$\log_{10}x$ 的快速算法及 DSP 实现

刘玉玲　康忠林
（华侨大学信息科学与工程学院，泉州 362021）

摘　要　本文针对仪器测量范围以及显示的需要，提出了一种 $\log_{10}x$ 的快速计算方法，并详细的讲述了算法原理以及实现的具体步骤，最后还使用 DSP 实现了该快速算法。此算法大大提高了计算效率，对需要快速计算对数的仪器有着重大意义。

关键词　对数运算；DSP 实现；最小二乘法；线性拟合

0　引言

随着数字化的不断成熟，越来越多的设备从模拟转向数字，其中仪器也不例外。现在很多仪器公司纷纷把仪器数字化[1]，这样可以降低成本、提高性能、减小体积和方便升级等等。对于用户，数字化的仪器价格更低、性能更好、操作更简单等等。但仪器数字化同时也给研发带来一定的困难，譬如说仪器所测量的信号范围很宽，就得采用对数单位—分贝（dB），或者简单点说就是用常用运算 $\log_{10}x$ 来将信号范围对数缩小，模拟仪器处理时，可以通过一个小小的对数放大器即可实现，数字化仪器中却不能这样，但可以采用数字信号处理器，如 DSP，这就涉及到另外一个问题，使用定点 DSP 实现 $\log_{10}x$ 计算效率太低，无法完成仪器中快速的运算需求。

本文就该问题提出一种实现以 10 为底的对数运算的快速算法，该算法主要是针对数字化仪器的需要进行快速对数运算。其实与该运算类似的运算 $\log_2x$ 在 C 语言的库函数中已经有定义，而现在的所有的硬件都有 C 语言汇编器，都能直接调用以 2 为底的对数运算，联合换底公式就可以计算出以 10 为底的对数运算。那为什么还需要研究该算法呢？原因就是直接调用这个对数函数消耗的时间过长，执行效率低。在高速处理环境中最常用的就是定点 DSP[2]（因为浮点 DSP 主频太低），使用定点 DSP 执行 $\log_{10}2500$ 需要 5725 个时钟周期，如果频率为 600MHz，那么每秒钟也只能处理 104803.5 个数据，也就是 0.1M，这相对于几兆或者十几兆每秒的数据流来说真的是太小了。

1　快速算法原理

本文提出的快速算法的主要思想就是用一个简单的多项式来代替对数运算，并保证误差在 10^{-4} 内，以满足一般计算的需要，多项式在 DSP 的实现要比直接调用 log() 函数稍显复杂，但是其执行效率却要远远高于直接调用对数函数。

这个多项式可以通过拟合 $y=\log_{10}x$ 曲线来得到，考虑到拟合范围过大的曲线带来不必要的复杂，因此本文在拟合之前对曲线进行了两次归一化的预处理，缩短拟合长度。第一次归一化是以 10^i 为除数，把数据归一化到 [1,10)，第二次归一化则是以 2^i 为除数，把 [1,10) 数据归一到 [1,2)，那么最后需要拟合的就只有 $y=\log_{10}x\ (1\leqslant x<2)$ 这一小段函数。这样既可以计算所有数据的对数，又可以降低拟合的阶数。

本文是采用最小二乘法[3-4]来拟合对数运算曲线，以此来获得所需要的多项式系数。但是在真正实现的过程中并不需要人工按照最小二乘法的理论来计算多项式的系数，而是可以借助 Matlab 这个数学工具。

2　最小二乘法

最小二乘法是一种数学优化技术，它是通过最小化误差的平方和找到一组数据的最佳函数匹配，它也是数理统计中一种常用的方法，在工业技术和其他科学研究中有广泛应用。

假设样点 $\varphi_0(x),\varphi_1(x),\cdots,\varphi_n(x)$ 是线性无关的，令

$q(x)=a_0\varphi_0(x)+a_1\varphi_1(x)+\cdots+a_n\varphi_n(x)$，求 $a_0,a_1,\cdots,a_n$ 使得

$$\varphi(a_0,a_1,\cdots,a_m)=\sum_{k=1}^{n}[q(x_k)-y_k]^2=\sum_{k=1}^{n}\left[\sum_{j=0}^{m}a_j\varphi_j(x_k)-y_k\right]^2 \quad (1)$$

取得最小值。现在 $\varphi_k(k)=x^k(k=0,1,\cdots,m)$，这时的拟合是 m 次最小二乘拟合多项式。

$$\begin{bmatrix}(\varphi_0,\varphi_0) & (\varphi_0,\varphi_1) & \cdots & (\varphi_0,\varphi_m)\\ (\varphi_1,\varphi_0) & (\varphi_1,\varphi_1) & \cdots & (\varphi_1,\varphi_m)\\ \vdots & \vdots & & \vdots\\ (\varphi_m,\varphi_0) & (\varphi_m,\varphi_1) & \cdots & (\varphi_m,\varphi_m)\end{bmatrix}\begin{bmatrix}a_0\\a_1\\\vdots\\a_m\end{bmatrix}=\begin{bmatrix}(y,\varphi_0)\\(y,\varphi_0)\\\vdots\\(y,\varphi_0)\end{bmatrix} \quad (2)$$

