Chapter 4 Introduction to Adaptive Signal Processing

I. Adaptive Signal Processing with the Least Mean-Squared (LMS) Algorithm

1) Concerns of Least-Squares (LS) Optimization
   (a) The LS optimization technique is not suitable for nonstationary environments.
   (b) For locally stationary environments, we need to estimate the autocorrelation in a short period when we use the LS optimization technique. It is important to select an appropriate length of window in this case. $\rightarrow$ "gating"

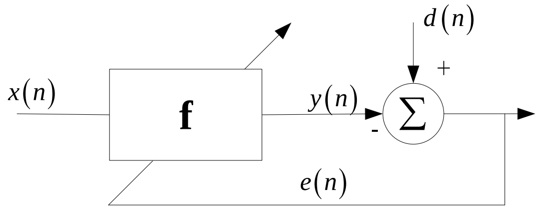
$$\mathbf{R}_1\mathbf{f}_1 = \mathbf{g}_1 \quad \mathbf{R}_2\mathbf{f}_2 = \mathbf{g}_2 \quad \mathbf{R}_3\mathbf{f}_3 = \mathbf{g}_3 \quad \mathbf{R}_4\mathbf{f}_4 = \mathbf{g}_4 \quad \cdots$$

$$\mathbf{f}_1 \quad \mathbf{f}_2 \quad \mathbf{f}_3 \quad \mathbf{f}_4 \quad \cdots$$

"Adaptive!" But the solutions of normal equations are still needed.

We want a filter that will iterate to approach the optimal solution as well as iterate to track changes in the optimal solution. $\Rightarrow$ "Adaptive filter."

2) Three Major Elements of Adaptive Filters
   (a) Filter structure: FIR, IIR, lattice …
   (b) Overall system configuration.

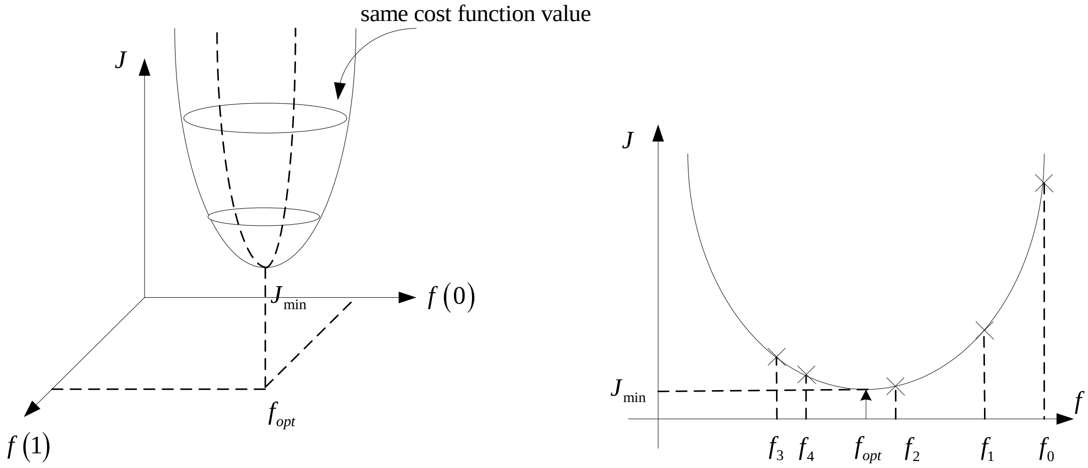


■ **Figure 4.1** Basic adaptive filtering system. Inputs to the filter are $x(n)$ and $d(n)$. $y(n)$ is the filter output and $e(n)$ represents the difference between the desired input $d(n)$ and the actual output $y(n)$ [1].

   (c) Performance criterion for adaptation.
   <u>*Example* 1</u>: LS minimization of the error signal.

3) Iterative Solutions to Normal Equations

Cost function: $J = E\left\{e^2(n)\right\} \rightarrow$ quadratic form.



■ **Figure 4.2** Quadratic performance surface *J* for a two-dimensional filter with coefficients *f*(0) and *f*(1) [1, p.159]. Its side view with different coefficient values is also presented.

Gradient search method

$$\mathbf{f}_{n+1} = \mathbf{f}_n - \alpha_n \mathbf{p}_n \quad \left(\alpha_n: \text{ step size; } \ \mathbf{p}_n: \text{gradient}\right) \tag{1}$$

$$\mathbf{p}_n = \mathbf{D}_n \cdot \nabla J_n, \quad \mathbf{D}_n: \text{ weight matrix} \tag{2}$$

$$\nabla J_n = \frac{\partial J}{\partial \mathbf{f}_n} = \left[\frac{\partial J}{\partial f(0)} \quad \frac{\partial J}{\partial f(1)} \quad \cdots \quad \frac{\partial J}{\partial f(L-1)}\right]^T \tag{3}$$

$$\mathbf{D}_n = \mathbf{I} \quad \left(\text{for steepest descent}\right) \tag{4}$$

$$\mathbf{p}_n = \nabla J_n = \frac{\partial E\left\{e^2(n)\right\}}{\partial \mathbf{f}_n} \tag{5}$$

$$\begin{aligned} \mathbf{f}_{n+1} &= \mathbf{f}_n - \alpha_n \cdot \nabla J_n \\ &= \mathbf{f}_n - \frac{\alpha}{2} \cdot 2\left(\mathbf{R}_n \mathbf{f}_n - \mathbf{g}_n\right) \end{aligned} \tag{6}$$

$$\mathbf{f}_{n+1} = \mathbf{f}_n - \alpha\left(\mathbf{R}_n \mathbf{f}_n - \mathbf{g}_n\right) \rightarrow \text{ Steepest descent algorithm}$$

where α is the step size or adaptation constant and its range for system convergence is given as follows:

$$0 < \alpha < \frac{2}{\lambda_{\max}} \Rightarrow \lim_{n\to\infty}\left\{f_n\right\} \rightarrow f^*, \ \ \textit{convergence} \tag{7}$$

where $\lambda_{\max}$ is the maximum eigenvalue of **R**.

