chap 7 最小二乘自适应滤波

💡 一 掌握以下概念：线性 LS 估计问题，正交原理，正则方程

二 理解标准 RLS 自适应滤波器算法原理，存在的问题

三 掌握：最小二乘自适应滤波器的矢量空间分析基本方法， 正向预测和后向预测误差滤波的矢量空间分析基本方法，时间更新和阶次更新思路、 方法及推导过程；

理解：最小二乘滤波器的矢量空间分析中的投影矩阵和正交投影矩阵,角参量的物理意义

了解：LS 准则下的预测误差滤波器的格形结构， 最小二乘格形（LSL）自适应算法

四 了解快速横向滤波（FTF）自适应算法的算法原理， 理解横向滤波算子， 增益滤波器的概念
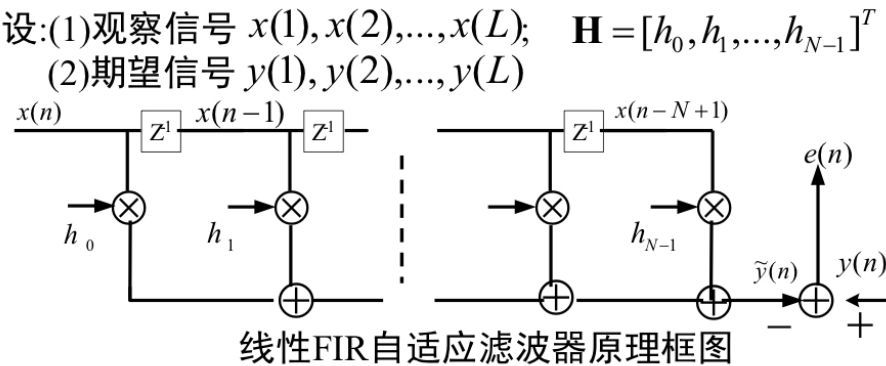
最小二乘法

线性LS估计问题

LMS的缺点：失调较大，需要信号平稳

解决方法：LS(Least Square，最小二乘)，以时间均值代替统计均值

$$J(\mathbf{H}) = \sum_{n=n_1}^{n=n_2} e^2(n) = \sum_{n=n_1}^{n=n_2} [(y(n) - \tilde{y}(n)]^2 = \min$$

$$J(\mathbf{H}) = E[e^2(n)] = \min$$

设:(1)观察信号 $x(1), x(2), ..., x(L)$; $\quad \mathbf{H} = [h_0, h_1, ..., h_{N-1}]^T$
(2)期望信号 $y(1), y(2), ..., y(L)$

线性FIR自适应滤波器原理框图
线性FIR自适应滤波器的输出:

$$\tilde{y}(n) = \sum_{k=0}^{N-1} h_k x(n-k) \quad e(n) = y(n) - \tilde{y}(n)$$

$$J(\mathbf{H}) = \sum_{n=n_1}^{n=n_2} e^2(n) = \sum_{n=n_1}^{n=n_2} [(y(n) - \tilde{y}(n)]^2 = \min \quad \mathbf{H}_{LS} \leftarrow \textbf{LS Filter}$$

正交原理

$$\frac{\partial J(\mathbf{H})}{\partial \mathbf{H}} = 0 \Rightarrow \sum_{n=n_1}^{n=n_2} x(n-k)e(n) = 0, k = 0,1,...,N-1 \quad \underset{k=0}{\rule{2cm}{0.4pt}}$$

正交原理:LS滤波器的输入$x(n-k)$和误差$e(n)$正交, $k = 0,1,…,N-1$

推论1:滤波器的输出和误差$e(n)$正交

$$\sum_{n=n_1}^{n=n_2} \tilde{y}(n)e(n) = 0$$

推论2:LS滤波等价于将期望信号$y(n)$进行正交分解

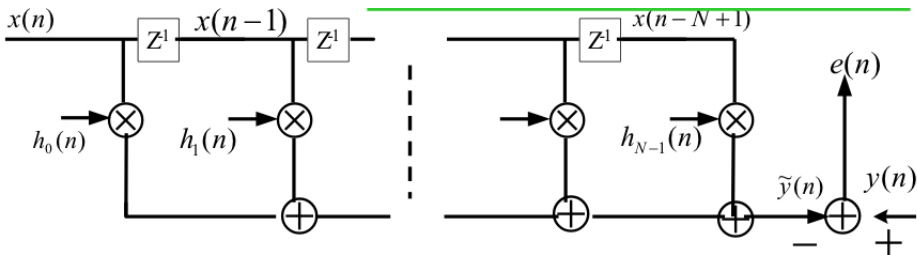$$\mathbf{y}(n) = \tilde{\mathbf{y}}(n) + \mathbf{e}(n)$$

