第一届(1984)全国信号处理学术会议

论　文　集

第四卷

(共四卷)

中国电子学会
中国仪器仪表学会　信号处理学会

1984 · 承德

用微处理器实现流水线FFT

哈尔滨工业大学无线电工程系
刘永坦、张　学、孙松岩

摘　要

FFT是数字信号处理的一个重要组成部分。如何实现实时的FFT处理，是许多应用部门亟需解决的问题。本文提出了一个以微处理器为基础实现流水线FFT的方案，其可以实时处理的信号频率范围为（复信号）：0～5KC，系统的动态范围大于70db。文中还给出了系统动态范围的计算机仿真结果，并指出了进一步提高系统主要指标的途径和方法。

一　引　言

FFT是DFT的快速算法[1]。经过不少人近些年的探讨，产生了许多FFT算法，在计算机模拟分析中得到了十分广泛的应用。但在通用计算机上软件实现FFT一般不能进行实时处理。因此，用硬件实现FFT，对高速实时处理而言是一个重要途径。

硬件实现FFT主要有两种方式：一种是串行处理，一种是并行处理[1]。串行处理硬件结构简单，成本低，但速度不易提高。并行处理则相反。在并行处理诸方式中，速度／成本比较高的是流水线方式。流水线处理的特点是：采用适量的通用器件就可达到较高的处理速度指标。

现以图1所示的8点FFT来说明流水线处理过程：将三个运算单元级联即可构成一个8点FFT流水线处理器。每个运算单元对应于信号流图中的一级。在每批（8个）数据的处理中，各运算单元始终处理位于该级的数据，并以适当的方式将结果送至下一级继续处理，之后，该级又开始处理下一批数据。当数据源源不断地送入后，第后

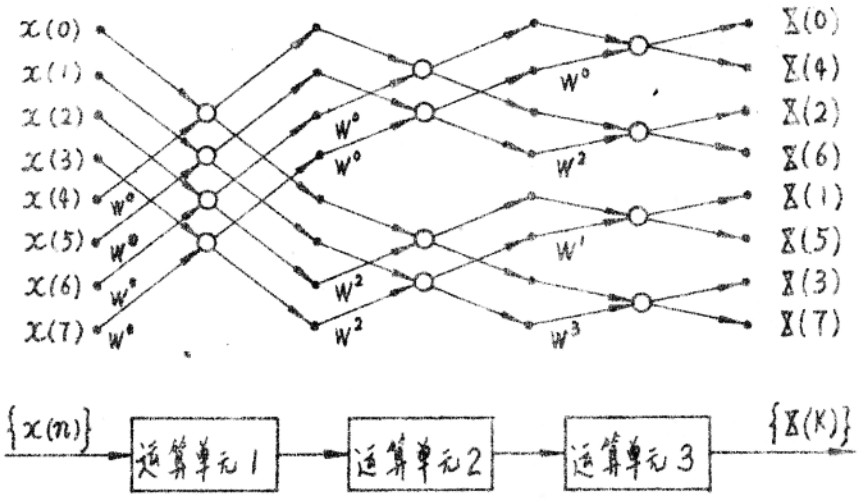


图1. 8点FFT流水线示意图

一级可将每批数据的处理结果连续地送出。当然，这是一个并行处理的过程，大大提高了数据的吞吐能力。

运算单元的结构与所采用的算法有很大的关系。基2算法具有简洁的蝶形结构，因而最适于硬件实现。

用微处理器和高速硬件乘法器构成运算单元，是一种比较理想的结构。微处理器功能强，使用灵活、方便，可使结构大大简化；硬件乘法器速度快，可弥补微处理器乘法运算慢的不足。二者的结合可使整个系统的性能/成本比达到较高的指标。

我们以基2算法为基础，选择Z80A CPU[2]和TDC1010J硬件乘法器[3]作为主要器件，构成一个八级流水线FFT处理系统(256点复数据处理)，采用16 bits二进制补码定点运算体制。分析表明，本系统可达到较高的速度和动态范围指标。

