

Method of Least Squares

The LS filtering method is a deterministic method. The performance criteria is the sum of squared errors produced by the filter over a finite set of (training) data

- The method is related to linear regression
- Optimization procedure results in a LS best fit for the filter over the samples in the optimization (training) set.

Consider the linear transversal filter

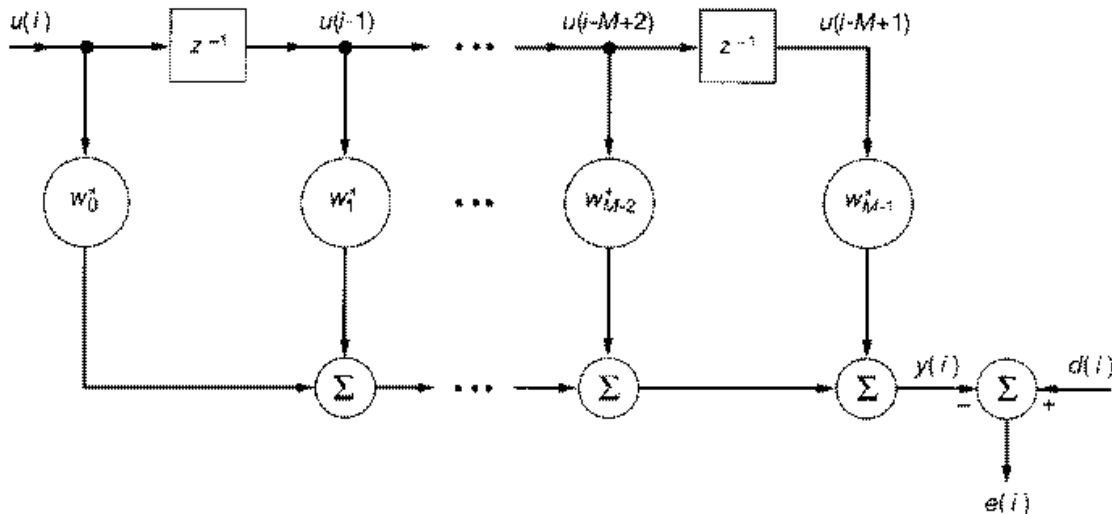


Figure 11.2 Linear transversal filter model.

Given

M — the number of taps in the filter

$\{x(i)\}$ — input sequence

$\{d(i)\}$ — desired output sequence

for $i = 1, 2, \dots, N$. The goal of the LS

method is to set the tap weights such that

sum of squared errors

$$\mathcal{E}(\mathbf{w}) = \sum_{i=M}^N |e(i)|^2$$

is minimized.

Let $\mathbf{w} = [w_0, w_1, \dots, w_{M-1}]^T$ be the weight vector and

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

be the observation vector for $M \leq i \leq N$.

Then the error of time i is

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

The full set of error values can be compiled into a vector.

Define

$$\boldsymbol{\varepsilon}^H = [e(M), e(M+1), \dots, e(N)]$$

Then $\boldsymbol{\varepsilon}$ is the $(N - M + 1) \times 1$ error vector.

Similarly, the $(N - M + 1) \times 1$ desired vector \mathbf{d} can be defined as

$$\mathbf{d}^H = [d(M), d(M + 1), \dots, d(N)]$$

If the filter output is denoted as $\hat{d}(i)$, then combining the filter output values in a vector yields

$$\begin{aligned}\hat{\mathbf{d}}^H &= [\hat{d}(M), \hat{d}(M + 1), \dots, \hat{d}(N)] \\ &= [\mathbf{w}^H \mathbf{x}(M), \mathbf{w}^H \mathbf{x}(M + 1), \dots, \mathbf{w}^H \mathbf{x}(N)] \\ &= \mathbf{w}^H [\mathbf{x}(M), \mathbf{x}(M + 1), \dots, \mathbf{x}(N)] \\ &= \mathbf{w}^H \mathbf{A}^H\end{aligned}$$

where

$$\mathbf{A}^H = [\mathbf{x}(M), \mathbf{x}(M + 1), \dots, \mathbf{x}(N)]$$

is the observation data matrix.

Expanding the data matrix

$$\begin{aligned}\mathbf{A}^H &= [\mathbf{x}(M), \mathbf{x}(M+1), \dots, \mathbf{x}(N)] \\ &= \begin{bmatrix} x(M) & x(M+1) & \cdots & x(N) \\ x(M-1) & x(M) & \cdots & x(N-1) \\ \vdots & \vdots & \ddots & \vdots \\ x(1) & x(2) & \cdots & x(N-M+1) \end{bmatrix}\end{aligned}$$

We see that \mathbf{A}^H is a $M \times (N - M + 1)$

rectangular toplitz matrix

Combining all the above we have

filter output vector: $\hat{\mathbf{d}}^H = \mathbf{w}^H \mathbf{A}^H$

desired output vector: \mathbf{d}^H

error vector: $\begin{aligned}\boldsymbol{\varepsilon}^H &= \mathbf{d}^H - \hat{\mathbf{d}}^H \\ &= \mathbf{d}^H - \mathbf{w}^H \mathbf{A}^H\end{aligned}$

The sum of the squared estimate errors can now be written as

$$\begin{aligned}\mathcal{E}(\mathbf{w}) &= \sum_{i=M}^N |e(i)|^2 \\ &= \boldsymbol{\varepsilon}^H \boldsymbol{\varepsilon} \\ &= (\mathbf{d}^H - \mathbf{w}^H \mathbf{A}^H)(\mathbf{d} - \mathbf{A}\mathbf{w}) \\ &= \mathbf{d}^H \mathbf{d} - \mathbf{d}^H \mathbf{A}\mathbf{w} - \mathbf{w}^H \mathbf{A}^H \mathbf{d} + \mathbf{w}^H \mathbf{A}^H \mathbf{A}\mathbf{w}\end{aligned}$$

minimizing with respect to \mathbf{w} ,

$$\frac{\partial \mathcal{E}(\mathbf{w})}{\partial \mathbf{w}} = -2\mathbf{A}^H \mathbf{d} + 2\mathbf{A}^H \mathbf{A}\mathbf{w}$$

setting to zero gives the optimal LS

weights $\hat{\mathbf{w}}$

$$\Rightarrow \quad \mathbf{A}^H \mathbf{A} \hat{\mathbf{w}} = \mathbf{A}^H \mathbf{d} \quad \left(\begin{array}{c} \text{Deterministic} \\ \text{normal equation} \end{array} \right)$$

while \mathbf{A} is not generally square, and thus not invertible, $\mathbf{A}^H \mathbf{A}$ is square and generally invertible.

