

一种基于二阶梯度估计的自适应算法

胡瑞敏 姚天任 黄铁侠

(华中理工大学电子与信息工程系, 武汉 430074)

摘 要

在自适应信号处理中得到广泛应用的 LMS 算法, 对信号模型及特性有着极其严格的限制, 这些限制在很多实际情况中并不能保证得到满足. 相对 LMS 算法, 基于中心差的梯度估计自适应算法, 其适应面则要广泛得多. 但是, 该算法存在着收敛速度慢, 所需采样点数多的缺点. 为此本文提出一种适应于平稳情况的新的估计算法, 除首次估计需做采样外, 在收敛过程中无需再做采样. 与传统的中心差算法相比, 本文算法具有较快的收敛速度和较好的失调性能.

关键词: 自适应信号处理, 自适应滤波, 语音处理, 神经网络.

一、引 言

在自适应信号处理领域中得到广泛应用的自适应线性组合器模型如图 1 所示.

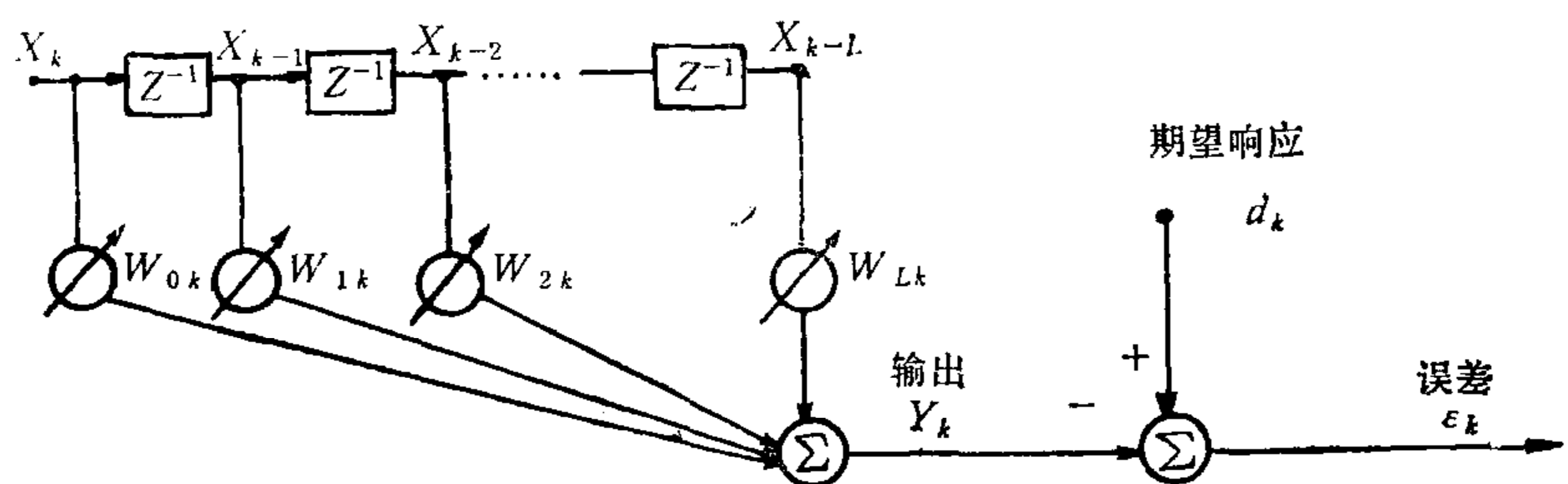


图 1 自适应线性组合器模型

图中输入矢量为

$$x_k = [x_k x_{k-1} \cdots x_{k-L}]^T, \quad (1)$$

输入输出关系为

$$y_k = \sum_{i=0}^L w_{ik} x_{k-i}, \quad (2)$$

设权矢量为

$$w_k = [w_{0k} w_{1k} \cdots w_{Lk}]^T, \quad (3)$$

则输入输出关系可以简单地用矢量形式表示为

$$y_k = x_k^T w_k = w_k^T x_k. \quad (4)$$

图中 d_k 为训练信号, d_k 与 y_k 的差称为误差信号, 即

$$\varepsilon_k = d_k - y_k = d_k - x_k^T w_k = d_k - w_k^T x_k. \quad (5)$$

设输入信号的自相关矩阵 R 定义为

$$R = E[x_k x_k^T] = E \begin{bmatrix} x_k^2 & x_k x_{k-1} & \cdots & x_k x_{k-L} \\ \vdots & \vdots & & \vdots \\ x_{k-L} x_k & x_{k-L} x_{k-1} & \cdots & x_{k-L}^2 \end{bmatrix}, \quad (6)$$

列矩阵 p 定义为

$$p = E[d_k x_k] = E[d_k x_k d_k x_{k-1} \cdots d_k x_{k-L}]^T, \quad (7)$$

这里假定 d_k , x_k 和 ε_k 都是统计平稳信号, 显然, 均方误差为

$$\text{MSE} = \xi = E[\varepsilon_k^2] = E[d_k^2] + w^T R w - 2p^T w. \quad (8)$$

算法的目的就是通过调整权矩阵 w , 使 MSE 最小. 如图 2 所示 w^* 就是最佳权矢量. 假定 R 是非奇异阵, 则最佳权值为其维纳解^[1,2],

$$w^* = R^{-1} p. \quad (9)$$

把(9)式代入(8)式得

$$\xi_{\min} = E[d_k^2] - p^T w^*, \quad (10)$$

把(10)式代回(8)式, 则有

$$\xi = \xi_{\min} + (w - w^*)^T R (w - w^*). \quad (11)$$

如果定义一偏差矢量 v , 则

$$v = w - w^*, \quad (12)$$

$$\xi = \xi_{\min} + v^T R v. \quad (13)$$

显然, 权值 w 对 w^* 的任何偏差都会使得 ξ 相对于 ξ_{\min} 增加一不为零的分量 $v^T R v$, 这称为超量均方误差. 对于输入信号的自相关矩阵 R , 它可以分解成