其中 $(\varphi_i,\varphi_j)=\sum_{k=1}^{n}\varphi_i(x_k)\varphi_j(x_k)$，$(y,\varphi_i)=\sum_{k=1}^{m}y_k\varphi_i(x_k)(i=0,1,\cdots,m)$。这方程称为法方程，按照此方程求得的解可以使式(1)达到最小值。

在 Matlab 中具体实现如下：设 $y=\log_{10}x$ $(1\leqslant x<2)$的结果矩阵为 sample，sample=log(j)/log10，其中，$j=1:0.0001:2$，即 j 取 1 到 10 以 0.0001 为步长的所有数据，因为在 Matlab 中的函数 log() 是以 2 为底的，为了计算出以 10 为底的样点，这样里需要“换底公式”计算出样点矩阵 sample。

计算出样点以后即可通过使用多项式拟合的函数 polyfit 来拟合曲线，计算出最小二乘法的多项式系数矩阵 p，p = polyfit(j, sample, 4)，为了达到精度的需要，本文通过一系列实验，最后采用 4 阶的拟合。因为拟合的情况很好，如果完全显示波形就看不出来差别，为了看出拟合误差，这里就显示了该段中误差最大的图形，具体图像如图 1 所示。

拟合绝对误差是拟合曲线与原始曲线差值的绝对值，各处的差值如图 2 所示，其中最大误差为 5.8×10^{-5}，由此可见完全符合 10^{-4} 精度的要求。若采用 5 阶拟合，最大误差为 0.9×10^{-5}，对一般仪器没有必要。通过最小二乘法拟合对数曲线可以计算出来的多项式各项系数 $p_1=[-0.0238,0.1893,-0.6266,1.2128,-0.7516]$，所以以 10 为底的对数运算对于计算 1～2 段的代替多项式可以表示如下：

$$y=\log_{10}(x)\approx-0.0238x^4+0.1893x^3-0.6266x^2+1.2128x-0.7516,(1\leqslant x<2) \quad (3)$$

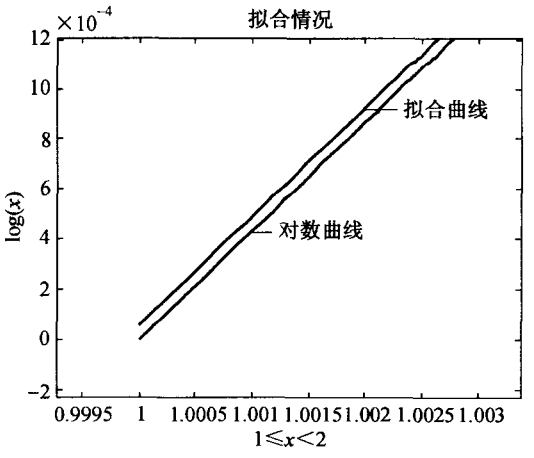

图 1　1≤x<2 时对数曲线和拟合曲线误差最大的一段波形

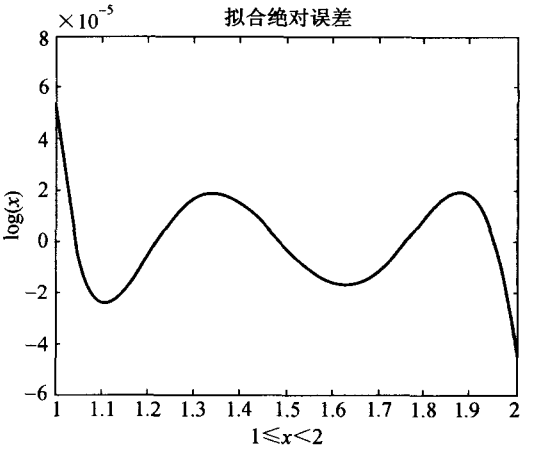

图 2　4 阶拟合的绝对误差

3　具体算法实现

下面通过一个计算实例来介绍该算法实现的具体步骤，算法流程图如图 3 所示。在此流程图中，a_1、a_2、a_3、a_4、a_5 是拟合的对数曲线的多项式系数，其取值如式(3)，$a_1=-0.0238$、$a_2=0.1893$、$a_3=-0.6266$、$a_4=1.2128$、$a_5=-0.7516$。

