

* 学术论文 *

并行流水线 LMS 算法及其性能分析*

尚 勇^{1,2**} 吴顺君² 项海格¹

1. 北京大学电子学系, 北京 100871; 2. 西安电子科技大学雷达信号处理重点实验室, 西安 710071

摘要 通过将并行处理方法引入流水线 LMS(PIPLMS)算法的设计, 提出一种新的并行 PIPLMS 算法. 与常规 PIPLMS 算法相比, 提出的算法具有更小的流水时延, 更高的数据吞吐率, 更快的收敛速度和更大的保证算法收敛的步长因子 μ 选择范围, 同时对具有相关性的输入具有一定的去相关作用, 特别适用于滤波器阶数高, 要求系统收敛速度快的应用场合. 也便于系统硬件设计.

关键词 并行处理 自适应滤波 并行流水线 LMS 算法 性能分析

目前对并行滤波的研究大多集中于变换域滤波技术^[1~3]. 而文献[4]提出了一种自适应滤波并行 LMS(PLMS)算法, 与常规 LMS 算法相比, PLMS 算法的数据吞吐率高, 收敛速度快, 保证算法收敛的步长因子取值范围比常规 LMS 算法大. 虽然 PLMS 算法具有许多优点, 但它与常规 LMS 算法一样, 算法本身都不具有流水线特性, 考虑其硬件实现时, 在权值更新和误差计算中存在着严重的计算瓶颈, 不能直接用于构造采用流水线技术和脉动阵技术的高速自适应滤波器结构. 为此, 人们对常规 LMS 算法进行了一些改进, 其中延时 LMS(DLMS)算法^[5,6]和 PIPLMS 算法^[7,8]都是为解决这一问题而提出的具有代表性的算法, 这些算法部分解决了计算瓶颈问题. 但收敛性却比 LMS 算法差. 本文的并行 PIPLMS 算法是以 PLMS 算法为基础, 结合常规 PIPLMS 算法而提出的. 与常规 PIPLMS 算法相比, 在相同条件下, 并行 PIPLMS 算法步长因子上限更大, 数据吞吐率高; 同时它也是研究并行流水线自适应滤波器高速实现结构的理论基础.

1 PIPLMS 算法

作为 DLMS 算法^[5,6]的一种合理推广, PIPLMS 算法采用了放松约束的预测技术. 它有多种放松条件可供选择, 这也导致了算法的不同收敛特性, 我们统称之为放松型预测技术. 它不要求保持严格的输入输出关系, 对硬件的节省量是很大的. 做为代价, 系统的收敛特性会变差.

下面给出基本的 PIPLMS 算法

$$\mathbf{h}_j = \mathbf{h}_{j-D_2} + \mu \sum_{k=0}^{L-1} e_{j-D_1-k} \mathbf{X}_{j-D_1-k}, \quad (1)$$

2000-12-12 收稿, 2001-02-23 收修改稿

* “九五”预研基金资助项目(批准号: 7.4.4.1)

** E-mail: shangyong@ele.pku.edu.cn

$$\begin{aligned} e_j &= d_j - \mathbf{h}_{j-1}^H \mathbf{X}_j \\ &= d_j - \left[\mathbf{h}_{j-D_2-1}^H + \mu \sum_{k=0}^{LA-1} e_{j-D_1-k-1} \mathbf{X}_{j-D_1-k-1}^H \right] \mathbf{X}_j, \end{aligned} \quad (2)$$

其中: $1 \leq LA \leq D_2$, 是算法中放松约束求和的项数. D_2 为权值预测的步长, D_1 为算法引入延时的量, j 为采样时刻, μ 为步长因子, \mathbf{h}_j , \mathbf{X}_j , e_j 和 d_j 分别为 j 时刻的权值矢量, 输入信号矢量, 误差信号和期望信号. H 表示矢量的共轭转置.