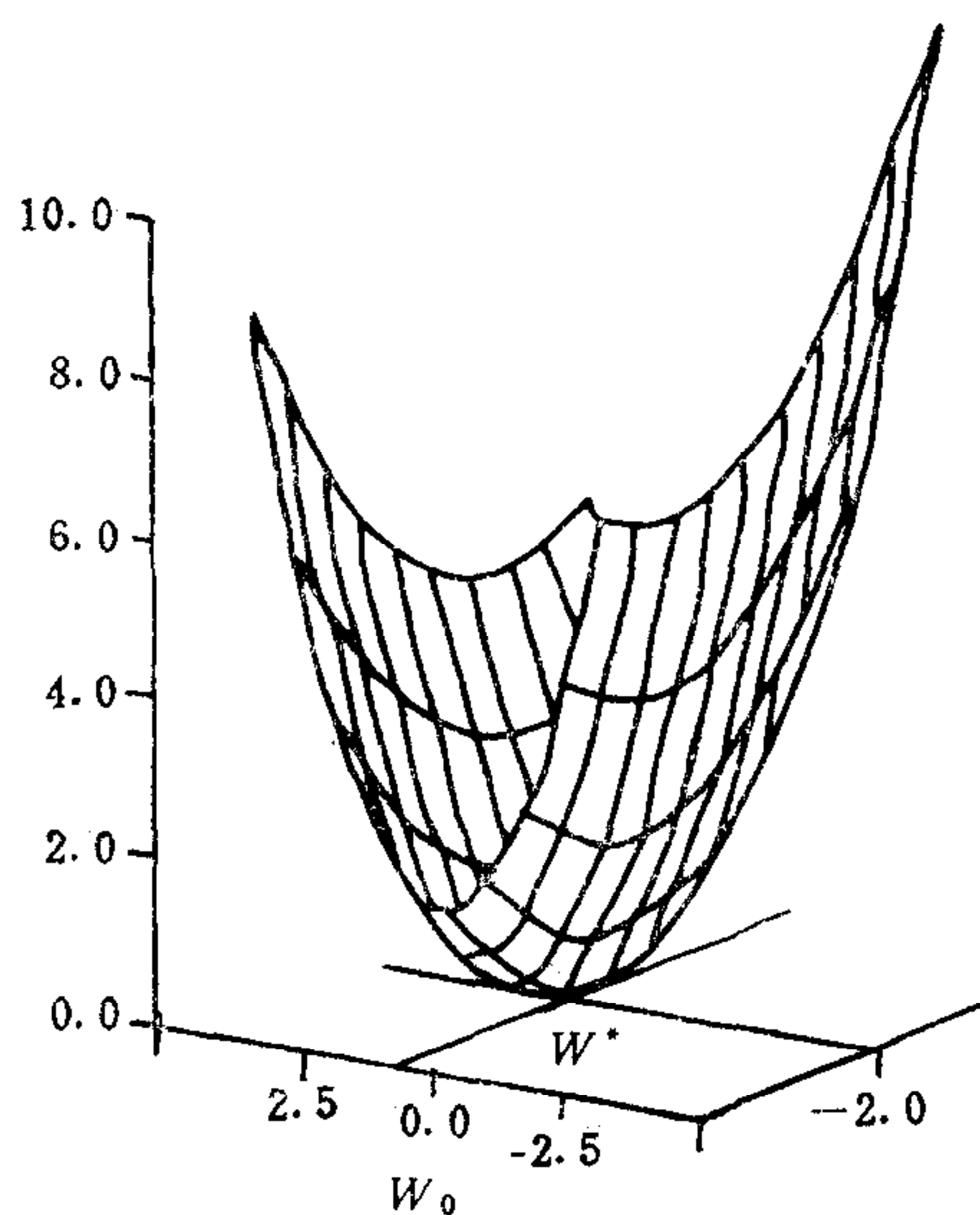


图 2 二维自适应系统误差曲面

$$R[q_0 q_1 \cdots q_L] = [q_0 q_1 \cdots q_L] \begin{bmatrix} \lambda_0 & \cdots & \cdots & 0 \\ & \lambda_1 & & \\ 0 & \cdots & \cdots & \lambda_L \end{bmatrix}, \quad (14)$$

这里 λ_i 是 R 矩阵的第 i 个特征值, 而 q_i 则是与之对应的特征矢量, 即 R 可分解成标准形^[13]

$$R = Q \Lambda Q^{-1}, \quad (15)$$

这里 Λ 是一由特征值构成的对角阵, Q 则是由特征矢量构成的标准正交阵, 它满足

$$Q Q^T = I, \quad (16)$$

$$Q^{-1} = Q^T. \quad (17)$$

二、算法的提出

为了方便起见,首先以单系数自适应线性组合器系统为例来加以讨论.这时,(11)式可简化为

$$\xi = \xi_{\min} + \lambda(w - w^*)^2 = \xi_{\min} + \lambda v^2, \quad (18)$$

式中 $v = w - w^*$, w^* 是系统的最佳维纳解,即 $w^* = R^{-1}p$, λ 是特征值,显然

$$d\xi/dv = 2\lambda v, \quad (19)$$

$$d^2\xi/dv^2 = 2\lambda. \quad (20)$$

传统算法是采用中心差法来估计 $d\xi/dv$ 的,即

$$d\xi/dv \approx [\xi(v + \delta) - \xi(v - \delta)]/(2\delta), \quad (21)$$

$$d^2\xi/dv^2 \approx [\xi(v + \delta) - 2\xi(v) + \xi(v - \delta)]/(\delta^2). \quad (22)$$

现在由(19)式,(20)式和(22)式得出一种新的梯度估计式来代替(21)式

$$\begin{aligned} \nabla &= d\xi/dv = 2\lambda v = (d^2\xi/dv^2)v \\ &\approx v \times \frac{\xi(v + \delta) - 2\xi(v) + \xi(v - \delta)}{\delta^2}. \end{aligned} \quad (23)$$

当误差函数 ξ 是二次函数时,式(23)精确成立,一般情况下仅当 δ 趋近于零时成立.

对于上述估计算法,下面依次讨论其收敛性、方差、失调和收敛速度.

