

多路 IIR 自适应滤波和高阶 IIR 自适应滤波的实现*

杨大成 王德隽
(北京邮电学院,北京)

摘要 本文提出了一种可供在多路 IIR 或者高阶 IIR 情况下进行联合过程估值的超稳定自适应算法。

关键词 自适应滤波, IIR 自适应滤波器, 超稳定自适应算法

一、引言

自适应滤波的方法在许多领域中都有着广泛的应用。迄今为止,人们对于 FIR 结构的自适应滤波器已经有了比较透彻的和多方面的研究^[1]。FIR 结构具有许多优点,其中最突出的优点是,这种结构的均方误差或最小平方误差曲面具有单一的全局最小点。这给基于梯度搜寻技术的自适应算法带来了很大好处。但是, FIR 结构的主要缺点是难以处理未知系统含有极点的情况。在实际应用中,对于含有极点的情况往往都是靠增加 FIR 滤波器的阶次来尽量逼近。显然,阶次的增加将使计算负担加重。

为了解决对具有极点的系统的自适应问题,需要研究 IIR 型或零极点型的自适应滤波问题。最初,有人仿照 FIR 梯度搜寻的思想,提出过 IIR 的梯度法^[2]。但是,通过对 IIR 结构的均方误差曲面或平方误差曲面的分析表明,这种结构所产生的误差曲面具有多个局部最小点而不具有全局最小点^[3,4]。这就使得基于梯度搜寻的自适应算法失去了效力。后来,有人根据控制论中的超稳定定理^[5],提出了超稳定自适应滤波器 (HARF) 及其简化形式 (SHARF)^[6,7]。这种自适应滤波器虽然从理论上讲要受到严格正实条件的限制,但由于这是一个充分条件,因此对于一些实际应用来说是可行的。

有两种情况是已有的超稳定自适应滤波器尚难以处理的。第一,当被估值的理想输出序列是由高阶的零极点结构产生的时候,众所周知,在自适应过程中,每迭代一次就需要计算一次自适应滤波器的输出,而这种计算实际上可视为滤波器的一次实现。理论分析已经表明,直接型的实现在高阶 IIR 的情况下误差性能最差^[8]。但是,若把一个高阶的零极点型结构化解为低阶的并联结构,则误差性能将大为改善。第二,在许多应用中,理想输出序列有可能是由一个具有多路输入的系统产生的。对于这种涉及多路输入的自适应问题, HARF 或 SHARF 算法都无法解决。

* 1987年10月9日收到,1988年7月4日修改定稿。

为了解决以上两个问题,本文根据超稳定原理,导出了一种适用于多路和并联型高阶零极点结构的自适应滤波器算法,并且就这两种情况给出几个计算机模拟结果。

二、系统模型

一个具有多路输入、单路输出的系统模型如图 1 所示。图中的系统由 s 路组成,第 i 路的输入为 $u_i(n)$ 。未知系统和自适应系统的输出分别为 $d_i(n)$ 和 $y_i(n)$ 。两个系统的总输出分别为 $d(n)$ 和 $y(n)$ 。我们的目的是希望通过调节自适应系统的参数使 $y(n)$ 逼近 $d(n)$,从而达到自适应联合过程估值的目的。在实际应用中,各路输出 $d_i(n)$ 一般是无法知道的。在此我们假定,各路滤波器都是零极点型的。

如果我们令各路的输入都相等,即

$$u_1(n) = u_2(n) = \cdots = u_s(n) \quad (1)$$

则图 1 所示的多路系统模型将化为图 2 所示的用并联结构实现的高阶零极点型滤波器。也就是说,图 2 中的高阶并联结构,是图 1 中多路输入结构在各路输入都相等情况下的一个特例。因此,下面我们将就一般的多路输入结构进行推导。