二、硬件结构

整个系统由三部分组成，如图2所示：

第一部分是输入接口，完成对模拟信号的采样及A/D变换；第二

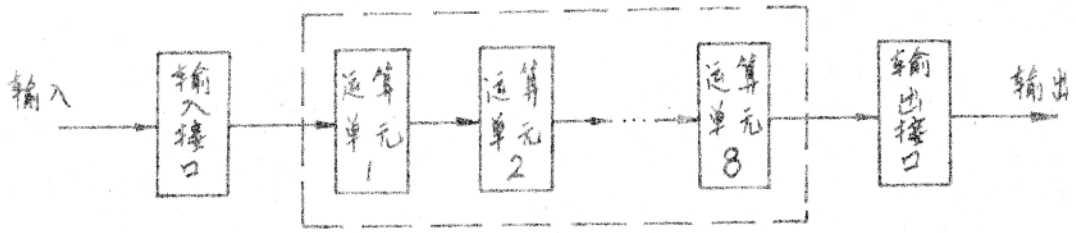


图2. 8级流水线FFT

部分是虚线框内的八级流水线处理器，这是系统的核心；最后是输出接口部分。下面主要讨论一下第二部分的构成。

流水线处理器由八个完全相同的运算单元级联而成。每个运算单元的结构如图3所示：

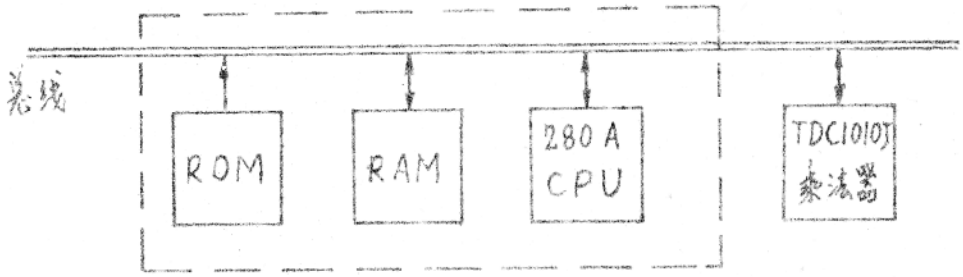


图3. 运算单元结构

图3中，ROM是只读存贮器，存放程序和FFT中的旋转因子。RAM是随机存贮器，存放处理数据。虚线框内部分与常见的单板计算机相同。乘法器直接接入系统总线，以免通过I/O口降低处理速度。

在这种流水线结构里，必须采用多总线结构（即各运算单元都有自己独立的总线）。为了使数据的传送尽量减少对处理速度的影响，我们采用存贮器交换的方式来完成两相邻运算单元之间的数据传送。这种方式的优点是使数据传送基本上不占用CPU时间，各级处理在时间上得到了最大的重合，从而提高了处理速度。

流水线处理中须要特别保持同步，否则将得不到正确的结果。

三、处理速度

对于流水线FFT，整个系统的处理速度与各级的处理速度相同。我们可以分析某一运算单元的运算过程，从而得出系统的处理速度。

本系统采用实型运算单元，完成一次蝶形运算需进行4次实数乘法，6次实数加、减法，取入6个数据，送出4个数据[4]。因为数据字长为16 bits，在处理中都采用Z80A的双字节操作指令。为了提高运算速度，利用Z80A内部的寄存器对和辅助寄存器对作为蝶形运算的中间结果寄存器，以减少对存储器的访问次数。按照以上方式，我们用汇编语言写出程序（其中包括运算过程的控制），通过查Z80指令表，计算出完成一次蝶形运算约需170 μS。

256点FFT每级有128个蝶形运算，因此，总运算时间约为：

$$T_1 \doteq 170\mu S \times 128 = 21.76\ mS \qquad \cdots\cdots (1)$$

系统的管理时间取运算时间的20%，则总处理时间为：

$$T = T_1(1+20\%) \doteq 26\ mS \qquad \cdots\cdots (2)$$

在处理单路复信号时，本系统能实时处理的信号频率范围约为0～5KC。对于实信号，则为0～10KC。

影响可处理信号频带宽度的主要因素是微处理器的速度，若采用更高时钟频率或16 bits的微处理器，可不同程度地提高系统处理速度。

例如，当采用Z80B CPU（6MC时钟）时，实时处理信号频率范围能扩展到0～7KC（复信号）。若采用Z8000 CPU（16bits、4MC时钟），实时范围能达到0～10KC（复信号）。若采用M68000 CPU，速度将有成倍地提高。

四、动态范围

字长和运算体制的选择对系统的性能及设备的复杂程度有很大地影响。一般要在处理速度、动态范围和成本等问题上折衷考虑。

图4示出了在FFT中对系统动态范围有影响的诸单元[5]，其影响是有限字长效应产生的。这些算法分别是：A/D变换，加权函数，余弦表（旋转因子），运算单元及FFT算法（即溢出控制方式）。

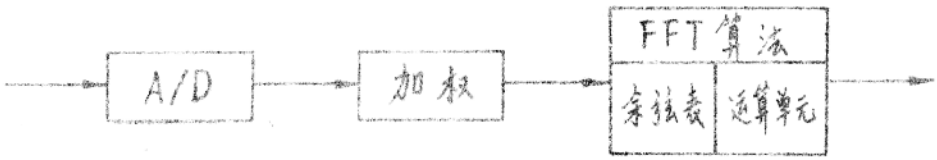


图4. FFT算法中舍化噪声的来源

我们根据前面讨论的硬件结构，对上述各单元对系统动态范围的影响用计算机进行了仿真实验。图5各曲线描述了在用一有限字长实现某单元，所引限字长实现某单元时，该单元对系统动态范围的贡献与字长的关系。图中，量化均采用舍入方式，溢出控制采用条件判断移位。纵座标为动态范围，横座标为bit数。