三、算法的收敛性

现采用 N 点最大似然估计 $\hat{\xi}_k$ 作为 ξ_k 的估计值,即

$$\hat{\xi}_k = \frac{1}{N} \sum_{i=1}^N (\varepsilon_{ki})^2, \quad (24)$$

显然, $\hat{\xi}_k$ 是 ξ_k 的无偏估计,这是因为

$$E[\hat{\xi}_k] = E\left[\frac{1}{N} \sum_{i=1}^N (\varepsilon_{ki})^2\right] = \frac{1}{N} \sum_{i=1}^N E(\varepsilon_{ki}^2) = E(\varepsilon_{ki}^2) = \xi_k. \quad (25)$$

设 k 步梯度估计误差为 n_k , 即

$$\hat{\nabla}_k = \nabla_k + n_k, \quad (26)$$

由于 $\hat{\xi}_k$ 是 ξ_k 的无偏估计,从式(23)知 $\hat{\nabla}_k$ 亦是 ∇_k 的无偏估计,即

$$E[\hat{\nabla}_k] = \nabla_k, \quad (27)$$

$$E[n_k] = 0. \quad (28)$$

对于估计式(23),下面分别讨论在牛顿法和最陡下降法中的收敛性.

1. 牛顿法

无噪递推公式为

$$w_{k+1} = w_k - \mu R^{-1} \nabla_k, \quad (29)$$

含噪递推公式为

$$w_{k+1} = w_k - \mu R^{-1} \nabla_k = w_k - \mu R^{-1} \nabla_k - \mu R^{-1} n_k, \quad (30)$$

$$v_{k+1} = v_k - \mu R^{-1} \nabla_k - \mu R^{-1} n_k. \quad (31)$$

将 $\nabla = 2Rv$ 及 $v = Qv'$ 代入上式得

$$v_{k+1} = (1 - 2\mu)v_k - \mu R^{-1} n_k, \quad (32)$$

$$\begin{aligned} Qv'_{k+1} &= (1 - 2\mu)Qv'_k - \mu Q\Lambda^{-1}Q^{-1}n_k, \\ v'_{k+1} &= (1 - 2\mu)v'_k - \mu\Lambda^{-1}(Q^{-1}n_k). \end{aligned} \quad (33)$$

如果定义 $Q^{-1}n$ 为 n' , 则

$$v'_{k+1} = (1 - 2\mu)v'_k - \mu\Lambda^{-1}n'_k, \quad (34)$$

这里 Λ^{-1} 是对角阵, 上式代表了一组差分方程

$$\begin{aligned} v'_1 &= (1 - 2\mu)v'_0 - \mu\Lambda^{-1}n'_0, \\ v'_2 &= (1 - 2\mu)^2v'_0 - \mu\Lambda^{-1}[(1 - 2\mu)n'_0 + n'_1], \\ &\vdots \\ v'_k &= (1 - 2\mu)^k v'_0 - \mu\Lambda^{-1} \sum_{n=0}^{k-1} (1 - 2\mu)^n n'_{k-n-1}. \end{aligned} \quad (35)$$

当 k 趋近于 ∞ 时, 假定 μ 在稳定范围以内, 那么系统的残差为

$$v'_k = -\mu\Lambda^{-1} \sum_{n=0}^{\infty} (1 - 2\mu)^n n'_{k-n-1}, \quad (36)$$

$$\begin{aligned} E[v'_k] &= -\mu\Lambda^{-1} \sum_{n=0}^{\infty} (1 - 2\mu)^n E[n'_{k-n-1}] \\ &= -\mu\Lambda^{-1} \sum_{n=0}^{\infty} (1 - 2\mu)^n Q^{-1}E[n_{k-n-1}]. \end{aligned} \quad (37)$$

将(28)式代入上式得

$$E[v'_k] = 0, \quad (38)$$

$$\begin{aligned} E[w_k] &= E[w^* + v_k] = w^* + E[v_k] \\ &= w^* + E[Qv'_k] = w^* + QE[v'_k] = w^*, \end{aligned} \quad (39)$$

所以 $E[w_k]$ 收敛于系统最佳维纳解 $R^{-1}p$.

2. 最陡下降法

无噪递推公式为

$$w_{k+1} = w_k - \mu \nabla_k, \quad (40)$$

含噪递推公式为

$$w_{k+1} = w_k - \mu \hat{\nabla}_k. \quad (41)$$

类似于牛顿法可得到当 k 趋近于 ∞ 时的残差为

$$v'_k = -\mu \sum_{n=0}^{\infty} (I - 2\mu\Lambda)^n n'_{k-n-1}, \quad (42)$$

显然估计式(23)用于最陡下降法时仍能保证算法的收敛性。

四、梯度估计方差

先给出采用估计式(23)后的算法梯度估计方差, 然后再讨论该方差对系数解的影响。

1. VAR[$\hat{\xi}$]

如前所述, ξ_k 的估计值 $\hat{\xi}_k$ 被定义成基于 ε_k^2 的 N 次采样的最大似然估计, 即(24)式. 由该式得

$$\begin{aligned} \text{VAR}[\hat{\xi}] &= E[(\hat{\xi} - \xi)^2] = E[\hat{\xi}^2 + \xi^2 - 2\xi \cdot \hat{\xi}] \\ &= E[\hat{\xi}^2] - \xi^2 = E\left[\frac{1}{N} \sum_{i=1}^N (\varepsilon_i)^2 \times \frac{1}{N} \sum_{j=1}^N (\varepsilon_j)^2\right] - \xi^2 \\ &= \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N E[(\varepsilon_i \varepsilon_j)^2] - \xi^2. \end{aligned} \quad (43)$$

注意到:

$$\begin{aligned} i = j \text{ 时, } E[(\varepsilon_i \varepsilon_j)^2] &= E[\varepsilon_i^4], \\ i \neq j \text{ 时, } E[(\varepsilon_i \varepsilon_j)^2] &= E[\varepsilon_i^2]E[\varepsilon_j^2] = \xi^2, \end{aligned} \quad (44)$$

所以

$$\begin{aligned} \text{VAR}[\hat{\xi}] &= \frac{1}{N^2} [(N^2 - N)\xi^2 + N \times E(\varepsilon_i^4)] - \xi^2 \\ &= \frac{1}{N} [E(\varepsilon_i^4) - \xi^2]. \end{aligned} \quad (45)$$

一般假定 ε_i 具有均值为零, 方差为 ξ 的正态分布^[5], 于是

$$\text{VAR}[\hat{\xi}] = 2\xi^2/N. \quad (46)$$

文献[5]指出, 当 ε_i 满足其它分布时, 对应的方差 $\text{VAR}[\hat{\xi}]$ 均小于(46)式的值.