正则方程

$$e(n) = y(n) - \widetilde{y}(n) = y(n) - \sum_{k=0}^{N-1} h_k x(n-k)$$

$$\sum_{n=n_1}^{n=n_2} x(n-k)e(n) = 0, k = 0,1,...,N-1 \qquad \downarrow \text{代入}$$

$$\sum_{m=0}^{N-1} h_m \sum_{n=n_1}^{n=n_2} x(n-k)x(n-m) = \sum_{n=n_1}^{n=n_2} x(n-k)y(n), k = 0,1,...,N-1$$

$$\sum_{m=0}^{N-1} h_m \underline{\Phi(m,k)} = \underline{z(-k)}, k = 0,1,...,N-1 \qquad \textbf{Normal Equation}$$

$$\mathbf{\Phi H} = \mathbf{z} \Rightarrow \mathbf{H}_{LS} = \mathbf{\Phi}^{-1}\mathbf{z}$$

标准 RLS 自适应滤波器

算法原理

基本思想: 假设在**n-1**时刻得到滤波器系数的**LS**估计,在**n**时刻新的数据到来后,按**LS**准则更新滤波器系数➔**RLS**

$$J[\mathbf{H}(n)] = \sum_{i=1}^{n} \lambda^{n-i} e^2(i) = \min \qquad \lambda \to \text{遗忘因子}$$

$$\mathbf{x}(i) = [x(i), x(i-1),...,x(i-N+1)]^T \qquad e(i) = y(i) - \widetilde{y}(i)$$

$$\mathbf{H}(n) = [h_0(n), h_1(n),...,h_{N-1}(n)]^T$$

$$\widetilde{y}(i) = \sum_{k=0}^{N-1} h_k(n)x(i-k) = \mathbf{H}^T(n)\mathbf{x}(i) = \mathbf{x}^T(i)\mathbf{H}(n)$$

初始化: $\mathbf{H}(0) = \mathbf{0}$

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

$$\delta = \begin{cases} small\ positive\ \text{contant for high SNR} \\ \text{large}\ positive\ \text{contant for low SNR} \end{cases}$$

叠代: n=1,2,... $\quad \boldsymbol{\pi}(n) = \mathbf{P}(n-1)\mathbf{x}(n), \qquad P(n) = \varphi^{-1}(n)$

$$\mathbf{k}(n) = \frac{\boldsymbol{\pi}(n)}{\lambda + \mathbf{x}^T(n)\boldsymbol{\pi}(n)}, \qquad \lambda: \text{遗忘因子}$$

$$\xi(n) = y(n) - \mathbf{H}^T(n-1)\mathbf{x}(n),$$

$$\mathbf{H}(n) = \mathbf{H}(n-1) + \mathbf{k}(n)\xi(n),$$

$$\mathbf{P}(n) = \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)$$

存在的问题

矢量空间分析

LS Adaptive Filter的矢量空间分析

$$\mathbf{X}_{0,N-1}(n) = \begin{bmatrix} x(1) & 0 & 0 & 0 \\ x(2) & x(1) & & \\ & & & \\ x(n-1) & x(n-2) & & x(n-N) \\ x(n) & x(n-1) & & x(n-N+1) \end{bmatrix}$$

以 $\mathbf{X}_{0,N-1}(\mathbf{n})$ 的N个列向量 $z^0\mathbf{x}(n), z^{-1}\mathbf{x}(n), ..., z^{-(N-1)}\mathbf{x}(n)$ 为基底，构成n维空间的N维子空间 $\{\mathbf{X}_{0,N-1}(\mathbf{n})\}$

$$\mathbf{x}(n) = [x(1), x(2), ..., x(n)]^T$$
$$z^{-j}\mathbf{x}(n) = [0, ..., x(1), x(2), ..., x(n-j)]^T$$
$$\mathbf{X}_{0,N-1}(n) = [z^0\mathbf{x}(n), z^{-1}\mathbf{x}(n), ..., z^{-(N-1)}\mathbf{x}(n)]$$
$$\hat{\mathbf{y}}(n) = \mathbf{X}_{0,N-1}(n)\mathbf{H}(n) = [z^0\mathbf{x}(n), z^{-1}\mathbf{x}(n), ..., z^{-(N-1)}\mathbf{x}(n)]\mathbf{H}(n)$$
$$= h_0(n)\mathbf{x}(n) + h_1(n)z^{-1}\mathbf{x}(n), ..., h_{N-1}(n)z^{-(N-1)}\mathbf{x}(n)$$