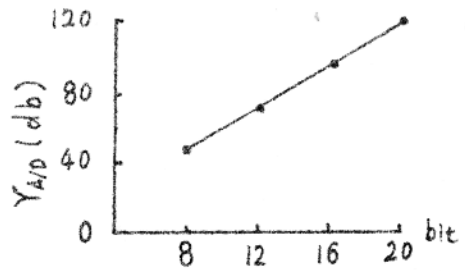


图5(a). A/D变换

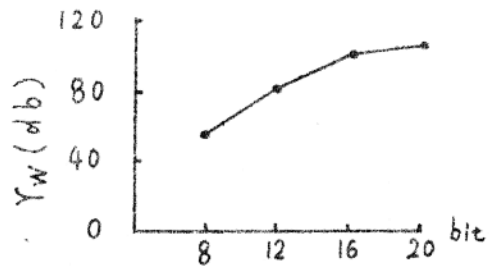


图5(b). 加权函数

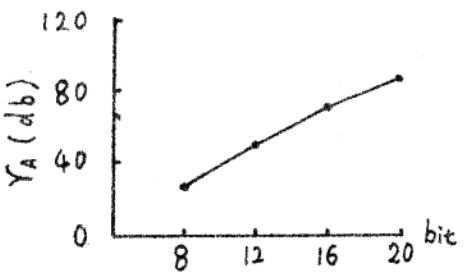


图5(c). 运算单元

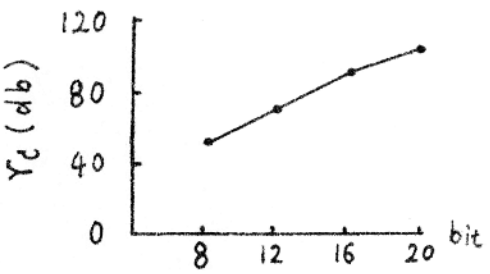


图5(d). 余弦表

由图5可见，当运算单元为16 bits，A/D为12 bits，加权函数为12 bits，余弦表为12 bits，溢出控制采用条件判断移位，另此采用舍入时，系统动态范围可达70 db 以上。图6给出了采用上述参数时该系统对双频率正弦信号的计算机模拟估谱结果，f_1处的弱信号比f_2处的强信号(满刻度)低70 db，但在图中清楚可见。

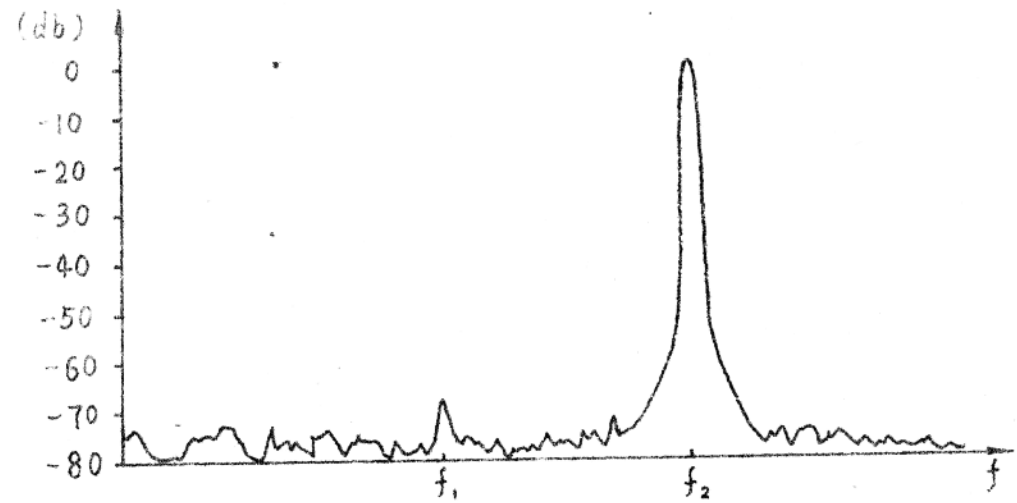


图6. 计算机模拟估谱结果

五. 结 语

由于采用微处理器，使本系统结构简单，应用灵活。流水线处理和乘法器的使用大大提高了系统的数据吞吐量。合理选择字长使系统获得了较大的动态范围。只要配以相应的接口，本系统可用于雷达、语音处理、信号频谱分析等许多领域。

参考文献：

[1] 《数字信号处理的原理及应用》，[美]，L.R.拉宾纳 等编 国防版

[2] 《280单板机使用手册》，北京工业大学电子厂等

[3] TRW LSI PRODUCTS 1979.

[4] 《快速付里叶变换》 清华大学

[5] 《A Nomogram for Determining FFT System Dynamic Range》
E.O. Brigham and L.R. Cecchini IEEE-ICASSP. 1977.

准最佳卡尔曼滤波器的设想

北方交通大学，研究生，李济生

摘要：本文介绍分段常增益和分段线性增益的方法，提出了一种计标量较少的卡尔曼滤波增益公式，它特别适用在微处理机上实现。

(I) 引言

自卡尔曼滤波问世以来，人们试图为减少计标量做了很多工作。但是，由于卡尔曼滤波要求的计标量较大，起初只限于在较大的计标机上实现。随着微处理机的发展，自然提出卡尔曼滤波是否能搬到微处理机上进行实时处理。基于这些考虑，本文提出了一种准最佳卡尔曼滤波四，並分别研究分段常增益和分段线性增益的计标方法。这两种方法假适用于观测速率大于计标机迭代周期的情况。然而，无论那一种方法，它们都要计标某一时刻的增益，若用一般的卡尔曼滤波方程来计标，需要的计标量很大。针对这个问题，本文也提出了一种简便的增益计标方法——降阶递推法，当观测阶数远小于系统阶数时，计标量节省得相当显著。这样，很容易在小容量的微处理机上实现。

(II) 基本问题：

考虑动态系统的离散状态方程为：

$$X_{k+1} = \Phi_k X_k + \Gamma_k W_k \tag{1}$$

观测方程为：

$$Y_k = H_k X_k + V_k \tag{2}$$

假设系统的初始状态X_0为随机变量，且

$$E(X_0) = \bar{X}_0 \,,\ \mathrm{Cov}(X_0, X_0) = P_0 \tag{3}$$

假设噪声W_k，V_k是不相关的另均值高斯随机序列，且具有已知方差，

$$\begin{aligned} &E\{W_k\} = 0 \,,\ \mathrm{Cov}\{W_k, W_j\} = Q_k \delta_{kj} \,,\ Q_k \geq 0 \\ &E\{V_k\} = 0 \,,\ \mathrm{Cov}\{V_k, V_j\} = R_k \delta_{kj} \,,\ R_k > 0 \\ &\mathrm{Cov}\{W_k, V_j\} = \mathrm{Cov}\{W_k, X_0\} = \mathrm{Cov}\{V_k, X_0\} = 0 \end{aligned} \tag{4}$$