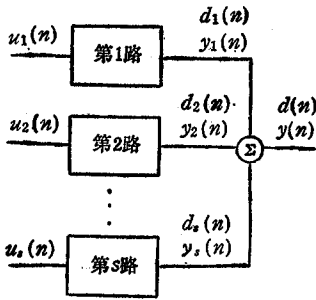


图 1 多路输入系统模型

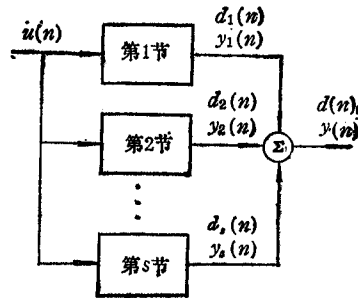


图 2 并联高阶零极点系统模型

假定第 i 路未知系统的输入输出关系为

$$d_i(n) = \sum_{j=1}^{N_i} a_{ij}d_i(n-j) + \sum_{j=0}^{M_i} b_{ij}u_i(n-j) \triangleq \mathbf{F}_i^T \mathbf{x}_i(n) \quad (2)$$

式中, $i = 1, 2, \dots, s$, 且上角标 T 表示转置,

$$\mathbf{F}_i = [a_{i1} \cdots a_{iN_i} b_{i0} \cdots b_{iM_i}]^T \quad (3)$$

$$\mathbf{x}_i(n) = [d_i(n-1) \cdots d_i(n-N_i) u_i(n) \cdots u_i(n-M_i)]^T \quad (4)$$

易知

$$d(n) = \sum_{i=1}^s d_i(n) \triangleq \mathbf{F}^T \mathbf{x}(n) \quad (5)$$

式中,

$$\mathbf{F} = [\mathbf{F}_1^T \mathbf{F}_2^T \cdots \mathbf{F}_s^T]^T \quad (6)$$

$$\mathbf{x}(n) = [\mathbf{x}_1^T(n) \mathbf{x}_2^T(n) \cdots \mathbf{x}_s^T(n)]^T \quad (7)$$

类似地,可以写出自适应系统的输入输出关系式。第 i 路的关系可以写为

$$y_i(n) = \mathbf{F}_i^T(n)\boldsymbol{\phi}_i(n) \quad (8)$$

式中, $i = 1, 2, \dots, s$, 且

$$\mathbf{F}_i(n) = [a_{i1}(n) \cdots a_{iN_i}(n)b_{i0}(n) \cdots b_{iM_i}(n)]^T \quad (9)$$

$$\boldsymbol{\phi}_i(n) = [z_i(n-1) \cdots z_i(n-N_i)u_i(n) \cdots u_i(n-M_i)]^T \quad (10)$$

其中的 $z_i(n)$ 定义为

$$z_i(n) = \mathbf{F}_i^T(n+1)\boldsymbol{\phi}_i(n) \quad (11)$$

自适应系统的总输出表示为

$$y(n) = \sum_{i=1}^s y_i(n) = \mathbf{F}^T(n)\boldsymbol{\phi}(n) \quad (12)$$

式中,

$$\mathbf{F}(n) = [\mathbf{F}_1^T(n)\mathbf{F}_2^T(n) \cdots \mathbf{F}_s^T(n)]^T \quad (13)$$

$$\boldsymbol{\phi}(n) = [\boldsymbol{\phi}_1^T(n)\boldsymbol{\phi}_2^T(n) \cdots \boldsymbol{\phi}_s^T(n)]^T \quad (14)$$

以上就是我们考虑的基本的系统模型. 需要指出的是, 与 FIR 结构不同, 在 IIR 结构的情况下, 自适应部分的输出 $y(n)$ 不是直接由输入和 $y(n)$ 过去的样值产生, 而要经过一个中间环节先求出 $z(n)$ 然后再产生. 这是由于需要利用超稳定原理的缘故.

三、算法推导

超稳定定理 图 3 所示的无激励、非线性时变反馈系统是超稳定的, 如果

(1) 线性前馈部分的传递函数 $H(Z)$ 稳定并且是严格正实的, 即

$$\operatorname{Re}\{H(z)\} > 0, \quad \forall |z| = 1 \quad (15)$$

(2) 非线性时变反馈部分的输入 $v(n)$ 和输出 $\bar{w}(n) = -\omega(n)$ 遵从关系式

$$\sum_{n=0}^{n_1} \bar{w}(n)v(n) \geq -\xi_0^2, \quad n_1 > 0 \quad (16)$$

当系统超稳定时, 随着 $n \rightarrow \infty$, $v(n) \rightarrow 0$.