➡ 线性组合，估计是数据矢量空间的一个矢量

投影矩阵和正交投影矩阵

角参量的物理意义

新息的度量

时间更新和阶次更新

正向预测和后向预测误差滤波的矢量空间分析

$$e(i|n) = y(i) - \sum_{k=0}^{N-1} h_k(n)x(i-k), i=1,..,n$$

正向预测误差：

$$e_N^f(i) = x(i) - \hat{x}_f(i) = x(i) - \sum_{k=1}^{N} a_{Nk}x(i-k), 1 \le i \le n$$

$$\varepsilon_N^f(n) = \sum_{i=1}^{n}[e_N^f(i)]^2 \overset{\{a_{Nk}\}_{k=1}^{N}}{\Rightarrow} \min$$

定义：
$$\mathbf{e}_N^f(n) = [e_N^f(1), ..., e_N^f(i), ..., e_N^f(n)]^T$$
$$\mathbf{x}(n-1) = [x(1), x(2), ..., x(n-1)]^T \Leftarrow \mathbf{x}(n) \quad 输入信号矢量$$
$$\mathbf{x}(n) = [x(1), x(2), ..., x(n)]^T \Leftarrow \mathbf{y}(n) \quad 参考信号矢量$$
$$\hat{\mathbf{x}}_f(n) = [\hat{x}_f(1), ..., \hat{x}_f(i), ..., \hat{x}_f(n)]^T \Leftarrow \hat{\mathbf{y}}(n)$$
$$\mathbf{A}_N(n) = [a_{N1}(n), a_{N2}(n), ..., a_{NN}(n)]^T \Leftarrow \mathbf{H}(n)$$

$$\mathbf{X}_{1,N}(n) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ x(1) & 0 & & \\ & & & \\ x(n-2) & x(n-3) & & x(n-N-1) \\ x(n-1) & x(n-2) & & x(n-N) \end{bmatrix}$$

$$= [z^{-1}\mathbf{x}(n), z^{-2}\mathbf{x}(n), ..., z^{-N}\mathbf{x}(n)]^T$$

则：
$$\hat{\mathbf{x}}_f(n) = \mathbf{X}_{1,N}(n)\mathbf{A}_N(n) = \mathbf{P}_{1,N}(n)\mathbf{x}(n)$$
$$\mathbf{e}_N^f(n) = \mathbf{x}(n) - \mathbf{X}_{1,N}(n)\mathbf{A}_N(n) = \mathbf{P}_{1,N}^\perp(n)\mathbf{x}(n)$$
$$\{\mathbf{X}_{1,N}(n)\} \begin{cases} \mathbf{P}_{1,N}(n) = \mathbf{X}_{1,N}(n)\langle \mathbf{X}_{1,N}, \mathbf{X}_{1,N}\rangle^{-1}\mathbf{X}_{1,N}^T \\ \mathbf{P}_{1,N}^\perp(n) = \mathbf{I} - \mathbf{P}_{1,N}(n) \end{cases}$$
$$e_N^f(n) = \boldsymbol{\pi}^T(n)\mathbf{e}_N^f(n) = \langle \boldsymbol{\pi}(n), \mathbf{P}_{1,N}^\perp\mathbf{x}(n)\rangle$$
$$\varepsilon_N^f(n) = \sum_{i=1}^{n}[e_N^f(i)]^2 = \langle \mathbf{e}_N^f(n), \mathbf{e}_N^f(n)\rangle$$

后向预测误差滤波的矢量空间分析

反向（后向）预测误差:
$$e(i|n) = y(i) - \sum_{k=0}^{N-1} h_k(n)x(i-k), i=1,..,n$$

$$e_N^b(i) = x(i-N) - \hat{x}_b(i-N) = x(i-N) - \sum_{k=1}^{N} b_{Nk}x(i-N+k)$$

$$1 \le i \le n$$

$$\varepsilon_N^f(n) = \sum_{i=1}^{n}[e_N^b(i)]^2 \overset{\{b_{Nk}\}_{k=1}^N}{\Rightarrow} \min$$