基中：状态X_k是n维向量，测量Y_k是r维向量，噪声W_k是m维向量。卡尔曼滤波方程为：

$$\hat{X}_k = \Phi_{k-1}\hat{X}_{k-1} + K_k(Y_k - H_k\Phi_{k-1}\hat{X}_{k-1}) \tag{5}$$

$$K_k = P_{k|k-1}H_k^T(H_kP_{k|k-1}H_k^T + R_k)^{-1} \tag{6}$$

$$P_{k|k-1} = \Phi_{k-1}P_{k-1}\Phi_{k-1}^T + \Gamma_{k-1}Q_{k-1}\Gamma_{k-1}^T \tag{7}$$

$$P_k = (1 - K_kH_k)P_{k|k-1} \tag{8}$$

(Ⅲ) 准最佳卡尔曼滤波器的设计

尽可能的减少计标量，而引起估计精度下降的尽可能小，是设计准最佳滤波器的核心。一般来讲，减少计标量是有条件的，它是相对于某一具体过程而言。本节考虑的准最佳滤波器是指非强烈随机振动系统，其估计精度基本上能满足要求。

(1) 分段常增益.

这种方法前人已做了很多探讨[1]，特别最近 shieh. L. S 等人[2]提出了另一新方法，他们把卡尔曼滤波问题在数学上完全等效成为一个调节器问题，其实质是分段常误差协方差矩阵。本小节以[1]，[2]为基础进行讨论，並对实时处理提出了新看法。

设观测时刻为 k，增益更新时刻为 lN（N 为某一固定值）则对区间 $[lN, (l+1)N)$ 内的观测值 Y_k 均采用增益

$$K_{k,c} \triangleq K_{lN} = P_{lN|lN-1}H_{lN}^T(H_{lN}P_{lN|lN-1}H_{lN}^T + R_{lN})^{-1} \tag{9}$$

处理，就可得到分段常增益估计。N的取值与要求的精度有关。由图(1)曲线(a)可知：用 lN 时刻的增益 K_{lN} 代替区间 $[lN, (l+1)N)$ 的增益，会产生较大的误差，为

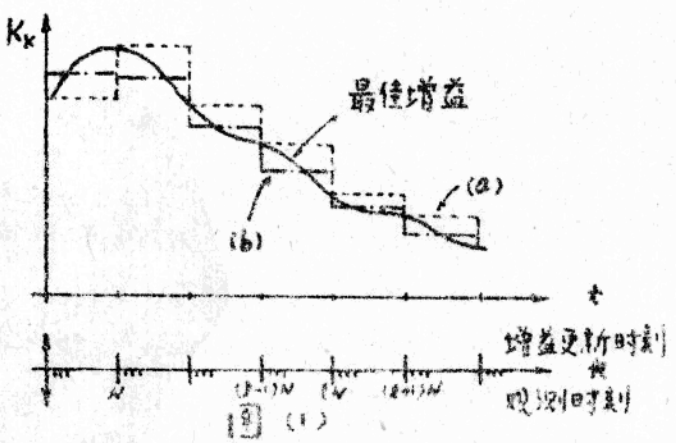


图 (1)

了减少误差，我们可取，

$$K_{k,c} = \frac{1}{2}\{K_{l_N+1} + K_{l_N}\}, \quad (lN \le k < (l+1)N) \tag{10}$$

其精度略有改进，见图(1)曲线(b)。但是它要求在计标了 K_{l_N+1} 之后才能估计区间 $[lN, (l+1)N)$ 的值，这必然会失去实时性。考察方程(6)——(8)可知，它们均与估值 X_k 和测量值 Y_k 无关，因此我们可以提前 N 个观测时刻来计标 K_{l_N+1}，保证在处理该区间的测量值之前计标出 K_{l_N+1}。这样就可以进行实时处理。

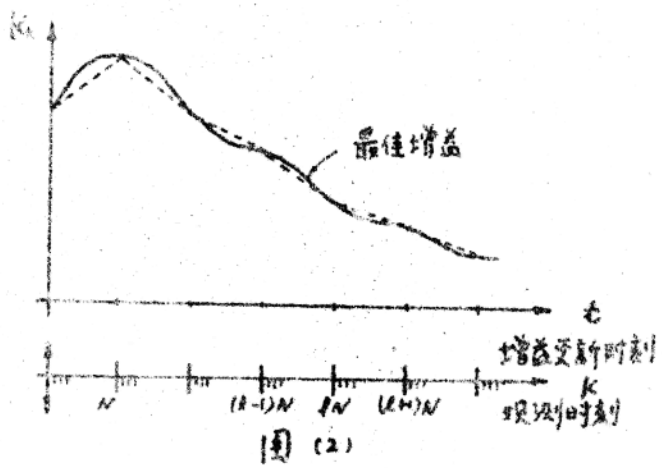


图 (2)

(2) 分段线性增益

最佳增益随时间而连续变化，因此，用分段常增益逼近最佳增益总是不够理想的。如果我们在两个增益更新点之间用线性函数而不用常数来逼近最佳增益，即，

$$K_{k,c} = K_{l_N} + \frac{K_{l_N+1} - K_{l_N}}{N}(k - lN), \quad (lN \le k < (l+1)N) \tag{11}$$

那么，估计精度可以得到很好改善，见图(2)。同样采用上述方法来保证实时性。然而，随之带来了计标量略有增加，但是，从精度提高与计标量减少折衷考虑，这种方法还是很可取的。

