
- 第二章 LMS自适应滤波

2.6 非平稳环境下的LMS算法

一 梯度失调误差和跟踪误差

对非平稳信号，自适应滤波器期待尽可能地跟踪上信号特性的变化，但是由于时间常数的关系，总存在延迟。因此超量误差有失调和跟踪两个误差组成。自适应的有效性取决于信号特性。显然最好情况下，信号是慢变化的，保证自适应过程能够跟踪上信号的变化。引起非平稳性的情况有：

- 参考信号是非平稳的
- AF输入信号是非平稳的
- 参考信号和输入信号都是非平稳的

由于 $H_{opt}(n)$ 是时变的，未知的，故系数误差矢量：

$$C(n) = H(n) - H_{opt}(n) = \{H(n) - E[H(n)]\} + \{E[H(n)] - H_{opt}(n)\}$$

其中 $C_1(n) = H(n) - E[H(n)]$

是梯度失调引起，相当于权系数矢量噪声

$$C_2(n) = E[H(n)] - H_{opt}(n)$$

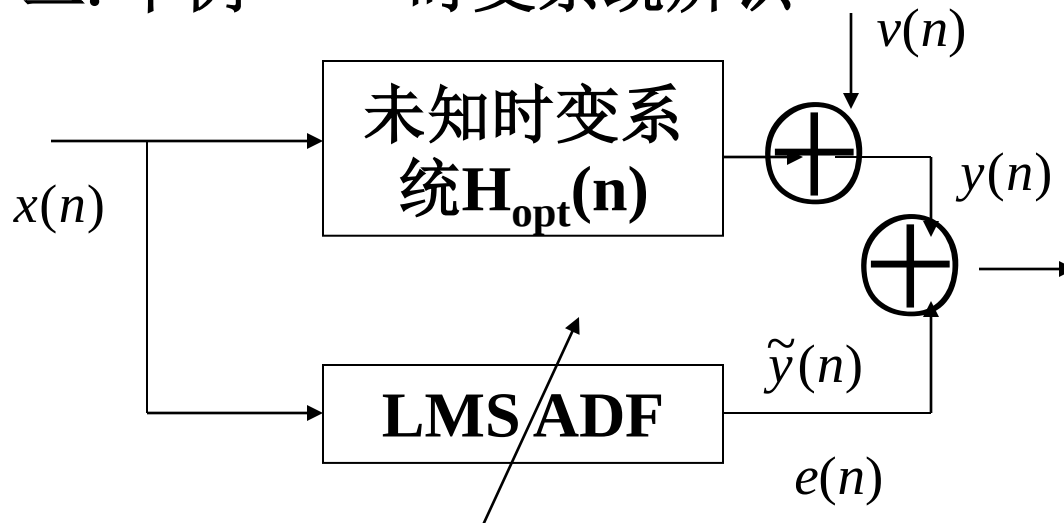
是跟踪误差，由于自适应过程的滞后引起，称为权系数矢量滞后误差。

若LMS算法被应用到平稳环境时， $H_{opt}(n)$ 是常数 H_{opt}

$$E[H(n)] = H_{opt} \quad C_2(n) = 0$$

即无跟踪误差。

二. 举例 —— 时变系统辨识



$$e_{ns,i}(n) \xrightarrow{\quad} \boxed{A_i(z)} \xrightarrow{\quad} h_{opt,i}(n)$$

$$A_i(z) = \frac{1}{1 - \gamma z^{-1}}$$

$\gamma < 1$ 但接近于 1
表示只有慢变化成分通过

设：

1) 未知系统的 $H_{opt}(n)$ 是由一组零均值 i. i. d (独立同分布)，方差为 σ_{ns}^2 的随机噪声 $e_{ns,i}(n)$ ，激励一个一阶低通滤波器模型所产生，有：

$$\mathbf{H}_{opt}(n+1) = \mathbf{H}_{opt}(n) + \mathbf{e}_{ns}(n+1)$$

这里

$$\mathbf{e}_{ns}(n+1) = \begin{bmatrix} e_{ns,0}(n+1) \\ e_{ns,1}(n+1) \\ \dots \\ e_{ns,N-1}(n+1) \end{bmatrix}$$

2)输入 $x(n)$ 是平稳的，具有零均值，自相关矩阵 $\mathbf{R}$

3)未知系统输出受白噪声 $\{v_n\}$ 污染（设为0均值，方差 σ_v^2 ），由于该噪声的存在和未知系统的时变性，使AF和未知系统之间匹配存在差别。

$$\begin{aligned}\mathbf{a}(n+1) &= \mathbf{Q}^T [\mathbf{H}(n+1) - \mathbf{H}_{opt}] = \mathbf{Q}^T [\mathbf{H}(n) + \delta e(n+1)\mathbf{X}(n+1) - \mathbf{H}_{opt}] \\ &= \mathbf{a}(n) + \delta \mathbf{Q}^T e(n+1)\mathbf{X}(n+1)\end{aligned}$$

由于 $\mathbf{H}_{opt}(n)$ 是自适应滤波器跟踪的目标，当

$$\mathbf{H}(n) = \mathbf{H}_{opt}(n)$$

时，则 $J_{\min} = \sigma_v^2$ 而 $y(n)$ 是未知系统的输出，它是时变的。

$$\mathbf{H}_{opt}(n+1) = \mathbf{H}_{opt}(n) + \mathbf{e}_{ns}(n+1)$$

从

$$\begin{aligned}[\mathbf{a}(n+1)] &= \mathbf{Q}^T [\mathbf{H}(n+1) - \mathbf{H}_{opt}(n+1)] \\ &= \mathbf{Q}^T [\mathbf{H}(n) + \delta e(n+1)\mathbf{X}(n+1) - \mathbf{H}_{opt}(n+1)] \\ &= [\mathbf{a}(n)] + \delta \mathbf{Q}^T \mathbf{X}(n+1)e(n+1) - \mathbf{Q}^T \mathbf{e}_{ns}(n+1)\end{aligned}$$

