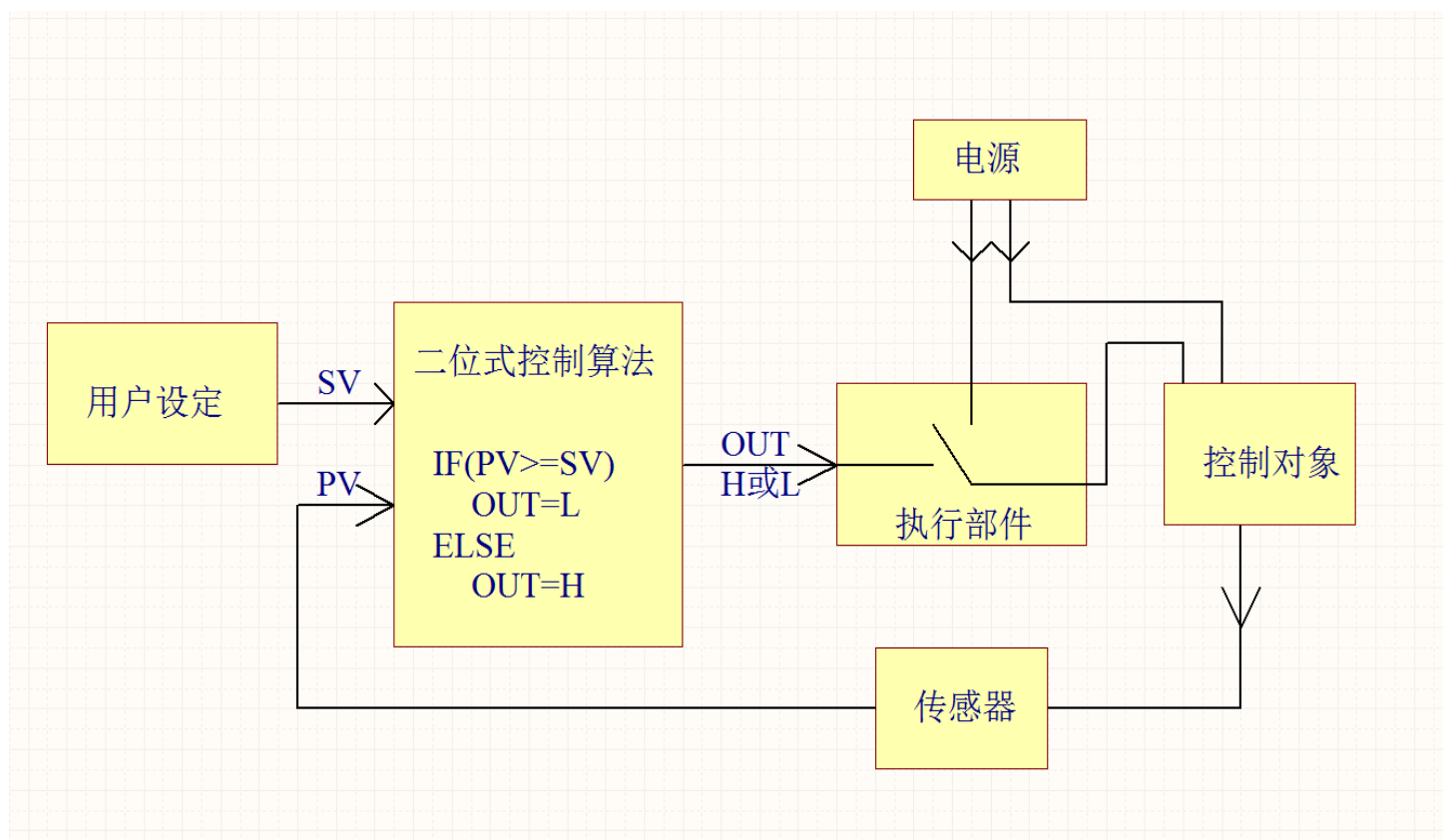
基本思想：

用户设定值 **SV** 与被控制对象当前的值 **PV** 两者同时送入由特定硬件电路模型或特定的软件算法组成的控制算法逻辑中，利用不同的控制算法对 **SV** 和 **PV** 进行分析、判断、处理，从而产生当前应该输出的控制信号 **OUT**，控制信号经过执行机构施加到控制对象