当 μ 值较小, 同时用 \mathbf{h}_{j-D_2} 代替 \mathbf{h}_{j-D_2-1} , 上式可近似写为

$$e_j = d_j - \mathbf{h}_{j-D_2}^H \mathbf{X}_j. \quad (3)$$

为了分析 PIPLMS 算法的性能, 可以考虑一些特殊情况. 文献[8]指出, 在 $LA = 1$, 同时约束 $D_1 = KD_2$, K 为正整数比例因子时, 在不同的近似条件下, 对应 PIPLMS 算法的步长因子 μ 的选择范围分别由下面各式确定

$$0 < \mu < \frac{2}{(\alpha M + 2K)\sigma^2}, \quad (4)$$

$$0 < \mu < \frac{\alpha M + 2K - \sqrt{(\alpha M + 2K)^2 - 8K(K+1)}}{2K(K+1)\sigma^2}, \quad (5)$$

$$\frac{2}{\sigma^2} - (\alpha M + 2K)\mu + K(K+1)\mu^2 - \frac{1}{3}K(K+1)(2K+1)\mu^3 > 0. \quad (6)$$

对应上面几种情况, 当系统达到稳态时, 其失调量 A 分别为

$$A = \frac{\alpha N b}{2 - (\alpha M + 2K)b}, \quad (7)$$

$$A = \frac{\alpha N b}{2 - (\alpha M + 2K)b + K(K+1)b^2}, \quad (8)$$

$$A = \frac{\alpha N b}{2 - (\alpha M + 2K)b + K(K+1)b^2 - \frac{1}{3}K(K+1)(2K+1)b^3}, \quad (9)$$

其中

$$\alpha = \frac{\lambda_{\text{rms}}^2}{\lambda_{\text{av}}^2}; \lambda_{\text{rms}}^2 = \frac{\left(\sum_{i=1}^N \lambda_i^2\right)}{N}; \lambda_{\text{av}} = \sigma^2 = \frac{\left(\sum_{i=1}^N \lambda_i\right)}{N}; M = N + v - 1. \quad (10)$$

这里 N 为自适应滤波器的长度; $\lambda_i (i = 1, \dots, N)$ 是输入信号自相关矩阵 \mathbf{R} 的第 i 个特征值; v 为旋转输入数据矢量 \mathbf{z}_j 的峰度; $\mathbf{z}_j = \mathbf{U}^H \mathbf{X}_j$; \mathbf{U} 是对矩阵 \mathbf{R} 进行酉变换的酉阵; 而 $b = \mu\sigma^2$.

2 并行 PIPLMS 算法

2.1 算法原理

下面结合同步滤波算法提出并行 PIPLMS 算法并进行性能分析. 对二输入二输出并行滤波结构^[9], 有如下输入输出关系

$$y_{0,j} = \mathbf{h}_{0,j}^H \mathbf{X}_{1,j} + \mathbf{h}_{1,j}^H \mathbf{X}_{0,j-1}, \quad (11)$$

$$y_{1,j} = \mathbf{h}_{0,j}^H \mathbf{X}_{1,j} + \mathbf{h}_{1,j}^H \mathbf{X}_{0,j}, \quad (12)$$

式中下标 0, 1 分别代表偶、奇支路, 而 $\mathbf{X}_{0,j} = [x_{2j}, x_{2j-2}, x_{2j-4}, \dots, x_{2j-N}]_{1 \times \frac{N}{2}}$, $\mathbf{X}_{1,j} = [x_{2j-1}, x_{2j-3}, x_{2j-5}, \dots, x_{2j-1-N}]_{1 \times \frac{N}{2}}$, N 为原型滤波器的长度.

对于(11)和(12)式, 由于有两条输出支路 $y_{0,j}$ 和 $y_{1,j}$, 计算误差时, 期望信号的选择与一般的滤波器不同. 为了容易得到并行自适应滤波的期望信号, 我们给出一种简单有效的选取方法, 即将对应原型滤波器的期望信号按奇、偶项分为两个序列, 分别当作对应 $y_{1,j}$ 支路和 $y_{0,j}$ 支路的期望信号.

下面以 $y_{1,j}$ 支路为例进行分析, 定义并行 PIPLMS 算法如下

$$\begin{bmatrix} \mathbf{h}_{0,j} \\ \mathbf{h}_{1,j} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_{0,j-D_2} \\ \mathbf{h}_{1,j-D_2} \end{bmatrix} + \mu \sum_{k=0}^{L-1} e_{1,j-D_1-k} \begin{bmatrix} \mathbf{X}_{1,j-D_1-k} \\ \mathbf{X}_{0,j-D_1-k} \end{bmatrix}, \quad (13)$$

$$\begin{aligned} e_{1,j} &= d_{1,j} - \mathbf{h}_{0,j-1}^H \mathbf{X}_{1,j} - \mathbf{h}_{1,j-1}^H \mathbf{X}_{0,j} \\ &= d_{1,j} - \left(\mathbf{h}_{0,j-D_2-1}^H + \mu \sum_{k=0}^{L-1} e_{1,j-D_1-k-1} \mathbf{X}_{1,j-D_1-k-1}^H \right) \mathbf{X}_{1,j} - \\ &\quad \left(\mathbf{h}_{1,j-D_2-1}^H + \mu \sum_{k=0}^{L-1} e_{1,j-D_1-k-1} \mathbf{X}_{0,j-D_1-k-1}^H \right) \mathbf{X}_{0,j}. \end{aligned} \quad (14)$$