2. VAR[$\partial \hat{\xi} / \partial v$]

从式(23)得到

$$\begin{aligned} \text{VAR}[\partial \hat{\xi} / \partial v] &= [v^2 / \delta^4] \times [\text{VAR}[\hat{\xi}(v + \delta)] + 4\text{VAR}[\hat{\xi}(v)] \\ &\quad + \text{VAR}[\hat{\xi}(v - \delta)]], \end{aligned} \quad (47)$$

把式(46)的结果代入式(47)得

$$\text{VAR}[\partial \hat{\xi} / \partial v] = \frac{v^2}{\delta^4} \left[\frac{2}{N} \xi(v + \delta) + 4 \frac{2}{N} \xi^2(v) + \frac{2}{N} \xi^2(v - \delta) \right]. \quad (48)$$

如果假定 δ 很小且自适应过程收敛于权系数 w^* , 那么 ξ 就近似等于 ξ_{\min} ^[6], 上式可简化为

$$\text{VAR} \left[\frac{\partial \hat{\xi}}{\partial v} \right] = \frac{2(v^*)^2}{N\delta^4} 6\xi_{\min}^2 = 12\xi_{\min}^2(v^*)^2 / (N\delta^4), \quad (49)$$

上式中 v^* 为自适应过程收敛后的 v , 即

$$v^* = w - w^* |_{w \rightarrow w^*} \approx 0. \quad (50)$$

由于在估计梯度中, 参数 N 和 δ 是相同的, 故在多维情况下有

$$\text{COV}[\hat{\nabla}_k] = E[(\hat{\nabla}_k - \nabla_k)(\hat{\nabla}_k - \nabla_k)^T] = 12(\xi_{\min}^2 / N\delta^4)T, \quad (51)$$

$$v^* = w - w^* |_{E[w] \rightarrow w^*} \approx 0, \quad (52)$$

T 为 v^* 对应的扩充对角矩阵, 即

$$T = \begin{bmatrix} (v_0^*)^2 & \cdots & 0 \\ \vdots & (v_1^*)^2 & \vdots \\ 0 & \cdots & (v_L^*)^2 \end{bmatrix}, \quad (53)$$

这是因为主对角线正好是 $E[(\partial \hat{\xi} / \partial \nu - \partial \xi / \partial \nu)^2] = \text{VAR}[\partial \hat{\xi} / \partial \nu]$, 交叉项则因彼此独立, 其积的期望为零.

进一步推证得

$$\begin{aligned} \text{cov}[n'_k] &= E[n'_k n'_k{}^T] = Q^{-1} E[n_k n_k{}^T] Q \\ &= Q^{-1} \text{cov}[\hat{V}_k] Q = 12[\xi_{\min}^2 / (N\delta^4)] T. \end{aligned} \quad (54)$$

五、失调 M

结合式(13), 自适应系统的超量均方误差定义为

$$\text{excess MSE} = E[\xi_k - \xi_{\min}] = E[\nu_k^T R \nu_k], \quad (55)$$

$$\text{excess MSE} = E[\nu_k'^T \Lambda \nu_k']. \quad (56)$$

设 ν_k' 的元素是 ν'_{nk} , 则

$$\text{excess MSE} = E \left[\sum_{n=0}^L \lambda_n \nu'_{nk}{}^2 \right] = \sum_{n=0}^L \lambda_n E[\nu'_{nk}{}^2]. \quad (57)$$

由于 $E[\nu'_{nk}{}^2]$ 正好是 $\text{cov}[\nu_k']$ 的对角元素, 所以应先求出 $\text{cov}[\nu_k']$. 因 ν_k' 是零均值, 故

$$\text{cov}[\nu_k'] = E[\nu_k' \nu_k'^T]. \quad (58)$$

1. 应用于牛顿法所造成的失调 M

从(34)式得

$$\begin{aligned} \nu_k' \nu_k'^T &= (1 - 2\mu)^2 \nu_{k-1}' \nu_{k-1}'^T + \mu^2 \Lambda^{-1} n_{k-1}' n_{k-1}'^T (\Lambda^{-1})^T \\ &\quad - \mu(1 - 2\mu) [\nu_{k-1}' n_{k-1}'^T (\Lambda^{-1})^T + \Lambda^{-1} n_{k-1}' \nu_{k-1}'^T]. \end{aligned} \quad (59)$$

整理得

$$\text{cov}[\nu_k'] = \mu [(\Lambda^{-1})^2 / 4(1 - \mu)] \text{cov}[n_k'], \quad (60)$$

把(54)式代入(60)式得

$$\text{cov}[\nu_k'] = 3\mu [(\Lambda^{-1})^2 / (1 - \mu) N\delta^4] \xi_{\min}^2 T, \quad (61)$$

其主对角线上的元素为

$$\frac{3\mu \xi_{\min}^2}{(1 - \mu) N\delta^4} \left(\frac{\nu_i^*}{\lambda_i^*} \right)^2,$$

其中 ν_i^* 为 ν^* 的第 i 个分量, 所以

$$\begin{aligned} \text{excess MSE} &= \sum_{n=0}^L \lambda_n \frac{1}{\lambda_n^2} \frac{3\mu \xi_{\min}^2 \nu_n^{*2}}{(1 - \mu) N\delta^4} \\ &= \frac{3\mu \xi_{\min}^2}{(1 - \mu) N\delta^4} \sum_{n=0}^L \lambda_n \frac{\nu_n^{*2}}{\lambda_n^2}, \end{aligned} \quad (62)$$

