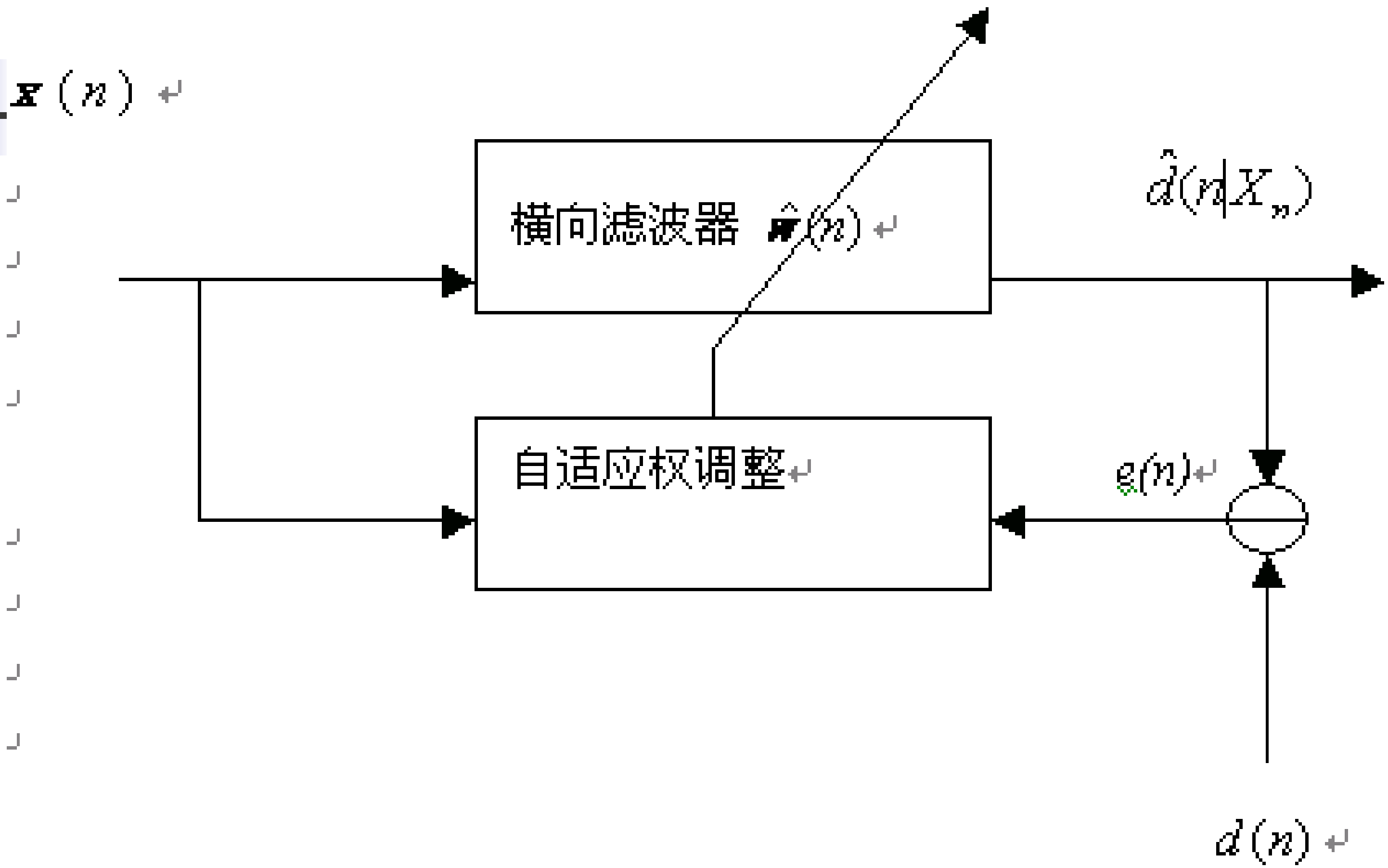
第5章　线性自适应滤波

大量的线性自适应滤波器能满足各类应用，且实现简单

许多非线性（如多项式、高阶量等）自适应滤波器是以线性为核心。

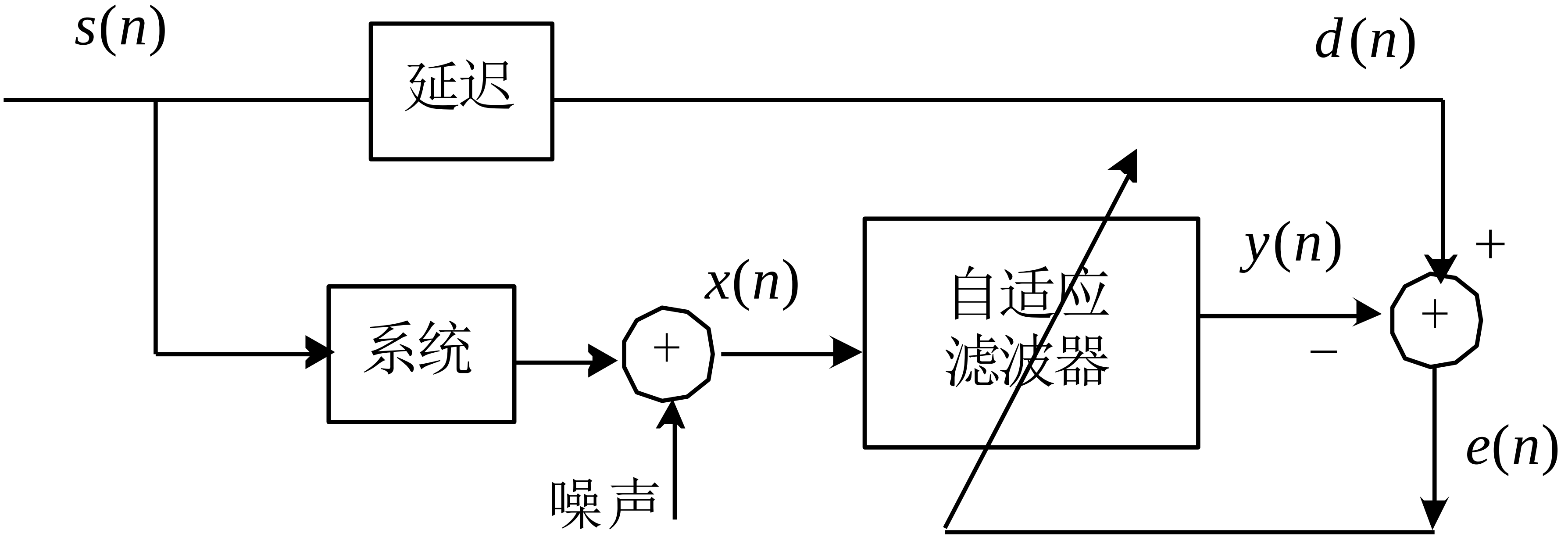
一大类人工神经网络是线性自适应滤波器---LMS算法的推广

盲均衡是线性自适应滤波的一种扩展

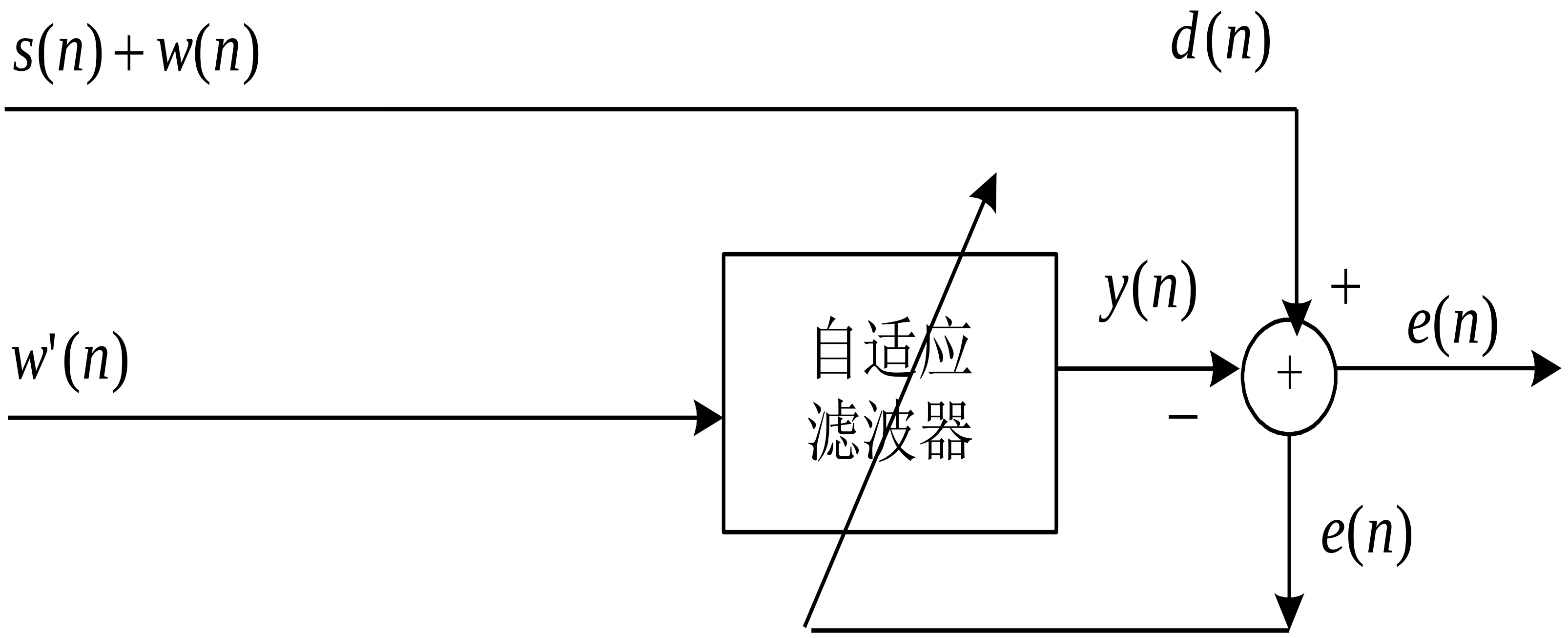


线性自适应滤波器的一般结构框图

应用类型3:自适应均衡



应用类型4:干扰对消



自适应地调整权矢量 $\hat{\mathbf{w}}(n)$，使滤波器达到最优（或接近最优）对期望响应的估计误差：

$$e(n)=d(n)-\hat{d}(n\,|\,\boldsymbol{x}_n)=d(n)-\boldsymbol{w}^H(n)\boldsymbol{x}(n)$$

对于任意权矢量 $\boldsymbol{w}(n)$，均方误差（开销函数）

$$J(n)=\sigma_d^2-\boldsymbol{w}^H(n)\boldsymbol{r}_{xd}-\boldsymbol{r}_{xd}^H\boldsymbol{w}(n)+\boldsymbol{w}^H(n)R\boldsymbol{w}(n)$$