考虑到一般 μ 值较小, 同时分别用 $\mathbf{h}_{0,j-D_2}$ 和 $\mathbf{h}_{1,j-D_2}$ 代替 $\mathbf{h}_{0,j-D_2-1}$ 和 $\mathbf{h}_{1,j-D_2-1}$. 上式可以简化为

$$e_{1,j} = d_{1,j} - \mathbf{h}_{0,j-D_2}^H \mathbf{X}_{1,j} - \mathbf{h}_{1,j-D_2}^H \mathbf{X}_{0,j}, \quad (15)$$

(13)和(15)式即为本文提出的并行 PIPLMS 算法的基本形式.

2.2 性能分析

作为一个特例, 下面分析当 $LA = 1, D_1 = KD_2$ 时的并行 PIPLMS 算法. 在这种情况下, 显然有

$$[E(e_{1,j}^2)]_{\min} = E(d_{1,j}^2) - \mathbf{h}_1^{*H} \mathbf{P}_{10} - \mathbf{h}_0^{*H} \mathbf{P}_{11}, \quad (16)$$

$$E(e_{1,j}^2) = [E(e_{1,j}^2)]_{\min} + \mathbf{V}_{0,j-D_2}^H \mathbf{R}_{11} \mathbf{V}_{0,j-D_2} + \mathbf{V}_{1,j-D_2}^H \mathbf{R}_{00} \mathbf{V}_{1,j-D_2}, \quad (17)$$

其中: \mathbf{h}_0^* 和 \mathbf{h}_1^* 分别为子滤波器 \mathbf{h}_0 和 \mathbf{h}_1 的 Wiener 解, 而

$$\mathbf{V}_{0,j-D_2} = \mathbf{h}_{0,j-D_2} - \mathbf{h}_0^*, \quad (18)$$

$$\mathbf{V}_{1,j-D_2} = \mathbf{h}_{1,j-D_2} - \mathbf{h}_1^*, \quad (19)$$

式中 $\mathbf{V}_{0,j}, \mathbf{V}_{1,j}$ 分别为 $\mathbf{h}_{0,j}, \mathbf{h}_{1,j}$ 的权值失调.

令: $\mathbf{V}'_{0,j} = \mathbf{Q}^{-1} \mathbf{V}_{0,j} = \mathbf{Q}_1^H \mathbf{V}_{0,j}, \mathbf{V}'_{1,j} = \mathbf{Q}_0^{-1} \mathbf{V}_{1,j} = \mathbf{Q}_0^H \mathbf{V}_{1,j}$ 其中: \mathbf{Q} 为对矩阵 \mathbf{R}_{00} 进行酉变换的酉阵, 而 \mathbf{Q}_1 为对矩阵 \mathbf{R}_{11} 进行酉变换的酉阵.

则

$$E(e_{1,j}^2) = [E(e_{1,j}^2)]_{\min} + \mathbf{V}'_{1,j-D_2}{}^H \Delta_{00} \mathbf{V}'_{1,j-D_2} + \mathbf{V}'_{0,j-D_2}{}^H \Delta_{11} \mathbf{V}'_{0,j-D_2}, \quad (20)$$

其中: Δ_{00}, Δ_{11} 均为对角阵. (17)式可重写为

$$E(e_{1,j}^2) = E(e_{1,j}^2)_{\min} + \epsilon_{j-D_2}, \quad (21)$$

其中稳态剩余误差为

$$\epsilon_{j-D_2} = E(\mathbf{V}'_{0,j-D_2}{}^H \Delta_{11} \mathbf{V}'_{0,j-D_2}) + E(\mathbf{V}'_{1,j-D_2}{}^H \Delta_{00} \mathbf{V}'_{1,j-D_2}). \quad (22)$$