和原来相比多了一项 $\mathbf{Q}^T \mathbf{e}_{ns}(n+1)$ ，且 $\mathbf{e}_{ns}(n+1)$ 和其他变量都不相关，因此 $E\{[\mathbf{a}(n+1)][\mathbf{a}(n+1)]^T\}$ 中多了一项

$$\mathbf{Q}^T E\{[\mathbf{e}_{ns}(n+1)][\mathbf{e}_{ns}(n+1)]^T\} \mathbf{Q} = \sigma_{ns}^2 \mathbf{I}_N$$

$$J(n) = J_{\min} + \sum_{i=0}^{N-1} \lambda_i E[\alpha_i^2(n)]$$

$$E\{\mathbf{a}(n+1)[\mathbf{a}(n+1)]^T\} = [\mathbf{I}_N - 2\delta \text{diag}(\lambda_i)] E\{\mathbf{a}(n)[\mathbf{a}(n)]^T\} +$$

$$\delta^2 E[e^2(n+1)] \text{diag}(\lambda_i) + \sigma_{ns}^2 \mathbf{I}_N$$

$$\boldsymbol{\beta}(n+1) = E[\mathbf{a}^2(n+1)] = \mathbf{B}\boldsymbol{\beta}(n) + \delta^2 J_{\min} \boldsymbol{\Lambda} + \sigma_{ns}^2 \mathbf{I}_N$$

$$b_{ij} = \begin{cases} (1 - \delta\lambda_i)^2 & i = j \\ \delta^2 \lambda_i \lambda_j & i \neq j \end{cases}$$

$$\beta_i(n+1) = [1 - \delta\lambda_i]^2 \beta_i(n) + \delta^2 \lambda_i \sum_{\substack{j=0 \\ j \neq i}}^{N-1} \lambda_j \beta_j(n) + \delta^2 J_{\min} \lambda_i + \sigma_{ns}^2$$

$$n \rightarrow \infty \text{ 时, 忽略 } \delta^2 \text{ 项, 则近似有: } \beta_i(\infty) = \frac{\delta}{2} J_{\min} + \frac{\sigma_{ns}^2}{2\delta} \lambda_i^{-1}$$

$$\therefore J(\infty) = J_{\min} + \sum_{i=0}^{N-1} \lambda_i \beta_i(\infty)$$

$$= J_{\min} \left(1 + \frac{\delta}{2} N \sigma_x^2\right) + \frac{\sigma_{ns}^2}{2\delta} N$$

失调误差 跟踪误差

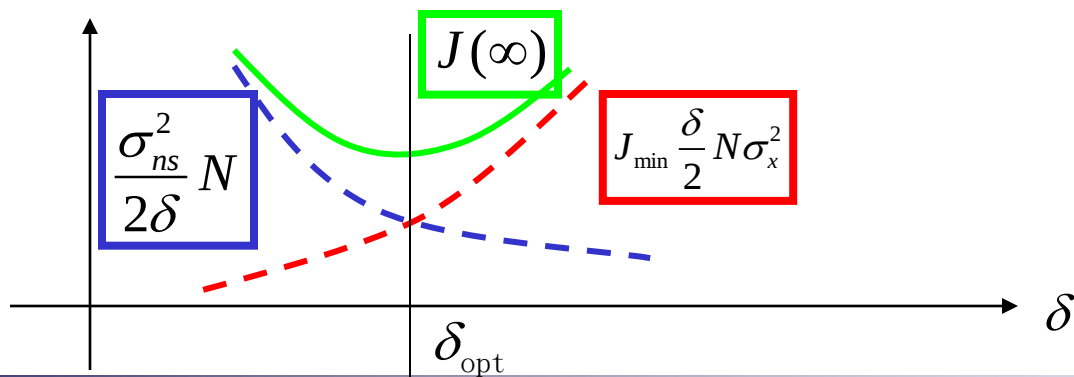
上式简洁地表示出参考信号非平稳时，失调误差和跟踪误差对总误差的贡献。显然这二项误差都与 δ 大小有关，但互相制约。

因为一项与 δ 成正比，一项与 δ 成反比， δ 的一种选择是使误差平衡分布。即

$$\frac{\delta}{2} N \sigma_x^2 J_{\min} = \frac{\sigma_{ns}^2}{2\delta} N \quad \therefore \delta_{opt} = \frac{\sigma_{ns}}{\sigma_x}$$

则
$$J(\infty) = J_{\min} \left(1 + \frac{N \sigma_{ns} \sigma_x}{2\sqrt{J_{\min}}} \right) + \frac{N \sigma_{ns} \sigma_x}{2\sqrt{J_{\min}}} \sqrt{J_{\min}}$$

以上模型是相对粗糙的，但是它对更复杂情况下的滤波器的行为给出了很多启示。



$$\tau_e \approx \frac{1}{\delta \sigma_x^2}$$

三. 归一化LMS方法 (NLMS)

如果输入信号的一些重要的参数可以实时估计的话，自适应滤波器的性能可以大大改善。

对LMS算法来讲，最重要的参数是输入信号的功率，因为它影响 δ 的大小。如果输入信号的功率可以实时估计的话，则可以采用归一化的LMS算法

$$\mathbf{H}(n+1) = \mathbf{H}(n) + \frac{\Delta}{\sigma_x^2} e(n+1) \mathbf{X}(n+1) \quad \sigma_x^2(n)$$

此时相当于step size为 $\delta = \frac{\Delta}{\sigma_x^2}$ ，是在浮动的。

优点是：

- 原来递推式中 δ 是固定值，要考虑到 σ_x^2 的变化，即 σ_x^2 最大时 δ 要满足的收敛条件（ $\delta < \frac{2}{N\sigma_x^2}$ ），因此，

