第七章　最小二乘自适应滤波

(Least-Squares Adaptive Filters)

Motivation

- **LMS失调较大；**
- 非平稳时存在较大的跟踪误差，因此要求信号平稳或在较长的时间内具有平稳性

--------》不要用误差的样本值代替误差的统计均值

一种解决途径：以时间均值代替统计均值

•采用最小二乘准则;

•按时间进行准确迭代

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

$$\Uparrow$$

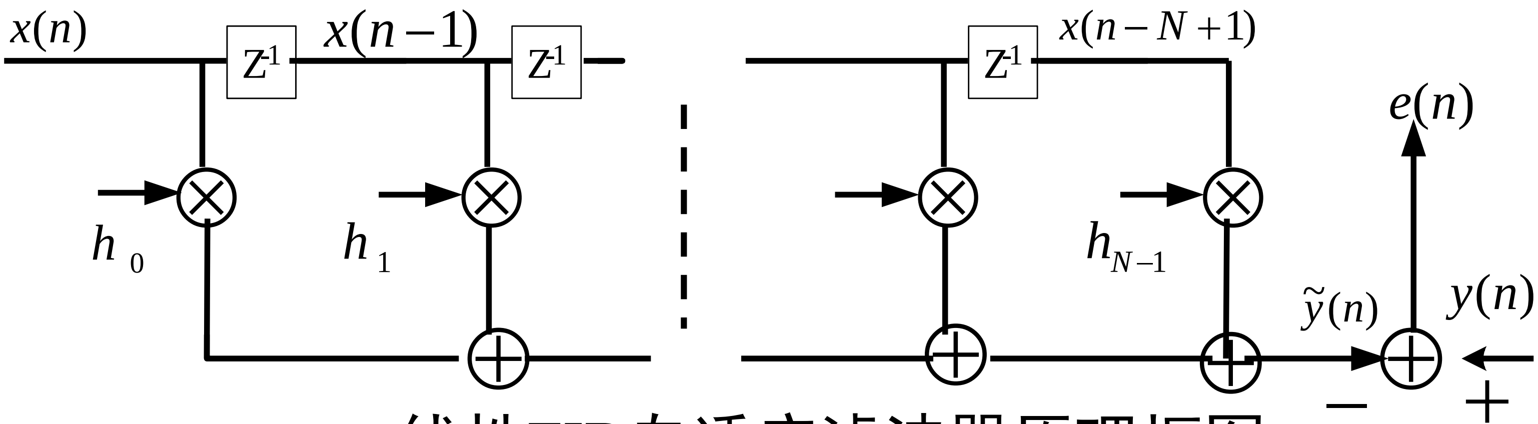
$$J(\mathbf{H})=E[e^2(n)]=\min$$

7.1　最小二乘法
(Method of Least-Squares)

一 线性LS估计问题

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

(2)期望信号 $y(1), y(2), \ldots, y(L)$



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:

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

$\mathbf{H}_{LS} \Leftarrow$ **LS Filter**

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

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

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

二 正交原理(Principle of Orthogonality)

$$\frac{\partial J(\mathbf{H})}{\partial \mathbf{H}}=0\Rightarrow\sum_{n=n_1}^{n=n_2}x(n-k)e(n)=0,k=0,1,\ldots,N-1$$

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

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

7.1　最小二乘法

$$\sum_{n=n_1}^{n=n_2} y^2(n) = \sum_{n=n_1}^{n=n_2} \tilde{y}^2(n) + \sum_{n=n_1}^{n=n_2} e^2(n)$$

$$\varepsilon_y = \varepsilon_{\tilde{y}} + J_{\min} \Rightarrow J_{\min} = \varepsilon_y - \varepsilon_{\tilde{y}}$$

三 正则方程(Normal Equation)