4) The LMS Adaptive Filter

$$\mathbf{f}_{n+1} = \mathbf{f}_n - \alpha\left(\mathbf{R}_n\mathbf{f}_n - \mathbf{g}_n\right) \quad \text{“time-dependent update equation”} \tag{8}$$

Gradient-based search method with $\mathbf{D}_n = \mathbf{I}$

$$\mathbf{f}_{n+1} = \mathbf{f}_n - \frac{\alpha}{2}\nabla J_n, \quad \nabla J_n \equiv \text{true gradient} = 2\left(\mathbf{R}_n\mathbf{f}_n - \mathbf{g}_n\right) \tag{9}$$

$$\mathbf{f}_{n+1} = \mathbf{f}_n - \frac{\alpha}{2}\nabla \hat{J}_n, \quad \nabla \hat{J}_n \equiv \text{gradient estimate} \tag{10}$$

$$\nabla J_n = \frac{\partial E\left\{e^2(n)\right\}}{\partial \mathbf{f}_n} \tag{11}$$

$$\begin{aligned} \Rightarrow \nabla \hat{J}_n = \frac{\partial e^2(n)}{\partial \mathbf{f}_n} &= 2e(n)\frac{\partial e(n)}{\partial \mathbf{f}_n} \\ &= 2e(n)\cdot\frac{\partial\left[d(n) - \mathbf{f}_n^T\mathbf{x}_n\right]}{\partial \mathbf{f}_n} \\ &= -2e(n)\mathbf{x}_n \end{aligned} \tag{12}$$

$$\therefore \mathbf{f}_{n+1} = \mathbf{f}_n - \frac{\alpha}{2}\left[-2e(n)\mathbf{x}_n\right] = \mathbf{f}_n + \alpha e(n)\mathbf{x}_n \tag{13}$$

Estimate the gradient based on the instantaneous error signal.

The procedure of the LMS algorithm:

$$\begin{cases} y(n) = \mathbf{f}_n^T\mathbf{x}_n \\ e(n) = d(n) - y(n) \\ \mathbf{f}_{n+1} = \mathbf{f}_n + \alpha e(n)\mathbf{x}_n \end{cases} \tag{14}$$

$$\lim_{n\to\infty} E\left\{\mathbf{f}_n\right\} \to \mathbf{f}^* \tag{15}$$

Advantages:

(a) No matrix inverse.

(b) No averaging.

Note: Three important parameters of the LMS adaptive algorithm.

1. Filter length: L
   $L\uparrow \Rightarrow$ convergence speed $\downarrow$.
2. Initial filter vector: $\mathbf{f}_0$
   Generally, $\mathbf{f}_0 = \mathbf{0}$.
3. Step size (or adaptation constant): α

$$\mathbf{f}_{n+1} = \mathbf{f}_n + \alpha e(n)\mathbf{x}_n \tag{16}$$

$$\mathbf{f}_{n+1} = \mathbf{f}_n - \frac{\alpha}{2}\nabla J_n \tag{17}$$

$$\mathbf{f}_{n+1} = \mathbf{f}_n - \frac{\alpha}{2}\nabla \hat{J}_n = \mathbf{f}_n - \frac{\alpha}{2}\Big(\nabla J_n + \underset{noise}{\mathbf{N}_n}\Big) \tag{18}$$

$\alpha\uparrow \Rightarrow$ convergence speed $\uparrow$. Noise ($\alpha\mathbf{N}_n$) is amplified.

$\Rightarrow$ faster convergence with noise enhancement

II. Performance of the LMS Adaptive Filter

1) Introduction

Essentially, the analysis of the filter performance can be sub-divided into two basic concerns:

(a) The stability and convergence of the algorithm (towards the least-squares optimum solution). The best we can hope for is that the filter coefficient vector will tend towards the optimum solution in the mean:

$$\lim_{n\to\infty} E\{\mathbf{f}_n\} \to \mathbf{f}^* \tag{19}$$

(b) The mean-squared error of the filter output.

2) Convergence in the Mean

(a) Independence assumption

$$\begin{cases} \mathbf{f}_n \text{ is independent of } \mathbf{x}_n \\ \left.\begin{array}{l} \mathbf{f}_n \text{ depends on } \mathbf{x}_{n-1} \text{ and } \mathbf{f}_{n-1} \\ \mathbf{f}_{n-1} \text{ depends on } \mathbf{x}_{n-2} \text{ and } \mathbf{f}_{n-2} \\ \qquad\qquad \vdots \end{array}\right\} \Rightarrow \mathbf{f}_n \text{ depends on } \mathbf{x}_{n-1}, \mathbf{x}_{n-2}, \ldots \end{cases}$$

$$\Rightarrow \mathbf{x}_n \text{ is independent of } \mathbf{x}_{n-1}, \mathbf{x}_{n-2}, \ldots$$

$$\Rightarrow E\left\{\mathbf{x}_n\mathbf{x}_m^T\right\} = \mathbf{0}, \quad n \neq m$$

This is a highly unrealistic assumption. Even white sequences cannot satisfy this assumption since

$$E\left\{\mathbf{x}_n\mathbf{x}_{n-1}^T\right\} = E\{\begin{bmatrix} x_n \\ x_{n-1} \\ \vdots \\ x_{n-L+1} \end{bmatrix}\begin{bmatrix} x_{n-1} & x_{n-2} & \cdots & x_{n-L} \end{bmatrix}\} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ \sigma_w^2 & 0 & \cdots & 0 & 0 \\ 0 & \sigma_w^2 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & \cdots & \sigma_w^2 & 0 \end{bmatrix} \neq \mathbf{0} \tag{20}$$

The filter coefficient vector becomes

$$\begin{aligned} \mathbf{f}_{n+1} &= \mathbf{f}_n + \alpha e(n)\mathbf{x}_n \\ &= \mathbf{f}_n + \alpha\left[d(n) - y(n)\right]\mathbf{x}_n = \mathbf{f}_n + \alpha\left[d(n) - \mathbf{f}_n{}^T\mathbf{x}_n\right]\mathbf{x}_n \\ &= \mathbf{f}_n + \alpha d(n)\mathbf{x}_n - \alpha\mathbf{x}_n\left(\mathbf{x}_n{}^T\mathbf{f}_n\right), \quad \mathbf{f}_n{}^T\mathbf{x}_n = \mathbf{x}_n{}^T\mathbf{f}_n \\ &= \left(\mathbf{I} - \alpha\mathbf{x}_n\mathbf{x}_n{}^T\right)\mathbf{f}_n + \alpha d(n)\mathbf{x}_n \end{aligned} \tag{21}$$

$$E\{\mathbf{f}_{n+1}\} = E\left\{\left(\mathbf{I} - \alpha\mathbf{x}_n\mathbf{x}_n{}^T\right)\mathbf{f}_n + \alpha d(n)\mathbf{x}_n\right\} \tag{22}$$

Using the independence assumption: (if $\mathbf{x}_n$ and $\mathbf{f}_n$ are independent.)