Thus

$$\begin{aligned}\mathbf{A}^H \mathbf{A} \hat{\mathbf{w}} &= \mathbf{A}^H \mathbf{d} \\ \Rightarrow \hat{\mathbf{w}} &= (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{d}\end{aligned}$$

Note that the deterministic can be rearranged as

$$\begin{aligned}\mathbf{A}^H \mathbf{A} \hat{\mathbf{w}} - \mathbf{A}^H \mathbf{d} &= \mathbf{0} \\ \mathbf{A}^H (\mathbf{A} \hat{\mathbf{w}} - \mathbf{d}) &= \mathbf{0} \quad \text{or using } \boldsymbol{\varepsilon}_{\min} = \mathbf{d} - \mathbf{A} \hat{\mathbf{w}} \\ \mathbf{A}^H \boldsymbol{\varepsilon}_{\min} &= \mathbf{0}\end{aligned}$$

Thus the LS orthogonality principle states that the estimate error $\boldsymbol{\varepsilon}_{\min}$ is orthogonal to the row vectors of the data matrix \mathbf{A}^H .

Or in its expanded form

$$\mathbf{A}^H \boldsymbol{\varepsilon}_{\min} = \mathbf{0}$$

$$\begin{bmatrix} x(M) & x(M+1) & \cdots & x(N) \\ x(M-1) & x(M) & \cdots & x(N-1) \\ \vdots & \vdots & \ddots & \vdots \\ x(1) & x(2) & \cdots & x(N-M+1) \end{bmatrix} \begin{bmatrix} e_{\min}(M) \\ e_{\min}(M+1) \\ \vdots \\ e_{\min}(N) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

We can also derive the minimum sum of squared errors, which results when $\hat{\mathbf{w}}$ is used.

$$\begin{aligned} e_{\min} &= \boldsymbol{\varepsilon}_{\min}^H \boldsymbol{\varepsilon}_{\min} \\ &= (\mathbf{d}^H - \hat{\mathbf{w}}^H \mathbf{A}^H)(\mathbf{d} - \mathbf{A}\hat{\mathbf{w}}) \\ &= \mathbf{d}^H \mathbf{d} - \hat{\mathbf{w}}^H \mathbf{A}^H \mathbf{d} - \mathbf{d}^H \mathbf{A}\hat{\mathbf{w}} + \hat{\mathbf{w}}^H \mathbf{A}^H \mathbf{A}\hat{\mathbf{w}} \end{aligned}$$

Utilizing the normal equations we have

$$\hat{\mathbf{w}}^H \mathbf{A}^H \mathbf{d} = \hat{\mathbf{w}}^H \mathbf{A}^H \mathbf{A} \hat{\mathbf{w}}$$

Thus,

$$\begin{aligned} e_{\min} &= \mathbf{d}^H \mathbf{d} - \underbrace{\hat{\mathbf{w}}^H \mathbf{A}^H \mathbf{d}}_{\hat{\mathbf{w}}^H \mathbf{A}^H \mathbf{A} \hat{\mathbf{w}}} - \mathbf{d}^H \mathbf{A} \hat{\mathbf{w}} + \hat{\mathbf{w}}^H \mathbf{A}^H \mathbf{A} \hat{\mathbf{w}} \\ &= \mathbf{d}^H \mathbf{d} - \mathbf{d}^H \mathbf{A} \hat{\mathbf{w}} \end{aligned}$$

or using $\hat{\mathbf{w}} = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{d}$

$$e_{\min} = \underbrace{\mathbf{d}^H \mathbf{d}}_{\sum_{i=M}^N |d(i)|^2} - \mathbf{d}^H \mathbf{A} (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{d}$$

energy of
desired response

Consider again the deterministic normal equation

$$\mathbf{A}^H \mathbf{A} \hat{\mathbf{w}} = \mathbf{A}^H \mathbf{d}$$

Note that

$$\mathbf{A}^H \mathbf{A} = [\mathbf{x}(M), \mathbf{x}(M+1), \dots, \mathbf{x}(N)] \begin{bmatrix} \mathbf{x}^H(M) \\ \mathbf{x}^H(M+1) \\ \vdots \\ \mathbf{x}^H(N) \end{bmatrix}$$

$$= \sum_{i=M}^N \mathbf{x}(i) \mathbf{x}^H(i)$$

$$= \mathbf{\Phi}$$

↑
Time averaged
correlation matrix
size $M \times M$

From $\Phi = \sum_{i=M}^N \mathbf{x}(i)\mathbf{x}^H(i)$

We see

- 1) Φ is Hermetian
- 2) Φ is nonnegative definite

since for any \mathbf{a}

$$\begin{aligned}\mathbf{a}^H \Phi \mathbf{a} &= \sum_{i=M}^N \mathbf{a}^H \mathbf{x}(i)\mathbf{x}^H(i)\mathbf{a} \\ &= \sum_{i=M}^N [\mathbf{a}^H \mathbf{x}(i)][\mathbf{a}^H \mathbf{x}(i)]^H \\ &= \sum_{i=M}^N |\mathbf{a}^H \mathbf{x}(i)|^2 \geq 0\end{aligned}$$

- 3) From 1) and 2) we can derive that the eigenvalues of Φ are real and nonnegative

The deterministic normal equation,

$$\mathbf{A}^H \mathbf{A} \hat{\mathbf{w}} = \mathbf{A}^H \mathbf{d}$$

also gives

$$\mathbf{A}^H \mathbf{d} = [\mathbf{x}(M), \mathbf{x}(M+1), \dots, \mathbf{x}(N)] \begin{bmatrix} d^*(M) \\ d^*(M+1) \\ \vdots \\ d^*(N) \end{bmatrix}$$

$$= \sum_{i=M}^N \mathbf{x}(i) d^*(i)$$

$$= \boldsymbol{\theta}$$

↑
Time averaged
cross-correlation vector
size $M \times 1$

Thus the deterministic normal equation can be expressed as

$$\mathbf{A}^H \mathbf{A} \hat{\mathbf{w}} = \mathbf{A}^H \mathbf{d}$$

$$\mathbf{\Phi} \hat{\mathbf{w}} = \boldsymbol{\theta}$$

and since $\mathbf{\Phi}$ is usually positive definite (always positive semi-definite)

$$\hat{\mathbf{w}} = \mathbf{\Phi}^{-1} \boldsymbol{\theta}$$

Also, e_{\min} can now be expressed as

$$e_{\min} = \mathbf{d}^H \mathbf{d} - \underbrace{\mathbf{d}^H \mathbf{A}}_{\boldsymbol{\theta}^H} \underbrace{(\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{d}}_{\hat{\mathbf{w}}}$$

$$= \mathbf{d}^H \mathbf{d} - \boldsymbol{\theta}^H \hat{\mathbf{w}}$$

$$= e_d - \boldsymbol{\theta}^H \hat{\mathbf{w}} = e_d - \boldsymbol{\theta}^H \mathbf{\Phi}^{-1} \boldsymbol{\theta}$$

↑
Energy of desired signal

Consider again the orthogonality principle

$$\mathbf{A}^H \boldsymbol{\varepsilon}_{\min} = \mathbf{0}$$

For $\hat{\mathbf{d}}$ the estimate of \mathbf{d} utilizing $\hat{\mathbf{w}}$, i.e.