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

$$\sum_{n=n_1}^{n=n_2} x(n-k)e(n) = 0, k = 0,1,\ldots,N-1$$

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

$$\sum_{m=0}^{N-1} h_m \Phi(m,k) = z(-k), k = 0,1,\ldots, N-1$$

Normal Equation

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

7.1　最小二乗法

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

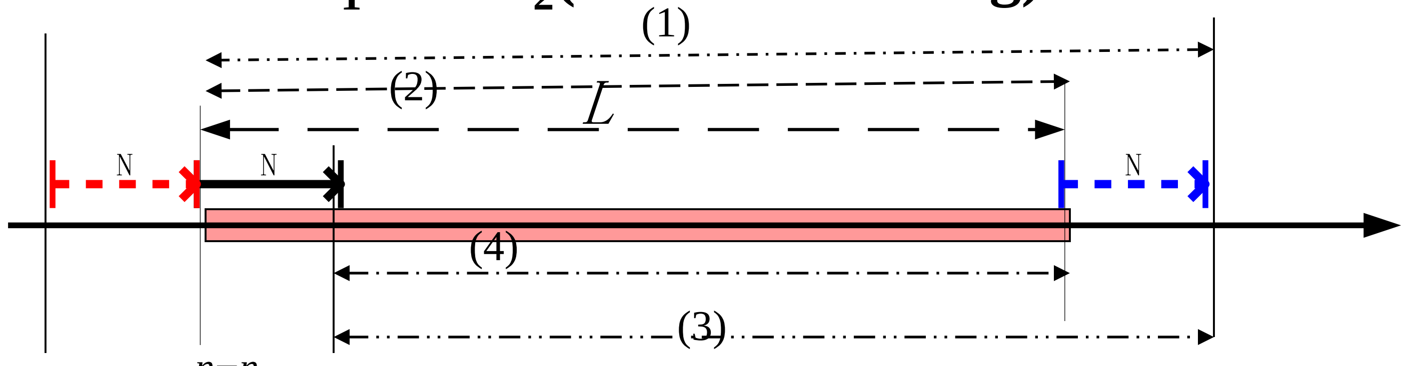
$$z(-k)=\sum_{n=n_1}^{n=n_2}x(n-k)y(n),k=0,1,\ldots,N-1$$

$$\mathbf{\Phi}=[\Phi(m,k)]_{N\times N}$$

$$\mathbf{z}=[z(0),z(-1),\ldots,z(-N+1)]^T$$

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

四 selection of n_1 and n_2(Data windowing)



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

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

1)n_1 =1, n_2=L+N-1,前后补零N-1(自相关法)
2)n_1 =1, n_2=L,前补零N-1,后不补零(前加窗法)
3)n_1 =N, n_2=L+N-1,前不补零后补零N-1 (后加窗法)
4)n_1 =N, n_2=L,前后不补零(协方差法)

五 Minimum Sum of Error Squares

$$\varepsilon_{\tilde{y}} = \sum_{n=n_1}^{n=n_2} \tilde{y}^2(n) = \sum_{n=n_1}^{n=n_2} [\sum_{\mathrm{k}=0}^{\mathrm{N}-1} h_k x(n-k)]^2$$

$$= \mathbf{H}^T \mathbf{\Phi} \mathbf{H} = \mathbf{H}^T \mathbf{z} = \mathbf{z}^T \mathbf{H}$$

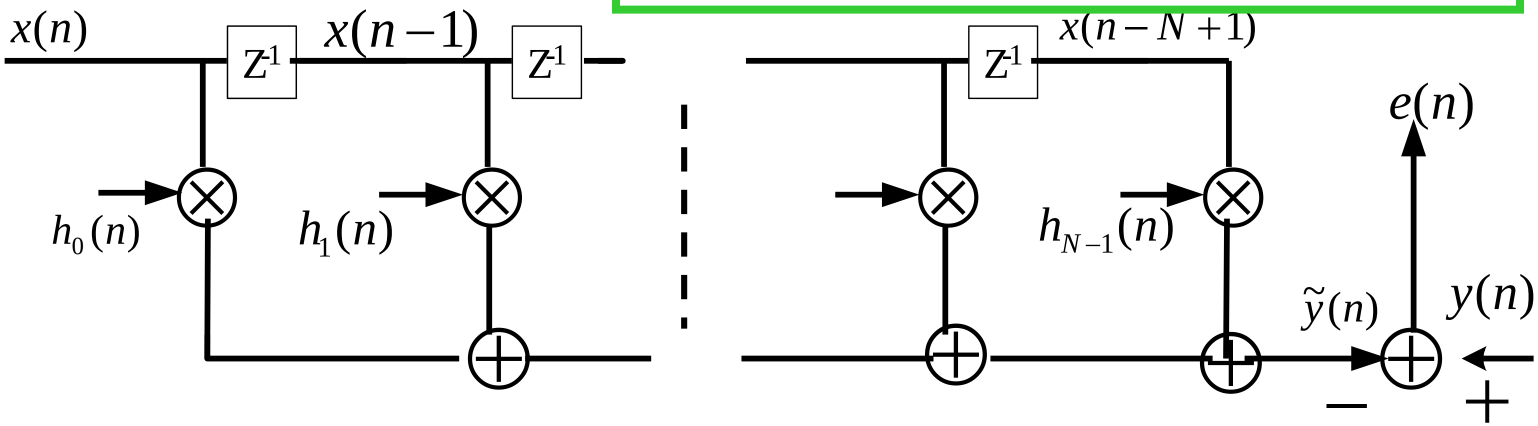
$$\boxed{\mathbf{H}_{LS} = \mathbf{\Phi}^{-1} \mathbf{z}}$$

$$J_{\min} = \varepsilon_y - \mathbf{z}^T \mathbf{H} = \varepsilon_y - \mathbf{z}^T \mathbf{\Phi}^{-1} \mathbf{z}$$