故失调可以写成

$$M = \frac{\text{excess MSE}}{\xi_{\min}} = \frac{3\mu \xi_{\min}}{(1 - \mu) N\delta^4} \sum_{n=0}^L \frac{\nu_n^{*2}}{\lambda_n}. \quad (63)$$

2. 应用于最陡下降法中的失调 M

类似于牛顿法, 可以从式(41)得到 ν_k' 的结果, 从而进一步得到 $\text{cov}[\nu_k']$ 的表达式如

下:

$$\text{cov}[v'_k] = (\mu/4)(\Lambda - \mu\Lambda^2)^{-1}\text{cov}[n'_k], \quad (64)$$

$$\text{cov}[v'_k] = (3\mu\xi_{\min}^2/(N\delta^4))(\Lambda - \mu\Lambda^2)^{-1}T. \quad (65)$$

注意到(65)式中的 $(\Lambda - \mu\Lambda^2)^{-1}$ 仍然为对角阵,且其主对角元素为 $1/\lambda_n - \mu\lambda_n^2$,而 T 亦为对角阵,故 $\text{cov}[v'_k]$ 的主对角元素为

$$\frac{3\mu\xi_{\min}^2}{N\delta^4} \cdot \frac{v_n^{*2}}{\lambda_n - \mu\lambda_n^2},$$

因此超量均方误差为

$$\text{excess MSE} = \sum_{n=0}^L \lambda_n \frac{v_n^{*2}}{\lambda_n - \mu\lambda_n^2} \cdot \frac{3\mu\xi_{\min}^2}{N\delta^4} = \frac{3\mu}{N\delta^4} \xi_{\min}^2 \sum_{n=0}^L \frac{v_n^{*2}}{1 - \mu\lambda_n}, \quad (66)$$

故失调可以写成

$$M = \frac{\text{excess MSE}}{\xi_{\min}} = \frac{3\mu\xi_{\min}}{N\delta^4} \sum_{n=0}^L \frac{v_n^{*2}}{1 - \mu\lambda_n}. \quad (67)$$

六、算法收敛速度

1. 采样点数

对于单系数自适应系统而言,传统算法为了估计它的梯度值 ∇_k , 需要分别用基于 N 点采样值的最大似然法来估计 $\xi(v + \delta)$ 和 $\xi(v - \delta)$, 见式(21)所示. 也就是说需要 $2N$ 点采样才能满足单系数自适应系统梯度估计的需要. 对于 $L + 1$ 维系统则需 $2N(L + 1)$ 点采样值才能满足梯度 ∇_k 估计的需要, 而且每当完成一次迭代, 改变权系数 v 值时, 就要重新采样 $2N(L + 1)$ 点以估计此时的梯度值. 若假设自适应过程需要 K 步迭代完成, 那么整个过程就需 $2NK(L + 1)$ 点采样值来估计各相关梯度值.

采用本文提出的算法后, 由于二阶导数是一常数, 故一经估计出后就不随权系数 v 的变化而变化, 也就不需在每一步迭代完成后再次作 N 点采样以估计此步的梯度值了. 从(23)式可以看出, 本算法只需 $2N(L + 1) + 1$ 点采样就可以估计自适应过程中任意一步的梯度值.

2. 时间常数

显然采用本文算法后, 权松弛时间常数 Γ 和学习时间 Γ_{mse} 与原算法是相同的, 所不同的是自适应时间常数 T_{mse} . 由于自适应时间常数的基本单位是数据采样, 对于传统算法而言, 前已说明其每一步均需 $2N(L + 1)$ 点采样来估计梯度值, 故其自适应时间常数为

牛顿法:

$$T_{\text{mse}} = 2(L + 1)N\Gamma_{\text{mse}} = N(L + 1)\Gamma, \quad (68)$$

最陡下降法:

$$(T_{\text{mse}})_n = 2(L + 1)N(\Gamma_{\text{mse}})_n = N(L + 1)(\Gamma)_n. \quad (69)$$

然而采用本文的算法后, 无需每次迭代后做采样, 故

牛顿法:

$$T_{\text{mse}} \approx \Gamma_{\text{mse}}, \quad (70)$$

最陡下降法：

$$(T_{\text{mse}})_n \approx (\Gamma_{\text{mse}})_n. \quad (71)$$

七、算法比较

下面以新旧估计式在牛顿法中的应用为例进行以较，可以证明应用于最陡下降法时也有类似的结果。

显然本算法和传统算法都保证了收敛性。从失调来看，传统算法的失调 M_0 为^[7]

$$M_0 = \frac{(L+1)\lambda_{av}(1/\lambda)_{av}}{8N\Gamma}, \quad (72)$$

(72)式中

$$\lambda_{av} = \frac{1}{L+1} \sum_{n=0}^L \lambda_n, \quad (73)$$

$$(1/\lambda)_{av} = \frac{1}{L+1} \sum_{n=0}^L \frac{1}{\lambda_n}, \quad (74)$$

$$P = \delta^2 \lambda_{av} / \xi_{\min}, \quad (75)$$

$$\Gamma = 1/(2\mu). \quad (76)$$

从式(63)式中知本文算法所引起的失调 M 为

$$M = \frac{3\mu\xi_{\min}}{(1-\mu)N\delta^4} \sum_{n=0}^L \frac{v_n^{*2}}{\lambda_n} = \frac{\lambda_{av}}{8N\Gamma} \frac{12}{(1-\mu)\delta^2} \sum_{n=0}^L \frac{v_n^{*2}}{\lambda_n}. \quad (77)$$

设 $v_{\max}^* = \max_{0 \leq i \leq L} \{v_i^*\}$ ，且考虑到 μ 一般很小， $1-\mu \approx 1$ ，则(77)式变成

$$\begin{aligned} M &= \frac{\lambda_{av}}{8N\Gamma} \frac{12}{\delta^2} \sum_{n=0}^L \frac{v_n^{*2}}{\lambda_n} \leq \frac{\lambda_{av}}{8N\Gamma} \frac{12}{\delta^2} v_{\max}^{*2} \sum_{n=0}^L \frac{1}{\lambda_n} \\ &= \frac{(L+1)\lambda_{av}(1/\lambda)_{av}}{8N\Gamma} \frac{12}{\delta^2} v_{\max}^{*2}. \end{aligned} \quad (78)$$