$\hat{\mathbf{d}} = \hat{\mathbf{w}}^H \mathbf{A}^H$, we have

$$\mathbf{A}^H \boldsymbol{\varepsilon}_{\min} = \mathbf{0}$$

$$\Rightarrow \hat{\mathbf{w}}^H \mathbf{A}^H \boldsymbol{\varepsilon}_{\min} = \hat{\mathbf{w}}^H \mathbf{0}$$

$$\hat{\mathbf{d}} \boldsymbol{\varepsilon}_{\min} = \mathbf{0}$$

Thus, the minimum estimation error vector,

$\boldsymbol{\varepsilon}_{\min}$, is orthogonal to the data matrix \mathbf{A}^H

and the LS estimate $\hat{\mathbf{d}}$.

Analysis of LS solution

Suppose that the true underlying system is linear,

$$\begin{aligned}d(i) &= \sum_{k=0}^{M-1} w_{0k}^* x(i-k) + e_0(i) \\ &= \mathbf{w}_0^H \mathbf{x}(i) + e_0(i)\end{aligned}$$

where $e_0(i)$ is the unobservable measurement error,

$$E\{e_0(i)\} = 0$$

and

$$E\{e_0(i)e_0^*(k)\} = \begin{cases} \sigma^2 & i = k \\ 0 & i \neq k \end{cases}$$

That is, $e_0(i)$ is white with zero mean and variance σ^2 .

Expressing the desired signal in vector form

$$\mathbf{d} = \mathbf{A}\mathbf{w}_0 + \boldsymbol{\varepsilon}_0$$

where $\boldsymbol{\varepsilon}_0^H = [e_0(M), e_0(M+1), \dots, e_0(N)]$

Recall that

$$\hat{\mathbf{w}} = (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{d}$$

using $\mathbf{d} = \mathbf{A}\mathbf{w}_0 + \boldsymbol{\varepsilon}_0$ in the above

$$\begin{aligned}\hat{\mathbf{w}} &= (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{A}\mathbf{w}_0 + (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \boldsymbol{\varepsilon}_0 \\ &= \mathbf{w}_0 + (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \boldsymbol{\varepsilon}_0\end{aligned}$$

Since \mathbf{A} is fixed, taking the expectation yields

$$\begin{aligned}E\{\hat{\mathbf{w}}\} &= \mathbf{w}_0 + (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H E\{\boldsymbol{\varepsilon}_0\} \\ &= \mathbf{w}_0\end{aligned}$$

- The LS estimate, $\hat{\mathbf{w}}$, is unbiased.

Consider next the covariance of $\hat{\mathbf{w}}$.

Note that

$$\begin{aligned}\hat{\mathbf{w}} &= \mathbf{w}_0 + (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \boldsymbol{\varepsilon}_0 \\ \Rightarrow \hat{\mathbf{w}} - \mathbf{w}_0 &= (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \boldsymbol{\varepsilon}_0\end{aligned}$$

and

$$\begin{aligned}\text{cov}[\hat{\mathbf{w}}] &= E\{(\hat{\mathbf{w}} - \mathbf{w}_0)(\hat{\mathbf{w}} - \mathbf{w}_0)^H\} \\ &= E\{(\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \boldsymbol{\varepsilon}_0 \boldsymbol{\varepsilon}_0^H \mathbf{A} (\mathbf{A}^H \mathbf{A})^{-1}\} \\ &= \mathbf{\Phi}^{-1} \mathbf{A}^H \underbrace{E\{\boldsymbol{\varepsilon}_0 \boldsymbol{\varepsilon}_0^H\}}_{\sigma^2 \mathbf{I}} \mathbf{A} \mathbf{\Phi}^{-1} \\ &= \sigma^2 \mathbf{\Phi}^{-1} \mathbf{\Phi} \mathbf{\Phi}^{-1} \\ &= \sigma^2 \mathbf{\Phi}^{-1}\end{aligned}$$

- The covariance of $\hat{\mathbf{w}}$ is proportional to the variance of the measurement noise and the inverse of the time average correlation matrix.

Now we will show that the LS estimate $\hat{\mathbf{w}}$ is the best linear unbiased estimate.

Consider any linear unbiased estimate $\tilde{\mathbf{w}}$, which we can write as

$$\tilde{\mathbf{w}} = \mathbf{B}\mathbf{d}$$

\uparrow
 $M \times (N - M + 1)$ matrix

Substituting $\mathbf{d} = \mathbf{A}\mathbf{w}_0 + \boldsymbol{\varepsilon}_0$ in the above,

$$\tilde{\mathbf{w}} = \mathbf{B}\mathbf{A}\mathbf{w}_0 + \mathbf{B}\boldsymbol{\varepsilon}_0$$

Taking the expectation,

$$\begin{aligned} E\{\tilde{\mathbf{w}}\} &= \mathbf{B}\mathbf{A}\mathbf{w}_0 \\ \Rightarrow \mathbf{B}\mathbf{A} &= \mathbf{I} \end{aligned}$$

for $\tilde{\mathbf{w}}$ to be unbiased.