δ 的值不敢取得太大, 而归一化后, 等效 δ 在浮动, 故 Δ 可取较大值; 实际中多数情况下, σ_x^2 小, 等效 δ 值也比较大, 故收敛快。

• 残差:
$$J(\infty) \approx J_{\min} \left(1 + \frac{\delta N \sigma_x^2}{2} \right)$$

$$= J_{\min} \left(1 + \frac{\Delta N}{2} \right)$$

时间常数:
$$\tau_e \approx \frac{1}{\delta \sigma_x^2} = \frac{1}{\Delta}$$

这里 $\delta \sigma_x^2 = \Delta$ 是常数, 故归一化的结果, 使残差 $J(\infty)$ 和时间常数 τ_e 近似不随信号的性质变化。

σ_x^2 的估计:

$$P_x(n) = (1 - \alpha)P_x(n-1) + \alpha x^2(n)$$

2.7 级联型FIR梯度自适应滤波器

在某些应用中，跟踪自适应滤波器的传输函数根的情况（是不是在单位圆内）是很重要的。例如，逆系统要实现时，为保证稳定性，根必须在单位圆内。因此把滤波器设计成一个二阶 $H(z)$ 的级联会带来方便。

一个二阶FIR滤波器：

$$H_l(z) = 1 + h_{1l}z^{-1} + h_{2l}z^{-2}$$

对实数 h_{1l}, h_{2l} ，根 z_l 是复的情况，有：

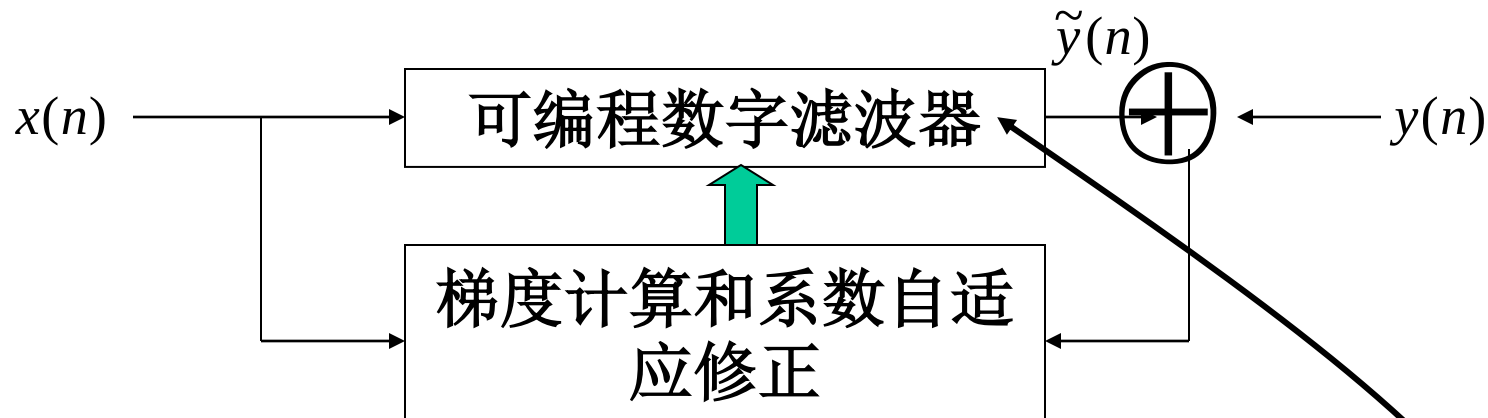
$$z_l = |z_l| e^{+j\alpha_l} \quad \text{且} \quad h_{1l} = -2\text{Re}(z_l) \quad h_{2l} = |z_l|^2$$

不难证明：

若 $|h_{2l}| < 1$ 满足，则复根将在单位圆内，即 $|z_l| < 1$ ，

若是实根， x_1, x_2 ，则要求 $|h_{1l}| < 1 + h_{2l}$ 满足，则有 $|x_1| < 1$ ， $|x_2| < 1$

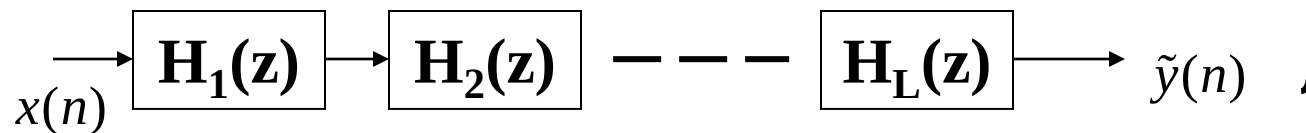
$$\mathbf{H} = [h_0, h_1, \dots, h_{N-1}]^T$$



梯度自适应滤波原理框图

$$\mathbf{H} = [h_{11}, h_{21}, h_{12}, h_{22}, \dots, h_{1L}, h_{2L}]^T$$

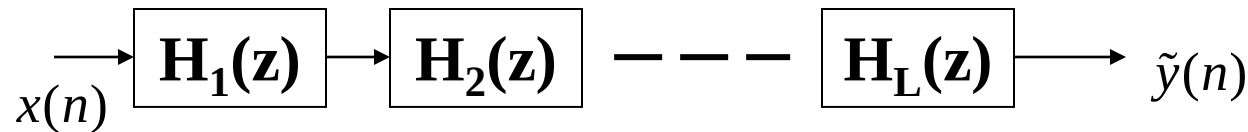
整个滤波器的传递函数 $H(z) = \prod_{l=1}^L (1 + h_{1l}z^{-1} + h_{2l}z^{-2})$



$$e(n+1) = y(n+1) - \tilde{y}(n+1)$$

$$H(z) = \prod_{l=1}^L (1 + h_{1l}z^{-1} + h_{2l}z^{-2})$$

$$\frac{\partial e^2(n+1)}{\partial \mathbf{H}(n)}$$



由于级联的关系，误差梯度矢量估计

$$\frac{\partial \tilde{y}(n+1)}{\partial \mathbf{H}(n)} \neq \mathbf{X}(n+1) \text{ 需修正}$$