从式(78)和(72)知，二种算法失调的比值 K 为

$$K = M_0/M > \frac{\delta^2}{12v_{\max}^{*2}}. \quad (79)$$

由于 v^* 是在自适应过程收敛于 w^* 后的值，故 v^* 是一趋近于零的矢量，则

$$\lim v_{\max}^* = 0, \quad (80)$$

而 δ 是事先选定的固定值，故

$$K \gg 1, \quad (81)$$

即新算法具有较小的失调。

最后来看算法的收敛速度，显然收敛速度取决于自适应时间常数 T_{mse} 。从式(68)和式(70)中知，新旧算法的收敛速度比为

$$(T_{\text{mse}})_{\text{new}} / (T_{\text{mse}})_{\text{old}} = \frac{1}{2N(L+1)} \ll 1, \quad (82)$$

故新算法的收敛速度明显加快。

八、模拟实验分析

为了验证算法的有效性,作者在 T&W386 上做了模拟实验,图 3 示为单系数自适应组合器模型。

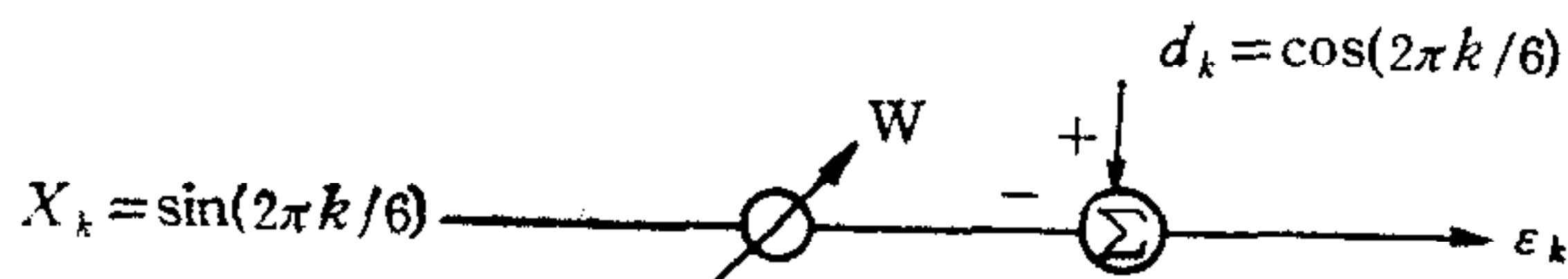


图 3 单系数线性组合器模型

为了验证方便,先计算相关量如下(除 R 外,实验中不使用这些数据):

$$R = [0.5], p = [0], w^* = w = 0, \xi_{\min} = 2, \xi = 2 + 0.5\nu^2.$$

实验中取 $\mu = 0.01, \delta = 0.01, w$ 的初值取为 1, 精度控制为 10^{-4} 。

模拟实验的结果如图 4 至图 6 所示。其中图 4 至图 6 中波动较大的曲线代表传统算法所得到的结果,而波动较小的曲线代表本文算法所对应的权系数收敛曲线。从中可见,传统算法在正确值上下作较大的波动,相对而言本文算法则具有收敛快且波动小的特点。从表 1 中更可以看出,在收敛时间上,传统算法比本文算法要慢好几个数量级。而且当精

表 1 算法收敛步数、时间与精度关系表(最下一行表示定步数比较)

精 度	旧 算 法		新 算 法	
	收敛步数	收敛时间	收敛步数	收敛时间
10^{-4}	5003	2:14.11	179	0.08
10^{-8}	>10000	>2:25.17	501	0.22
10^{-12}	>10000	>2:25.22	794	0.28
	5000	2:14.09	5000	1.97