上，从而产生预期的控制效果。

2. 常用控制算法：---位式控制

1). 二位式控制算法



特点：

a. 二位式控制算法输出的控制量只有高低 2 种状态。

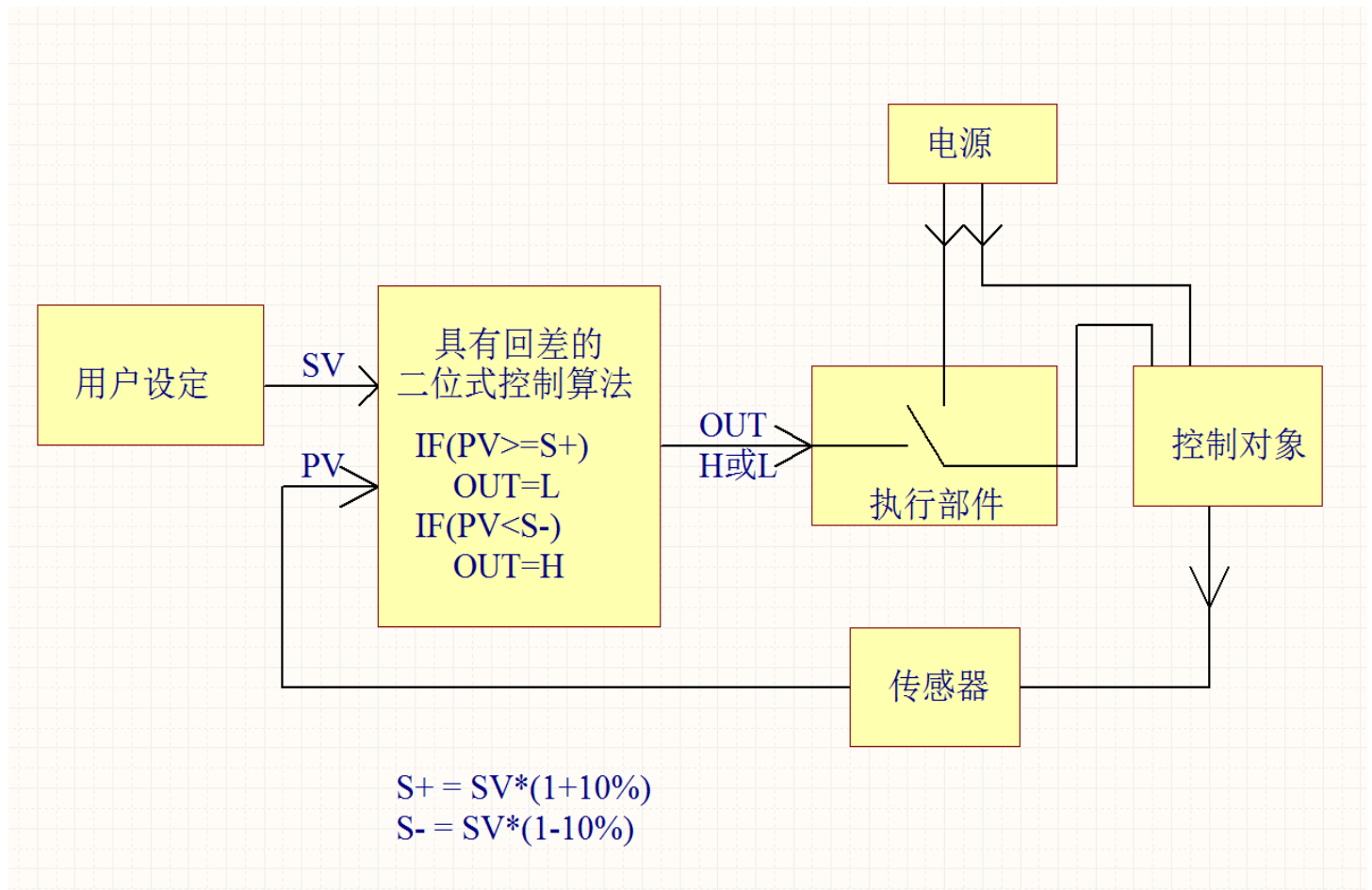
b. 执行机构使控制对象要不全额工作，要不就停止工作。当 PV 低于 SV 时全额工作， PV 大于或等于 SV 时就彻底停止工作。如果控制对象是一个 $1000W$ 的加热器，温度不到时就 $1000W$ 全功率运行，温度达到时就停止工作。