当达到最优时： $\boldsymbol{w}(n)=\boldsymbol{w}_0$，满足：

$$R\boldsymbol{w}_0=\boldsymbol{r}_{xd} \qquad J_{\min}=\sigma_d^2-\boldsymbol{r}_{xd}^H\boldsymbol{w}_0$$

5.1 梯度下降法

对任意初始值 $\boldsymbol{w}(0)$，用传统的优化算法，最徒下降法计算权更新：

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\frac{1}{2}\mu[-\nabla J(n)]$$

并由矩阵函数求导公式：

$$\nabla J(n)=\frac{\partial J(n)}{\partial \boldsymbol{w}(n)}=-2\boldsymbol{r}_{xd}+2R\boldsymbol{w}(n)$$

权更新公式：

$$W(n+1)=W(n)+\mu[\boldsymbol{r}_{xd}-RW(n)]$$

注意到

$$\boldsymbol{w}(n+1)=\boldsymbol{w}(n)+\delta\boldsymbol{w}(n)$$

$$\delta\boldsymbol{w}(n)=\mu[\boldsymbol{r}_{xd}-R\boldsymbol{w}(n)]$$

$$=\mu E[\boldsymbol{x}(n)e^*(n)]$$

最陡下降算法的稳定性分析：

将权更新式两侧减 w_0，并带入：$r_{xd} = Rw_0$

得 $w(n+1) - w_0 = w(n) - \mathbf{w}_0 + \mu[Rw_0 - Rw(n)]$

设 $c(n) = w(n) - w_0$

$$c(n+1) = (I - \mu R)c(n)$$

注意带入：$R = Q\Lambda Q^H$

得：$c(n+1) = (I - \mu Q\Lambda Q^H)c(n)$

上式两边乘 Q^H，并注意 $Q^H = Q^{-1}$，得到

$$Q^H\mathbf{c}(n+1) = (\mathbf{I} - \mu\Lambda)Q^H\mathbf{c}(n)$$

设 $\boldsymbol{v}(n)=Q^{H}\boldsymbol{c}(n)=Q^{H}[\boldsymbol{w}(n)-\mathbf{w}_{0}]$

$$\boldsymbol{v}(n+1)=(I-\mu\Lambda)\boldsymbol{v}(n)$$

从 $\mathbf{v}(0)$ 开始递推：$\boldsymbol{v}(n)=(\mathbf{I}-\mu\Lambda)^{n}\boldsymbol{v}(0)$