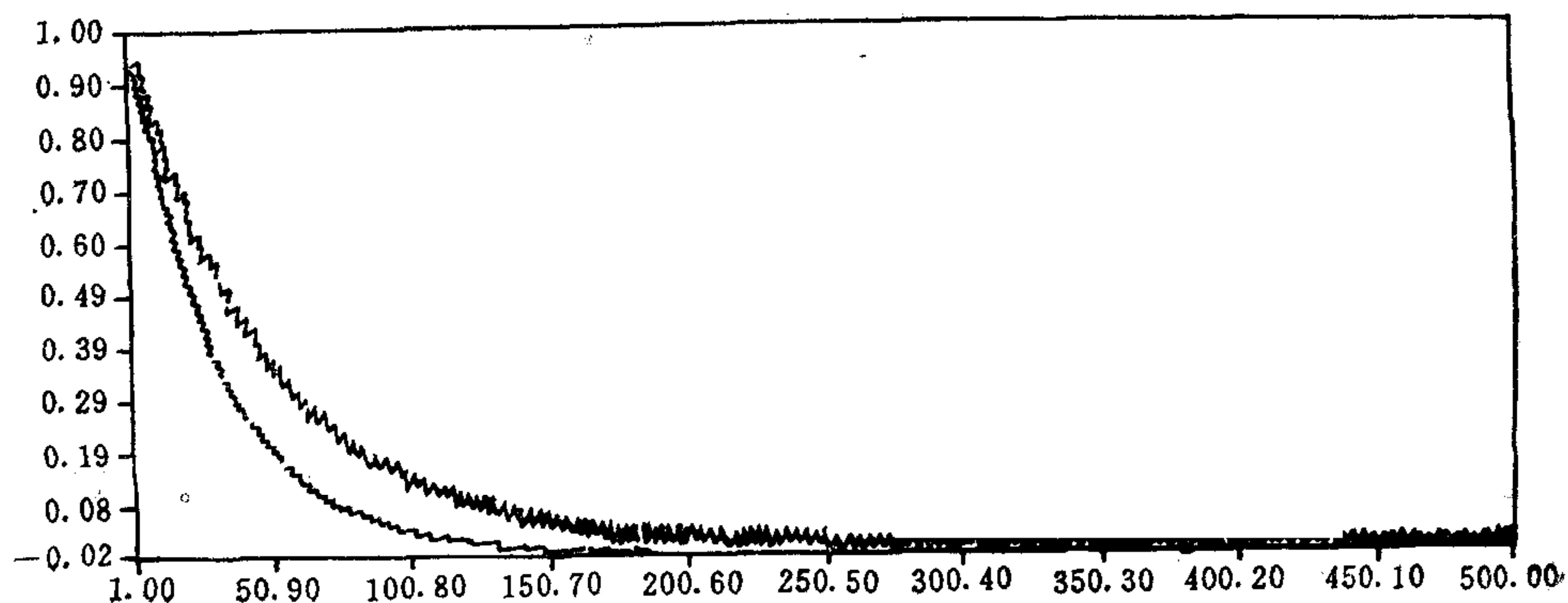


图 4 两种算法迭代开始部分权系数收敛曲线比较

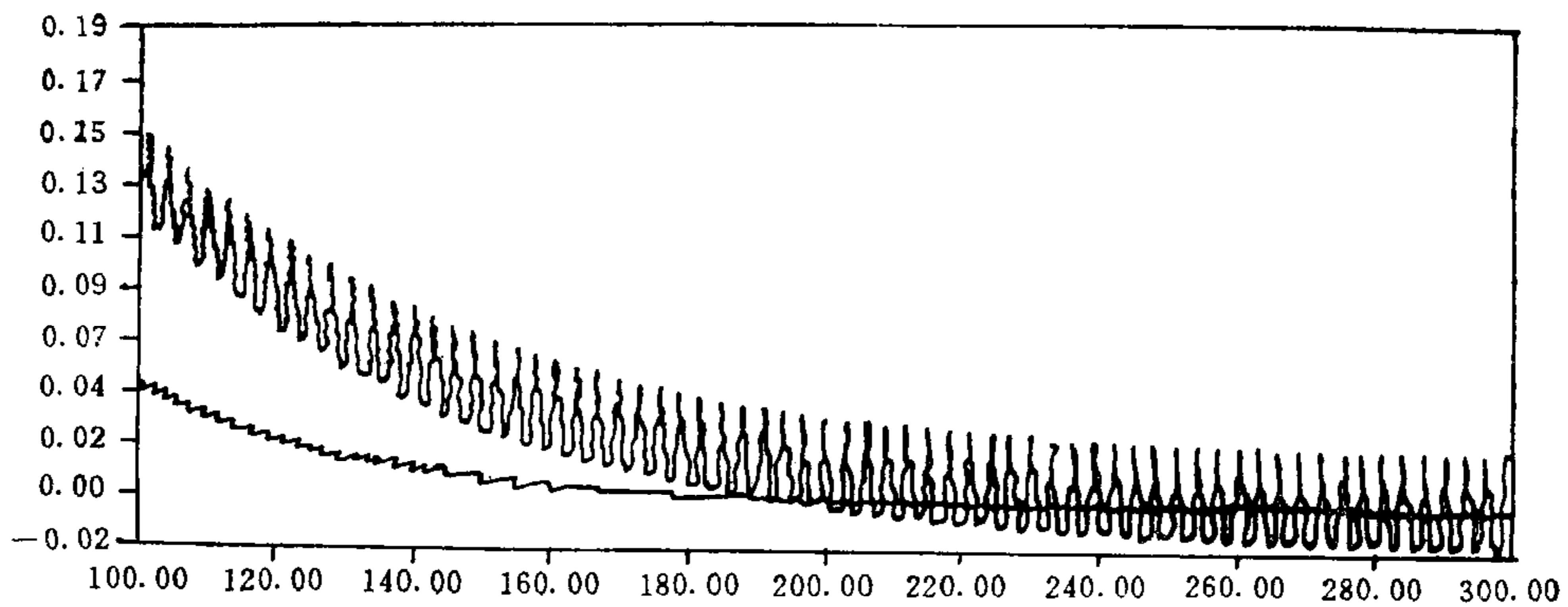


图 5 两种算法(100 点—300 点间)收敛曲线比较

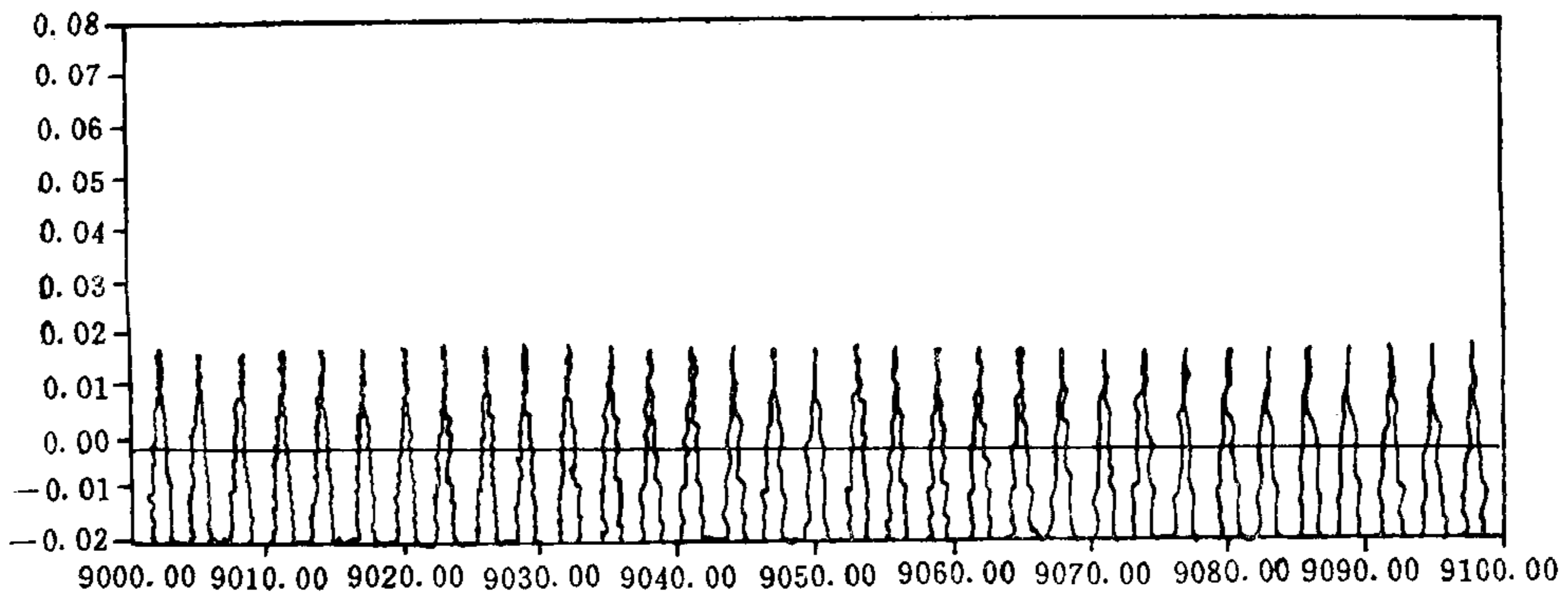


图 6 两种算法收敛后权系数收敛曲线比较

度控制较严时,传统算法就可能因无法达到要求而无限迭代下去(因为算法只能保证均方收敛,故离散程度太大就有可能达不到控制精度要求),所以,模拟实验结果证明了理论推导的正确性。

参 考 文 献

- [1] Wiener, N., Extrapolation, Interpolation and Smoothing of Stationary Time Series, with Engineering Applications. New York: Wiley, 1949.
- [2] Bode, H. W. and Shannon, C. E., A simplified derivation of linear least squares smoothing and prediction theory. Proc. IRE, 38(1950), 417—425.
- [3] Murray R. Spiegel, Theory and Problems of Advanced Mathematics for Engineers and Scientists, McGraw-Hill Book Company, 1971.
- [4] Bernard Widrow and Samned D. Stearns, Adaptive Signal Processing, New Jersey: Prentice-Hall Inc., 1985.
- [5] Bernard Widrow and Samned D. Stearns, Adaptive Signal Processing, New Jersey: Prentice-Hall, Inc., 1985.
- [6] Bernard Widrow and Samned D. Stearns, Adaptive Signal Processing, New Jersey: Prentice-Hall, Inc., 1985.
- [7] Bernard Widrow and Samned D. Stearns, Adaptive Signal Processing, New Jersey: Prentice-Hall, Inc., 1985.
- [8] 王宏禹,随机数字信号处理,科学出版社,1988.

AN ADAPTIVE ALGORITHM BASED ON THE SECOND DERIVATIVE

HU RUIMIN YAO TIANREN HUANG TIEXIA

(Dept. of Electronics & Information Engineering, Huazhong University of Science & Technology,
Wuhan 430074, P. R. China)

ABSTRACT