超稳定自适应滤波器的基本思想是设法把自适应滤波器的输出与理想输出之间的误差表示为 $v(n)$, 然后设法构造一个图 3 所示的非线性时变反馈系统, 并且证明这个系统满足超稳定定理.

为此, 首先定义 $v(n)$ 为一个动均误差

$$v(n) = d(n) - z(n) + \sum_{j=1}^p c_j [d(n-j) - z(n-j)] \quad (17)$$

式中, $d(n)$ 由 (5) 式定义,

$$z(n) = \mathbf{F}^T(n+1)\boldsymbol{\phi}(n) \quad (18)$$

于是我们得到如下的定理.

定理 1 在多路输入的零极点型自适应滤波器中, 若每一路都满足超稳定定理, 则当 $n \rightarrow \infty$ 时, $y(n) \rightarrow d(n)$.

证明 (17) 式显然可以写为

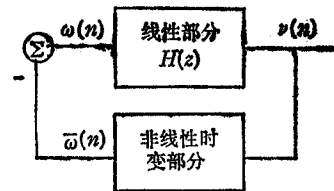


图 3 无激励非线性时变反馈系统模型

$$v(n) = \sum_{i=1}^s v_i(n) \quad (19)$$

式中,

$$v_i(n) = d_i(n) - z_i(n) + \sum_{j=1}^p c_j [d_i(n-j) - z_i(n-j)] \quad (20)$$

并且它仅与第 i 路滤波器有关.

文献[6]已证明, 对于一个单路输入自适应滤波器, 若它满足超稳定定理, 则当 $n \rightarrow \infty$ 时, $v_i(n) \rightarrow 0$, 并且 $y_i(n) \rightarrow d_i(n)$. 利用这一点和前面的关系容易证明, 若每一路都满足超稳定定理, 则必有

$$\lim_{n \rightarrow \infty} [y(n) - d(n)] = \sum_{i=1}^s \lim_{n \rightarrow \infty} [y_i(n) - d_i(n)] = 0 \quad (21)$$

因此, 只要找到一个系数更新公式, 使每一路都是超稳定的, 则总的估值误差将是收敛的.

定理 2 (多路 HARF 算法——MCHARF 算法) 如果第 i 路滤波器的系数更新公式为

$$F_i(n+1) = F_i(n) + \frac{\Gamma_i \phi_i(n) \gamma(n)}{1 + \phi_i^T(n) \Gamma_i \phi_i(n)}, \quad i = 1, 2, \dots, s \quad (22)$$

则当 $n \rightarrow \infty$ 时, $y_i(n) \rightarrow d_i(n)$, 式中,

$$\Gamma_i = \text{diag}[\alpha_i^{(1)} \alpha_i^{(2)} \dots \alpha_{N_i}^{(i)} \beta_0^{(i)} \beta_i^{(1)} \dots \beta_{M_i}^{(i)}] \quad (23)$$

$$\gamma(n) = \sum_{i=1}^s \gamma_i(n) \quad (24)$$

$$\gamma_i(n) = d_i(n) - y_i(n) + \sum_{j=1}^p c_j [d_i(n-j) - z_i(n-j)] \quad (25)$$

其中, α 和 β 为步长; c_j 是平滑系数, 它的选择应使每一个

$$H_i(Z) = \left(1 + \sum_{j=1}^p c_j Z^{-j}\right) / \left(1 - \sum_{j=1}^{N_i} a_{ij} Z^{-j}\right), \quad i = 1, 2, \dots, s \quad (26)$$

均满足严格正实条件. 在此需要注意, 由于(25)式中的 $d_i(n-j)$ 一般无法得到, 因此各路的平滑系数不能独立选取. 不过, 模拟结果表明, 在很多情况下, 平滑系数也可以不用.

(23) 式中的 $\alpha_j^{(i)}$ 和 $\beta_j^{(i)}$, $j = 1, 2, \dots, N_i$ 和 M_i 分别为前馈和反馈部分的步长因子, 一般它们的取值都是较小的正实数. 在实际应用中, 为方便起见, 所有的 $\alpha_j^{(i)}$ 可取一个值, $\beta_j^{(i)}$ 也是如此. 下面证明这个定理.