由矩阵解耦性质，得：$v_{k}(n)=(1-\mu\lambda_{k})^{n}v_{k}(0)$

为使 $\boldsymbol{v}(n)$ 收敛到零（$\boldsymbol{w}(n)\rightarrow\boldsymbol{w}_{0}$）要求：

$|1-\mu\lambda_{k}|<1$， 对所有 k，

收敛条件：$$0<\mu<\frac{2}{\lambda_{\max}}$$

由： $w(n) = w_0 + Qv(n)$

得： $$w_i(n) = w_{0i} + \sum_{k=1}^{M} q_{ki} v_k(0)(1-\mu\lambda_k)^n$$

q_{ki} 是 R 第 k 个特征矢量 q_k 的第 i 个元素

对于每一个指数衰减项，可以定义一个时间常数 τ_k

$$ce^{-\frac{\tau}{\tau_k}}\Big|_{\tau=1} = c(1-\mu\lambda_k)$$

时间常数为 $$\tau_k = \frac{1}{\ln(1-\mu\lambda_k)}$$

步长μ取的很小时，时间常数近似为 $\tau_k \approx \dfrac{1}{\mu\lambda_k}$

最大时间常数来刻画算法的收敛时间，

$$\tau_{\max} \approx \frac{1}{\mu\lambda_{\min}}$$

由收敛条件取 $\mu = \alpha \cdot \dfrac{2}{\lambda_{\max}} \qquad 0 < \alpha < 1$

得到最大时间常数为 $\tau_{\max} \approx \dfrac{1}{2\alpha}\dfrac{\lambda_{\max}}{\lambda_{\min}}$

均方误差为：

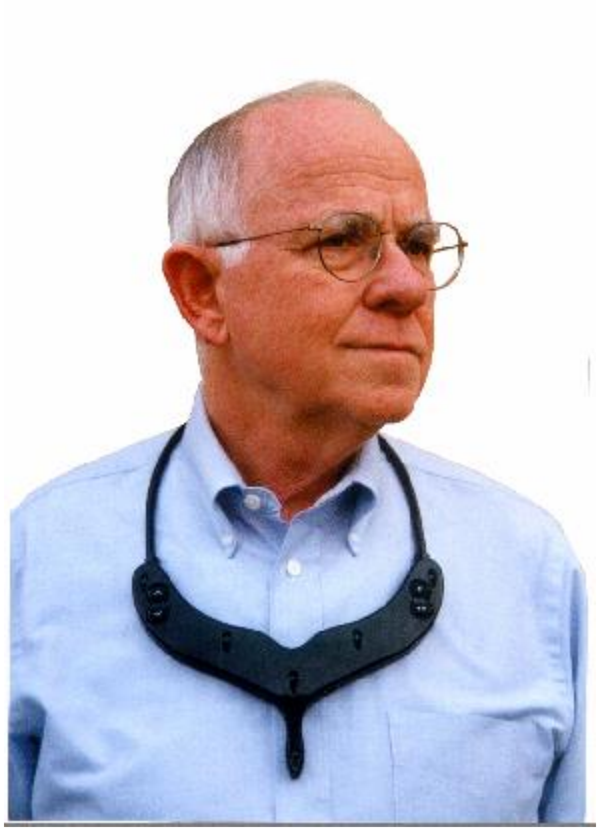
$$J(n) = J_{\min} + \sum_{k=1}^{M} \lambda_k \left| v_k(n) \right|^2$$

$$= J_{\min} + \sum_{k=1}^{M} \lambda_k (1 - \mu \lambda_k)^{2n} \left| v_k(0) \right|^2$$

若满足收敛条件， $J(n) \rightarrow J_{\min}$

5.2 LMS算法（Least-Mean-Square）

- B. Widrow,1960
  - MTI:S.B. 1952
  - S.M. 1953
  - SC.D 1956
- NOW
  - Professor of Stanford Univ.



LMS算法（Least-Mean-Square）

为了构造真正的自适应算法，需要使用估计递度。由于

$$\nabla J(n) = -2\boldsymbol{r}_{xd} + 2R\boldsymbol{w}(n)$$

需估计 $\boldsymbol{r}_{xd}$ 和 R，一种估计方法是：

$$\hat{R}(n) = \boldsymbol{x}(n)\boldsymbol{x}^{H}(n)$$

$$\hat{\boldsymbol{r}}_{xd}(n) = \boldsymbol{x}(n)d^{*}(n)$$

得到递度估计为：

$$\hat{\nabla}J(n) = -2\boldsymbol{x}(n)d^*(n) + 2\boldsymbol{x}(n)\boldsymbol{x}^H(n)\hat{\boldsymbol{w}}(n)$$

$$= -2\boldsymbol{x}(n)(d^*(n) - \boldsymbol{x}^H(n)\hat{\boldsymbol{w}}(n))$$

$$= -2\boldsymbol{x}(n)e^*(n)$$

得到权递推：

$$\hat{\boldsymbol{w}}(n+1) = \hat{\boldsymbol{w}}(n) + \mu\boldsymbol{x}(n)(d^*(n) - \boldsymbol{x}^H(n)\hat{\boldsymbol{w}}(n))$$