The current LMS algorithms, widely applied to adaptive signal processing, have some rigorous restrictions on the model and its properties, which in practice may not be guaranteed. Adaptive algorithms based on central-difference gradient estimation, on the other hand, is better suited for many applications, but it requires extensive samples and the convergence is slow. This paper proposes a new algorithm under stationary condition. The algorithm does not require sampling during convergence, converges fast, and has better property to cope with misadjustment.

Key words: Adaptive signal processing; adaptive filter; speech processing; neural networks.



胡瑞敏 1984年毕业于南京邮电学院无线电工程系,1990年在南京邮电学院获通信与电子系统工学硕士学位,现为华中理工大学电子与信息工程系博士研究生。主要工作涉及数字信号处理、人工智能、语音识别和神经网络等领域。已在国际、国内会议和杂志上发表了十几篇论文。



姚天任 1962年毕业于北京清华大学无线电电子学系,1982年至1984年为美国辛辛那提大学访问学者,现任华中理工大学电子与信息工程系主任,副教授。主要著作有《高频电子线路》、《数字信号处理》和《数字语音处理》。现在主要从事通信与电子系统、信号和信息处理等专业的科研、教学和指导研究生的工作。



黄铁侠 1956年毕业于华中工学院。现任华中理工大学电子与信息工程系教授，通信与电子系统专业博士生导师。多年来从事电子学与通信学科的科研、教学和指导研究生的工作。

中国自动化学会第九届青年学术年会 征 文 通 知

为了促进自动化领域青年科技工作者的学术交流，经中国自动化学会批准，由中南工业大学自动控制工程系和计算机应用研究所承办“中国自动化学会第九届青年学术年会”，暂定于1993年8月中旬在湖南省慈利县(张家界)召开，欢迎全国广大青年科技工作者踊跃投稿，论文录用后将编入正式出版的论文集。

一、征文范围

自动化领域的理论和技术及其应用。征文主要内容包括：1. 控制理论及应用；2. 非线性系统控制；3. 大系统控制理论及应用；4. 过程控制与自动化仪器仪表；5. 电气自动化；6. 计算机应用；7. 机器人学及应用；8. 模式识别与智能控制；9. 人工智能及专家系统；10. 人工神经网络；11. 计算机图形学及辅助设计；12. 综合办公室自动化；13. 管理系统工程；14. 经济系统的建模、控制与决策；15. 计算机集成制造技术；16. 遥测遥感遥控技术；17. 空间及运动体控制；18. 系统仿真；19. 生物控制理论及技术；20. 离散事件系统理论及应用以及其它与自动化领域相关的论文。

二、征文要求

1. 反映国内先进水平，未在公开刊物或学术会议上发表过。2. 来稿请写清作者的年龄和联系地址，并在信封上注明“青年年会”字样。

三、论文截止日期

1993年3月20日前提交两份1000~1500字的摘要或6000字以内的全文。1993年4月20日前发出论文录用通知。1993年5月20日前提交论文全文。若在1993年4月31日后仍未接收到录取通知，可自行处理稿件。

来稿请寄湖南省长沙市中南工业大学自动控制工程系(410083)夏学锋,吴敏收。

中国自动化学会第九届青年学术年会筹委会
中南工业大学自动控制工程系