级联滤波器的输出可以从逆 z 变换求得：

$$\tilde{y}(n) = \frac{1}{2\pi j} \oint_z z^{(n-1)} \prod_{l=1}^L (1 + h_{1l}z^{-1} + h_{2l}z^{-2}) X(z) dz$$

LMS算法系数修正中用到梯度矢量估计 $\frac{\partial e^2(n+1)}{\partial \mathbf{H}(n)}$

$$e(n+1) = y(n+1) - \tilde{y}(n+1)$$

$$\tilde{y}(n) = \frac{1}{2\pi j} \oint_z z^{(n-1)} \prod_{l=1}^L (1 + h_{1l} z^{-1} + h_{2l} z^{-2}) X(z) dz$$

这样 $\frac{\partial e^2(n+1)}{\partial \mathbf{H}(n)} = 2e(n+1) \frac{\partial e(n+1)}{\partial \mathbf{H}(n)} = -2e(n+1) \frac{\partial \tilde{y}(n+1)}{\partial \mathbf{H}(n)}$

即 $\frac{\partial e(n+1)}{\partial h_{ki}} = -\frac{\partial \tilde{y}(n+1)}{\partial h_{ki}} \quad 1 \leq i \leq L \quad k=1,2$

$$H(z) = \prod_{l=1}^L (1 + h_{1l} z^{-1} + h_{2l} z^{-2})$$

变换微积分次序

$$\frac{\partial e(n+1)}{\partial h_{ki}} = -\frac{1}{2\pi j} \oint_z z^{(n+1)-1} z^{-k} \prod_{\substack{l=1 \\ l \neq i}}^L (1 + h_{1l} z^{-1} + h_{2l} z^{-2}) X(z) dz$$

为简洁表示, 令

$$g_{ki}(n+1) = -\frac{\partial e(n+1)}{\partial h_{ki}} = \frac{1}{2\pi j} \oint_z z^n z^{-k} \frac{H(z)}{1 + h_{1i} z^{-1} + h_{2i} z^{-2}} X(z) dz$$

得到修正公式: $h_{k,i}(n+1) = h_{k,i}(n) + \delta g_{ki}(n+1) e(n+1) \quad 1 \leq i \leq L \quad k=1,2$

由前面公式可见, $g_{ki}(n+1)$ 看成一个滤波器的输出, 它的传递函数是两个函数的级联: $H(z)$ 和 $H_i(z)$, 而它的输入是信号矢量 $\mathbf{X}(n+1)$

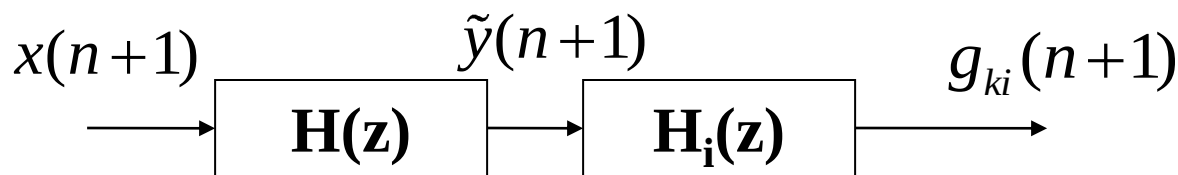
$$g_{ki}(n+1) = \frac{1}{2\pi j} \oint_z \frac{z^{-k}}{1 + h_{1i} z^{-1} + h_{2i} z^{-2}} H(z) X(z) z^n dz$$

$$g_{ki}(n+1) = \frac{1}{2\pi j} \oint_z \frac{z^{-k}}{1 + h_{1i}z^{-1} + h_{2i}z^{-2}} H(z) X(z) z^n dz$$

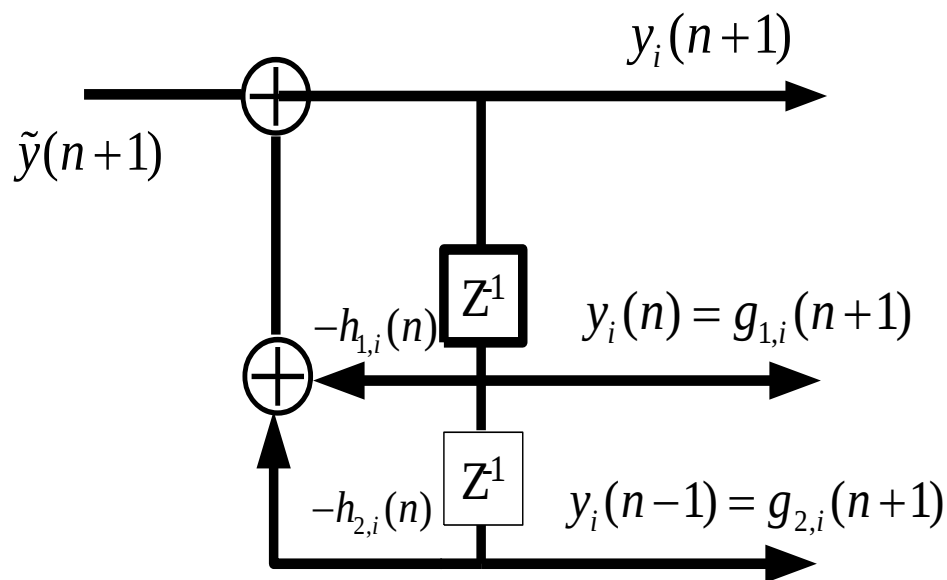
$$H(z) = \prod_{l=1}^L (1 + h_{1l}z^{-1} + h_{2l}z^{-2})$$

$$H_i(z) = \frac{z^{-k}}{(1 + h_{1l}z^{-1} + h_{2l}z^{-2})}$$

由于 $\tilde{y}(z) = H(z)X(z)$ ，即 $\tilde{y}(n+1)$ 是 $X(n+1)$ 输入到整个滤波器的输出，然后将 $\tilde{y}(n+1)$ 输入到第二个滤波器 $H_i(z)$ 。 $H_i(z)$ 是一个二阶IIR滤波器，它的传递函数是第i节的倒数乘以 z^{-k} 。