以上讨论知，无论那一种方法，都要用(6)——(8)式计标某一时刻的增益。然而，这个计标过程需要较大的计标量。为了减少计标量，下面提出一种简单计标增益的方法，它的基本思想是将一个高阶系统化简为一个低阶系统来递推计标增益。

(四) 降阶递推法的结构，流图及计标量比较

由于卡尔曼滤波的计算量主要是增益的计算量，因此，减少增益的计算是减化算法的根本。这里所讨论的是一种降阶法，它利用了测量矩阵的一种特殊结构，减少了很多运算，而这种结构是用线性变换得到的。

(1) 降阶递推法的结构：

对原系统作线性变换[3]，$X_k = M_k Z_k$，(1)——(2)式成为，

$$Z_{k+1} = A_k Z_k + B_k W_k \tag{12}$$

$$Y_k = C_k Z_k + V_k \tag{13}$$

其中：$$M_k = \left[\begin{array}{c|c} (H_k^1)^{-1} & -(H_k^1)^{-1} H_k^2 \\ \hline 0 & I \end{array}\right] \tag{14}$$

$$A_k = M_{k+1}^{-1} \Phi_k M_k, \quad B_k = M_{k+1}^{-1} \Gamma_k \tag{15}$$

$$C_k = [\, I_r \mid 0 \,] \tag{16}$$

且，$H_k = [H_k^1 \mid H_k^2]$，H_k^1是$r \times r$矩阵，H_k^2是$r \times (n-r)$矩阵，M_k^{11}，M_k^{12}，M_k^{22}有相应的阶数。

这样，估计X_k的问题就转变成估计Z_k的问题，它们之间的关系为，

$$\begin{aligned} X_k^1 &= M_k^{11} Z_k^1 + M_k^{12} Z_k^2 \\ X_k^2 &= Z_k^2 \end{aligned} \tag{17}$$

当r相对n很小时，由变换引入的计算量可忽略不计，参见表(1)。

由于观测矩阵具有(16)式的形式，相应的把$P_{k|k-1}$和P_k分块为[4]

$$P_{k|k-1} = \left[\begin{array}{c|c} P_{k|k-1}^{11} & P_{k|k-1}^{12} \\ \hline P_{k|k-1}^{21} & P_{k|k-1}^{22} \end{array}\right], \quad P_k = \left[\begin{array}{c|c} P_k^{11} & P_k^{12} \\ \hline P_k^{21} & P_k^{22} \end{array}\right] \tag{18}$$

且，$$P_k^{11} = [I - P_{k|k-1}^{11} (P_{k|k-1}^{11} + R_k)^{-1}] P_{k|k-1}^{11} \tag{19}$$

$$P_k^{12} = [I - P_{k|k-1}^{11} (P_{k|k-1}^{11} + R_k)^{-1}] P_{k|k-1}^{12} \tag{20}$$

$$P_k^{22} = P_{k|k-1}^{22} - (P_{k|k-1}^{12})^T (P_{k|k-1}^{11} + R_k)^{-1} P_{k|k-1}^{12} \tag{21}$$

其中，$P_{k|k-1}^{11}$可逆，定义，

$$G_k \triangleq P_k^{11} (P_{k|k-1}^{11})^{-1} \tag{22}$$

$$D_k \triangleq (P_{k|k-1}^{11})^{-1} (I - G_k) \tag{23}$$

得，$$P_k=\left[\begin{array}{c|c} P_k'' & G_k P_{k|k-1}^{12} \\ \hline (P_{k|k-1}'')^T G_k^T & P_{k|k-1}^{22}-(P_{k|k-1}^{12})^T D_k P_{k|k-1}^{12} \end{array}\right] \tag{24}$$

这里，P_k''是$r\times r$矩阵，P_k^{12}是$r\times(n-r)$矩阵，P_k^{22}是$(n-r)\times(n-r)$矩阵，$P_{k|k-1}''$，$P_{k|k-1}^{12}$，$P_{k|k-1}^{22}$有相应的阶数。由(24)式知，求P_k必须要知道P_k''和$P_{k|k-1}$。

另外，增益方程和预估误差协方差方程可分别写成。

$$K_k^1=P_{k|k-1}''(P_{k|k-1}''+R_k)^{-1} \tag{25}$$

$$K_k^2=P_{k|k-1}^{21}(P_{k|k-1}''+R_k)^{-1} \tag{26}$$

$$P_{k|k-1}=A_{k-1}P_{k-1}A_{k-1}^T+B_{k-1}Q_{k-1}B_{k-1}^T \tag{27}$$

由方程(19)，(22)——(27)构成了用降阶递推法求增益K_k的递推公式。

(2) 计标量比较

表(1)列出了用降阶递推法((19)，(22)——(27)式)和一般卡尔曼滤波法((6)——(8)式)计标增益以及由变换引入的计标量[5]。

	增益的计标量		变换引入的计标量	
	乘法次数	加法次数	乘法次数	加法次数
降阶递推法	$2n^3+5r^2+n^2m+m^2n+n^2r+r^2n$	$2n^3+n^2m+m^2n+5r^3+n^2r+nr^2-mn-2n^2-3nr+r^2$	nr	$nr-2r$
一般卡尔曼滤波法	$3n^2+r^3+n^2m+m^2n+2n^2r+2r^2n$	$3n^3+n^2m+mn^2+r^3+2n^2r+2nr^2-mn-3n^2-2nr$	0	0

表(1)