$$\begin{aligned} E\{\mathbf{f}_{n+1}\} &= (\mathbf{I}-\alpha\mathbf{R})E\{\mathbf{f}_n\}+\alpha\mathbf{g} \quad \left(\mathbf{R}=E\{\mathbf{x}_n\mathbf{x}_n{}^T\},\ \ \mathbf{g}=E\{d(n)\mathbf{x}_n\}\right) \\ &= (\mathbf{I}-\alpha\mathbf{R})E\{\mathbf{f}_n\}+\alpha\mathbf{R}\mathbf{f}^* \end{aligned} \tag{23}$$

$$\Rightarrow \quad \begin{aligned} E\{\mathbf{f}_{n+1}\}-\mathbf{f}^* &= (\mathbf{I}-\alpha\mathbf{R})E\{\mathbf{f}_n\}+\alpha\mathbf{R}\mathbf{f}^*-\mathbf{f}^* \\ &= (\mathbf{I}-\alpha\mathbf{R})\left[E\{\mathbf{f}_n\}-\mathbf{f}^*\right] \end{aligned} \tag{24}$$

Let $\mathbf{u}_n = E\{\mathbf{f}_n\}-\mathbf{f}^* \Rightarrow \mathbf{u}_{n+1}=(\mathbf{I}-\alpha\mathbf{R})\mathbf{u}_n$.

$\lim_{n\to\infty} E\{\mathbf{f}_n\}\to\mathbf{f}^*$ is equivalent to $\lim_{n\to\infty}\mathbf{u}_n\to\mathbf{0}$.

(b) Decoupling

$$\mathbf{u}_{n+1}=(\mathbf{I}-\alpha\mathbf{R})\mathbf{u}_n \tag{25}$$

Express $\mathbf{R}$ as $\mathbf{R}=\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, i.e., diagonalization, where

$$\mathbf{\Lambda}=\begin{bmatrix} \lambda_0 & & & \mathbf{0} \\ & \lambda_1 & & \\ & & \ddots & \\ \mathbf{0} & & & \lambda_{L\text{-}1} \end{bmatrix},\ \ \lambda_i:\text{ eigenvalue of }\mathbf{R}; \tag{26}$$

$\mathbf{Q}=[\mathbf{q}_1 \quad \mathbf{q}_2 \quad \cdots \quad \mathbf{q}_L]$, $\mathbf{q}_i$ is the eigenvector corresponding to λ_i and $\mathbf{Q}$ is an orthonormal matrix, $\mathbf{Q}^T\mathbf{Q}=\mathbf{I}$.

$$\therefore \mathbf{u}_{n+1}=(\mathbf{I}-\alpha\mathbf{R})\mathbf{u}_n \tag{27}$$

$$\begin{aligned} \mathbf{Q}^T\mathbf{u}_{n+1} &= \mathbf{Q}^T\left(\mathbf{I}-\alpha\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T\right)\mathbf{u}_n \\ &= \mathbf{Q}^T\mathbf{u}_n-\alpha\mathbf{\Lambda}\mathbf{Q}^T\mathbf{u}_n \end{aligned} \tag{28}$$

Let $\mathbf{u}'_n=\mathbf{Q}^T\mathbf{u}_n \Rightarrow \mathbf{u}'_{n+1}=\mathbf{u}'_n-\alpha\mathbf{\Lambda}\mathbf{u}'_n=(\mathbf{I}-\alpha\mathbf{\Lambda})\mathbf{u}'_n$.

$$\begin{bmatrix} u'_{n+1}(0) \\ u'_{n+1}(1) \\ \vdots \\ u'_{n+1}(L-1) \end{bmatrix} = \begin{bmatrix} 1-\alpha\lambda_0 & & & \mathbf{0} \\ & 1-\alpha\lambda_1 & & \\ & & \ddots & \\ \mathbf{0} & & & 1-\alpha\lambda_{L-1} \end{bmatrix} \begin{bmatrix} u'_n(0) \\ u'_n(1) \\ \vdots \\ u'_n(L-1) \end{bmatrix} \tag{29}$$

$$\Rightarrow u'_{n+1}(j)=(1-\alpha\lambda_j)u'_n(j),\quad j=0,\ 1,\ 2,\ldots,\ L-1 \tag{30}$$

$$\Rightarrow u'_n(j)=(1-\alpha\lambda_j)^n u'_0(j),\quad j=0,\ 1,\ 2,\ldots,\ L-1 \tag{31}$$

(c) Condition for convergence/stability

For $\lim_{n\to\infty} u'_n(j) \to 0$ or $\lim_{n\to\infty} u_n(j) \to 0$,

$$|1-\alpha\lambda_j| < 1 \text{ is required.} \tag{32}$$

$$\Leftrightarrow -1 < 1-\alpha\lambda_j < 1 \tag{33}$$

$$\Leftrightarrow 0 < \alpha < \frac{2}{\lambda_j} \tag{34}$$

Thus $\lim_{n\to\infty} u_n(j) \to 0 \quad \forall j$, if

$$0 < \alpha < \frac{2}{\lambda_{\max}}, \tag{35}$$

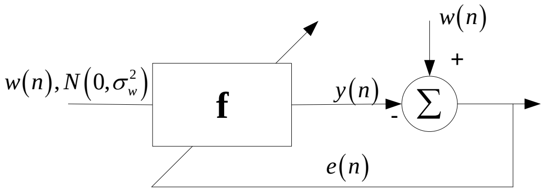
where $\lambda_{\max}$ is the largest eigenvalue of **R**. Using the fact that

$$\lambda_{\max} \le (\lambda_0 + \lambda_1 + \cdots + \lambda_{L-1}) = tr(\mathbf{R}). \tag{36}$$

we have the following alternative stability/convergence condition:

$$0 < \alpha < \frac{2}{\sum_i \lambda_i} = \frac{2}{tr(\mathbf{R})} = \frac{2}{L \cdot \underbrace{r(0)}_{=E\{x^2(n)\}}} = \frac{2}{L \cdot (\text{input signal power})} \tag{37}$$

<u>*Example* 2</u>: [1, p.173]



■ **Figure 4.3** Adaptive filter setup for stability trials. $w(n)$ is a zero-mean iid Gaussian sequence with $\sigma_w^2 = 0.04$.

The solution that minimizes the mean-squared error is $f_n(0) = 1$, $f_n(i) = 0$ for $i \neq 0$, which gives a zero error. The purpose here is to examine the conditions under which the filter will converge to this solution. Given $\mathbf{f}_0 = \mathbf{0}$ and the filter length L = 5, 10, 20, 50, 100, the ‘observed’ instability is defined as the occurrence of any error with magnitude in excess of 1×10^{10} during the 10,000 sample trial.