LMS 自适应滤波算法由以下 3 部分组成：

（1）滤波器输出：

$$y(n)=\hat{w}^H(n)x(n)$$

（2）估计误差：

$$e(n)=d(n)-y(n)$$

（3）权自适应更新：

$$\hat{w}(n+1)=\hat{w}(n)+\mu x(n)e^*(n)$$

注意：

（a）LMS 算法也可以由另一种方法推导出，令

$$J(n)=|e(n)|^2$$

为开销函数，在每一个时刻 n，令 J(n) 最小，得到相同的递推结果。

（b）每次迭代，LMS 算法需要 2M+1 复数乘法，2M 次复数加法，因此，它的运算量是：$O(M)$

稳定性分析

①为收敛，必须满足：$|1-\mu\lambda_i|^2<1$

即：$$0<\mu<\frac{2}{\lambda_{\max}}$$

实际实现时考虑：

取：$$0<\mu<\frac{2}{tr(R)}$$

$$tr(R)=Mr(0)=\sum_{k=0}^{M-1}E[|x(n-k)|^2]=ME[|x(n)|^2]$$

② 失调

$$J(n) = J_{\min} + J_{ex}(n)$$

$$J_{ex}(\infty) = J_{\min} \sum_{i=1}^{M} \frac{\lambda_i \mu}{2 - \mu\lambda_i}$$

定义失调参数

$$U = \frac{J_{ex}(\infty)}{J_{\min}} = \sum_{i=1}^{M} \frac{\mu\lambda_i}{2 - \mu\lambda_i}$$

改进LMS算法:正则LMS算法（Normalized LMS, NLMs）

取

$$\hat{w}(n+1)=\hat{w}(n)+\frac{\tilde{\mu}}{\|x(n)\|^2}x(n)e^*(n)$$

相当于 $\mu(n)=\dfrac{\tilde{\mu}}{\|x(n)\|^2}$ 的时变 μ 的 LMS 算法，

或相当于将输入信号能量正则化的 LMS 算法。

$0<\tilde{\mu}<2$ 为其收敛条件，$\tilde{\mu}$ 提前可以确定

为了避免在 $\|x(n)\|^2$ 较小的时刻，$\mu(n)$ 太大，进一步限制和改进 NLMS 算法如下：

$$\hat{W}(n+1)=\hat{w}(n)+\frac{\tilde{\mu}}{a+\|\boldsymbol{x}(n)\|^2}\boldsymbol{x}(n)e^*(n)$$

a 为大于零的校正量

稀疏 LMS 算法

识别的线性系统表示为

$$d(n) = w_o^H x(n) + v(n)$$

令开销函数为

$$J_1(n) = |e(n)|^2 + \gamma \|w(n)\|_1$$

其中

$$\|w(n)\|_1 = \sum_{i=0}^{M-1} |w_i(n)|$$

ZA-LMS 算法

则权更新公式可求得为

$$
\begin{aligned}
\boldsymbol{w}(n+1) &= \boldsymbol{w}(n) + \frac{1}{2}\mu\left[-\nabla J(n)\right] \\
&= \boldsymbol{w}(n) - \frac{1}{2}\mu\frac{\partial J_1(n)}{\partial \boldsymbol{w}(n)} \\
&= \boldsymbol{w}(n) + \mu e(n)\boldsymbol{x}(n) - \rho\,\mathrm{sgn}\left[\boldsymbol{w}(n)\right]
\end{aligned}
$$

$\rho = \gamma\mu/2$ 是一个新的控制参数。

最后一项"零吸引子（zero attractor, ZA)"

块LMS算法

- 变换域LMS算法
  - DFT－LMS
  - DCT－LMS
- 子带自适应滤波

5.3 最小二乘方法（Method of Least squares）

- *最陡下降算法*：假设$J(n)=E[|e(n)|^2]$，使之最小求递度，对$\boldsymbol{w}(n)$进行迭代，由于汇集平均E，使得递度中有R，$\boldsymbol{r}_{xd}$等量出现，实际中无法实现。
- *LMS 算法*：由于假设$J(n)=|e(n)|^2$，使之最小求递度，由于瞬间操作，递度随机性很大，收敛慢，有相对“较大”的多余误差$J_{ex}(\infty)$。
- *LS 算法*：设$J(n)=\sum_{i=i_1}^{i_2}|e(n)|^2$，是两者之间的折衷，比 LMS 更好的性能。

递推 LS 算法（RLS）

目的：已知 $\boldsymbol{w}(n-1)=[w_0(n-1),w_1(n-1),\cdots w_{M-1}(n-1)]^T$ 递推估计 $\boldsymbol{w}(n)$，使之满足最小二乘解。

用加权开销函数：

$$\xi(n)=\sum_{i=1}^{n}\beta(n,i)\,|\,e(i)\,|^2$$

根据 LS 算法原则，对于时间 n 和 $\boldsymbol{w}$(n)，要评价观测窗

$i=i_1$ 和 $i=i_2$ 之间（本节取 $i_1=1,\ i_2=n$）的估计误差

$$e(i) = d(i) - y(i)$$

$$= d(i) - \boldsymbol{w}^H(i)\boldsymbol{x}(i)$$

$$\boldsymbol{x}(i) = [x(i), x(i-1), \cdots, x(i-M+1)]^T$$

使得

$$\xi(n) = \sum_{i=1}^{n} \beta(n,i)\,|\,e(i)\,|^2$$

最小.