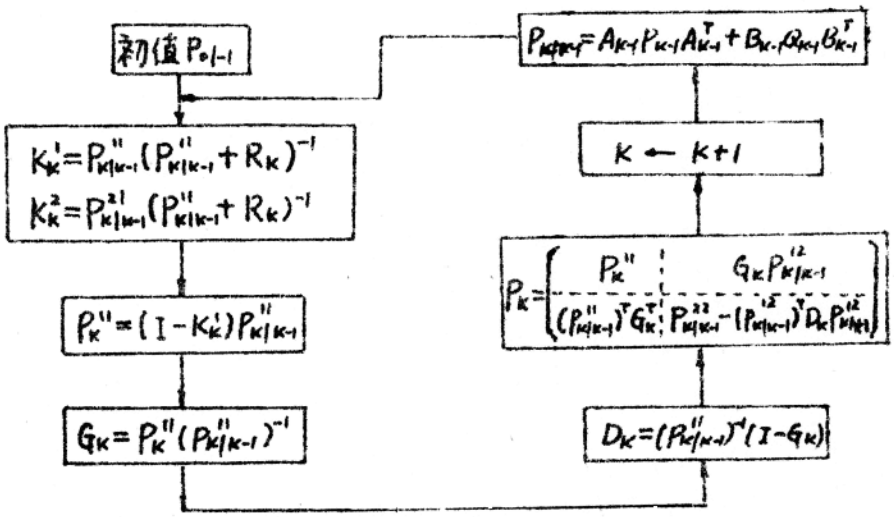
由此可见，用降阶递推法可以减少计标量。

例：

		增益的计标量		变换引入的计标量	
		乘法次数	加法次数	乘法次数	加法次数
降阶递推法	$n=m=8, r=4$	2752	2480	32	24
	$n=m=8, r=2$	2248	2012	16	12
一般卡尔曼滤波法	$n=m=8, r=4$	3392	3072	0	0
	$n=m=8, r=2$	2888	2600	0	0

表(2)

(3) 计算流程：



流程图

(四) 结论：

当观测速率大于计算机迭代周期，并且 $r \ll n$ 时，用这些方法进行实时处理很有效，故它适用于在小容量的微处理机上实现。

注：本文经由北方交通大学张树京教授，黄绵坤付教授认真审核，并对拙稿给予了热情的指导和帮助，对此，本人表示衷心的感谢。

主要参考文献：

[1]. Applied Optimal Estimation, the MIT Press. Cambridge. Massachusetts. A. Gelb; J. F. Kasper; R. A. Nash; 1974.

[2]. "Approximate Kalman Filters" Joint Technology Autmatic Control Conference (part of the ASME Century 2: Emerging Technology Conference). 1980.

[3]. "Suboptimal Desige Of Dicrete Kalman Filter And Smoother With Redurdant Measurement" IEEE Trans. Auto. Contr. No 2 P561—562. 1981.

[4]. "Data Compressron in Kalman Filter" 21st Israel Amual Conferercе on Aviation and Astronautics. 1979.

[5]. "Computational Requirements for A Discrete Kalman Filter" IEEE Trans, Auto. Contr. No 6 1971.

一种修正LMS自适应算法的研究

西北电讯工程学院 张玉洪 保 铮

一、引言

目前，自适应滤波技术的发展日趋成熟，并已广泛应用于通信、雷达、声纳、系统测辨、地球物理学……等许多领域。一般的自适应滤波如图1所示。图中输入矢量为 $\underline{X}(n)=[x_{1n}, x_{2n}, \cdots x_{Mn}]^T$，权矢量为 $\underline{W}(n)=[w_{1n}, w_{2n}, \cdots w_{Mn}]^T$，在 n 时刻的输出误差为

$$e(n)=d(n)-y(n)=d(n)-\underline{W}^T(n)\underline{X}(n) \tag{1}$$

大家知道，在最小均方(LMS)误差意义下的最佳权矢量由维纳(Wiener)解给出

$$\underline{W}_{opt}=R_x^{-1}r_{dx} \tag{2}$$

式中，$R_x \triangleq E[\underline{X}(n)\underline{X}^T(n)]$，$r_{dx} \triangleq E[d(n)\underline{X}(n)]$.

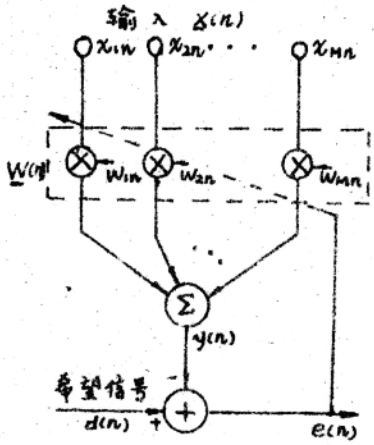


图1 自适应滤波原理图

有好几种实时的自适应迭代算法都收敛于维纳解[1]。但最简单也是最常用的还是Widrow-Hoff LMS算法，它定义为

$$\delta\underline{W}(n) \triangleq \underline{W}(n)-\underline{W}(n-1)=-\mu\hat{\nabla}(n-1) \tag{3a}$$

$$\hat{\nabla}(n-1) \triangleq \nabla_W e^2(n-1)=-2e(n-1)\underline{X}(n-1) \tag{3b}$$

这种算法的收敛性主要取决于步长 μ(常数)的选取。为保证收敛，通常总是把 μ 选得很小(这也是减少权噪声的需要)，因而LMS法存在收敛慢的缺点。当然，LMS法也可用于要求收敛快(μ值取得较大)的场合，不过，这时μ值的选取往往很困难，有时甚至无法确定μ的适当值。