$$\log_{10}x=\log_{10}(10^m\times2^n\times B)=m+n\log_{10}2+\log_{10}B$$

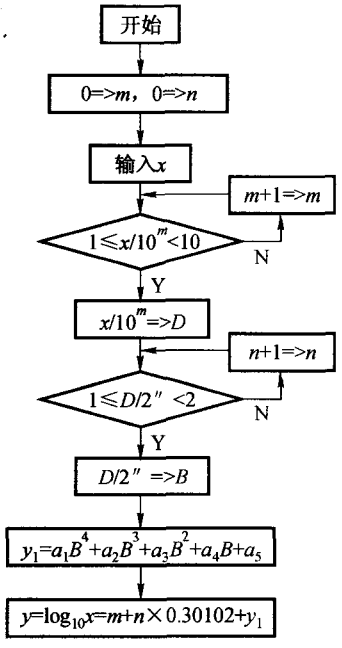

图 3　算法流程图

$$= m + 0.30102n + a_1B^4 + a_2B^3 + a_3B^2 + a_4B + a_5 \quad (4)$$

假设需要进行对数运算的数据为 2500，其对数运算的最终结果为 $y = \log_{10}2500 = 3.39794$。使用本文提出的方法计算如下：

首先经过两次归一化处理，得到下式

$$\log_{10}x = \log_{10}2500 = \log_{10}(1000 \times 2 \times 1.25) = 3 + 0.30102 + \log_{10}1.25$$

上式需要计算的只有 $\log_{10}1.25$，这可以利用式(3)的多项式计算，得出 1.25 的对数值 $y_1 = 0.09695$。最后，$y = 3 + 0.3010 + y_1 = 3 + 0.30102 + 0.09695 = 3.39797$，计算其误差为 3×10^{-5}。

4　DSP 实现

在 DSP 实现时，前面两次的归一化处理是通过判断来实现，本设计使用 C 语言编写，这样做既方便编程又没有影响执行效率，例如 $x = 2500$，则 $i = 3$，归一化结果为 2.5，第二次归一化到 1～2，可以通过 2.5/2 = 1.25 来实现，那么 $\log_{10}2.5 = \log_{10}(1.25 \times 2) = \log_{10}1.25 + \log_{10}2$，$\log_{10}2 = 0.30102$ 也是一个常量了。

在整个计算中最费时间的就是通过拟合的多项式计算 1～2 段之间的对数了，由于大部分高速 DSP 都是定点的，本设计采用的 TMS320F2812 也是定点 DSP，为了提高执行效率，本设计采用汇编语言实现，把多项式的小数系数，通过左移 12 位，也就是乘以 4096 都变成整数。1～2 段的被求对象也是同样通过左移 12 位转换成整数再代入拟合多项式计算结果。在计算完整个多项式以后再通过一次右移 24 位来还原计算结果。这样操作既保证了精度又便于 DSP 运算，因为定点 DSP 计算一个浮点数所花的时间要远远超过计算一个整数所花的时间。

5　实验结果

在 Matlab 处理数据，由图 2 可知，4 阶拟合后的最大误差为 5.8×10^{-5}，在各种仪器中需要以分贝为单位的计算一般为 $y = 10\log_{10}x$，即使是使用 dBV 表示电压，其计算公式也就是 $y = 20\log_{10}x$，所以通过误差传递公式、公式(3)和(4)，可计算出使用本文提出的快速算法进行对数转换最大的误差值为 $\varepsilon = 20 \times 5.8 \times 10^{-5} = 1.16 \times 10^{-3}$，这足以满足一般仪器的精度要求，即使是“Agilent”的仪器，其精度一般也都在 0.01dB。

本文提出的改进算法需要的计算量只有 15 次乘法计算和 5 次加减运算，而且这些运算非常适合 DSP 的处理。使用 C 语言定义的“math.h”文件中的对数计算 $\log_2 2500/\log_2 10$ 需要 5725 个时钟周期，而采用本文提出的算法进行计算 $\log_{10}2500$ 只需要 264 个时钟周期。速度足足提高了 21 倍多，这种提高对于低速率的处理，可能没有太大的意义，但是如果对于需要高速的数据计算场合，这种速度的提高就难能可贵了。

该算法已经使用在新开发的频谱仪中，并取得了良好的效果。

参考文献

[1] Yih - Chyun Jenq. Direct digital synthesizer with jittered clock. IEEE Transactions on Instrumentation and Measurement, 1997, 46(3)

[2] TMS320C6414, TMS320C6415, TMS320C6416FIXED - POINT-DIGITALSIGNALPROCESSORS.SPRS146N - FEBRUARY 2001 - REVISED MAY 2005

[3] JORGE NOCEDAL(美), STEPHEN WRIGHT.NUMERICAL OPTIMIZATION. 北京:科学出版社,2006

[4] 叶明凤．关于快速计算 2 基对数与生成随机数函数的设计与实现．计算机与信息技术,2007(7)

[5] 单明,聂燕萍．线性拟合中的逐差法和最小二乘法的比较．大学物理实验,2005(6)