令 $\mathbf{z}_{1,j} = \mathbf{Q}_1^H \mathbf{X}_{1,j}$, $\mathbf{z}_{0,j} = \mathbf{Q}_0^H \mathbf{X}_{0,j}$. 则(15)式可以进一步写为

$$\begin{aligned} e_{1,j} &= d_{1,j} - (\mathbf{V}_{1,j-D_2} + \mathbf{h}_1^*)^H \mathbf{X}_{0,j} - (\mathbf{V}_{0,j-D_2} + \mathbf{h}_h^*)^H \mathbf{X}_{1,j} \\ &= e_{1,j}^* - \mathbf{V}'_{1,j-D_2}{}^H \mathbf{z}_{0,j} - \mathbf{V}'_{0,j-D_2}{}^H \mathbf{z}_{1,j}, \end{aligned} \quad (23)$$

其中

$$e_{1,j}^* = d_{1,j} - \mathbf{h}_1^{*H} \mathbf{X}_{0,j} - \mathbf{h}_0^{*H} \mathbf{X}_{1,j}, \quad (24)$$

而

$$\mathbf{V}'_{0,j} = \mathbf{V}'_{0,j-D_2} + \mu e_{1,j-D_1} \mathbf{z}_{1,j-D_1}, \quad (25)$$

$$\mathbf{V}'_{1,j} = \mathbf{V}'_{1,j-D_2} + \mu e_{1,j-D_1} \mathbf{z}_{0,j-D_1}, \quad (26)$$

将(23)式代入(25),(26)式可得

$$\begin{aligned} \mathbf{V}'_{0,j} &= \mathbf{V}'_{0,j-D_2} + \mu (e_{1,j}^* - \mathbf{V}'_{1,j-D}{}^H \mathbf{z}_{0,j-D_1} - \mathbf{V}'_{0,j-D}{}^H \mathbf{z}_{1,j-D_1}) \mathbf{z}_{1,j-D_1} \\ &= \mathbf{V}'_{0,j-D_2} + \mu (e_{1,j}^* - \mathbf{z}_{0,j-D_1}^H \mathbf{V}'_{1,j-D} - \mathbf{z}_{1,j-D_1}^H \mathbf{V}'_{0,j-D}) \mathbf{z}_{1,j-D_1}, \end{aligned} \quad (27)$$

$$\begin{aligned} \mathbf{V}'_{1,j} &= \mathbf{V}'_{1,j-D_2} + \mu (e_{1,j}^* - \mathbf{V}'_{1,j-D}{}^H \mathbf{z}_{0,j-D_1} - \mathbf{V}'_{0,j-D}{}^H \mathbf{z}_{1,j-D_1}) \mathbf{z}_{0,j-D_1} \\ &= \mathbf{V}'_{1,j-D_2} + \mu (e_{1,j}^* - \mathbf{z}_{0,j-D_1}^H \mathbf{V}'_{1,j-D} - \mathbf{z}_{1,j-D_1}^H \mathbf{V}'_{0,j-D}) - \mathbf{z}_{0,j-D_1}, \end{aligned} \quad (28)$$

其中 $D = D_1 + D_2$.

考虑到(22)式的关系可得

$$\begin{aligned} \epsilon_j &= E \{ [\mathbf{V}'_{0,j-D_2} + \mu (e_{1,j}^* - \mathbf{z}_{0,j-D_1}^H \mathbf{V}'_{1,j-D} - \mathbf{z}_{1,j-D_1}^H \mathbf{V}'_{0,j-D}) \mathbf{z}_{1,j-D_1}]^H \Lambda_{11} \cdot \\ &\quad [\mathbf{V}'_{0,j-D_2} + \mu (e_{1,j}^* - \mathbf{z}_{0,j-D_1}^H \mathbf{V}'_{1,j-D} - \mathbf{z}_{1,j-D_1}^H \mathbf{V}'_{0,j-D}) \mathbf{z}_{1,j-D_1}] \} + \\ &\quad E \{ [\mathbf{V}'_{1,j-D_2} + \mu (e_{1,j}^* - \mathbf{z}_{0,j-D_1}^H \mathbf{V}'_{0,j-D} - \mathbf{z}_{0,j-D_1}^H \mathbf{V}'_{1,j-D}) \mathbf{z}_{0,j-D_1}]^H \Lambda_{00} \cdot \\ &\quad [\mathbf{V}'_{1,j-D_2} + \mu (e_{1,j}^* - \mathbf{z}_{0,j-D_1}^H \mathbf{V}'_{0,j-D} - \mathbf{z}_{0,j-D_1}^H \mathbf{V}'_{1,j-D}) \mathbf{z}_{0,j-D_1}] \}, \end{aligned} \quad (29)$$