Table I Comparison of LMS Stability Criteria

L	$0 < \alpha < \frac{2}{\lambda_{\max}}$	$0 < \alpha < \frac{2}{L \cdot (\text{input signal power})}$	Observed
5	50	10	9.37
10	50	5	4.56
20	50	2.5	2.4
50	50	1	1.07
100	50	0.5	0.59

3) The Eigenvalue Disparity Problem (Eigenvalue Dispread)

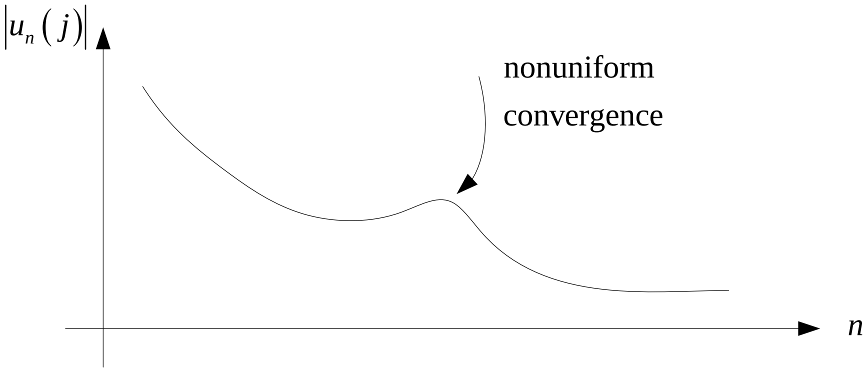
$$u'_{n+1}(j) = (1-\alpha\lambda_j)u'_n(j), \quad 0 < \alpha < \frac{2}{\lambda_{\max}} \tag{38}$$

$\lambda_j \uparrow \ \Rightarrow$ faster convergence.

However, the convergence of $u_n(j)$ does not usually take place uniformly.

$u_n(j) \ \Rightarrow$ nonuniform convergence (due to the eigenvalue disparity)

($n \uparrow \Rightarrow \ |u_n(j)| \ \downarrow$, this is uniform convergence.)



■ **Figure 4.4** Nonuniform convergence for filter coefficients.

(a) Eigenvalue ratio and spectrum

$$\frac{\lambda_{\max}}{\lambda_{\min}} \le \frac{S_{\max}}{S_{\min}} \tag{39}$$

where $S_{\max}$ and $S_{\min}$ are the maximum and the minimum components in the power spectrum of the input signal. Two extreme cases:

for white noise,

$$\frac{\lambda_{\max}}{\lambda_{\min}} = 1 = \frac{S_{\max}}{S_{\min}}; \tag{40}$$

for a sinusoid input (autocorrelation matrix is singular),

$$\frac{\lambda_{\max}}{\lambda_{\min}} = \frac{S_{\max}}{S_{\min}} = \infty\ . \tag{41}$$

(b) Eigenvalue ratio and filter length

The eigenvalue ratio is monotonically non-decreasing function of the filter length. (We cannot expect to improve the eigenvalue disparity by increasing the filter length. But, the convergence error, $J_{\min}$, will decrease as the filter length increases!)

Eigenvalue disparity ($\frac{\lambda_{\max}}{\lambda_{\min}}$) $\uparrow$

$\Rightarrow$ Uniformity of convergence of the LMS algorithm $\downarrow$.

(c) Time constant for convergence

Time constant: the required time interval that a component decays to 1/*e* of its initial value.

The mean convergence of the LMS algorithm is often quantified by means of time-constants.

$$u'_{n+1}(j) = (1-\alpha\lambda_j)u'_n(j) \tag{42}$$

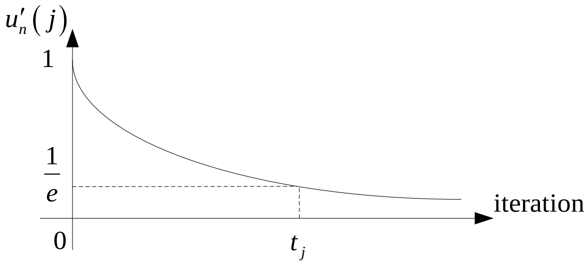
$$u'_n(j) = (1-\alpha\lambda_j)^n u'_0(j) \tag{43}$$

$$u'_{t_j}(j) = (1-\alpha\lambda_j)^{t_j} u'_0(j) = \frac{1}{e}u'_0(j) \tag{44}$$

$$(1-\alpha\lambda_j)^{t_j} = \frac{1}{e} \Rightarrow t_j \cdot \ln(1-\alpha\lambda_j) = -1 \tag{45}$$

$$t_j \cdot (-\alpha\lambda_j) \approx -1 \ (\text{if } 0 < \alpha\lambda_j \ll 1) \tag{46}$$

$$\Rightarrow t_j \approx \frac{1}{\alpha\lambda_j} \tag{47}$$

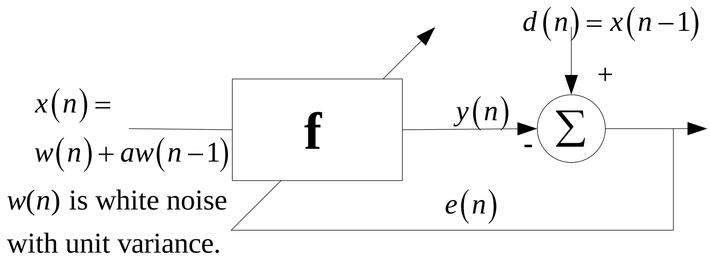


■ **Figure 4.5** The time-constant for adaptation for an decoupled mean coefficient error component $u'_n(j)$.

$$\begin{cases} \lambda_j \uparrow \ \Rightarrow t_j \downarrow \ \Rightarrow \text{ convergence speed } \uparrow \\ \lambda_j \downarrow \ \Rightarrow t_j \uparrow \ \Rightarrow \text{ convergence speed } \downarrow \end{cases}.$$

The overall convergence of the LMS algorithm is dominated by the slowest mode of convergence. (stepping from the smallest eigenvalue)

<u>*Example* 3</u>: Time-delay estimation



■ **Figure 4.6** Time-delay estimation using an adaptive filter [1, p.176].