在 RLS 问题中，$\beta(n,i)$ 取法应考虑给“较新的时刻”更大的比例，“较远的时刻”更小的比例，一种指数“忘却”因子如下：

$$\beta(n,i)=\lambda^{n-i} \qquad i=1,2,\cdots n$$

$$0<\lambda\le 1$$

加权开销函数的 LS 解为：

$$\Phi(n)\hat{w}(n)=z(n) \quad \overset{\text{解为}}{\Longrightarrow} \quad \hat{w}(n)=\Phi^{-1}(n)z(n)$$

其中：

$$\Phi(n)=\sum_{i=1}^{n}\lambda^{n-i}x(i)x^{H}(i)$$

$$z(n)=\sum_{i=1}^{n}\lambda^{n-i}x(i)d^{*}(i)$$

系数矩阵和矢量有递推关系：

$$\Phi(n)=\lambda\Phi(n-1)+\boldsymbol{x}(n)\boldsymbol{x}^H(n)$$

和

$$\boldsymbol{z}(n)=\lambda\boldsymbol{z}(n-1)+\boldsymbol{x}(n)d^*(n)$$

关键问题是，由 $\Phi^{-1}(n-1)$ 递推得到 $\Phi^{-1}(n)$，故此，应用矩阵反引理

矩阵逆引理

若

$$\mathbf{A} = \mathbf{B} + \mathbf{C}\mathbf{D}\mathbf{C}^{H}$$

则

$$\mathbf{A}^{-1} = \mathbf{B}^{-1} - \mathbf{B}^{-1}\mathbf{C}(\mathbf{D}^{-1} + \mathbf{C}^{H}\mathbf{B}^{-1}\mathbf{C})^{-1}\mathbf{C}^{H}\mathbf{B}^{-1}$$

若已知 B 和 D 的逆，并且 $D^{-1}+C^{H}B^{-1}C$ 阶数较小，应用矩阵反引理可以有效计算 A 的逆。

$\Phi^{-1}(n)$ 的递推公式：

为使用矩阵反引理，定义：

$$A = \Phi(n) \qquad B = \lambda\Phi(n-1)$$

$$C = \boldsymbol{x}(n) \qquad D = 1$$

带入 A^{-1} 公式得：

$$\Phi^{-1}(n) = \lambda^{-1}\Phi^{-1}(n-1) - \frac{\lambda^{-2}\Phi^{-1}(n-1)\boldsymbol{x}(n)\boldsymbol{x}^{H}(n)\Phi^{-1}(n-1)}{(1+\lambda^{-1}\boldsymbol{x}^{H}(n)\Phi^{-1}(n-1)\boldsymbol{x}(n))}$$

为表示方便令：

$$P(n) \overset{\Delta}{=} \Phi^{-1}(n)$$

$$\boldsymbol{k}(n) = \frac{\lambda^{-1}P(n-1)\boldsymbol{x}(n)}{(1+\lambda^{-1}\boldsymbol{x}^{H}(n)P(n-1)\boldsymbol{x}(n))}$$

由此得$P(n)$递推方程为：

$$P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}\boldsymbol{k}(n)\boldsymbol{x}^{H}(n)P(n-1)$$

这个方程称为 RLS 的 Riccati 方程

$\boldsymbol{k}(n)$ 称为增益矢量，它可以写为：（将其定义式分母两边同乘，并整理）

$$\boldsymbol{k}(n)=\lambda^{-1}P(n-1)\boldsymbol{x}(n)-\lambda^{-1}\boldsymbol{k}(n)\boldsymbol{x}^{H}(n)P(n-1)\boldsymbol{x}(n)$$

$$=[\lambda^{-1}P(n-1)-\lambda^{-1}\boldsymbol{k}(n)\boldsymbol{x}^{H}(n)P(n-1)]\boldsymbol{x}(n)$$