Thus $\tilde{\mathbf{w}} = \mathbf{w}_0 + \mathbf{B}\boldsymbol{\varepsilon}_0$

or

$$\tilde{\mathbf{w}} - \mathbf{w}_0 = \mathbf{B}\boldsymbol{\varepsilon}_0$$

and

$$\begin{aligned}\text{cov}[\tilde{\mathbf{w}}] &= E\{(\tilde{\mathbf{w}} - \mathbf{w}_0)(\tilde{\mathbf{w}} - \mathbf{w}_0)^H\} \\ &= E\{\mathbf{B}\boldsymbol{\varepsilon}_0\boldsymbol{\varepsilon}_0^H\mathbf{B}^H\} \\ &= \sigma^2\mathbf{B}\mathbf{B}^H\end{aligned}$$

Now define

$$\boldsymbol{\psi} = \mathbf{B} - (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H$$

which yields

$$\begin{aligned}\boldsymbol{\psi} &= [\mathbf{B} - \boldsymbol{\Phi}^{-1}\mathbf{A}^H][\mathbf{B}^H - \mathbf{A}\boldsymbol{\Phi}^{-1}] \\ &= \mathbf{B}\mathbf{B}^H - \underbrace{\mathbf{B}\mathbf{A}\boldsymbol{\Phi}^{-1}}_{\mathbf{I}} - \boldsymbol{\Phi}^{-1}\underbrace{\mathbf{A}^H\mathbf{B}^H}_{\mathbf{I}} + \underbrace{\boldsymbol{\Phi}^{-1}\mathbf{A}^H\mathbf{A}\boldsymbol{\Phi}^{-1}}_{\boldsymbol{\Phi}^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^{-1}} \\ &= \mathbf{B}\mathbf{B}^H - \boldsymbol{\Phi}^{-1} - \boldsymbol{\Phi}^{-1} + \boldsymbol{\Phi}^{-1} \\ &= \mathbf{B}\mathbf{B}^H - \boldsymbol{\Phi}^{-1} = \mathbf{B}\mathbf{B}^H - (\mathbf{A}^H\mathbf{A})^{-1}\end{aligned}$$

Note that the diagonal elements at $\Psi\Psi^H$ must be ≥ 0 .

$$\text{Thus } \Psi\Psi^H = \mathbf{B}\mathbf{B}^H - (\mathbf{A}^H \mathbf{A})^{-1}$$

$$\Rightarrow \text{diag}[\mathbf{B}\mathbf{B}^H] \geq \text{diag}[(\mathbf{A}^H \mathbf{A})^{-1}]$$

or multiplying by σ^2

$$\text{diag}[\sigma^2 \mathbf{B}\mathbf{B}^H] \geq \text{diag}[\sigma^2 (\mathbf{A}^H \mathbf{A})^{-1}]$$

but

$$\text{cov}[\hat{\mathbf{w}}] = \sigma^2 (\mathbf{A}^H \mathbf{A})^{-1}$$

$$\text{cov}[\tilde{\mathbf{w}}] = \sigma^2 \mathbf{B}\mathbf{B}^H$$

Thus the above states

$$\text{variance}[\tilde{w}_i] \geq \text{variance}[\hat{w}_i] \quad i = 1, 2, \dots, M$$

Thus the weights in $\hat{\mathbf{w}}$ have lower variance than any other linear estimates.

- The LS estimate $\hat{\mathbf{w}}$ is the Best Linear Unbiased Estimate (BLUE).

Recursive Least Squares (RLS) estimateion

In the LS approach, we chose the M element weight vector \mathbf{w} to minimize

$$\mathcal{E}(\mathbf{w}) = \sum_{i=M}^N |e(i)|^2$$

where $e(i) = d(i) - \mathbf{w}^H \mathbf{x}(i)$

Letting the observation vectors be written as

$$\mathbf{A}^H = [\mathbf{x}(M), \mathbf{x}(M+1), \dots, \mathbf{x}(N)]$$

$$= \begin{bmatrix} x(M) & x(M-1) & \cdots & x(N) \\ x(M-1) & x(M-2) & \cdots & x(N-1) \\ \vdots & \vdots & \ddots & \vdots \\ x(N) & x(N-1) & \cdots & x(N-M+1) \end{bmatrix}$$

and given the desired output

$$\mathbf{d}^H = [d(M), d(M+1), \dots, d(N)]$$

The LS solution is defined by

$$\mathbf{A}^H \mathbf{A} \hat{\mathbf{w}} = \mathbf{A}^H \mathbf{d}$$

or

$$\begin{aligned} \hat{\mathbf{w}} &= (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{d} \\ &= \mathbf{\Phi}^{-1} \boldsymbol{\theta} \end{aligned}$$

where

$$\mathbf{\Phi} = \sum_{i=M}^N \mathbf{x}(i) \mathbf{x}^H(i) \quad \boldsymbol{\theta} = \sum_{i=M}^N \mathbf{x}(i) d^*(i)$$

We now wish to form a recursive update for the weights such that we do not have to recompute $(\mathbf{A}^H \mathbf{A})^{-1}$ for each new observation sample

- $(\mathbf{A}^H \mathbf{A})$ is $M \times M$
- inversion requires $O(M^3)$ multiplications and additions
- we can use the matrix inversion lemma to reduce the number of computations
- weights are now a function of n

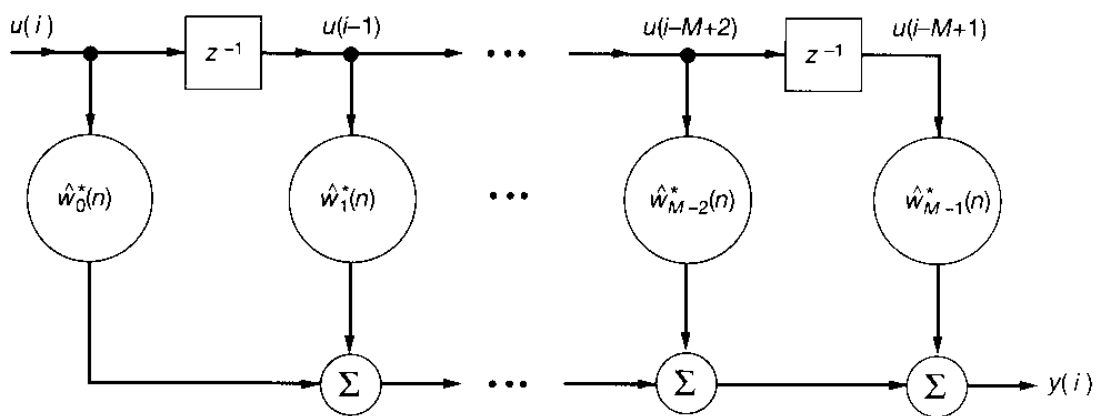


Figure 13.1 Transversal filter.

Let the observation sequence be $x(1), x(2), \dots, x(n)$ where we assume $x(l) = 0$ for $l \leq 0$.

Then the error is defined as

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

where

$$e(i) = d(i) - \mathbf{w}^H(n) \mathbf{x}(i)$$

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

$$\mathbf{w}(n) = [w_0(n), w_1(n), \dots, w_{M-1}(n)]^T$$

and $\beta(n, i) \in (0, 1]$ is a forgetting factor used in non-stationary statistic cases.