(1) LS solution

$$\mathbf{R}\mathbf{f}^* = \mathbf{g}$$

$$\begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \end{bmatrix} = \begin{bmatrix} g(0) \\ g(1) \end{bmatrix}$$

$$r(0) = E\{x^2(n)\} = E\left\{\left[w(n) + aw(n-1)\right]^2\right\} = 1 + a^2$$

$$r(1) = E\{x(n)x(n-1)\} = a$$

$$g(0) = E\{d(n)x(n)\} = E\{x(n-1)x(n)\} = a$$

$$g(1) = E\{d(n)x(n-1)\} = E\{x(n-1)x(n-1)\} = 1 + a^2$$

$$\begin{bmatrix} 1+a^2 & a \\ a & 1+a^2 \end{bmatrix} \begin{bmatrix} f(0) \\ f(1) \end{bmatrix} = \begin{bmatrix} a \\ 1+a^2 \end{bmatrix}$$

$$\Rightarrow \begin{bmatrix} f(0) \\ f(1) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \Rightarrow \text{unit delay operator}$$

(2) stability limit

step-size range: $0 < \alpha < \dfrac{2}{L \cdot r(0)} = \dfrac{2}{2(1+a^2)} = \dfrac{1}{1+a^2}$

(3) nonuniform convergence

$$\mathbf{R} = \begin{bmatrix} 1+a^2 & a \\ a & 1+a^2 \end{bmatrix}, \quad \begin{aligned} &\det(\mathbf{R} - \lambda\mathbf{I}) = 0 \\ &\Rightarrow \lambda^2 - 2\lambda(1+a^2) + (a^4 + a^2 + 1) = 0 \\ &\Rightarrow \lambda_1 = 1 + a + a^2, \lambda_2 = 1 - a + a^2 \end{aligned}$$

$$\begin{cases} \mathbf{R}\mathbf{q}_1 = \lambda_1\mathbf{q}_1, \ \mathbf{q}_1: \text{ eigenvector corresponding to } \lambda_1. \\ \mathbf{R}\mathbf{q}_2 = \lambda_2\mathbf{q}_2, \ \mathbf{q}_2: \text{ eigenvector corresponding to } \lambda_2. \end{cases}$$

$$\mathbf{R} - \lambda_1\mathbf{I} = \begin{bmatrix} (1+a^2) - (1+a+a^2) & a \\ a & (1+a^2) - (1+a+a^2) \end{bmatrix}$$

$$= \begin{bmatrix} -a & a \\ a & -a \end{bmatrix}$$

$$\Rightarrow \mathbf{q}_1 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}$$

Similarly, $\mathbf{q}_2 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix} \Rightarrow \mathbf{Q} = [\mathbf{q}_1 \quad \mathbf{q}_2] = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix}$.

$$\mathbf{u}'_n = \mathbf{Q}^T\mathbf{u}_n \Rightarrow \mathbf{u}_n = \mathbf{Q}\mathbf{u}'_n \quad \left(\mathbf{Q}^T = \mathbf{Q}^{-1}\right)$$

$$
\begin{aligned}
\mathbf{u}_n &= \mathbf{Q}\mathbf{u}'_n \\
&= \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} u'_n(0) \\ u'_n(1) \end{bmatrix} = \begin{bmatrix} u_n(0) \\ u_n(1) \end{bmatrix}
\end{aligned}
$$

$$
\Rightarrow \begin{cases} u_n(0) = 1/\sqrt{2}\,u'_n(0) + 1/\sqrt{2}\,u'_n(1) \\ \qquad\quad = 1/\sqrt{2}(1-\alpha\lambda_1)^n u'_0(0) + 1/\sqrt{2}(1-\alpha\lambda_2)^n u'_0(1) \\ u_n(1) = 1/\sqrt{2}\,u'_n(0) - 1/\sqrt{2}\,u'_n(1) \\ \qquad\quad = 1/\sqrt{2}(1-\alpha\lambda_1)^n u'_0(0) - 1/\sqrt{2}(1-\alpha\lambda_2)^n u'_0(1) \end{cases}
$$

$$
\mathbf{f}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \Rightarrow \begin{cases} \mathbf{u}_0 = E\{\mathbf{f}_0\} - \mathbf{f}^* = \mathbf{f}_0 - \mathbf{f}^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \\ \mathbf{u}'_0 = \mathbf{Q}^T\mathbf{u}_0 = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} -1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix} = \begin{bmatrix} u'_0(0) \\ u'_0(1) \end{bmatrix} \end{cases}
$$

$$
\Rightarrow \begin{cases} u_n(0) = -1/2(1-\alpha\lambda_1)^n + 1/2(1-\alpha\lambda_2)^n \\ u_n(1) = -1/2(1-\alpha\lambda_1)^n - 1/2(1-\alpha\lambda_2)^n \end{cases}
$$

■

4) Steady State Mean-Squared Error for Random Inputs

$J_n : J_0,\ J_1,\ J_2, \ldots,\ J_\infty$

$$
J_\infty = J_{\min} + (\text{excess mean-squared error}) = J_{\min} + J_{excess} \tag{48}
$$

(a) Mean-squared error at iteration n: J_n (Recall page 3-14 in Chapter 3)

$$
J = J_{\min} + \mathbf{v}^T\mathbf{R}\mathbf{v}, \quad \mathbf{v} = \mathbf{f} - \mathbf{f}^* \tag{49}
$$

$$
J_n = J_{\min} + E\{\mathbf{v}_n^T\mathbf{R}\mathbf{v}_n\}, \quad \mathbf{v}_n = \mathbf{f}_n - \mathbf{f}^* \tag{50}
$$

Let $\mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, $\mathbf{v}'_n = \mathbf{Q}^T\mathbf{v}_n$

$$
\begin{aligned}
J_n &= J_{\min} + E\{\mathbf{v}_n^T\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T\mathbf{v}_n\} \\
&= J_{\min} + E\{\mathbf{v}'^T_n\mathbf{\Lambda}\mathbf{v}'_n\} \\
&= J_{\min} + \sum_{j=0}^{L-1}\lambda_j E\{v'^2_n(j)\} \\
&= J_{\min} + \sum_{j=0}^{L-1}\lambda_j\gamma_n(j), \quad \gamma_n(j) = E\{v'^2_n(j)\}
\end{aligned}
\quad , \quad \mathbf{v}'_n = \begin{bmatrix} v'_n(0) \\ v'_n(1) \\ \vdots \\ v'_n(L-1) \end{bmatrix} \tag{51}
$$

J_n: The mean-squared error of the LMS algorithm at iteration n.