c. 由于环境因素或控制系统传输延时或者控制对

象本身的惯性等因素，控制效果往往是 PV 在 SV 的上下有一个较大的波动。

d.在 PV 接近 SV 的临界点时，控制输出信号 OUT 往往在 H 和 L 之间频繁转换，导致执行部件的触点频繁开关动作，易产生干扰及缩短执行部件的寿命。

2).具有回差的二位式控制算法



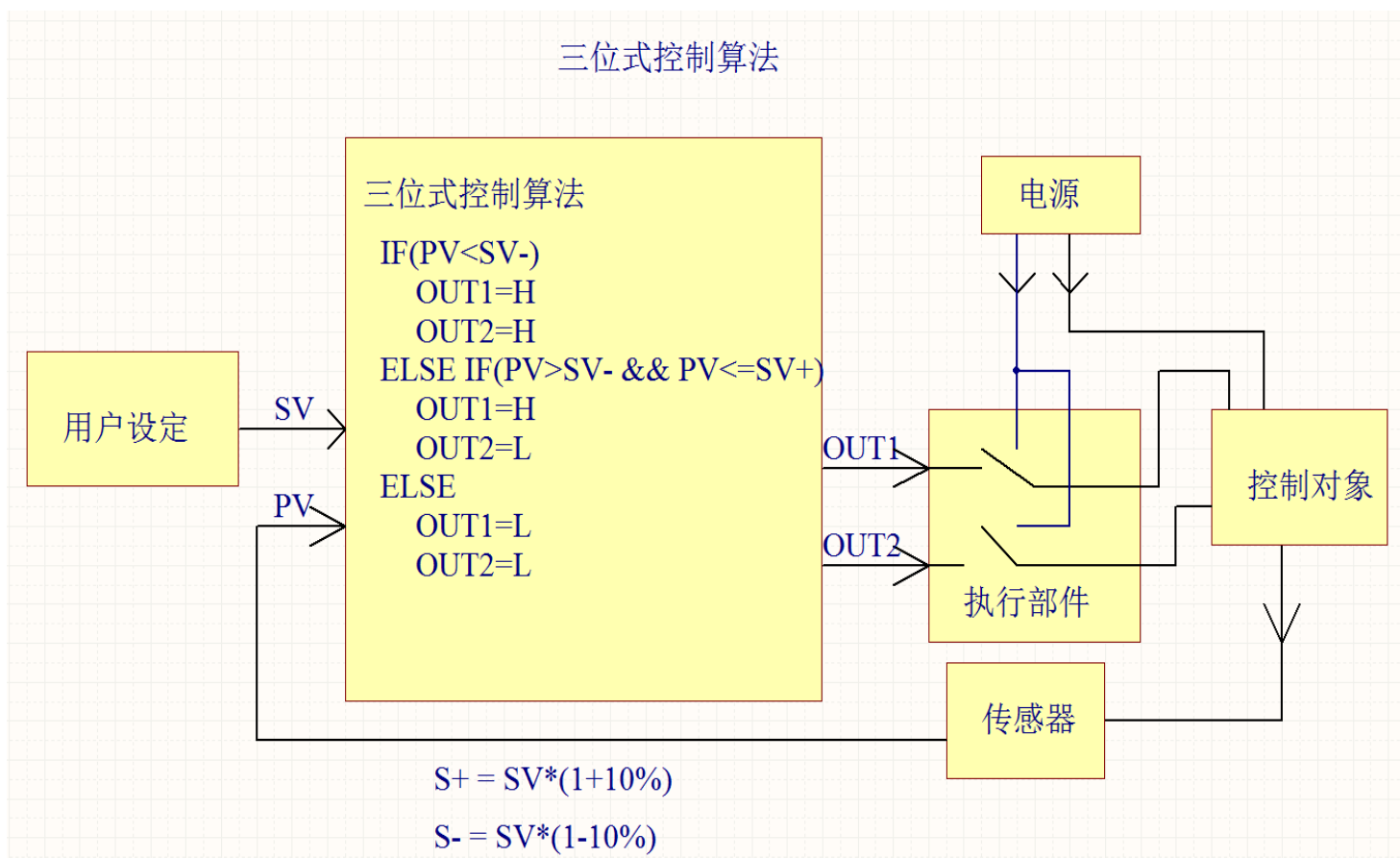
特点：

- a. 取 SV 的正负 10%左右作为回差调节上下限，高于上限才开始输出 L,低于下限才开始输出 H;
- b.避免了一般二位式控制算法在临界点时执行部

件频繁动作。

c.因为控制对象只有全额运行或不运行两种状态，仍然存在一般二位式控制算法的缺点：PV 总是在 SV 附近波动。

3).三位式控制算法