$$=P(n)\boldsymbol{x}(n)$$

$$=\Phi^{-1}(n)\boldsymbol{x}(n)$$

$\boldsymbol{k}(n)$ 由 $\boldsymbol{x}(n)$ 经由一个 $\Phi^{-1}(n)$ 线性变换而得

滤波器权系数更新公式：

由

$$\hat{\boldsymbol{w}}(n)=\Phi^{-1}(n)\boldsymbol{z}(n)$$

$$=P(n)\boldsymbol{z}(n)$$

$$=P(n)[\lambda\boldsymbol{z}(n-1)+\boldsymbol{x}(n)d^*(n)]$$

带入$P(n)$的迭代公式：（仅对第一项）

$$\hat{\boldsymbol{w}}(n)=P(n-1)\boldsymbol{z}(n-1)-\boldsymbol{k}(n)\boldsymbol{x}^{H}(n)P(n-1)z(n-1)$$
$$+P(n)\boldsymbol{x}(n)d^{*}(n)$$

$$=\Phi^{-1}(n-1)\boldsymbol{z}(n-1)-\boldsymbol{k}(n)\boldsymbol{x}^{H}(n)\Phi^{-1}(n-1)\boldsymbol{z}(n-1)+\boldsymbol{k}(n)d^{*}(n)$$

$$=\hat{\boldsymbol{w}}(n-1)-\boldsymbol{k}(n)\boldsymbol{x}^{H}(n)\hat{\boldsymbol{w}}(n-1)+\boldsymbol{k}(n)d^{*}(n)$$

$$=\hat{\boldsymbol{w}}(n-1)-\boldsymbol{k}(n)[d^{*}(n)-\boldsymbol{x}^{H}(n)\hat{\boldsymbol{w}}(n-1)]$$

$$\overset{\Delta}{=}\hat{\boldsymbol{w}}(n-1)+\boldsymbol{k}(n)\varepsilon^{*}(n)$$

$$\varepsilon(n)=d(n)-\boldsymbol{x}^T(n)\hat{\boldsymbol{w}}^*(n-1)$$

为前验估计误差，它用上一次迭代时刻的权系数 $\hat{w}(n-1)$ 估计当前时刻误差。

权更新递推公式：

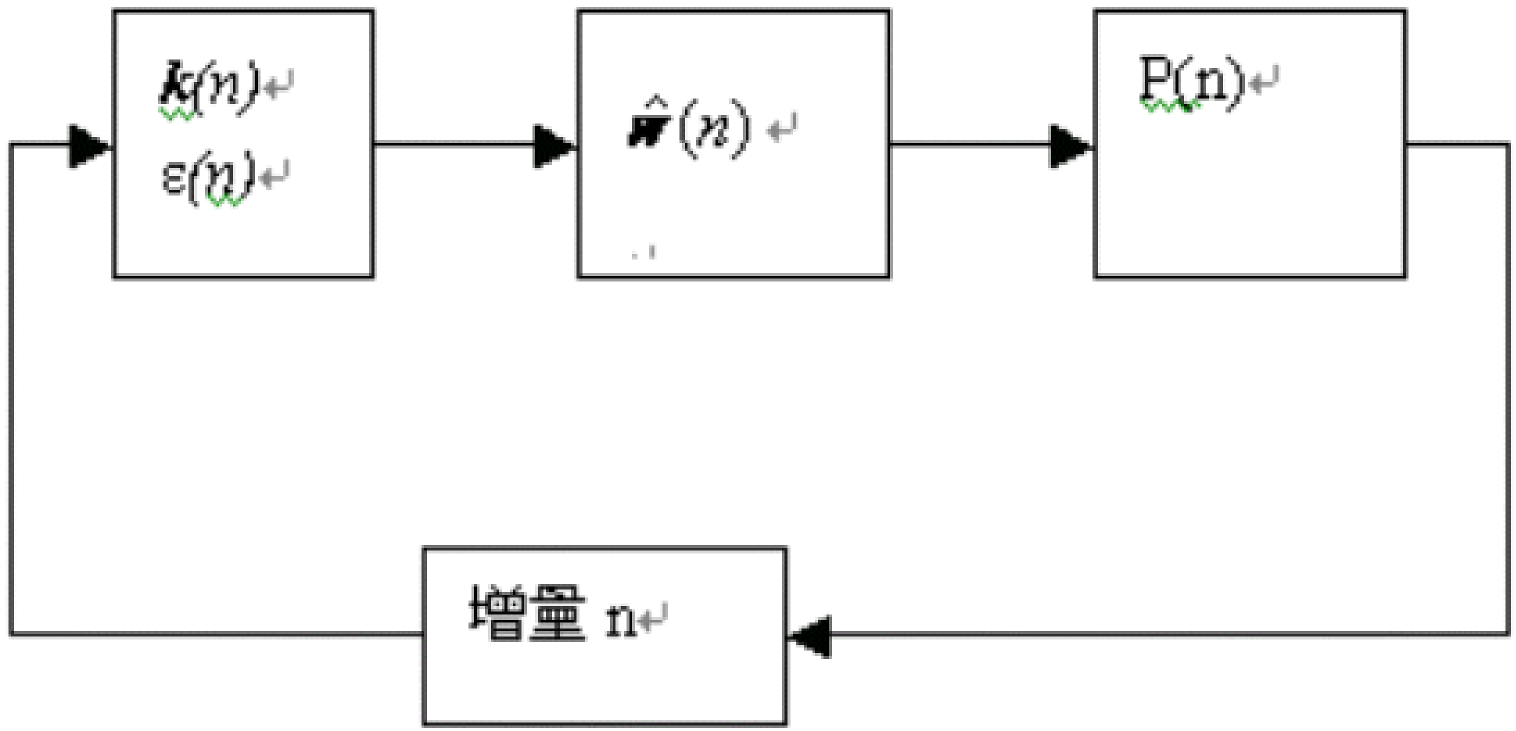
$$\hat{\boldsymbol{w}}(n)=\hat{\boldsymbol{w}}(n-1)+\boldsymbol{k}(n)\varepsilon^*(n)$$