$$H_i(z) = \frac{z^{-k}}{(1 + h_{1i}z^{-1} + h_{2i}z^{-2})}$$



最后我们得到

注：这里的 $h_{k,i}$ 均是 n 时刻的 $H(n)$ 的分量

$$y_i(n+1) = \tilde{y}(n+1) - h_{1i}y_i(n) - h_{2i}y_i(n-1)$$

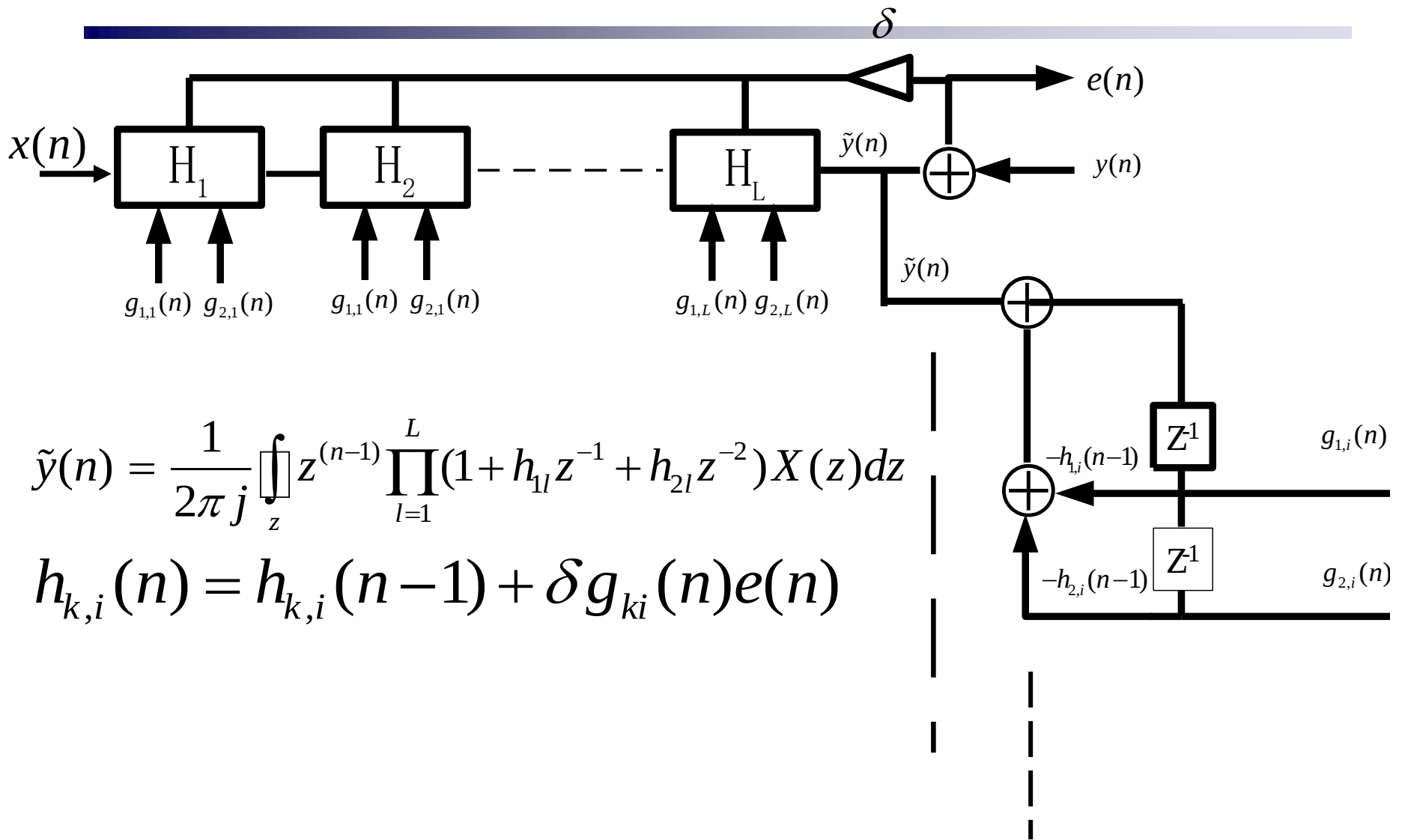
$$Y_i(z) = \frac{\tilde{Y}(z)}{(1 + h_{1i}z^{-1} + h_{2i}z^{-2})}$$

由图中 $y_i(n) = g_{1,i}(n+1)$ 满足

$$\frac{z^{-1}}{(1 + h_{1i}z^{-1} + h_{2i}z^{-2})} \quad k=1$$

$y_i(n-1) = g_{2,i}(n+1)$ 满足

$$\frac{z^{-2}}{(1 + h_{1i}z^{-1} + h_{2i}z^{-2})} \quad k=2$$



讨论：

这种滤波器虽然要比一般横向滤波器要复杂，但是它提供了一种简单发现和跟踪根的方法。由于有递归部分存在（指算 $g_{k,i}$ 时），故根应该在Z平面的单位园内，以保证稳定。然而也存在一些实现上的问题：每一节必须具有适当的特性，以保证整个滤波器有效地工作。

2.8 IIR梯度自适应滤波器

一般来讲，IIR滤波器可以用少于FIR的阶数实现所需的性能。所以IIR也是一种重要的滤波器

IIR滤波器输出：
$$\tilde{y}(n) = \sum_{l=0}^L a_l x(n-l) + \sum_{k=1}^K b_k \tilde{y}(n-k)$$

传递函数
$$H(z) = \frac{\sum_{l=0}^L a_l z^{-l}}{1 - \sum_{k=1}^K b_k z^{-k}} = \frac{N(z)}{D(z)}$$

