

单片机数字滤波算法研究

牛余朋, 成 曙

(第二炮兵工程学院 502 教研室, 陕西 西安市 710025)

摘 要: 本文以 51 系列单片机为背景, 针对电子系统中经常出现的随机干扰, 介绍了几种使用较为普遍的克服随机干扰信号的单片机数字滤波算法, 并给出了相应的算法程序。同时对这几种算法进行了比较, 并指出了每一种算法的具体适用范围和注意事项。

关键词: 单片机; 随机干扰; 数字滤波

中图分类号: TP368.1

文献标识码: A

文章编号: 1672-4984(2005)06-0097-03

Research on algorithm of digital filter

NIU Yur-peng, CHENG Shu

(The Second Artillery Engineering College, Xi'an 710025, China)

Abstract: Aiming at the random interruptions that usually appears in the electronic systems, this article introduced several algorithms of digital filter and program based on MCU of 51 series to inhibit the random interruptions. Comparison among the several algorithms and application field were also mentioned.

Key words: MCU; Random interruption; Digital filter

1 引 言

在单片机应用中经常需要对输入的信号进行处理, 当由于存在随机干扰使被测信号中混入了无用的成分时, 可以采用滤波器滤掉信号中的无用成分, 从而提高信号质量。模拟滤波器在其低频和甚低频时实现是比较困难的, 而数字滤波器是不存在这些问题的。它具有精度高、高可靠性和高稳定性的特点, 因此在单片机应用系统中被广泛用于克服随机误差。采用数字滤波算法克服随机误差主要有如下优点:

(1) 数字滤波是由软件程序实现的, 不需要硬件, 因此不存在阻抗匹配的问题。

(2) 对于多路信号输入通道, 可以共用一个软件“滤波器”, 从而降低仪表的设计成本。

(3) 只要只当改变滤波器程序或元算参数, 就能方便的改变滤波特性, 这对于低频脉冲干扰和随机噪声的克服特别有效。

2 算法介绍

为便于以下的程序书写方便, 这里做一个假设: 假定从 8 位 AD 中读取数据 (如果是更高位的 AD 可定义数据类型为 int), 子程序为 get_ad()。

2.1 一阶滞后滤波法

硬件滤波电路中最常用的是一阶惯性 RC 模拟低通滤波器, 当采用这种模拟滤波器来抑制低频干扰时, 要求滤波器有较大的时间常数和高精度的 RC 网络, 增大时间常数有求增大 R 值, 其漏电流也随之增大, 从而降低了滤波效果。而采用数字滤波算法来实现动态的 RC 滤波, 则能很好的克服上述模拟滤波器的缺点。在模拟常数要求较大的场合这种方法显得更为实用, 一阶惯性滤波算法对于周期干扰具有良好的抑制作用, 其不足之处是带来了相位滞后, 导致灵敏度低。同时它不能滤除频率高于采样频率二分之一 (称为奈奎斯特频率) 的干扰信号。例如采样频率为 100Hz, 则它不能滤除 50Hz 以上的干扰信号。对于高于奈奎斯特频率的干扰信号, 应该采用模拟滤波器。假设 X 为输入, Y 为滤波后的输出值, 则其公式为:

$$y_n = a * x_n + (1 - a) * y_{n-1}$$

式中 a 为与 RC 值有关的一个参数, 当采样间隔 t 足够小时, $a = t/(RC)$ 。

下面是实现一阶惯性滤波算法的程序。

```
/* 为加快程序处理速度假定基数为 100, a = 0  
~ 100 */  
# define a 50
```

收稿日期: 2004-11-14; 收到修改稿日期: 2005-01-09

```

char value ;
char filter ()
{
char new _ value ;
new _ value = get _ ad () ;
return (100 - a) * value + a * new _ value ;
}

```

2.2 限副滤波法

由于单片机系统中可能存在随机脉冲干扰,或由于变送器不可靠而将脉冲干扰引入输入端,从而造成测量信号的严重失真。对于这种随机干扰,限副滤波是一种有效的方法。其基本方法是比较相邻(n 和 $n-1$ 时刻的两个采样值 y_n 和 y_{n-1})。然后根据经验确定两次采样允许的最大偏差。如果两次采样值 y_n 和 y_{n-1} 的差值超过了所允许的最大偏差范围,则认为发生可随机干扰,并认为后一次采样值 y_n 为非法值,应予删除,删除 y_n 后,可用 y_{n-1} 代替 y_n 。若未超过所允许的最大偏差范围,则认为本次采样值有效。

下面是实现限副滤波法的程序。

```

/* A 值可根据实际情况调整
value 为有效值 ,new _ value 为当前采样值
滤波程序返回有效的实际值 */
# define A 10
char value ;
char filter ()
{
char new _ value ;
new _ value = get _ ad () ;
if ((new _ value - value > A) (value - new _
value > A)
return value ;
return new _ value ;
}

```

2.3 中位值滤波法

中位值滤波法是对某一被测参数连续采样 n 次(一般 n 取奇数),然后再把采样值按大小排列,取中间值为本次采样值。中位值滤波法能有效克服偶然因素引起的波动或采样不稳定引起的误码等脉冲干扰。对温度、液位等缓慢变化的被测参数用此法能收到良好的滤波效果,但是对于流量压力等快速变化的参数一般不宜采用中位值滤波法。