(b) LMS algorithm in terms of $\mathbf{v}_n = \mathbf{f}_n - \mathbf{f}^*$

$$\begin{aligned}
&\mathbf{f}_{n+1} = \mathbf{f}_n + \alpha e(n)\mathbf{x}_n \\
&\mathbf{f}_{n+1} - \mathbf{f}^* = \mathbf{f}_n - \mathbf{f}^* + \alpha e(n)\mathbf{x}_n \\
&\Rightarrow \mathbf{v}_{n+1} = \mathbf{v}_n + \alpha e(n)\mathbf{x}_n
\end{aligned} \tag{52}$$

Let $\underset{\substack{\text{estimated}\\\text{gradient}}}{\nabla \hat{J}_n} = \underset{\substack{\text{true}\\\text{gradient}}}{\nabla J_n} + \underset{\text{noise}}{2\mathbf{N}_n}$,

$$-2e(n)\mathbf{x}_n = \nabla J_n + 2\mathbf{N}_n \tag{53}$$

using $\nabla J_n = 2(\mathbf{R}\mathbf{f}_n - \mathbf{g})$

$$\Rightarrow -2e(n)\mathbf{x}_n = 2(\mathbf{R}\mathbf{f}_n - \mathbf{g}) + 2\mathbf{N}_n \tag{54}$$

$$\begin{aligned}
\Rightarrow e(n)\mathbf{x}_n &= -(\mathbf{R}\mathbf{f}_n - \mathbf{g}) - \mathbf{N}_n \\
&= -(\mathbf{R}\mathbf{f}_n - \mathbf{R}\mathbf{f}^*) - \mathbf{N}_n \\
&= -\mathbf{R}\mathbf{v}_n - \mathbf{N}_n
\end{aligned} \tag{55}$$

$$\Rightarrow \alpha e(n)\mathbf{x}_n = -\alpha\mathbf{R}\mathbf{v}_n - \alpha\mathbf{N}_n \tag{56}$$

$$\begin{aligned}
\Rightarrow \mathbf{v}_{n+1} &= \mathbf{v}_n - \alpha\mathbf{R}\mathbf{v}_n - \alpha\mathbf{N}_n \\
&= (\mathbf{I} - \alpha\mathbf{R})\mathbf{v}_n - \alpha\mathbf{N}_n
\end{aligned} \tag{57}$$

Note: $\nabla \hat{J}_n$ is an unbiased gradient estimate.

$$\begin{aligned}
E\{e(n)\mathbf{x}_n\} &= E\{[d(n) - y(n)]\mathbf{x}_n\} = E\{[d(n) - \mathbf{f}_n^T\mathbf{x}_n]\mathbf{x}_n\} \\
&= E\{d(n)\mathbf{x}_n - \mathbf{x}_n\mathbf{x}_n^T\mathbf{f}_n\} = \mathbf{g} - \mathbf{R}\mathbf{f}_n \\
&(\because \text{independence assumption})
\end{aligned} \tag{58}$$

$$E\{-2e(n)\mathbf{x}_n\} = E\{\nabla \hat{J}_n\} = 2(\mathbf{R}\mathbf{f}_n - \mathbf{g}) = \nabla J_n \quad (\therefore \mathrm{E}\{\mathbf{N}_n\} = \mathbf{0}) \tag{59}$$

(c) LMS Algorithm in terms of $\mathbf{v}'_n = \mathbf{Q}^T\mathbf{v}_n$

$$\mathbf{v}_{n+1} = (\mathbf{I} - \alpha\mathbf{R})\mathbf{v}_n - \alpha\mathbf{N}_n \tag{60}$$

$$\begin{aligned}
&\mathbf{v}_n = \mathbf{Q}\mathbf{v}'_n \ , \ \mathbf{v}'_n = \mathbf{Q}^T\mathbf{v}_n \\
&\mathbf{N}_n = \mathbf{Q}\mathbf{N}'_n \ , \ \mathbf{N}'_n = \mathbf{Q}^T\mathbf{N}_n \ , \ \mathbf{R} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \\
&\Rightarrow \mathbf{Q}^T\mathbf{v}_{n+1} = \mathbf{Q}^T(\mathbf{I} - \alpha\mathbf{R})\mathbf{v}_n - \alpha\mathbf{Q}^T\mathbf{N}_n \\
&\qquad\qquad = \mathbf{Q}^T(\mathbf{I} - \alpha\mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T)\mathbf{v}_n - \alpha\mathbf{Q}^T\mathbf{N}_n
\end{aligned} \tag{61}$$

$$\Rightarrow \mathbf{v}'_{n+1} = (\mathbf{I} - \alpha\mathbf{\Lambda})\mathbf{v}'_n - \alpha\mathbf{N}'_n \tag{62}$$

$$\Rightarrow v'_{n+1}(j) = (1 - \alpha\lambda_j)v'_n(j) - \alpha N'_n(j), \ j = 0, 1, 2, \cdots, L-1 \tag{63}$$

$$J_n = J_{\min} + \sum_{j=0}^{L-1} \lambda_j E\{v'^2_n(j)\} \tag{64}$$

(d) Recursive form of $\gamma_n(j) = E\left\{v_n'^2(j)\right\}$

$$\begin{aligned}\gamma_{n+1}(j) &= E\left\{v_{n+1}'^2(j)\right\} \\ &= \gamma_n(j)\left(1-\alpha\lambda_j\right)^2 + \alpha^2 E\left\{N_n'^2(j)\right\} - 2\alpha\left(1-\alpha\lambda_j\right)E\left\{v_n'(j)N_n'(j)\right\}\end{aligned} \tag{65}$$

Assume $E\left\{v_n'(j)N_n'(j)\right\} = 0$

$$\Rightarrow \gamma_{n+1}(j) = E\left\{v_{n+1}'^2(j)\right\} = \gamma_n(j)\left(1-\alpha\lambda_j\right)^2 + \alpha^2 E\left\{N_n'^2(j)\right\} \tag{66}$$

(e) Determination of $E\left\{N_n'^2(j)\right\}$

For the steady state, $E\{\mathbf{f}_n\} \to \mathbf{f}^*$

$$\nabla\hat{J}_n = \nabla J_n + 2\mathbf{N}_n \tag{67}$$

∇J_n can be assumed to be zero in the steady state. $\left(\nabla J_n = 0\right)$

$$\Rightarrow \nabla\hat{J}_n \approx 2\mathbf{N}_n \tag{68}$$

$$\Rightarrow -2e(n)\mathbf{x}_n \approx 2\mathbf{N}_n \Rightarrow \underbrace{e(n)\mathbf{x}_n}_{\text{known}} \approx -\mathbf{N}_n \tag{69}$$

$$\begin{aligned}E\left\{\mathbf{N}_n\mathbf{N}_n^T\right\} &= E\left\{e^2(n)\mathbf{x}_n\mathbf{x}_n^T\right\} \\ &= E\left\{e^2(n)\right\}E\left\{\mathbf{x}_n\mathbf{x}_n^T\right\} \quad \left(\because\ e(n),\, \mathbf{x}_n \text{ are independent}\right) \\ &= J_{\min}\mathbf{R}\end{aligned} \tag{70}$$

$$\begin{aligned}E\left\{\mathbf{N}_n'\mathbf{N}_n'^T\right\} &= E\left\{\mathbf{Q}^T\mathbf{N}_n\mathbf{N}_n^T\mathbf{Q}\right\} = \mathbf{Q}^T E\left\{\mathbf{N}_n\mathbf{N}_n^T\right\}\mathbf{Q} = \mathbf{Q}^T J_{\min}\mathbf{R}\mathbf{Q} \\ &= J_{\min}\mathbf{Q}^T\mathbf{R}\mathbf{Q} = J_{\min}\mathbf{\Lambda},\end{aligned} \tag{71}$$

where $\mathbf{\Lambda} = \begin{pmatrix}\lambda_0 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \lambda_{L-1}\end{pmatrix}$.