对误差梯度矢量的估计，可通过对滤波器输出求导来计算

$$\frac{\partial \hat{y}(n)}{\partial a_l} = x(n-l) + \sum_{k=1}^K b_k \frac{\partial \hat{y}(n-k)}{\partial a_l} \quad 0 \leq l \leq L$$

$$\frac{\partial \hat{y}(n)}{\partial b_k} = \hat{y}(n-k) + \sum_{i=1}^K b_i \frac{\partial \hat{y}(n-i)}{\partial b_k} \quad 1 \leq k \leq K$$

$$H(z) = \frac{\sum_{l=0}^L a_l z^{-l}}{1 - \sum_{k=1}^K b_k z^{-k}} = \frac{N(z)}{D(z)}$$

如前节采用的方法, 因为

$$\hat{y}(n) = \frac{1}{2\pi j} \oint_z z^{(n-1)} H(z) X(z) dz$$

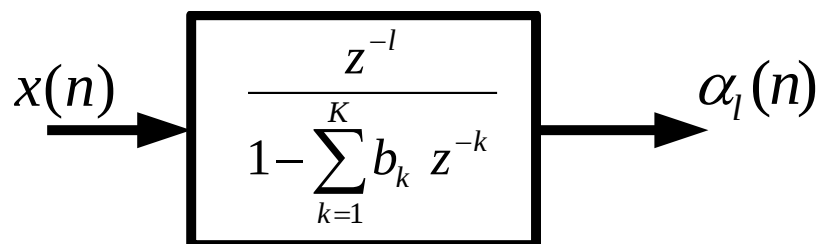
$$\text{令 } \alpha_l(n) = \frac{\partial \hat{y}(n)}{\partial a_l} = \frac{1}{2\pi j} \oint_z z^{(n-1)} z^{-l} \frac{X(z)}{D(z)} dz$$

$$\beta_k(n) = \frac{\partial \hat{y}(n)}{\partial b_k} = \frac{1}{2\pi j} \oint_z z^{(n-1)} z^{-k} \frac{1}{D(z)} \underbrace{H(z) X(z)}_{\tilde{Y}(z)} dz$$

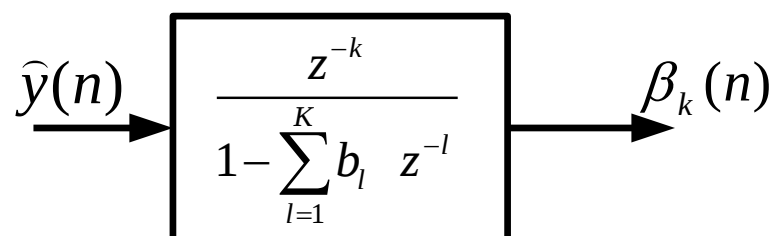
这样梯度矢量估计可以分别将

$x(n)$ 输入到 $\frac{1}{D(z)}$, 将输出适当延迟 (l) 得 $\frac{\partial \hat{y}(n)}{\partial a_l}$

$\tilde{y}(n)$ 输入到 $\frac{1}{D(z)}$, 将输出适当延迟 (k) 得 $\frac{\partial \hat{y}(n)}{\partial b_k}$



$$\alpha_l(n) = x(n-l) + \sum_{k=1}^K b_k \alpha_l(n-k)$$



$$\beta_k(n) = \tilde{y}(n-k) + \sum_{l=1}^K b_l \beta_k(n-l)$$

$$e(n+1) = y(n+1) - [\mathbf{A}^T(n), \mathbf{B}^T(n)] \begin{bmatrix} \mathbf{X}(n+1) \\ \tilde{\mathbf{y}}(n) \end{bmatrix}$$

$$\tilde{y}(n) = \sum_{l=0}^L a_l x(n-l) + \sum_{k=1}^K b_k \tilde{y}(n-k)$$

式中

$$\mathbf{A}(n) = \begin{bmatrix} a_0(n) \\ a_1(n) \\ \vdots \\ a_L(n) \end{bmatrix}$$

$$\mathbf{B}(n) = \begin{bmatrix} b_1(n) \\ b_2(n) \\ \vdots \\ b_K(n) \end{bmatrix}$$