定义：

$$\mathbf{e}_N^b(n) = [e_N^b(1),...,e_N^b(i),...,e_N^b(n)]^T$$

$$\mathbf{x}(n) = [x(1),x(2),...,x(n)]^T \quad \text{输入信号矢量}$$

$$\mathbf{x}_b(n-N) = [x(1-N),...,x(i-N),...,x(n-N)]^T \Leftarrow \mathbf{y}(n)$$

$$\hat{\mathbf{x}}_b(n-N) = [\hat{x}_b(1-N),...,\hat{x}_b(i-N),...,\hat{x}_b(n-N)]^T \Leftarrow \hat{\mathbf{y}}(n)$$

$$\mathbf{B}_N(n) = [b_{NN}(n),b_{N(N-1)}(n-1),...,b_{N1}(n)]^T \Leftarrow \mathbf{H}(n)$$

$$\mathbf{X}_{0,N-1}(n) = \begin{bmatrix} x(1) & 0 & 0 & 0 \\ x(2) & x(1) & & \\ & & & \\ x(n-1) & x(n-2) & & x(n-N) \\ x(n) & x(n-1) & & x(n-N+1) \end{bmatrix}$$

$$= [z^0\mathbf{x}(n), z^{-1}\mathbf{x}(n),...,z^{-(N-1)}\mathbf{x}(n)]^T$$

则：
$$\hat{\mathbf{x}}_b(n-N) = \mathbf{X}_{0,N-1}(n)\mathbf{B}_N(n) = \mathbf{P}_{0,N-1}(n)z^{-N}\mathbf{x}(n)$$

$$\mathbf{e}_N^b(n) = z^{-N}\mathbf{x}(n) - \hat{\mathbf{x}}_b(n-N) = \mathbf{P}_{0,N-1}^\perp z^{-N}\mathbf{x}(n)$$

$$\{\mathbf{X}_{0,N-1}(n)\}\begin{cases} \mathbf{P}_{0,N-1}(n) = \mathbf{X}_{0,N-1}(n)\langle \mathbf{X}_{0,N-1}, \mathbf{X}_{0,N-1}\rangle^{-1}\mathbf{X}_{0,N-1}^T \\ \mathbf{P}_{0,N-1}^\perp(n) = \mathbf{I} - \mathbf{P}_{0,N-1}(n) \end{cases}$$

$$e_N^b(n) = \boldsymbol{\pi}^T(n)\mathbf{e}_N^b(n) = \langle \boldsymbol{\pi}(n), \mathbf{P}_{0,N-1}^\perp z^{-N}\mathbf{x}(n)\rangle$$

$$\varepsilon_N^b(n) = \sum_{i=1}^{n}[e_N^b(i)]^2 = \langle \mathbf{e}_N^b(n), \mathbf{e}_N^b(n)\rangle$$

LS 准则下的预测误差滤波器的格形结构

$$e_{N+1}^f(n) = e_N^f(n) - k_{N+1}^b e_N^b(n-1)$$

$$e_{N+1}^b(n) = e_N^b(n-1) - k_{N+1}^f e_N^f(n)$$

最小二乘格形（LSL）自适应算法

算法总结：

1）初始化，N=1,2,,...P

$$e_N^b(0) = 0, \Delta_N(0) = 0, \gamma_N(0) = 1, \varepsilon_N^b(0) = \varepsilon_N^f(0) = \delta$$

For n=1,2,3,... Repeat 2) and 3):

2）n时刻初始化(零阶预测)（n=1,2,3,...）

$$e_0^f(n) = e_0^b(n) = x(n)$$

$$\varepsilon_0^b(n) = \varepsilon_0^f(n) = \varepsilon_0^f(n-1) + x^2(n)$$

$$\gamma_0(n) = 1$$

3）n时刻的阶次迭代（N=0,1,2,,...P-1）

$$\Delta_{N+1}(n) = \Delta_{N+1}(n-1) + \frac{e_N^f(n)e_N^b(n-1)}{\gamma_N(n-1)}$$

$$k_{N+1}^b(n) = \frac{\Delta_{N+1}(n)}{\varepsilon_N^b(n-1)} \quad k_{N+1}^f(n) = \frac{\Delta_{N+1}(n)}{\varepsilon_N^f(n)}$$

$$e_{N+1}^f(n) = e_N^f(n) - k_{N+1}^b e_N^b(n-1)$$

$$e_{N+1}^b(n) = e_N^b(n-1) - k_{N+1}^f e_N^f(n)$$

$$\varepsilon_{N+1}^f(n) = \varepsilon_N^f(n) - k_{N+1}^b(n)\Delta_{N+1}(n)$$

$$\varepsilon_{N+1}^b(n) = \varepsilon_N^b(n) - k_{N+1}^f(n)\Delta_{N+1}(n)$$

$$\gamma_{N+1}(n-1) = \gamma_N(n-1) - \frac{[e_N^b(n-1)]^2}{\varepsilon_N^b(n-1)}$$