证明 首先定义

$$e_i(n) = d_i(n) - z_i(n), \quad i = 1, 2, \dots, s \quad (27)$$

$$e(n) = \sum_{i=1}^s e_i(n) \quad (28)$$

于是有

$$v_i(n) = e_i(n) + \sum_{i=1}^p c_i e_i(n-i) \quad (29)$$

$$v(n) = e(n) + \sum_{i=1}^p c_i e(n-i) \quad (30)$$

$$r_i(n) = d_i(n) - y_i(n) + \sum_{i=1}^p c_i e_i(n-i) \quad (31)$$

$$r(n) = d(n) - y(n) + \sum_{i=1}^p c_i e(n-i) \quad (32)$$

$e_i(n)$ 显然可以写为

$$e_i(n) = \sum_{i=1}^{N_i} a_i e_i(n-i) + w_i(n) \quad (33)$$

式中,

$$w_i(n) = [\mathbf{F}_i - \mathbf{F}_i(n+1)]^T \boldsymbol{\phi}_i(n) = -\bar{w}_i(n) \quad (34)$$

将每一路的 $w_i(n)$ 和 $v_i(n)$ 都构成一个图 3 所示的闭环系统,再利用文献[6]中的方法,通过适当选择每一路的平滑系数的个数和数值,可以使每一路对应的前馈传递函数 $H_i(Z) = V_i(Z)/W_i(Z)$ 满足严格正实条件。在此, $V_i(Z)$ 和 $W_i(Z)$ 分别为 $v_i(n)$ 和 $w_i(n)$ 的 Z 变换。下面证明,利用(22)式可以使每一路均满足(16)式,为此,还需要首先定义以下各个量。

$$\tilde{r}_i(n) = r(n) - r_i(n) = \sum_{i=1}^i r_i(n) \quad (35)$$

$$\tilde{v}_i(n) = v(n) - v_i(n) = \sum_{i=1}^i v_i(n) \quad (36)$$

$$v'_i(n) = v_i(n) + \tilde{r}_i(n) \quad (37)$$

$$\tilde{\mathbf{F}}_i(n) = \mathbf{F}_i(n) - \mathbf{F}_i \quad (38)$$

利用(22), (34) 和以上各式,得到

$$\bar{w}_i(n) = \tilde{\mathbf{F}}_i^T(n+1) \boldsymbol{\phi}_i(n) \quad (39)$$

$$\tilde{\mathbf{F}}_i(n+1) = \tilde{\mathbf{F}}_i(n) + \Gamma_i \boldsymbol{\phi}_i(n) v'_i(n) \quad (40)$$

利用以上二式,可以得到

$$\bar{w}_i(n) v_i(n) = \tilde{\mathbf{F}}_i^T(n) \boldsymbol{\phi}_i(n) v_i(n) + \boldsymbol{\phi}_i^T(n) \Gamma_i \boldsymbol{\phi}_i(n) v'_i(n) v_i(n) \quad (41)$$

经过适当简化后有

$$\begin{aligned} \bar{w}_i(n) v_i(n) = & -\boldsymbol{\phi}_i^T(n) \tilde{\mathbf{F}}_i(n) \tilde{r}_i(n) + \frac{1}{2} [f_i(n+1) - f_i(n)] \\ & + \sigma_i^2(n) - \boldsymbol{\phi}_i^T(n) \Gamma_i \boldsymbol{\phi}_i(n) v'_i(n) \tilde{r}_i(n) \end{aligned} \quad (42)$$

式中,

$$f_i(n) = \tilde{\mathbf{F}}_i^T(n) \Gamma_i^{-1} \tilde{\mathbf{F}}_i(n) \quad (43)$$

$$\sigma_i^2(n) = \frac{1}{2} \boldsymbol{\phi}_i^T(n) \Gamma_i \boldsymbol{\phi}_i(n) v_i^2(n) \quad (44)$$

显然, $f_i(n)$ 和 $\sigma_i^2(n)$ 都是非负的. 利用(42)式可以得到

$$\begin{aligned} \bar{w}_i(n)v_i(n) \geq & -|\phi_i^T(n)\tilde{F}_i(n)\tilde{r}_i(n)| + \frac{1}{2}[f_i(n+1) - f_i(n)] \\ & + \sigma_i^2(n) - \phi_i^T(n)\Gamma_i\phi_i(n)|v_i(n)\tilde{r}_i(n)| \end{aligned} \quad (45)$$