A commonly used forgetting factor is the exponential forgetting factor

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

where $\lambda \in (0, 1]$

Thus,

$$\varepsilon(n) = \sum_{i=1}^n \lambda^{n-i} |e(i)|^2$$

The LS solution to this problem is given by the normal equation

$$\Phi(n) \hat{\mathbf{w}}(n) = \boldsymbol{\theta}(n)$$

where now

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

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

The deterministic normal equation components can be updated recursively,

$$\begin{aligned}
 \Phi(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) \mathbf{x}^H(i) \\
 &= \lambda \underbrace{\left[\sum_{i=1}^{n-1} \lambda^{(n-1)-i} \mathbf{x}(i) \mathbf{x}^H(i) \right]}_{\Phi(n-1)} + \mathbf{x}(n) \mathbf{x}^H(n) \\
 &= \lambda \Phi(n-1) + \mathbf{x}(n) \mathbf{x}^H(n)
 \end{aligned}$$

Similarly

$$\begin{aligned}
 \theta(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) d^*(i) \\
 &= \lambda \left[\sum_{i=1}^{n-1} \lambda^{(n-1)-i} \mathbf{x}(i) d^*(i) \right] + \mathbf{x}(n) d^*(n) \\
 &= \lambda \theta(n-1) + \mathbf{x}(n) d^*(n)
 \end{aligned}$$

In order to obtain the LS solution, we need the inverse of $\Phi(n)$

matrix inversion lemma gives

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

size: $M \times M$ $M \times M$ $M \times L$ $L \times L$ $L \times M$

where \mathbf{A} , \mathbf{B} and \mathbf{D} are positive definite (non-singular)

Then

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B} \mathbf{C} [\mathbf{D} + \mathbf{C}^H \mathbf{B} \mathbf{C}]^{-1} \mathbf{C}^H \mathbf{B}$$

For

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

we have

$$\begin{aligned} \mathbf{A} &= \Phi(n) & (M \times M) & & \mathbf{C} &= \mathbf{x}(n) & (M \times 1) \\ \mathbf{B}^{-1} &= \lambda \Phi(n-1) & (M \times M) & & \mathbf{D} &= 1 & (1 \times 1) \end{aligned}$$

For

$$\begin{aligned}\mathbf{A} &= \Phi(n) & \mathbf{C} &= \mathbf{x}(n) \\ \mathbf{B}^{-1} &= \lambda\Phi(n-1) & \mathbf{D} &= 1\end{aligned}$$

and

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}[\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C}]^{-1}\mathbf{C}^H\mathbf{B}$$

Note that

$$[\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C}]^{-1} = [1 + \lambda^{-1}\mathbf{x}^H(n)\Phi^{-1}(n-1)\mathbf{x}(n)]^{-1}$$

is a scalar. Thus the above yields

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

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

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

Then ↑
└── Gain vector

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

The gain vector can be simplified further

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

\Rightarrow

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

Therefore

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

we must now derive an update for the tap weight vector. Recall,

$$\hat{\mathbf{w}}(n) = \mathbf{\Phi}^{-1}(n) \boldsymbol{\theta}(n) = \mathbf{P}(n) \boldsymbol{\theta}(n)$$

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

we have

$$\hat{\mathbf{w}}(n) = \lambda \mathbf{P}(n) \boldsymbol{\theta}(n-1) + \mathbf{P}(n) \mathbf{x}(n) d^*(n)$$

Using the update

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

in the first $\mathbf{P}(n)$ term of

$$\hat{\mathbf{w}}(n) = \lambda\mathbf{P}(n)\boldsymbol{\theta}(n-1) + \mathbf{P}(n)\mathbf{x}(n)d^*(n) \text{ gives}$$

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

$$+ \mathbf{P}(n)\mathbf{x}(n)d^*(n)$$

$$= \underbrace{\mathbf{P}(n-1)\boldsymbol{\theta}(n-1)}_{\hat{\mathbf{w}}(n-1)} - \mathbf{k}(n)\mathbf{x}^H(n)\underbrace{\mathbf{P}(n-1)\boldsymbol{\theta}(n-1)}_{\hat{\mathbf{w}}(n-1)}$$

$$+ \mathbf{P}(n)\mathbf{x}(n)d^*(n)$$

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

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

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

where $\alpha(n) = d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{x}(n)$

Notice the difference between $e(n)$ and $\alpha(n)$

$e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{x}(n)$ = a posteriori error

$\alpha(n) = d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{x}(n)$ = a priori error

Summary of RLS algorithm:

- 1) Given a new sample $\mathbf{x}(n)$, update the gain vector

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

- 2) Update the innovation

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

- 3) Update the tap weight vector

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

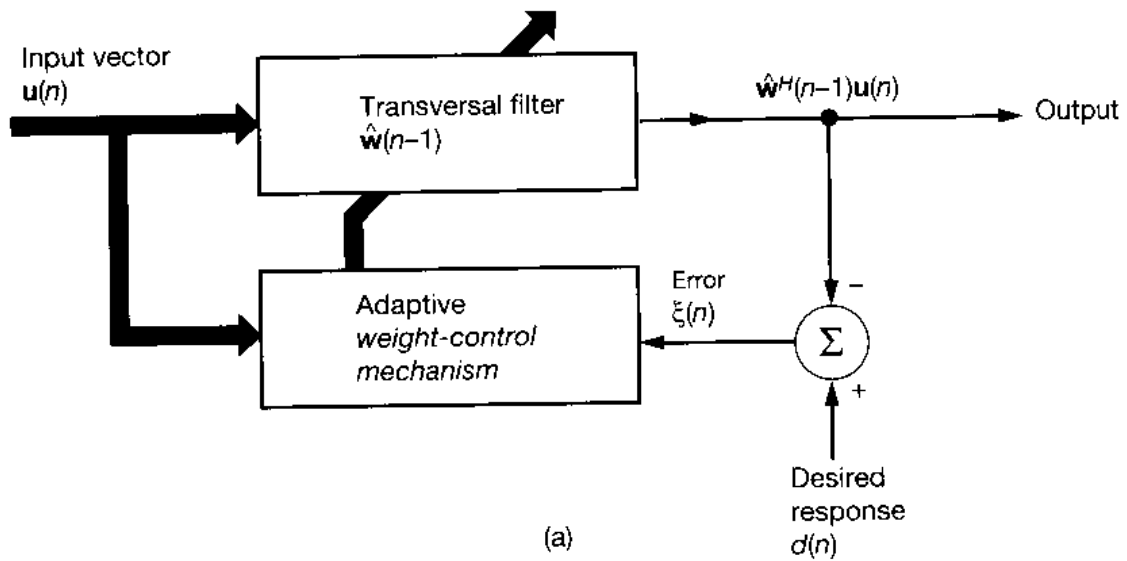
- 4) Update inverse correlation matrix

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

Initial conditions: $\hat{\mathbf{w}}(0) = \mathbf{0}$ and $\mathbf{P}(0) = \delta\mathbf{I}$

where δ is a small positive constant,

$$\delta \approx 0.01\sigma_x^2$$



Comparison between RLS and LMS algorithm terms:

Entity	RLS	LMS
Error	$\alpha(n) = d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{x}(n)$ (a priori error)	$e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{x}(n)$ (a posteriori error)
Weight Update	$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\alpha^*(n)$	$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{x}(n)e^*(n)$
Gain of error update	$\left(\frac{\lambda^{-1}\mathbf{P}(n-1)}{1 + \lambda^{-1}\mathbf{x}^H(n)\mathbf{P}(n-1)\mathbf{x}(n)} \right) \mathbf{x}(n)$	$(\mu)\mathbf{x}(n)$

Consider the complexity (number of additions and multiplies) for LMS, LS, and RLS algorithms.

- Assume the data is real and the filter is of size M .

For the LMS algorithm,

$$1) \quad \hat{d}(n) = \mathbf{w}^T(n)\mathbf{x}(n)$$

$$2) \quad e(n) = d(n) - \hat{d}(n)$$

$$3) \quad \mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{x}(n)e(n)$$

Complexity:

Stage	O_{\times}	O_{+}
1)	M	$M-1$
2)	0	1
3)	$M+1$	M
Total complexity per iteration	$O_{\times}(2M+1)$	$O_{+}(2M)$

The LS algorithm is given by

$$\mathbf{\Phi}(n)\hat{\mathbf{w}}(n) = \mathbf{\theta}(n).$$

For each new sample we have

$$1) \quad \mathbf{\Phi}(n+1) = \mathbf{\Phi}(n) + \mathbf{x}(n+1)\mathbf{x}^T(n+1)$$

$$2) \quad \mathbf{\theta}(n+1) = \mathbf{\theta}(n) + \mathbf{x}(n+1)d(n+1)$$

$$3) \quad \hat{\mathbf{w}}(n+1) = \mathbf{\Phi}^{-1}(n+1)\mathbf{\theta}(n+1)$$

Complexity:

Stage	O_{\times}	O_{+}
1)	M^2	M^2
2)	M	M
3)	$M^3 + M^2$	$M^3 + M(M-1)$
Total complexity per iteration	$O_{\times}(M^3 + 2M^2 + M)$	$O_{+}(M^3 + 2M^2)$

The RLS algorithm is (assume $\lambda = 1$) given by:

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

$$2) \quad \alpha(n) = d(n) - \hat{\mathbf{w}}^T(n-1)\mathbf{x}(n)$$

$$3) \quad \hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\alpha(n)$$

$$4) \quad \mathbf{P}(n) = \mathbf{P}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)$$

Note repeated operation:

$\mathbf{x}^T(n)\mathbf{P}(n-1)$ repeated steps are underlined complexity:

Stage	O_{\times}	O_{+}
1) numerator denominator division	M^2 <u>$M^2 + M$</u> M	$M(M-1)$ <u>$M(M-1) + M$</u>
2)	M	M
3)	M	M
4)	<u>$M^2 + M^2$</u>	<u>$M(M-1) + M^2$</u>
Total complexity per iteration	$O_{\times}(3M^2 + 4M)$	$O_{+}(3M^2 + M)$

Analysis of the RLS algorithm

Assume again that the desired signal is formed by the regression model

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

where $e_0(n)$ is white with variance σ^2

Assume $\lambda = 1$ and $n \geq M$, then

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

where

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

and

$$\boldsymbol{\theta}(n) = \sum_{i=1}^n \mathbf{x}(i)d^*(i)$$

substituting into $d(i)$ from above

$$\begin{aligned}\boldsymbol{\theta}(n) &= \sum_{i=1}^n \mathbf{x}(i)[e_0^*(i) + \mathbf{x}^H(i)\mathbf{w}_0] \\ &= \sum_{i=1}^n \mathbf{x}(i)\mathbf{x}^H(i)\mathbf{w}_0 + \sum_{i=1}^n \mathbf{x}(i)e_0^*(i) \\ &= \mathbf{\Phi}(n)\mathbf{w}_0 + \sum_{i=1}^n \mathbf{x}(i)e_0^*(i)\end{aligned}$$

Thus

$$\begin{aligned}\hat{\mathbf{w}}(n) &= \mathbf{\Phi}^{-1}(n)\boldsymbol{\theta}(n) \\ &= \mathbf{\Phi}^{-1}(n)\left[\mathbf{\Phi}(n)\mathbf{w}_0 + \sum_{i=1}^n \mathbf{x}(i)e_0^*(i)\right] \\ &= \mathbf{w}_0 + \mathbf{\Phi}^{-1}(n)\sum_{i=1}^n \mathbf{x}(i)e_0^*(i)\end{aligned}$$

note that $E\{A\} = E\{E\{A|B\}\}$ Thus

$$\begin{aligned}E\{\hat{\mathbf{w}}(n)\} &= \mathbf{w}_0 + E\left\{E\left\{\mathbf{\Phi}^{-1}(n)\sum_{i=1}^n \mathbf{x}(i)e_0^*(i) \mid x(i), i = 1, 2, \dots, n\right\}\right\} \\ &= \mathbf{w}_0 + E\left\{\mathbf{\Phi}^{-1}(n)\sum_{i=1}^n \mathbf{x}(i)E\{e_0^*(i)\}\right\} \\ &= \mathbf{w}_0\end{aligned}$$

Since $e_0(i)$ is independent of all observations and are the $x(i)$ terms are given, $\mathbf{\Phi}(n)$ is uniquely determined

- The RLS algorithm is convergent in the mean for $n \geq M$.
- How does this compare to the LMS algorithm?