$\because \mathbf{\Lambda}$ is a diagonal matrix, $\mathbf{N}' = \begin{bmatrix}N_n'(0) \\ N_n'(1) \\ \vdots \\ N_n'(L-1)\end{bmatrix}$

$\therefore E\left\{\mathbf{N}_n'\mathbf{N}_n'^T\right\} = \begin{pmatrix}E\left\{N_n'^2(0)\right\} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & E\left\{N_n'^2(L-1)\right\}\end{pmatrix}$ is a diagonal matrix.

$$\Rightarrow E\left\{N_n'^2(j)\right\} = J_{\min}\lambda_j \ , \ j = 0,\ 1,\ 2, \cdots,\ L-1 \tag{72}$$

(f) Determination of $r_n(j)$

$$\begin{aligned}\gamma_{n+1}(j) &= \gamma_n(j)\left(1-\alpha\lambda_j\right)^2 + \alpha^2 E\left\{N_n'^2(j)\right\} \\ &= \gamma_n(j)\left(1-\alpha\lambda_j\right)^2 + \alpha^2 J_{\min}\lambda_j\end{aligned} \tag{73}$$

$$\gamma_1(j) = \left(1-\alpha\lambda_j\right)^2 \gamma_0(j) + \alpha^2 J_{\min}\lambda_j \tag{74}$$

$$\begin{aligned}\gamma_2(j) &= \left(1-\alpha\lambda_j\right)^2 \gamma_1(j) + \alpha^2 J_{\min}\lambda_j \\ &= \left[\left(1-\alpha\lambda_j\right)^2 \gamma_0(j) + \alpha^2 J_{\min}\lambda_j\right]\left(1-\alpha\lambda_j\right)^2 + \alpha^2 J_{\min}\lambda_j \\ &= \left(1-\alpha\lambda_j\right)^4 \gamma_0(j) + \alpha^2 J_{\min}\lambda_j\left(1-\alpha\lambda_j\right)^2 + \alpha^2 J_{\min}\lambda_j\end{aligned} \tag{75}$$

$$\therefore \gamma_n(j) = \gamma_0(j)\left(1-\alpha\lambda_j\right)^{2n} + \alpha^2 \sum_{i=0}^{n-1}\left(1-\alpha\lambda_j\right)^{2i} J_{\min}\lambda_j \tag{76}$$

If $\left|1-\alpha\lambda_j\right| < 1 \Rightarrow 0 < \alpha < \dfrac{2}{\lambda_j}$ then $\left(1-\alpha\lambda_j\right)^{2n} \xrightarrow{n\to\infty} 0$

$$\gamma_\infty(j) = \lim_{n\to\infty}\gamma_n(j) = 0 + \alpha^2 \frac{1}{1-\left(1-\alpha\lambda_j\right)^2} J_{\min}\lambda_j = \frac{\alpha J_{\min}}{2-\alpha\lambda_j} \tag{77}$$

(g) Steady-state excess mean-squared error

From the results of (c),

$$\begin{aligned}J_{excess} = J_\infty - J_{\min} &= \sum_{j=0}^{L-1}\lambda_j\gamma_\infty(j) \\ &= \sum_{j=0}^{L-1}\frac{\alpha J_{\min}\lambda_j}{\left(2-\alpha\lambda_j\right)} = \sum_{j=0}^{L-1}\frac{\alpha J_{\min}\lambda_j}{2}\cdot\frac{1}{\left(1-\alpha\lambda_j/2\right)} \\ &= \sum_{j=0}^{L-1}\frac{\alpha J_{\min}\lambda_j}{2}\left[1+\frac{\alpha\lambda_j}{2}+\left(\frac{\alpha\lambda_j}{2}\right)^2+\cdots\right]\end{aligned} \tag{78}$$

For $\alpha \ll \dfrac{2}{\lambda_j} \Rightarrow \dfrac{\alpha\lambda_j}{2} \ll 1$

$$\begin{aligned}J_{excess} = J_\infty - J_{\min} &= \sum_{j=0}^{L-1}\frac{\lambda_j\alpha J_{\min}}{2}\left[1+\frac{\alpha\lambda_j}{2}+\left(\frac{\alpha\lambda_j}{2}\right)^2+\cdots\right] \\ &\approx \sum_{j=0}^{L-1}\frac{\alpha J_{\min}\lambda_j}{2} = \frac{\alpha J_{\min}}{2}\sum_{j=0}^{L-1}\lambda_j = \frac{\alpha J_{\min}}{2}tr(\mathbf{R}) \\ &= \frac{\alpha J_{\min}}{2}\left(L\times\text{input signal power}\right)\end{aligned} \tag{79}$$

$$\begin{aligned}&\Rightarrow J_\infty = J_{\min}\left[1+\frac{\alpha}{2}L\cdot\left(\text{input signal power}\right)\right] \\ &\left(L\uparrow\ , J_{\min}\downarrow\ ;\ L\leftrightarrow J_{\min}\ \text{tradeoff}\right)\end{aligned} \tag{80}$$

5) Performance Analysis of the LMS Algorithm for Deterministic Inputs

$$
\begin{aligned}
&\mathbf{f}_{n+1} = \mathbf{f}_n + \alpha e(n)\mathbf{x}_n \\
&\mathbf{f}_0 = 0 \\
&\mathbf{f}_1 = \alpha e(0)\mathbf{x}_0 \\
&\mathbf{f}_2 = \alpha e(0)\mathbf{x}_0 + \alpha e(1)\mathbf{x}_1 \\
&\qquad\vdots \\
&\mathbf{f}_n = \alpha\sum_{i=0}^{n-1} e(i)\mathbf{x}_i, \text{ where } \mathbf{x}_i = \begin{bmatrix} x(i) & x(i-1) & \cdots & x(i-L+1) \end{bmatrix}^T
\end{aligned} \tag{81}
$$

$$
\Rightarrow y(n) = \mathbf{f}_n^T\mathbf{x}_n = \alpha\sum_{i=0}^{n-1} e(i)\mathbf{x}_i^T\mathbf{x}_n, \tag{82}
$$

where $\mathbf{x}_n = \begin{bmatrix} x(n) & x(n-1) & \cdots & x(n-L+1) \end{bmatrix}^T$.