和

$$\varepsilon(n)=d(n)-\hat{\boldsymbol{w}}^H(n-1)\boldsymbol{x}(n)$$

RLS 递推算法示意:

已知 $P(n-1)$ 和 $\hat{w}(n-1)$，如下递推次序



递推公式集合:

$$\boldsymbol{k}(n)=\frac{\lambda^{-1}P(n-1)\boldsymbol{x}(n)}{(1+\lambda^{-1}\boldsymbol{x}^{H}(n)P(n-1)\boldsymbol{x}(n))}$$

$$\varepsilon(n)=d(n)-\hat{\boldsymbol{w}}^{H}(n-1)\boldsymbol{x}(n)$$

$$\hat{\boldsymbol{w}}(n)=\hat{\boldsymbol{w}}(n-1)+\boldsymbol{k}(n)\varepsilon^{*}(n)$$

$$P(n)=\lambda^{-1}P(n-1)-\lambda^{-1}\boldsymbol{k}(n)\boldsymbol{x}^{H}(n)P(n-1)$$

$$y(n)=\hat{\boldsymbol{w}}^{H}(n)\boldsymbol{x}(n)$$

RLS 递推的初始值

设 δ 是一个很小值，一般 $\delta \le 0.01\sigma_x^2$，取

$$\Phi(0) = \delta \cdot I$$

(按预加窗的定义， $\Phi(0)=0, P(0)\to\infty$)

或：

$$P(0) = \delta^{-1} I$$

和

$$\hat{w}(0) = 0$$

RLS 算法的收敛性分析

假设，期望响应和输入 $\boldsymbol{x}(n)$ 满足线性递归模型

$$d(n)=e_0(n)+\boldsymbol{w}_0^H\boldsymbol{x}(n)$$

$e_0(n)$ 是测量误差，方差为 σ^2，为一白噪声过程，$\boldsymbol{w}_0$ 为待估计的系数，迭代算法的权误差矢量：

$$\boldsymbol{\varepsilon}'(n)=\hat{\boldsymbol{w}}(n)-\boldsymbol{w}_0$$

$$K(n)=E[\boldsymbol{\varepsilon}'(n)\cdot\boldsymbol{\varepsilon}'^H(n)]$$

另一个评价准则是：前验估计误差的均方差：

$$J'(n)=E[|\varepsilon(n)|^2]$$

由独立性假设，可以推证如下结论：

① $E[\hat{w}(n)] = w_0 \qquad n \geq M$

② $K(n) = \dfrac{\sigma^2}{n-M-1} R^{-1} \qquad n > M+1$

$$E[\varepsilon'^H(n)\varepsilon'(n)] = tr[K(n)] = \frac{\sigma^2}{n-M-1}\sum_{i=1}^{M}\frac{1}{\lambda_i}$$

如果 $\lambda_{\min}$ 很小，收敛较慢。

③$J'(n)=\sigma^2+tr[RK(n-1)]$

$$=\sigma^2+\frac{M\sigma^2}{n-M-1} \qquad n>M+1$$

结论：

①RLS 是收敛的，且不存在额外误差项 $J_{ex}(\infty)$

②一般情况下，$n=2M$ 大约可以收敛，收敛到明显小于 LMS 的最终误差值。

③RLS 算法运算量明显大于 LMS 算法。

快速RLS算法 和其他发展

- 格型类算法
- QR和反QR算法
- 快速阵列算法（ Fast Array Algorithm）
- Robust Adaptive Filters
- H^{∞} Filter

深入阅读

- A. H. Sayed, Adaptive Filters, Wiley-Interscience, 2008
- S. Haykin, Adaptive Filter Theory, Fouth Edition,2001