对上式逐项展开后共有 32 项. 我们采用文献[10]中的假设, 即设序列 $\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_0, \mathbf{V}_1$ 及 e_1^* 是统计独立的, $E(\mathbf{X}_0) = 0, E(\mathbf{X}_1) = 0$. 且 \mathbf{X}_0 和 \mathbf{X}_1 的统计特性一样. 此时, $\mathbf{z}_0, \mathbf{z}_1, \mathbf{V}'_0, \mathbf{V}'_1, e_1^*$ 也是统计独立的. 而 $E(e_1^*) = 0$. 故(29)式中一些项为 0. 将这些等于 0 的项去掉后仍有 12 项. 将这些项分别展开, 可以得到 ϵ_j 的简化形式. 限于篇幅, 相应的分析从略, 下面直接给出 ϵ_j 的表示式.

当系统达到稳态时

$$\epsilon_j = \frac{Nb\alpha E(e_{1,j}^2)_{\min}}{2 - \left(\alpha M + 2K + \frac{N}{2}\alpha \right) b + K(K+1)b^2 - \dots}, \quad (30)$$

其中参数 α, M 和 b 的选取与常规 PIPLMS 算法不同, 区别在于这里与参数 α 和 b 有关的参数 λ_i 为矩阵 \mathbf{R}_{11} 的第 i 个特征值, 而 $M = \frac{N}{2} + v_1 - 1, v_1$ 为对应旋转输入数据矢量 $\mathbf{z}_{1,j}$ 的峰度.

此时可得在各种近似条件下保证算法收敛的并行 PIPLMS 算法步长因子 μ 的范围分别为

$$0 < \mu < \frac{2}{\left(\alpha M + 2K + \frac{N}{2}\alpha \right) \sigma^2}, \quad (31)$$

$$0 < \mu < \frac{\alpha M + 2K + \frac{N}{2}\alpha - \sqrt{\left(\alpha M + 2K + \frac{N}{2}\alpha\right)^2 + 8K(K+1)}}{2K(K+1)\sigma^2}, \quad (32)$$

$$\frac{2}{\sigma^2} - \left(\alpha M + 2K + \frac{N}{2}\alpha\right)\mu + K(K+1)\mu^2 - \frac{1}{3}K(K+1)(2K+1)\mu^3 > 0. \quad (33)$$

与之相应的失调量分别为

$$A = \frac{Nb\alpha}{2 - \left(\alpha M + 2K + \frac{N}{2}\alpha\right)b}, \quad (34)$$

$$A = \frac{Nb\alpha}{2 - \left(\alpha M + 2K + \frac{N}{2}\alpha\right)b + K(K+1)b^2}, \quad (35)$$

$$A = \frac{\alpha Nb}{2 - \left(\alpha M + 2K + \frac{N}{2}\alpha\right)b + K(K+1)b^2 - \frac{1}{3}K(K+1)(2K+1)b^3}. \quad (36)$$

当用并行滤波器结构的 $y_{0,j}$ 支路进行自适应滤波时,也可得到与上面分析相似的结论.

3 性能仿真

下面通过仿真实验比较并行 PIPLMS 算法和常规 PIPLMS 算法的收敛特性. 实验条件分别如下.

对应图 1: 输入信号为一组不相关的数据; 步长因子 $\mu = 0.01$; 对应常规 PIPLMS 算法, 设延时量 $D_1 = 22, D_2 = 2$; 对应并行 PIPLMS 算法, 各子滤波器长度为 11, 故此时延时量为 $D_1 = 11, D_2 = 2$.

对应图 2: 输入数据为一组不相关的数据通过系数为 $[1 \ 0.5]$ 的 MA 系统的输出. 步长因子 $\mu = 0.006$; 延时量与图 1 相同.