下面是实现中位值滤波法的程序。

```

/* N 值可根据实际情况调整
排序采用冒泡法 */

```

```

# define N 11
char filter ()
{
char value _ buf [N] ;
char count ,i ,j ,temp ;
for (count = 0 ;count < N ;count ++ )
{
value _ buf [count ] = get _ ad () ;
delay () ;
}
for (j = 0 ;j < N - 1 ;j ++ )
{
for (i = 0 ;i < N - j ;i ++ )
{
if (value _ buf [i] > value _ buf [i + 1])
{
temp = value _ buf [i] ;
value _ buf [i] = value _ buf [i + 1] ;
value _ buf [i + 1] = temp ;
}
}
}
return value _ buf [(N - 1) / 2] ;
}

```

2.4 算术平均滤波法

算术平均滤波法适用于对一般的具有随机干扰的信号进行滤波。这种信号的特点是信号本身在某一数值范围附近上下波动,如测量流量、液位时经常遇到这种情况。

算术平均滤波法是要按输入的 N 个采样数据,寻找这样一个 Y ,使得 Y 与各个采样值之间的偏差的平方和最小。

在编写算术平均滤波法程序时严格注意两点:一是对于定时测量,为了减小数据的存储容量,可对测得的值直接进行计算,但由于某些应用场合,为了加快数据测量的速度,可采用先测量数据,并存放在存储器中,测完 N 点后,再对 N 个数据进行平均值计算。二是选取适当的数据格式,也就是说采用定点数还是采用浮点数。采用浮点数计算比较方便,但计算时间较长,采用定点数可加快计算速度,但必须考虑累加时是否产生溢出。例如数据为14位二进制定点数时,采用双字节运算,当 $N > 4$ 时,就可能产生溢出。

下面是实现算术平均滤波法的程序。

```

# define N 12

```

```

char filter()
{
int sum = 0;
for (count = 0 ;count < N ;count ++ )
{
sum + = get _ ad() ;
delay() ;
}
return (char) (sum/N) ;
}

```

2.5 递推平均滤波法(又称滑动平均值滤波法)

上面介绍的算术平均值滤波,每计算一次数据,需要测量 N 次。对于测量速度较慢或要求计算速度较高的实时系统,该方法是无法使用的。例如某 ADC 芯片的转换速率为每秒钟 10 次,而要求每秒输入 4 次数据时,则不能大于 2。下面介绍一种只需要进行一次测量,就能得到一个新的算术平均值的方法,即滑动平均值滤波法。

滑动平均值滤波法采用队列作为测量数据存储单元,设队列的长度为 N ,每进行一次测量,把测量结果放于队尾,而扔掉原来队首的一个数据,这样在队列中始终就有 N 个“最新”的数据。当计算平均值时,只要把队列中的 N 个数据进行算术平均,就可得到新的算术平均值。这样每进行一次测量,就可得到一个新的算术平均值。

滑动平均值滤波法中的队列一般采用循环队列来实现。

下面是实现递推平均滤波法的程序。

```

# define N 12
char value _ buf[N] ;
char i = 0 ;
char filter()
{
char count ;
int sum = 0 ;
value _ buf[i ++ ] = get ad() ;
if (i = N) i = 0 ;
for (count = 0 ;count < N ,count ++ )
sum = value _ buf[count] ;
return (char) (sum/N) ;
}

```

```

}

```

2.6 加权滑动平均滤波法

在算术平均滤波和滑动平均滤波法中, N 次采样值在输出结果中的比重是均等的,即 $1/N$,用这样的滤波算法,对于时变信号会引入滞后, N 越大,滞后越严重。为了增加新的采样数据在滑动平均滤波法中的比重,以提高系统对当前采样值中多受干扰的灵敏度,可以采用加权滑动平均滤波法。实质上它是前面介绍的滑动平均滤波算法一种改进,即对不同时刻的数据加以不同的权,通常越接近现时刻的数据,权取得越大。加权滑动平均滤波算法适用于由较大纯滞后时间常数的对象和采样周期较短的系统,而对于纯滞后时间常数较小,采样周期较长,变化缓慢的信号,则不能迅速反映系统当前所受干扰的严重程度,滤波效果较差。

2.7 复合滤波法

单片机系统在实际应用中,所受到的随机干扰往往不是单一的,又是既要消除脉冲干扰的影响,又要做数据平滑处理。因此,在实际应用中常常把前面介绍的两种以上的方法结合起来使用,形成所谓的复合滤波。例如,防脉冲扰动平均值滤波算法就是一种实例。这种算法的特点是先中位值滤波算法滤掉采样值中的脉冲干扰,然后把剩下的各个采样值再进行滑动平均滤波。由于这种滤波算法兼容了中位值平均滤波算法和滑动平均滤波算法的优点,所以无论是对缓慢变化的过程变量还是对快速变化的过程变量都能起到较好的滤波效果。

3 结束语

本文介绍了几种使用较为普遍的克服随机干扰的软件算法,但是在一个具体的单片机系统中究竟应该选用哪种算法,还取决于系统的使用场合以及在过程中所含有的随机干扰情况。

参考文献

- [1] 余永权. 单片机在工业控制中的应用[M]. 北京:电子工业出版社,2003.
- [2] 胡伟,季晓衡. 单片机 C 程序设计及应用实例[M]. 北京:人民邮电出版社,2003.
- [3] 刘经燕. 测试技术及应用[M]. 广州:华南理工大学出版社出版,2002.