对上式两端从 0 至 n_1 求和, 得到

$$\sum_{n=0}^{n_1} \bar{w}_i(n)v_i(n) \geq \frac{1}{2}f_i(n_1+1) + \sum_{n=0}^{n_1} \sigma_i^2(n) - \xi_{i0}^2 \quad (46)$$

式中,

$$\xi_{i0}^2 = \sum_{n=0}^{n_1} [|\phi_i^T(n)\tilde{F}_i(n)\tilde{r}_i(n)| + \phi_i^T(n)\Gamma_i\phi_i(n)|v_i(n)\tilde{r}_i(n)|] + \frac{1}{2}f_i(0) \quad (47)$$

显然, ξ_{i0}^2 是 n_1+2 个非负实数之和, 因此它必定也是非负的. 又由于 $f_i(n_1+1)$ 和 $\sigma_i^2(n)$ 也都是非负的, 因此我们最后得到

$$\sum_{n=0}^{n_1} \bar{w}_i(n)v_i(n) \geq -\xi_{i0}^2 \quad (48)$$

这样我们就证明了, 当各路自适应滤波器的系数按(22)式更新时, 通过适当选择每一路的平滑系数, 可以使超稳定原理得到满足, 从而使得当 $n \rightarrow \infty$ 时, $y_i(n) \rightarrow d_i(n)$.

我们注意到, 多路情况和单路情况的主要差别就在于(22)式系数更新式中的 $r(n)$. 也就是说, 在多路情况下, 每一路系数的更新不仅要受其本身的影响, 而且还要受到所有其它各路的影响.

应注意到, 在以上各步证明过程中, 并未对各路输入信号有什么特殊的限制, 因此, 所得到的结论完全适用于并联结构的高阶零极点型滤波器.

四、计算机模拟结果

我们用超稳定自适应算法对并联高阶结构和多路输入结构进行了大量的计算机模拟. 在模拟实验中, 输入序列采用均值为零的高斯白噪声序列. 为简化计, 高阶结构的每一个并环节和多路结构的每一路都采用有两个极点和一个零点的二阶节, 并且所有二阶节的步长因子都取为相同的数值. 所有的模拟实验都没有使用平滑系数, 因为在实际应用中无法选取平滑系数. 下面介绍几个例子.

例 1 假定理想输出序列 $d(n)$ 是由一个四阶的 IIR 系统产生, 其传递函数为

$$G_1(Z) = \frac{1 + 0.28Z^{-1}}{1 - 1.6Z^{-1} + 0.7076Z^{-2}} + \frac{1 + 0.8Z^{-1}}{1 - 0.2Z^{-1} + 0.641Z^{-2}}$$

当不选择平滑系数时, 易知整个 $G_1(Z)$ 以及它的第一个二阶节都不满足严格正实条件. 图 4 为这个例子的模拟结果. 横坐标为迭代次数, 纵坐标为误差的 MSE 值的 dB 值, 定义为

$$\text{MSE} = 10 \lg\{E[d(n) - y(n)]^2\} \quad (\text{dB}) \quad (49)$$

系数的初始值随机地选取, 本例的初始值选为 [0.1, -0.4, 1.0, 0.6, 1.2, -0.4, 1.0, 0.15]. 值得注意的是, 当我们采用并联二阶节模型时, 每一个二阶节中分子的第一个系

数总是 1, 而分母的第一个系数 (常数 1) 不属于自适应范围 (参见前面的定义式), 因此它们都不需要进行调节。另外, 图 4 所示的三条曲线是在三种步长情况下的模拟结果。

从这个例子可以看出, 步长的变化对算法的收敛速度有很大的影响, 尽管平稳后的 MSE 值很接近。另外这个例子还表明, 和常规的单路输入超稳定 IIR 滤波器一样, 在本文提出的算法中, 严格正实条件也仅是一个充分条件, 而不是一个必要条件。