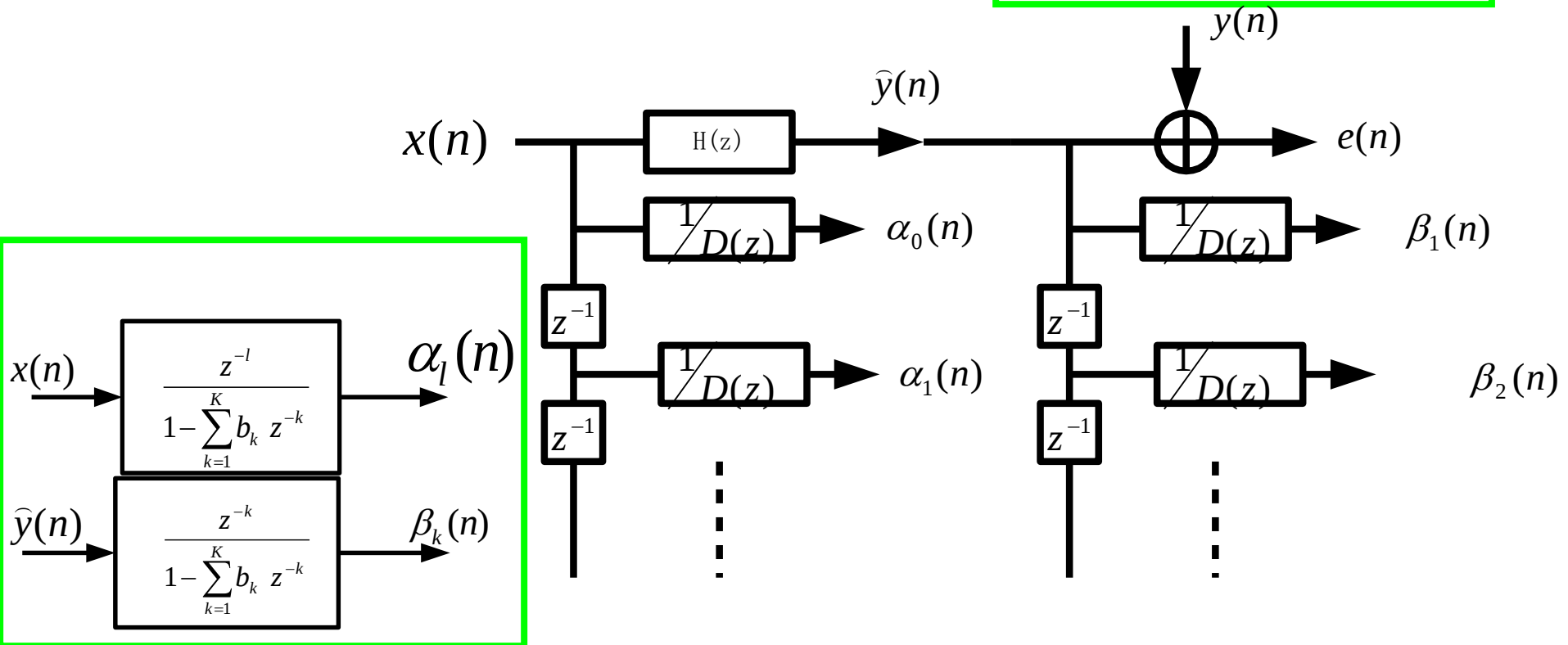
$$\tilde{\mathbf{y}}(n) = \begin{bmatrix} \tilde{y}(n) \\ \tilde{y}(n-1) \\ \vdots \\ \tilde{y}(n-K+1) \end{bmatrix} \quad \mathbf{X}(n+1) = \begin{bmatrix} x(n+1) \\ x(n) \\ \vdots \\ x(n-L+1) \end{bmatrix}$$

$$\mathbf{A}(n+1) = \mathbf{A}(n) + \delta e(n+1)\mathbf{a}(n+1)$$

$$\mathbf{B}(n+1) = \mathbf{B}(n) + \mu e(n+1)\mathbf{\beta}(n+1)$$

$$\mathbf{a}(n) = \begin{bmatrix} \alpha_0(n) \\ \alpha_1(n) \\ \vdots \\ \alpha_L(n) \end{bmatrix}$$

$$\mathbf{\beta}(n) = \begin{bmatrix} \beta_1(n) \\ \beta_2(n) \\ \vdots \\ \beta_K(n) \end{bmatrix}$$



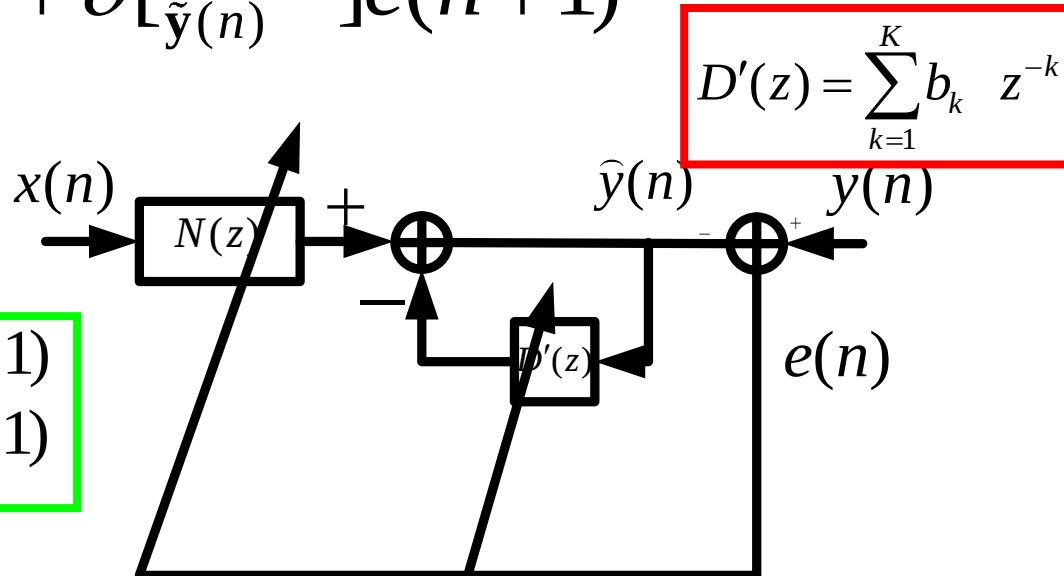
$$\alpha_l(n) = x(n-l) + \sum_{k=1}^K b_k \alpha_l(n-l-k)$$

$$\beta_k(n) = \tilde{y}(n-k) + \sum_{l=1}^K b_l \beta_k(n-k-l)$$

为了实现简单，忽略 $\alpha_l(n)$ 和 $\beta_k(n)$ 中求和项

$$\begin{bmatrix} \mathbf{A}(n+1) \\ \mathbf{B}(n+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(n) \\ \mathbf{B}(n) \end{bmatrix} + \delta \begin{bmatrix} \mathbf{X}(n+1) \\ \tilde{\mathbf{y}}(n) \end{bmatrix} e(n+1)$$

$$\begin{aligned} \mathbf{A}(n+1) &= \mathbf{A}(n) + \delta e(n+1) \boldsymbol{\alpha}(n+1) \\ \mathbf{B}(n+1) &= \mathbf{B}(n) + \mu e(n+1) \boldsymbol{\beta}(n+1) \end{aligned}$$



这称为**并联IIR梯度自适应滤波器**。要分析这样的滤波器不是一件容易的事，因为系统出现了 $\tilde{y}(n)$ ，即最近滤波器输出矢量。

$$e(n+1) = y(n+1) - [\mathbf{A}^T(n), \mathbf{B}^T(n)] \begin{bmatrix} \mathbf{X}(n+1) \\ \tilde{\mathbf{y}}(n) \end{bmatrix}$$