7.2　标准RLS自适应滤波器

(Standard RLS Adaptive Filters)

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



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

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

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

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

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

$$\tilde{y}(i) = \mathbf{H}^T(n)\mathbf{x}(i) = \mathbf{x}^T(i)\mathbf{H}(n)$$

$$e(i) = y(i) - \tilde{y}(i)$$

$$\frac{\partial J[\mathbf{H}(n)]}{\partial \mathbf{H}(n)} = 0 \Rightarrow \sum_{i=1}^{n} \lambda^{n-i} e(i)\mathbf{x}(i) = 0$$

$$\underbrace{\left[\sum_{i=1}^{n} \lambda^{n-i} \mathbf{x}(i)\mathbf{x}^T(i)\right]}\mathbf{H}(n) = \underbrace{\sum_{i=1}^{n} \lambda^{n-i} y(i)\mathbf{x}(i)}$$

$$\mathbf{\Phi}(n)\mathbf{H}(n) = \mathbf{z}(n)$$

1) $\mathbf{\Phi}(n), \mathbf{z}(n)$的递推计算

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

$$= \lambda \sum_{i=1}^{n-1} \lambda^{n-1-i} \mathbf{x}(i)\mathbf{x}^T(i) + \mathbf{x}(n)\mathbf{x}^T(n)$$

$$\boxed{\mathbf{\Phi}(n) = \lambda\mathbf{\Phi}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n)}$$

$$\mathbf{z}(n)=\sum_{i=1}^{n}\lambda^{n-i}y(i)\mathbf{x}(i)=\lambda\sum_{i=1}^{n-1}\lambda^{n-1-i}\mathbf{x}(i)y(i)+\mathbf{x}(n)y(n)$$

$$\boxed{\mathbf{z}(n)=\lambda\mathbf{z}(n-1)+\mathbf{x}(n)y(n)}$$

2) $\mathbf{\Phi}^{-1}(n)$的递推计算

$$\boxed{\mathbf{\Phi}(n)=\lambda\mathbf{\Phi}(n-1)+\mathbf{x}(n)\mathbf{x}^{T}(n)}$$

矩阵恒等式: $(\mathbf{A}+\mathbf{BCD})^{-1}=\mathbf{A}^{-1}-\mathbf{A}^{-1}\mathbf{B}(\mathbf{C}+\mathbf{DA}^{-1}\mathbf{B})^{-1}\mathbf{DA}^{-1}$

$$\mathbf{P}(n)=\mathbf{\Phi}^{-1}(n) \qquad \mathbf{A}=\lambda\mathbf{\Phi}(n-1)$$

$$\mathbf{B}=\mathbf{x}(n);\mathbf{C}=1,\mathbf{D}=\mathbf{x}^{T}(n)$$

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

$$\mathbf{P}(n)=\lambda^{-1}\mathbf{P}(n-1)-\frac{\lambda^{-2}\mathbf{P}(n-1)\mathbf{x}(n)\mathbf{x}^{T}(n)\mathbf{P}(n-1)}{[1+\lambda^{-1}\mathbf{x}^{T}(n)\mathbf{P}(n-1)\mathbf{x}(n)]}$$

$$\mathbf{P}(n)=\lambda^{-1}\mathbf{P}(n-1)-\frac{\lambda^{-2}\mathbf{P}(n-1)\mathbf{x}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)}{[1+\lambda^{-1}\mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)]}$$

$$\mathbf{z}(n)=\lambda\mathbf{z}(n-1)+\mathbf{x}(n)y(n)$$

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

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

$$\mathbf{k}(n)=\mathbf{P}(n)\mathbf{x}(n)$$

3) $\mathbf{H}(n)$的递推计算

$$\mathbf{H}(n)=\mathbf{\Phi}^{-1}(n)\mathbf{z}(n)=\mathbf{P}(n)\mathbf{z}(n)$$

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

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

$$e(n)=y(n)-\mathbf{H}^T(n)\mathbf{x}(n)$$ ($e(n)$验后误差)