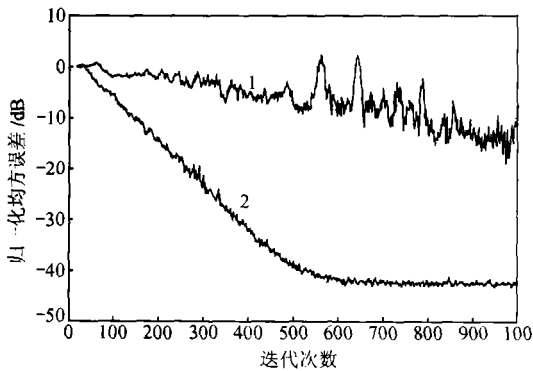


图 1 非相关输入情况下常规 PIPLMS 算法和并行 PIPLMS 算法收敛曲线

1 为常规 PIPLMS 算法收敛曲线; 2 为并行 PIPLMS 算法收敛曲线

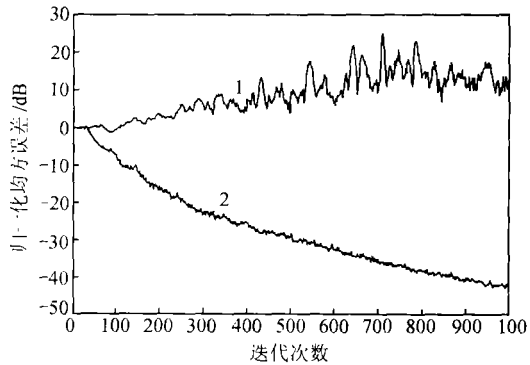


图 2 相关输入情况下常规 PIPLMS 算法和并行 PIPLMS 算法收敛曲线

1 为常规 PIPLMS 算法收敛曲线; 2 为并行 PIPLMS 算法收敛曲线

由图可见,并行 PIPLMS 算法和常规 PIPLMS 算法相比,在算法收敛的情况下,并行 PIPLMS 算法的收敛速度比常规 PIPLMS 算法快。而当步长因子 μ 进一步增大时,当常规 PIPLMS 算法已不能收敛时,并行 PIPLMS 算法仍能很好地收敛。换句话说,在保证算法收敛的前提下,并行 PIPLMS 算法具有更宽的 μ 值选择范围。通过上面的仿真实验也可以看到实验结果与前面的理论分析一致。

4 结论

如何将高性能的算法应用到实际工程中的研究已越来越受到人们普遍关注。这其中,关于高速并行自适应滤波算法及其实现结构的研究更是信号处理领域的一个研究重点和热点。本文通过将并行处理方法引入 PIPLMS 算法的设计,提出了一种新的并行 PIPLMS 算法,并分析了该算法的性能。计算机仿真结果比较了并行 PIPLMS 算法和常规 PIPLMS 算法。通过理论分析和仿真比较发现,并行 PIPLMS 算法比常规 PIPLMS 算法具有更快的收敛速度;更宽的 μ 值选择范围;其数据吞吐率也比常规 PIPLMS 算法大;同时并行 PIPLMS 算法是可以直接映射到高度流水并行自适应滤波器结构的自适应算法。因此对其研究具有比较重要的理论和工程实践意义。

参 考 文 献

- 1 Fiore P D. Low-complexity implementation of a polyphase filter bank. *Digital Signal Processing*, 1998, 8: 126
- 2 Berberidis K, et al. A new fast block adaptive algorithm. *IEEE Trans on Signal Processing*. 1999, 47(1): 75
- 3 Haykin S. *Adaptive Filter Theory*. New Jersey: Prentice Hall, Inc, 1996. 445 ~ 482
- 4 Shang Y, et al. Design of parallel adaptive FIR filters. *IEEE APCCAS'98*, Chiangmai, Thailand, 1998: 81
- 5 Meyer M D, et al. A high sampling rate delayed LMS filter architecture. *IEEE Trans on Circuits and Systems- II*, 1993, 40(11): 727
- 6 Herzberg H, et al. A systolic array realization of an LMS adaptive filter and the effects of delayed adaptation. *IEEE Trans on Signal Processing*, 1992, 40(11): 2799
- 7 Shanbhag N R, et al. Low-power adaptive filter architectures and their application to 51.84 Mb/s ATM-LAN. *IEEE Trans on Signal Processing*, 1997, 45(5): 1276
- 8 Shanbhag N R, et al. Relaxed look-ahead pipelined LMS adaptive filters and their application to ADPCM coder. *IEEE Trans on Circuits and Systems- II*, 1993, 40(12): 753
- 9 Parker D A, et al. Low-area/power parallel FIR digital filter implementations. *Journal of VLSI Signal Processing*, 1997, 17: 75
- 10 Long G Z, et al. The LMS algorithm with delayed coefficient adaptation. *IEEE Trans on Signal Processing*, 1989, 37(9): 1397