快速横向滤波（FTF）自适应算法

算法原理

FTF自适应算法流程：

1 初始化

$$\mathbf{A}_N(0) = \mathbf{0}, \mathbf{B}_N(0) = \mathbf{0}, \mathbf{H}_N(0) = \mathbf{0}, \mathbf{G}_N(0) = 0, \gamma_N(0) = 1.0$$

$$\varepsilon^f(0) = \varepsilon^b(0) = \delta, 0 < \delta < 1$$

2 按时间叠代计算（n=1,2,…）

(1)前向预测误差滤波器参量的时间更新

$$e^f(n|n-1) = x(n) - \mathbf{x}_N^T(n-1)\mathbf{A}_N(n-1)$$

$$e^f(n|n) = \gamma_N(n-1)e^f(n|n-1)$$

$$\varepsilon^f(n) = \varepsilon^f(n-1) + e^f(n|n)e^f(n|n-1)$$

$$\mathbf{A}_N(n) = \mathbf{A}_N(n-1) + e^f(n|n-1)\mathbf{G}_N(n-1)$$

(2) N+1阶角参量的时间更新和阶次更新

$$\gamma_{N+1}(n) = \frac{\varepsilon^f(n-1)}{\varepsilon^f(n)}\gamma_N(n-1)$$

(3)N+1阶增益滤波器权矢量的时间更新和阶次更新

$$\mathbf{G}_{N+1}(n) = \begin{bmatrix} \mathbf{k}_N(n) \\ k(n) \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{G}_N(n-1) \end{bmatrix} + \frac{e^f(n|n)}{\varepsilon^f(n)}\begin{bmatrix} 1 \\ -\mathbf{A}_N(n) \end{bmatrix}$$

（4）后向预测误差滤波器参量，N阶角参量，N阶增益滤波器权矢量的时间更新

$$e^b(n|n-1) = x(n-N) - \mathbf{x}_N^T(n)\mathbf{B}_N(n-1)$$

$$\gamma_N(n) = [1 - k(n)e^b(n|n-1)]^{-1}\gamma_{N+1}(n)$$

$$e^b(n|n) = \gamma_N(n)e^b(n|n-1)$$

$$\varepsilon^b(n) = \varepsilon^b(n-1) + e^b(n|n)e^b(n|n-1)$$

$$\mathbf{G}_N(n) = [\mathbf{k}_N(n) + k(n)\mathbf{B}_N(n-1)]\frac{\gamma_N(n)}{\gamma_{N+1}(n)}$$

$$\mathbf{B}_N(n) = \mathbf{B}_N(n-1) + e^b(n|n-1)\mathbf{G}_N(n)$$

（5）最小二乘横向滤波器权矢量的时间更新

$$e(n|n-1) = y(n) - \mathbf{x}_N^T(n)\mathbf{H}_N(n-1)$$

$$\mathbf{H}(n) = \mathbf{H}(n-1) + e(n|n-1)\mathbf{G}_N(n)$$

FTF比LMS算法收敛速度快

运算量： 8N

横向滤波算子

$$\mathbf{H}(n) = [\mathbf{X}_{0,N-1}^{T}(n)\mathbf{X}_{0,N-1}(n)]^{-1}\mathbf{X}_{0,N-1}^{T}(n)\mathbf{y}(n)$$

$$\mathbf{K}_{0,N-1}(n) = [\mathbf{X}_{0,N-1}^{T}(n)\mathbf{X}_{0,N-1}(n)]^{-1}\mathbf{X}_{0,N-1}^{T}(n) \quad \text{横向滤波算子}$$

$$\mathbf{H}(n) = \mathbf{K}_{0,N-1}(n)\mathbf{y}(n) \quad \Longrightarrow \quad \mathbf{H}(n-1) = \mathbf{K}_{0,N-1}(n-1)\mathbf{y}(n-1) \overset{?}{\Longrightarrow} \mathbf{H}(n)$$

增益滤波器

$$\mathbf{G}_{N}(n) = \mathbf{K}_{0,N-1}(n)\boldsymbol{\pi}(n)$$

$\mathbf{G}_{N}(n)$是数据矩阵$\mathbf{X}_{0,N-1}(n)$对$\boldsymbol{\pi}(n)$的最小二乘估计器