只是为了简化运算，(3)式在权修正量 δW 与瞬时误差面梯度之间人为地引入了一个时延。文献[1]提出把这个时延校正回来，即把(3)修改为

$$\delta\underline{W}(n)=-\mu\hat{\nabla}(n)=2\mu e(n)\underline{X}(n) \tag{4}$$

并称之为修正LMS算法(简称MLMS法)。显然，(4)式是收敛于维纳解的。通过对一阶($M=1$)滤波器的简单研究，文献[1]发现在阶跃输入时MLMS法的收敛是无条件的。但在向高阶推广时，他们遇到了计算上的困难，

没有能导出一种可实用的算法。本文完成了一般M阶MLMS算法递推公式的推导，并对这种算法作了比较全面的研究。

二．MLMS算法递推公式的推导

由(4)式，$$\underline{W}(n)=[I+2\mu\underline{X}(n)\underline{X}^T(n)]^{-1}[\underline{W}(n-1)+2\mu d(n)\underline{X}(n)] \tag{5}$$

式中，I为MxM么阵。利用矩阵求逆恒等式

$$[A+\alpha\underline{z}\underline{z}^T]^{-1}=A^{-1}-A^{-1}\underline{z}\underline{z}^TA^{-1}/(\alpha^{-1}+\underline{z}^TA^{-1}\underline{z}) \tag{6}$$

可求得 $$[I+2\mu\underline{X}(n)\underline{X}^T(n)]^{-1}=I-2\mu\underline{X}(n)\underline{X}^T(n)/(1+2\mu\underline{X}^T(n)\underline{X}(n)) \tag{7}$$

故有 $$\underline{W}(n)=\underline{W}(n-1)+\frac{2\mu}{1+2\mu\underline{X}^T(n)\underline{X}(n)}[d(n)-\underline{W}^T(n-1)\underline{X}(n)]\underline{X}(n) \tag{8a}$$

$$\triangleq\underline{W}(n-1)+G_M(n)e_M(n)\underline{X}(n) \tag{8b}$$

式中$e_M(n)\triangleq d(n)-\underline{W}^T(n-1)\underline{X}(n)$是用第n-1时刻的滤波器权系数对新到来的数据处理后的预测误差。而$G_M(n)\triangleq 2\mu/(1+2\mu\underline{X}^T(n)\underline{X}(n))$是可变增益，它随输入瞬时功率$\underline{X}^T(n)\underline{X}(n)$变化。输出误差仍由(1)定义。

在实际应用中，常用到的输出有两类。第一类是权系数$\underline{W}(n)$，第二类为误差$e(n)$(或$y(n)$)。由于(8)式中的权调整仅取决于$e_M(n)$而与$e(n)$无关，故对第一类情况，就可不计算$e(n)$而直接把$e_M(n)$看作误差输出，这样可使MLMS法的运算量与LMS法相比增加不多(只需多计算一次$G_M(n)$)。实际上对第二类情况，为简化运算，也可用$e_M(n)$作为输出，代价是加大了误差功率。本文把这种简化的方法称为MLMS2法，而把文[1]的MLMS算法称做MLMS1法。从(4)和(8)的信号流图容易看出，MLMS1滤波器更接近于理想的Howells-Appleboum自适应环[1]的数字化。而MLMS2与LMS的差别仅在于前者具有可变的增益$G_M(n)$。

三．收敛性

由于讨论μ值较大的情况，权系数均值的收敛是我们特别关心的。

由(8)得 $$\underline{W}(n)[1+2\mu\underline{X}^T(n)\underline{X}(n)]=\underline{W}(n-1)[1+2\mu\underline{X}^T(n)\underline{X}(n)]+2\mu[d(n)-\underline{W}^T(n-1)\underline{X}(n)]\underline{X}(n) \tag{9}$$

对上式两边取数学期望，设$\underline{W}(n-1)$与$\underline{X}(n)$统计相互，而$\underline{W}(n)$与$\underline{X}^T(n)\underline{X}(n)$一般是相关的，为简化分析，我们忽略这种相关性。这样有

$$E[\underline{W}(n)] = E[\underline{W}(n-1)] + 2\mu\{[\underline{\delta}x - R_x E[\underline{W}(n-1)]\}/E[1+2\mu\underline{X}^T(n)\underline{X}(n)] \quad (10)$$

由于 R_x 是对称的，故它与其特征值对角阵 $\Lambda = diag.[\lambda_1, \lambda_2, \cdots \lambda_M]$ 相似，即

$$R_x = Q\Lambda Q^{-1} = Q\Lambda Q^T \quad (11)$$

Q 是 R_x 的正交模态矩阵。简记 $E[\cdot]$ 为 $[\cdot]$，并注意到平均特征值

$$\lambda_{av} \triangleq \frac{1}{M}\sum_{i=1}^{M}\lambda_i = \frac{1}{M} tr R_x = \frac{1}{M} E[\underline{X}^T(n)\underline{X}(n)] \quad (12)$$

则有 $$\underline{W}(n) = Q[I - 2\mu\Lambda/(1+2\mu M\lambda_{av})]Q^{-1}\underline{W}(n-1) + 2\mu[\delta x/(1+2\mu M\lambda_{av}) \quad (13)$$

令
$$\left.\begin{aligned} \underline{W}'(n) &= Q^{-1}\underline{W}(n) = [\bar{W}_{1n}', \bar{W}_{2n}', \cdots, \bar{W}_{Mn}']^T \\ [\delta x &= Q^{-1}[\delta x = [r_1', r_2', \cdots r_M']^T \\ D &= I - 2\mu\Lambda/(1+2\mu M\lambda_{av}) \triangleq diag.[d_1, d_2, \cdots, d_M] \end{aligned}\right\} \quad (14)$$