特点：在二位式控制的基础上对控制对象的功率分成 0 功率（停止工作）、半功率、全功率三种情况（即三位）。

当前值低于设定值一定比例（一般 10%）时 OUT1 和 OUT2 同时起控制作用，控制对象全功率运行；

当前值在设定值的正负 10%范围时，OUT1 单独

作用，工作于半功率状态；

当前值达到或超过设定值时 OUT1 和 OUT2 都停止输出，控制对象停止工作。

相对一般二位式控制算法，三位式算法对控制对象的当前状态值做了简单的分析，并根据不同的当前状态值输出不同的控制信号。能够较好的对输出产生控制效果。

小结：位式控制的主要特征：

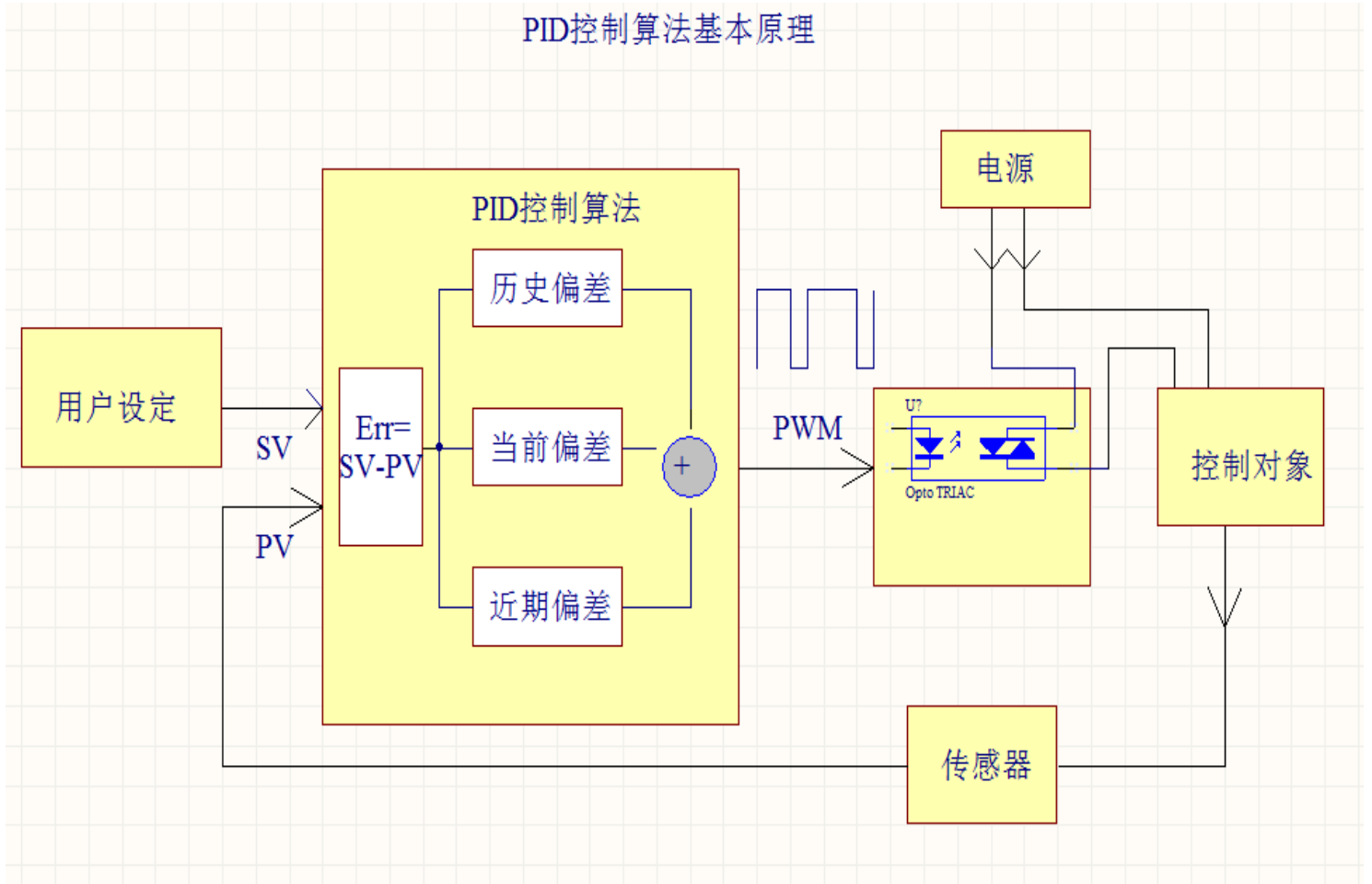
1.控制算法只关注控制当前的状态值（PV）与设定值之间的差值---二者当前有差值就输出控制信号，二者当前无差值就不输出控制信号。