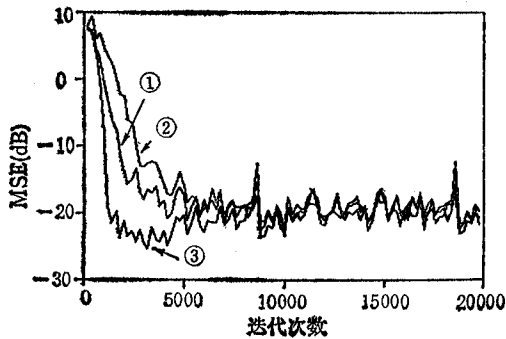


图 4 例 1 的模拟结果

- ① $\alpha_i^{(j)} = \alpha = \beta_i^{(j)} = \beta = 0.001$
- ② $\alpha_i^{(j)} = \alpha = \beta_i^{(j)} = \beta = 0.0005$
- ③ $\alpha_i^{(j)} = \alpha = 0.0015, \beta_i^{(j)} = \beta = 0.0005$

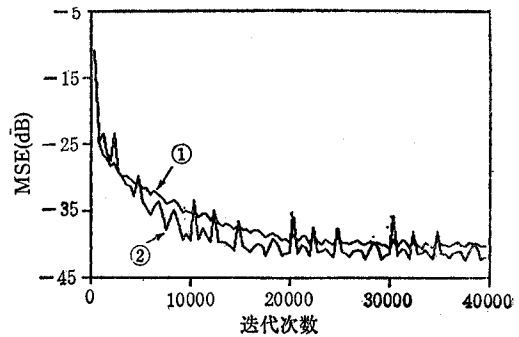


图 5 例 2 的模拟结果

例 2 假定理想输出序列 $d(n)$ 是由一个十阶的 IIR 系统产生, 其传递函数为

$$G_2(Z) = \frac{1 + 0.4Z^{-1}}{1 - 0.2Z^{-1} + 0.2Z^{-2}} + \frac{1 - 0.2Z^{-1}}{1 + 0.6Z^{-1} + 0.18Z^{-2}} \\ + \frac{1 - 0.1Z^{-1}}{1 - 1.0Z^{-1} + 0.5Z^{-2}} + \frac{1 + 0.3Z^{-1}}{1 + 0.8Z^{-1} + 0.32Z^{-2}} \\ + \frac{1 - 0.4Z^{-1}}{1 - 0.5Z^{-1} + 0.1225Z^{-2}}$$

这个例子进行的两次模拟都选用同一组初始值 $[0.1, -0.1, 1.0, 0.7, -0.3, -0.4, 1.0, -0.5, 0.8, -0.8, 1.0, -0.3, -0.4, -0.6, 1.0, 0.1, 0.302, -0.222, 1.0, -0.22]$ 。模拟结果示于图 5。曲线 ① 选用的步长是 $\alpha = \beta = 0.05$, 曲线 ② 所选用的步长是 $\alpha = \beta = 0.02$ 。从这个例子我们看到, 步长的加大有助于加快收敛速度, 但是平稳性较差。

例 3 假定理想输出序列 $d(n)$ 是由一个两路输入的 IIR 系统产生, 传递函数分别为

$$G_{31}(Z) = \frac{1 - 0.1Z^{-1}}{1 - 0.8Z^{-1} + 0.4304Z^{-2}}, \quad G_{32}(Z) = \frac{1 - 0.55Z^{-1}}{1 + 0.22Z^{-1} + 0.17Z^{-2}}$$

模拟所用的初值为 $[1.2, -0.85, 1.0, -0.2, -0.6, -0.45, 1.0, -0.75]$ 。模拟结果如图 6 所示。从这个例子可看出, 对于多路输入模型, 步长的选择对收敛性能同样有较大的影响。

例 4 假定理想输出序列 $d(n)$ 是由一个四路输入的 IIR 系统产生, 传递函数分别为

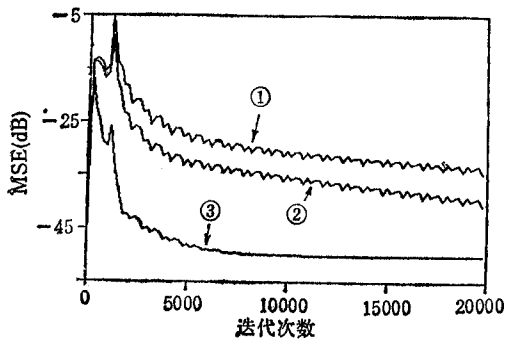


图 6 例 3 的模拟结果

- ① $\alpha = \beta = 0.005$
- ② $\alpha = \beta = 0.01$
- ③ $\alpha = \beta = 0.02$