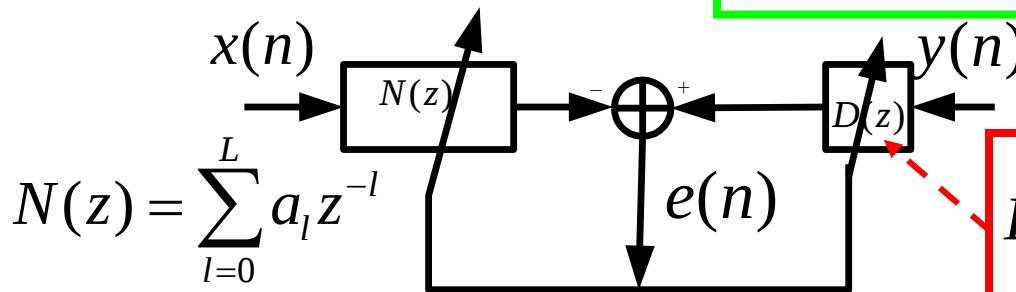
另一种简单的方法，考虑到收敛后误差信号一般较小， $\tilde{\mathbf{y}}(n)$ 接近 $y(n)$ ，因此系统方程中，滤波器输出矢量用参考矢量代替：

$$e(n+1) = y(n+1) - [\mathbf{A}^T(n), \mathbf{B}^T(n)] \begin{bmatrix} \mathbf{X}(n+1) \\ y(n) \end{bmatrix}$$

$$\begin{bmatrix} A(n+1) \\ B(n+1) \end{bmatrix} = \begin{bmatrix} A(n) \\ B(n) \end{bmatrix} + \delta \begin{bmatrix} X(n+1) \\ y(n) \end{bmatrix} e(n+1)$$

此种滤波器称为**串并型**：

$$\begin{bmatrix} A(n+1) \\ B(n+1) \end{bmatrix} = \begin{bmatrix} A(n) \\ B(n) \end{bmatrix} + \delta \begin{bmatrix} X(n+1) \\ \tilde{\mathbf{y}}(n) \end{bmatrix} e(n+1)$$



$$N(z) = \sum_{l=0}^L a_l z^{-l}$$

$$D(z) = 1 - \sum_{k=1}^K b_k z^{-k}$$

这种情况下，**只有FIR结构出现**，不存在滤波器的稳定性问题。稳定性和性能分析可以类似以前那样进行。

$$e(n+1) = y(n+1) - [\mathbf{A}^T(n), \mathbf{B}^T(n)] \begin{bmatrix} \mathbf{X}(n+1) \\ \mathbf{y}(n) \end{bmatrix}$$

例：将 $N\sigma_x^2$ 用 $L\sigma_x^2 + K\sigma_y^2$ 代替： $0 < \delta < \frac{2}{L\sigma_x^2 + K\sigma_y^2}$

可简单比较并联和串并型的性能，考虑收敛后的递推部分（分母）系数均值：

$$\begin{bmatrix} \mathbf{A}(n+1) \\ \mathbf{B}(n+1) \end{bmatrix} = \begin{bmatrix} \mathbf{A}(n) \\ \mathbf{B}(n) \end{bmatrix} + \delta \begin{bmatrix} \mathbf{X}(n+1) \\ \tilde{\mathbf{y}}(n) \end{bmatrix} e(n+1)$$

$$\mathbf{B}_\infty = \lim_{n \rightarrow \infty} E[\mathbf{B}(n+1)] \quad \mathbf{A}(n) \text{ 是一样的。}$$

$$\begin{aligned} \text{并型} \quad \mathbf{B}(n+1) &= \mathbf{B}(n) + \delta \tilde{\mathbf{y}}(n) e(n+1) \\ &= \mathbf{B}(n) + \delta \tilde{\mathbf{y}}(n) [y(n+1) - \mathbf{A}^T(n) \mathbf{X}(n+1) - \tilde{\mathbf{y}}^T(n) \mathbf{B}(n)] \end{aligned}$$

$$\therefore \mathbf{B}_\infty = E[\tilde{\mathbf{y}}(n) \tilde{\mathbf{y}}^T(n)]^{-1} E\{\tilde{\mathbf{y}}(n) [y(n+1) - \mathbf{A}^T(n) \mathbf{X}(n+1)]\}$$

串并型产生类似结果，改成 $E[\mathbf{y}(n) \mathbf{y}^T(n)]^{-1} e(n+1) = y(n+1) - \tilde{\mathbf{y}}(n+1)$

如果输出误差用一个功率为 σ_e^2 白噪声近似。

$$E[\mathbf{y}(n) \mathbf{y}^T(n)] = \sigma_e^2 \mathbf{I}_N + E[\tilde{\mathbf{y}}(n) \tilde{\mathbf{y}}^T(n)]$$

$$E[\mathbf{y}(n)\mathbf{y}^T(n)] = \sigma_e^2 \mathbf{I}_N + E[\tilde{\mathbf{y}}(n)\tilde{\mathbf{y}}^T(n)]$$

- 因此，串并型中，递归部分系数被引入一个偏量，所以残差要大一些。 $\mathbf{B}_\infty = E[\tilde{\mathbf{y}}(n)\tilde{\mathbf{y}}^T(n)]^{-1} E\{\tilde{\mathbf{y}}(n)[y(n+1) - \mathbf{A}^T(n)\mathbf{X}(n+1)]\}$
- 但是并型中，使用 $\tilde{\mathbf{y}}(n)$ ， $E[\tilde{\mathbf{y}}(n)\tilde{\mathbf{y}}^T(n)]$ 可能是奇异阵，出现稳定性问题。
- 二者自适应速度差不多，特别是对小的step size，二者初始时刻的误差序列几乎相同，都是 $y(n+1)$

注意：还有其他各种结构形式，在应用中，IIR梯度自适应滤波器是一种有吸引力的结构。