增量式 PID 控制 C 语言代码

增量式 PID 控制公式：

$$\Delta u(k) = k_p(e(k) - e(k-1)) + k_i e(k) + k_d(e(k) - 2e(k-1) + e(k-2))$$

上面△u(k)是控制量增量，“增量式 PID”就是直接以这个增量进行控制。

至于参数的整定，根据响应的情况调，比如，响应慢了，我就增大 kp，或者减小 kd，超调大了，就减小 kp 或增大点 kd，这个规律你可以看看 PID 三个参数的作用：）

```
//////////////////////////////////////////////////////
// 定义 PID 参数结构体
//////////////////////////////////////////////////////
typedef struct PID {                    //结构体定义
        int   SetPoint                  //设定值
        int   Proportion;               // Proportion 比例系数
        int   Integral;                 // Integral   积分系数
        int   Derivative;               // Derivative  微分系数
        int   LastError;                // Error[-1]  前一拍误差
        int   PreError;                 // Error[-2]  前两拍误差
} PID;

main()
{
  PID vPID;                             //定义结构变量名
  PIDInit ( &vPID );                    //Initialize Structure
  vPID.Proportion = 10;                 //Set PID Coefficients
  vPID.Integral   = 10;                 // Set PID Integral
  vPID.Derivative = 10;                 // Set PID Derivative
  vPID. SetPoint =                      //根据实际情况设定

  while(1)
  {
      Verror=Measure();                 //得到 AD 的输出值
      Error =vPID. SetPoint- Verror;    //与设定值比较，得到误差值
      tempi=PIDCal(&vPID, Error;
```

```
        laser.Value+=tempi;              // Value 与 Num[2]为共同体，共同体名
laser
        LASERH=laser.Num[0];
        LASERL=laser.Num[1];
  }
}

//////////////////////////////////////////////////////////////////
//Title:PID 参数初始化
//Description:  Proportion="0"
//                Integral=0
//                LastError=0
//Input:  PID 的 P、I 控制常数和之前的误差量（PID  *pp）
//Return:
//////////////////////////////////////////////////////////////////
void  PIDInit  (PID  *pp)                              //PID 参数初始化，都置 0
   {
     memset  (  pp,0,sizeof(PID));

//memset()的函数，  它可以一字节一字节地把整个数组设置为一个指定的值。
//  memset()函数在 mem.h 头文件中声明，它把数组的起始地址作为其第一个参
数，
//第二个参数是设置数组每个字节的值，第三个参数是数组的长度(字节数，不是
元素个数)。
//其函数原型为：    void  *memset(void*，int，unsigned)；
//头文件<string.h>
   }

//////////////////////////////////////////////////////////////////
//Title:增量式 PID 算法程序
//Description:给出一个误差增量
//Input:  PID 的 P、I 控制常数和之前的误差量（PID  *pp）&  当前误差量（This
Error）
//Return:  误差增量 templ
//////////////////////////////////////////////////////////////////
int  PIDCal(  PID  *pp,  int  ThisError  ){
 //增量式 PID 算法（需要控制的不是控制量的绝对值，而是控制量的增量）
     int  pError,dError,iError;
```

```
    long  templ;
    pError  =  ThisError-pp->LastError;
    iError  =  ThisError;
    dError  =  ThisError-2*(pp->LastError)+pp->PreError;

    //增量计算
    templ=pp->Proportion*pError  +  pp->Integral*iError+pp->Derivative*dErro
r;   //增量

    //存储误差用于下次运算
    pp->PreError    =  pp->LastError;
    pp->LastError  =  ThisError;

    return  ((int)(templ>>8));
}
```

参考自：http://blog.sina.com.cn/s/blog_408540af0100asu3.html