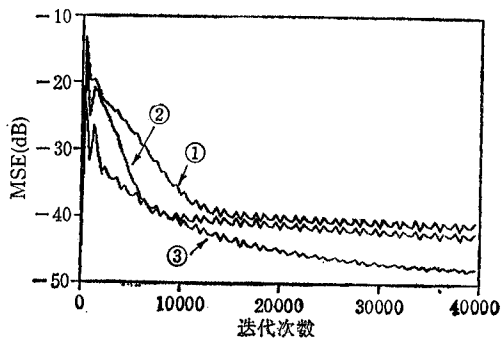


图 7 例 4 的模拟结果

- ① $\alpha = \beta = 0.05$
- ② $\alpha = \beta = 0.01$
- ③ $\alpha = \beta = 0.1$

$$G_{41}(Z) = \frac{1 + 0.38Z^{-1}}{1 - 0.6Z^{-1} + 0.25Z^{-2}}, \quad G_{42}(Z) = \frac{1 + 0.32Z^{-1}}{1 - 0.8Z^{-1} + 0.25Z^{-2}}$$

$$G_{43}(Z) = \frac{1 - 0.38Z^{-1}}{1 + 0.6Z^{-1} + 0.25Z^{-2}}, \quad G_{44}(Z) = \frac{1 - 0.32Z^{-1}}{1 + 0.8Z^{-1} + 0.25Z^{-2}}$$

模拟时所用的初值为 $[0.4, -0.15, 1.0, 0.5, 0.4, -0.35, 1.0, 0.5, -0.8, -0.4, 1.0, -0.6, -0.4, -0.4, 1.0, -0.2]$ 。模拟结果如图 7 所示。

以上四个例子我们都迭代了充分多的次数，但是并没有出现算法收敛以后又重新发散的现象。这表明算法本身是稳定的。另外，通过以上几个例子我们可以看出，步长的选择对算法有很大的影响，而且一般的规律仍然是较大的步长产生较快的收敛速度。其实，初值的选择对算法的性能也同样会有很大影响，这已为大量的模拟结果所证实。不过到目前为止还未找到一个有效的选择初值的理论公式或经验公式。不过，在非实时处理的条件下，可以通过选择不同的初值来获得较好的识别性能。同样，步长的选择在本文中也是试探性的。因此，如何选择有效的初值和步长问题，以及各路之间相互影响的问题，都是本文所提的算法在今后仍需继续探索的课题。

参 考 文 献

- [1] M. L. Honig, D. G. Messerschmitt, Adaptive Filters, Kluwer Academia Publishers, 1984.
- [2] P. L. Feintuch, *Proc. IEEE*, 64(1976)11, 1622—1624.
- [3] S. D. Stearns, *IEEE Trans. on ASSP*, ASSP 29(1981)3, 763—766.
- [4] C. R. Johnson, Jr., *IEEE Trans. on IT*, IT-30(1984)2, 237—250.
- [5] C. R. Johnson, Jr., *Hyperstability of Control Systems*, Berlin: Springer—Verlag, 1973.
- [6] C. R. Johnson, Jr., *IEEE Trans. on IT*, IT-25(1979)6, 745—749.
- [7] M. G. Larimore, et al., *IEEE Trans. on ASSP*, ASSP-28(1980)4, 428—440.
- [8] A. V. Oppenheim, R. W. Schaffer, *Digital signal Processing*, Englewood Cliffs, NJ: Prentice-Hall, 1975.

REALIZATION OF MULTICHANNEL AND HIGH-ORDER IIR ADAPTIVE FILTERS

Yang Dacheng Wang Dejung

(Beijing University of Posts and Telecommunications, Beijing)

Abstract It is developed that a new hyperstable adaptive algorithm which can be used in the multichannel IIR configuration and/or the high-order IIR configuration for joint process estimation.

Key words Adaptive filtering; IIR adaptive filter; Hyperstable adaptive algorithm