Next, consider the convergence in the mean square. Using

$$\hat{\mathbf{w}}(n) = \mathbf{w}_0 + \mathbf{\Phi}^{-1}(n) \sum_{i=1}^n \mathbf{x}(i) e_0^*(i)$$

we have

$$\boldsymbol{\varepsilon}(n) = \hat{\mathbf{w}}(n) - \mathbf{w}_0 = \mathbf{\Phi}^{-1}(n) \sum_{i=1}^n \mathbf{x}(i) e_0^*(i)$$

and the weight error correlation matrix is

$$\mathbf{K}(n) = E\{\boldsymbol{\varepsilon}(n)\boldsymbol{\varepsilon}^H(n)\}$$

$$= E\left\{ \mathbf{\Phi}^{-1}(n) \left(\sum_{i=1}^n \sum_{j=1}^n \mathbf{x}(i) e_0^*(i) e_0(j) \mathbf{x}^H(j) \right) \mathbf{\Phi}^{-1}(n) \right\}$$

using $E\{A\} = E\{E\{A|B\}\}$ again,

$$\begin{aligned} \mathbf{K}(n) &= E\left\{ \mathbf{\Phi}^{-1}(n) \left(\sum_{i=1}^n \sum_{j=1}^n \mathbf{x}(i) \underbrace{E\{e_0^*(i) e_0(j)\}}_{\sigma^2 \delta(i-j)} \mathbf{x}^H(j) \right) \mathbf{\Phi}^{-1}(n) \right\} \\ &= \sigma^2 E\left\{ \mathbf{\Phi}^{-1}(n) \left(\sum_{i=1}^n \mathbf{x}(i) \mathbf{x}^H(i) \right) \mathbf{\Phi}^{-1}(n) \right\} \end{aligned}$$

$$\mathbf{K}(n) = \sigma^2 E\{\mathbf{\Phi}^{-1}(n)\mathbf{\Phi}(n)\mathbf{\Phi}^{-1}(n)\} = \sigma^2 E\{\mathbf{\Phi}^{-1}(n)\}$$

The matrix $\mathbf{\Phi}(n)$ has a Wishart distribution and the expectation of $\mathbf{\Phi}^{-1}(n)$ is

$$E\{\mathbf{\Phi}^{-1}(n)\} = \frac{1}{n-M-1} \mathbf{R}^{-1} \quad n > M+1$$

thus

$$\mathbf{K}(n) = \frac{\sigma^2}{n-M-1} \mathbf{R}^{-1} \quad n > M+1$$

and using the trace

$$\begin{aligned} E\{\|\boldsymbol{\varepsilon}(n)\|^2\} &= E\{\boldsymbol{\varepsilon}^H(n)\boldsymbol{\varepsilon}(n)\} \\ &= E\{\text{trace}[\boldsymbol{\varepsilon}^H(n)\boldsymbol{\varepsilon}(n)]\} = E\{\text{trace}[\boldsymbol{\varepsilon}(n)\boldsymbol{\varepsilon}^H(n)]\} \\ &= \text{trace}E\{\boldsymbol{\varepsilon}(n)\boldsymbol{\varepsilon}^H(n)\} = \text{trace}[\mathbf{K}(n)] \\ &= \frac{\sigma^2}{n-M-1} \text{trace}[\mathbf{R}^{-1}] \\ &= \frac{\sigma^2}{n-M-1} \sum_{i=1}^M \frac{1}{\lambda_i} \quad n > M+1 \end{aligned}$$

- The weight vector MSE is initially proportional to $1/\lambda_{\min}$
- The weight vector converges linearly in the mean squared sense

Now consider the learning (error) curve of the RLS algorithm. Recall the a priori estimation error

$$\begin{aligned}
 \alpha(n) &= d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{x}(n) \\
 &= e_0(n) + \mathbf{w}_0^H \mathbf{x}(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{x}(n) \\
 &= e_0(n) - \boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)
 \end{aligned}$$

Now consider the MSE of $\alpha(n)$

$$\begin{aligned}
 J'(n) &= E\{|\alpha(n)|^2\} \\
 &= E\{[e_0^*(n) - \mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)] \\
 &\quad [e_0(n) - \boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)]\} \\
 &= E\{|e_0(n)|^2\} - E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)e_0(n)\} \\
 &\quad - E\{\boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)e_0^*(n)\} \\
 &\quad + E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)\}
 \end{aligned}$$

Consider each term:

$$E\{|e_0(n)|^2\} = \sigma^2$$

Invoking the independence theorem, $\boldsymbol{\varepsilon}(n-1)$ is independent of $\mathbf{x}(n)$ and $e_0(n)$. Thus,

$$E\{\boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)e_0^*(n)\} = E\{\boldsymbol{\varepsilon}^H(n-1)\}E\{\mathbf{x}(n)e_0^*(n)\}$$

which is 0 by the orthogonality principle.

Thus

$$E\{\boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)e_0^*(n)\} = E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)e_0^*(n)\} = 0$$

Now consider the last term

$$\begin{aligned} & E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)\} \\ &= E\{\text{trace}[\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)]\} \\ &= E\{\text{trace}[\mathbf{x}(n)\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)]\} \end{aligned}$$

by the independence theorem

$$\begin{aligned} &= \text{trace}[E\{\mathbf{x}(n)\mathbf{x}^H(n)\}E\{\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)\}] \\ &= \text{trace}[\mathbf{R}\mathbf{K}(n-1)] \end{aligned}$$

Putting the pieces together

$$J'(n) = \sigma^2 + \text{trace}[\mathbf{R}\mathbf{K}(n-1)]$$

or since

$$\mathbf{K}(n-1) = \frac{\sigma^2}{n-M-2} \mathbf{R}^{-1}$$

we have

$$J'(n) = \sigma^2 + \frac{M\sigma^2}{n-M-2} \quad n > M+1$$

- The ensemble average learning curve of the RLS converges in about $2M$ iterations, which is typically an order of magnitude faster than the LMS
- $\lim_{n \rightarrow \infty} J'(n) = \sigma^2$ thus there is no excess MSE
- Convergence of the RLS algorithm is independent the eigenvalues of $\Phi(n)$