$$\boxed{\mathbf{z}(n) = \lambda \mathbf{z}(n-1) + \mathbf{x}(n)y(n)}$$

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

$$\begin{aligned}
\mathbf{H}(n) &= \mathbf{\Phi}^{-1}(n)\mathbf{z}(n) = \mathbf{P}(n)\mathbf{z}(n) \\
&= \{\underbrace{\lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)}\}\{\lambda \mathbf{z}(n-1) + \mathbf{x}(n)y(n)\} \\
&= \mathbf{P}(n-1)\mathbf{z}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{z}(n-1) + \mathbf{P}(n)\mathbf{x}(n)y(n) \\
&= \mathbf{H}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\mathbf{H}(n-1) + \mathbf{k}(n)y(n) \\
&= \mathbf{H}(n-1) + \mathbf{k}(n)[y(n) - \mathbf{H}^T(n-1)\mathbf{x}(n)]
\end{aligned}$$

$$\boxed{\mathbf{z}(n)=\lambda\mathbf{z}(n-1)+\mathbf{x}(n)y(n)} \quad \boxed{\mathbf{H}(n)=\mathbf{H}(n-1)+\mathbf{k}(n)\xi(n)}$$

4) $J[\mathbf{H}(n)]=\min\ =J_{RLS}$ 的递推计算

$$J_{\min}=\varepsilon_y-\mathbf{z}^T\mathbf{H}_{LS} \qquad\qquad \varepsilon_y=\sum_{i=1}^{n}\lambda^{n-i}y^2(i)$$

$$J_{\min}(n)=\varepsilon_y(n)-\mathbf{z}^T(n)\mathbf{H}(n) \qquad \{\mathbf{H}_{RLS}(n)\}$$

$$\varepsilon_y(n)=\lambda\varepsilon_y(n-1)+y^2(n)$$

$$J_{\min}(n)=\varepsilon_y(n)-\mathbf{z}^T(n)\mathbf{H}_{RLS}(n)$$

$$=\lambda\varepsilon_y(n-1)+y^2(n)-\underbrace{[\lambda\mathbf{z}(n-1)+\mathbf{x}(n)y(n)]^T}$$

$$[\mathbf{H}(n-1)+\mathbf{k}(n)\xi(n)]$$

$$=\lambda[\varepsilon_y(n-1)-\mathbf{z}^T(n-1)\mathbf{H}(n-1)]$$

$$+y(n)[y(n)-\mathbf{x}^T(n)\mathbf{H}(n-1)]-\mathbf{z}^T(n)\mathbf{k}(n)\xi(n)$$

$$\mathbf{z}^T(n)\mathbf{k}(n) = \mathbf{z}^T(n)\mathbf{\Phi}^{-1}(n)\mathbf{x}(n)$$
$$= [\mathbf{\Phi}^{-1}(n)\mathbf{z}(n)]^T \mathbf{x}(n)$$
$$= \mathbf{H}^T(n)\mathbf{x}(n)$$

$$\boxed{J_{\min}(n) = \varepsilon_y(n) - \mathbf{z}^T(n)\mathbf{H}(n)}$$

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

$$J_{\min}(n) = \lambda[\varepsilon_y(n-1) - \mathbf{z}^T(n-1)\mathbf{H}(n-1)]$$
$$+ y(n)[y(n) - \mathbf{x}^T(n)\mathbf{H}(n-1)] - \mathbf{H}^T(n)\mathbf{x}(n)\xi(n)$$
$$= \lambda J_{\min}(n-1) + y(n)\xi(n) - \mathbf{H}^T(n)\mathbf{x}(n)\xi(n)$$
$$= \lambda J_{\min}(n-1) + [y(n) - \mathbf{H}^T(n)\mathbf{x}(n)]\xi(n)$$
$$= \lambda J_{\min}(n-1) + e(n)\xi(n)$$

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

$$\boxed{J_{\min}(n) = \lambda J_{\min}(n-1) + e(n)\xi(n)}$$

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

5) RLS算法总结

初始化： $\mathbf{H}(0) = \mathbf{0}$

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

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

叠代：n=1,2,… $\boldsymbol{\pi}(n) = \mathbf{P}(n-1)\mathbf{x}(n),$

$$\mathbf{k}(n) = \frac{\boldsymbol{\pi}(n)}{\lambda + \mathbf{x}^T(n)\boldsymbol{\pi}(n)},$$

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

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

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