2.位式控制算法的输出信号状态单一，只输出了高低两种状态，这两种状态对应着控制对象的工作与不工作----如果是温度控制系统，就是加热器加热与不加热。

3.由于控制系统自身的延时或者控制对象自身的惯性，位式控制算法只能使控制对象当前的状态值在设定值附件波动，不能很好的跟踪在设定值的附近甚至相等。

二、PID 控制算法

1.PID 控制算法的基本思想



PID 算法是一种具有预见性的控制算法，其核心思想是

1>. PID 算法不但考虑控制对象的当前状态值（现在状态），而且还考虑控制对象过去一段时间的状态值（历史状态）和最近一段时间的状态值变化（预期），由这 3 方面共同决定当前的输出控制信号；

2>.PID 控制算法的运算结果是一个数，利用这个数来控制被控对象在多种工作状态（比如加热器的多种功率，阀门的多种开度等）工作，一般输出形式为 PWM，基本上

满足了按需输出控制信号,根据情况随时改变输出的目的。

2.PID 算法分析:

设某控制系统: 用户设定值为 SV (也就是希望通过 PID 控制算法使被控制对象的状态值保持在 SV 的附件)。

1>从系统投入运行开始,控制算法每隔一段时间对被控制对象的状态值进行采样。由此,可得到开机以来由各个采样时间点被控制对象的状态值所形成的数据序列:

$$X_1, X_2, X_3, X_4, \dots, X_{k-2}, X_{k-1}, X_k$$

说明:

X_1 : 开机以来的第一次采样值

X_k : 目前的采样值 (最近一次的采样值)

2> 从这个采样值序列中提取出三方面信息:

①当前采样值 X_k 与用户设定值 SV 之间的差值: E_k

$$E_k = S_v - X_k$$

分析 E_k :

E_k {
 >0:说明当前状态值未达标
 =0: 说明当前控制状态值正好满足要求
 <0: 说明当前状态值已经超标

结论: E_k 反应了控制对象当前值与设定值的偏差程度, 可以根据 E_k 的大小对输出信号 OUT 进行调整: 偏差程度大 OUT 增大, 偏差程度小 OUT 减小。即输出信号的强弱与当前偏差程度的大小成比例, 所以根据 E_k 的大小来给出控制信号 OUT 的当前值的算法称为比例控制 (Proportion)。用数学模型可以表示为:

$$POUT = (K_p * E_k) + Out_0$$

K_p :一般称之为比例系数, 可以理解为硬件上的放大器 (或衰减器), 适当选取 K_p 将当前误差值 E_k 按一定的增益放大或缩小, 以提高控制算法的相应速度。

Out_0 :是一个常数, 目的是为了当 E_k 为 0 时, 确保输出信号不为 0, 以不至于在当前值与设定值相等时控制器输出信号 OUT 为 0, 系统处于无

控制信号的失控状态。

②将投入运行以来的各个采样值都与设定值相减，可得到开机以来每个采样时刻的偏差序列数据：

$E_1, E_2, E_3 \dots E_{k-2}, E_{k-1}, E_k$

说明：

E_1 ：开机的第一个采样点与设定值的偏差

$$E_1 = SV - X_1;$$

$$E_2 = SV - X_2;$$

.....

$$E_{k-2} = SV - X_{k-2};$$

$$E_{k-1} = SV - X_{k-1};$$

E_k ：当前的采样值与设定值的偏差

$$E_k = SV - X_k$$

分析开机以来的误差序列：

每个偏差值可能有： >0 ， <0 ， $=0$ 这三种可能的

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/747110012035006056>