(13)式变为 $$\underline{W}'(n) = D\underline{W}'(n-1) + 2\mu[\delta x/(1+2\mu M\lambda_{av})$$

即 $$\bar{W}_{in}' = d_i\bar{W}_{i(n-1)}' + 2\mu r_i'/(1+2\mu M\lambda_{av}),\ i=1,2,\cdots,M \quad (15)$$

上式可看成一个一阶全极点滤波器的状态方程。因此，MLMS 算法的收敛条件(即为全极点滤波器的稳定条件)为 $|d_i| = |1-2\mu\lambda_i/(1+2\mu M\lambda_{av})| < 1$

因为 $\lambda_{max} \geq \lambda_i \geq 0$，$i=1,2,\cdots M$，故收敛条件可简化为

$$|1 - 2\mu\lambda_{max}/(1+2\mu M\lambda_{av})| < 1 \quad (16)$$

考虑到 $\lambda_{max} \leq tr R_x = M\lambda_{av}$，就可看出，只要 $\mu > 0$，(16)式总是满足的。这就意味着 MLMS 算法是无条件收敛的。

我们仍然沿用文献[1]中的方法定义 MLMS 滤波器的自适应时常数为

$$\tau_i = \ell n^{-1}|(1+2\mu M\lambda_{av})/(1+2\mu(M\lambda_{av} - \lambda_i))| \quad (17)$$

MLMS算法的无条件收敛性也可从图2(单个极的情况)中清楚地看出。对LMS 法来说，随着 μ 的不同，$W(n+1)$ 的位置可以在 a 点左边的任何地方。如果在 0 点以左，就会出现所谓的"振铃现象"，甚至不收敛。而 MLMS 法则不同，由(4)式知，无论 μ 取何值，它都不可能一步到达 0 点(因这里神经为 0)，更不会到达 0 点之左(因这里神经为负)，因而它始终也不会发散，振铃现象也不会发生。图3 给出的是一种特定条件下的

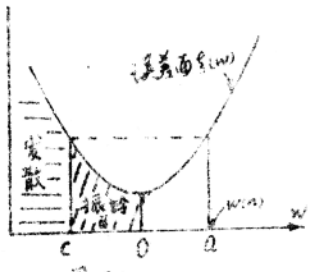


图2

自适应时常数 T 与步长 μ 的关系曲线也说明了这一点。

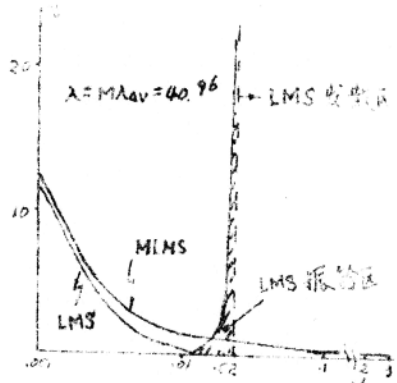


图3 时常数 T 一步长 μ 关系曲线

四、权噪声和输出均方误差

与 LMS 法一样，MLMS法也存在有权噪声的问题。如果作与文献[3]相同的假设，即设输入为异时不相关的高斯过程（注：这种假设通常很难满足，但结果一般还可用），可得考虑 μ 影响后的 LMS 权方差为

$$Vare[\underline{W}(n)] = [I-\mu\Lambda]^{-1}\mu\xi_{min} \tag{18}$$

式中

$$\xi_{min} = E[(d(n)-\underline{W}_{opt}^T\underline{X}(n))^2] \tag{19}$$

记权扰动为

$$\underline{V}(n) = \underline{W}(n)-\underline{W}_{opt} \tag{20}$$

则误调率为

$$M_e \triangleq E[\underline{V}^T(n)\underline{X}(n)\underline{X}^T(n)\underline{V}(n)]/\xi_{min} = \sum_{p=1}^{M}\mu\lambda_p/(1-\mu\lambda_p) \tag{21a}$$

$$\approx \mu tr R_x/(1-\mu tr R_x) \tag{21b}$$

后一近似式的条件为 R_x 中只有一个大特征值。总的输出均方误差为

$$\xi_e = E[e^2(n)] = [1+M_e]\xi_{min} \tag{22}$$

用类似的方法可求得MLMS法的权噪声方差和误调率为

$$Varme[\underline{W}(n)] = \mu\xi_{min}[I-\mu\Lambda/(1+2\mu M\lambda_{AV})]^{-1}/(1+2\mu M\lambda_{AV}) \tag{23}$$

$$M_{me} = \sum_{p=1}^{M}\mu\lambda_p/(1+2\mu M tr R_x-\mu\lambda_p) \approx \mu tr R_x/(1+\mu tr R_x) \tag{24}$$

比较(23)和(18)以及(24)和(21)可看到，在LMS算法中存在的权噪声对步长（即收敛速度）和输入噪声功率、误调率对收敛速度与滤波阶数及输入功率的强敏感性在MLMS法中都得到了一定的调和或减弱。应当指出，两种MLMS方法的稳态均方误差特性是不一样的。对于MLMS2法，与LMS相类似，有

$$\xi_{me2} = E[e_M^2(n)] = [1+M_{me}]\xi_{min} \tag{25}$$

对于MLMS1法，由于它是实时地朝着 $e^2(n)$ 减小的方向调整，追踪瞬时最佳点（$e^2(n)$ 的最小点）的，故由权噪声引起的误差功率增量始终不大于零，由此可证明，输入矢量 $X(n)$ 对于最佳权时的输出误差 $[d(n)-W_{opt}^TX(n)]$ 和由权扰动引起的输出 $X^T(n)V(n)$ 有强的相关性，若设它们完