Example: Consider again the channel equalization problem

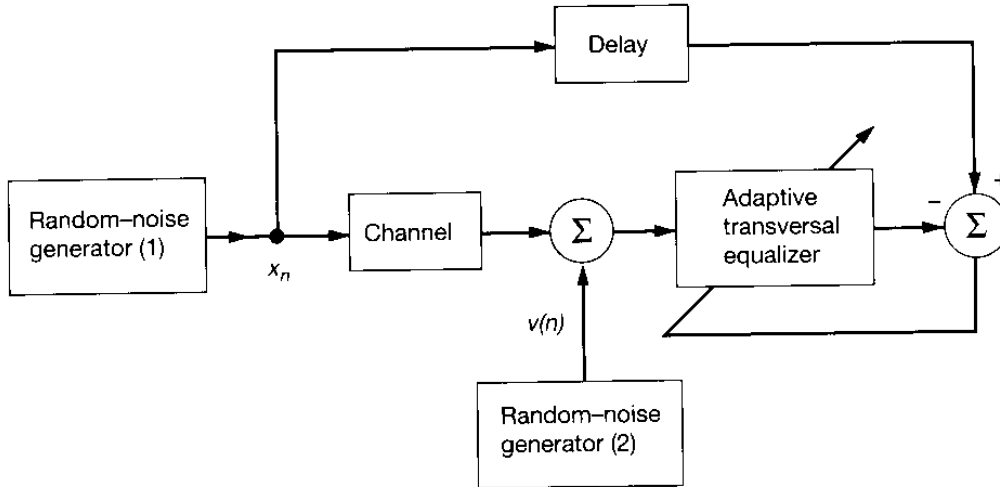
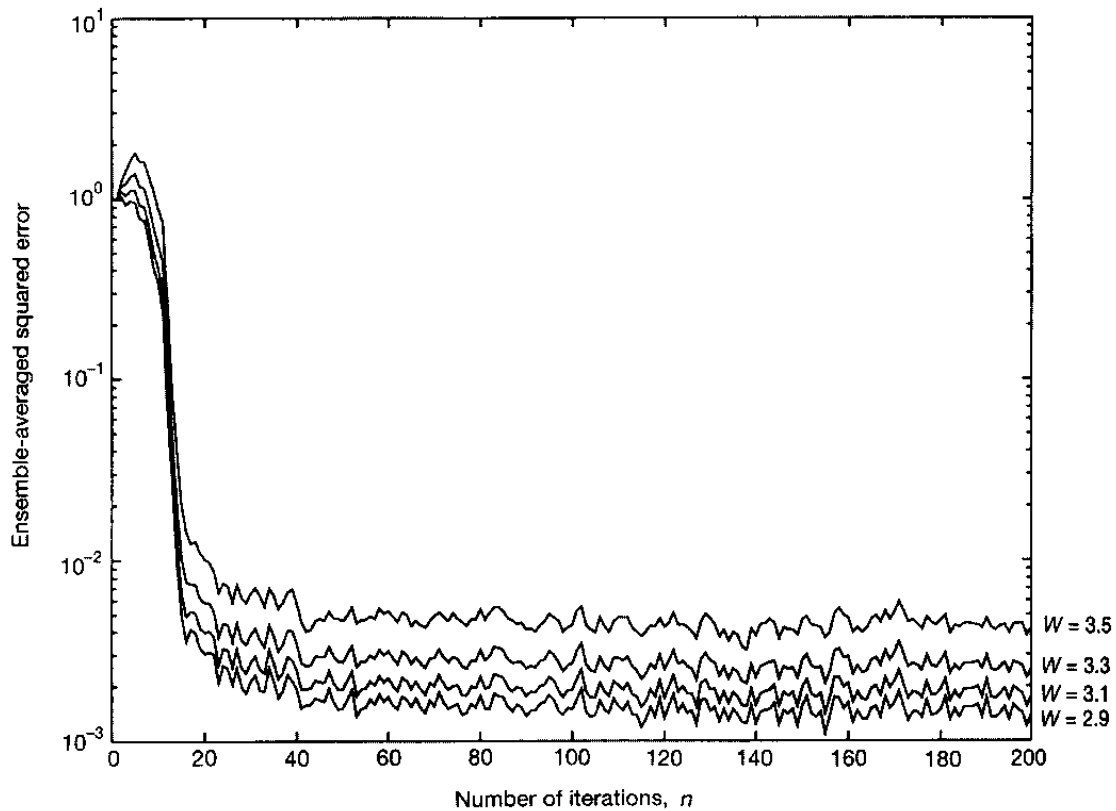


Figure 13.5 Block diagram of adaptive equalizer for computer experiment.

Where

$$h_n = \begin{cases} \frac{1}{2} [1 + \cos(\frac{2\pi}{W}(n-1))] & n = 1, 2, 3 \\ 0 & \text{Otherwise} \end{cases}$$

As before a 11-tap filter is used. The SNR is fixed at 30dB and W is varied to control the eigenvalue spread.



- The RLS algorithm converges in about 20 iterations (twice the number of filter taps).
- The convergence is insensitive to the eigenvalue spread.

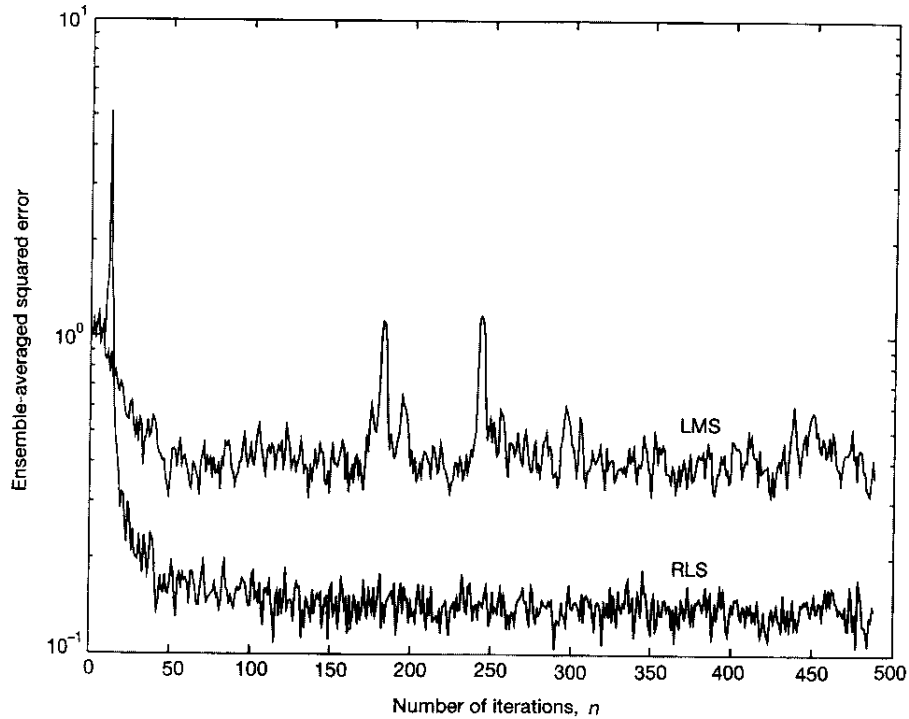


Figure 13.7 Learning curves for the RLS and LMS algorithms for $W = 3.1$ (i.e., eigenvalue spread $\chi(\mathbf{R}) = 11.1238$), and SNR = 10 dB. RLS: $\delta = 0.004$ and $\lambda = 1.0$. LMS: Step size parameter $\mu = 0.075$.

- The RLS algorithm converges faster than the LMS algorithm
- The RLS algorithm has lower steady state error than the LMS algorithm.