Let

$$
\begin{aligned}
r_{i,n} &= \frac{1}{L}\mathbf{x}_i^T\mathbf{x}_n = \frac{1}{L}\sum_{j=0}^{L-1} x(i-j)x(n-j) \\
&= \frac{1}{L}\sum_{k=i}^{i-L+1} x(k)x(k+n-i), \text{ (let } k = i-j \text{)} \\
&\approx \frac{1}{L}\sum_{k=0}^{-L+1} x(k)x(k+n-i), \text{ (if } L\to\infty) \\
&= r(n-i)
\end{aligned} \tag{83}
$$

$$
\therefore \mathbf{x}_i^T\mathbf{x}_n = L\cdot r(n-i) \tag{84}
$$

$$
\Rightarrow y(n) = \alpha L\sum_{i=0}^{n-1} e(i)r(n-i) \tag{85}
$$

$$
\begin{aligned}
e(n) &= d(n) - y(n) = d(n) - \alpha L\sum_{i=0}^{n-1} e(i)r(n-i) \\
&= d(n) - \alpha L\left[e(n)*r(n)\right]
\end{aligned} \tag{86}
$$

Let $r(n) = 0$ for $n < 1$ and $e(n) = 0$ for $n < 0$

$$
\Rightarrow e(n)*r(n) = \sum_{i=0}^{n-1} e(i)r(n-i) \tag{87}
$$

$$
E(z) = D(z) - \alpha LE(z)R(z),\ R(z) = r(1)z^{-1} + r(2)z^{-2} + r(3)z^{-3} + \cdots \tag{88}
$$

$$
\Rightarrow E(z)\left[1 + \alpha LR(z)\right] = D(z) \tag{89}
$$

$$
\Rightarrow \frac{E(z)}{D(z)} = \frac{1}{1+\alpha LR(z)} = H(z) \tag{90}
$$

$$
\begin{aligned}
&Y(z) = D(z) - E(z) \\
&\Rightarrow \frac{Y(z)}{D(z)} = 1 - \frac{E(z)}{D(z)} = 1 - \frac{1}{1+\alpha LR(z)} = \frac{\alpha LR(z)}{1+\alpha LR(z)}
\end{aligned} \tag{91}
$$

6) Special Case: Periodic Inputs

$$r_{i,n} = \frac{1}{L}\sum_{k=i}^{i-L+1} x(k)x(k+n-i) \tag{92}$$

Assume $x(k)$ is a periodic signal with period *N* and *L* is an integer multiple of *N*.

$$r_{i,n} = \frac{1}{L}\sum_{k=0}^{-L+1} x(k)x(k+n-i) = r(n-i) \tag{93}$$

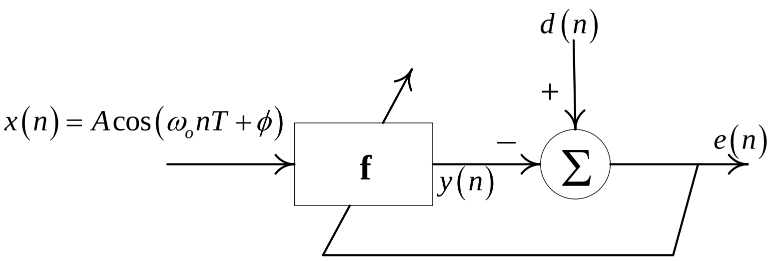
$$\begin{aligned}
R(z) &= r(1)z^{-1} + r(2)z^{-2} + \cdots + r(N-1)z^{-N+1} + r(N)z^{-N} \\
&\quad + \underbrace{r(N+1)}_{=r(1)}z^{-N-1} + \underbrace{r(N+2)}_{=r(2)}z^{-N-2} + \cdots \\
&= r(1)z^{-1}\left(1+z^{-N}+z^{-2N}+\cdots\right) \\
&\quad + r(2)z^{-2}\left(1+z^{-N}+z^{-2N}+\cdots\right) + \cdots + r(N)z^{-N}\left(1+z^{-N}+z^{-2N}+\cdots\right) \\
&= r(1)z^{-1}\frac{1}{1-z^{-N}} + r(2)z^{-2}\frac{1}{1-z^{-N}} + \cdots + r(N)z^{-N}\frac{1}{1-z^{-N}} \\
&= \frac{\hat{R}(z)}{1-z^{-N}},
\end{aligned} \tag{94}$$

where $\hat{R}(z) = r(1)z^{-1} + r(2)z^{-2} + \cdots + r(N-1)z^{-N+1} + r(N)z^{-N}$.

$$\Rightarrow \frac{E(z)}{D(z)} = \frac{1}{1+\alpha L R(z)} = \frac{1-z^{-N}}{1-z^{-N}+\alpha L\hat{R}(z)}, \quad \left(\because R(z) = \frac{\hat{R}(z)}{1-z^{-N}}\right) \tag{95}$$

<u>*Example* 4</u>:

$x(n) = A\cos(\omega_0 nT + \phi)$, $\omega_0 = 2\pi f_0$, T: sampling interval, $T_0 = 1/f_0$, $N = T_0/T$: period of the input signal.



■ **Figure 4.7** Adaptive filter with a sinusoidal input.

$$\begin{aligned}
r_{n,n-i} &= \frac{1}{L}\mathbf{x}_n^T\mathbf{x}_{n-i} = \frac{1}{L}\sum_{m=0}^{L-1}\left[A^2\cos\left(\omega_0(n-m)T+\phi\right)\cos\left(\omega_0(n-i-m)T+\phi\right)\right] \\
&= \frac{A^2}{2L}\sum_{m=0}^{L-1}\left\{\cos\left[\omega_0(2n-2m-i)T+2\phi\right] + \cos\left(\omega_0 iT\right)\right\}
\end{aligned}$$

Period of the input signal $T_0 = \dfrac{1}{f_0}$

Assume $\left(\frac{T_0}{T}\right)l = L$ (L is an integer multiple of the period of the sinusoid), then $\sum_{m=0}^{L-1} e^{j2m\omega_0 T} = 0 \Rightarrow \sum_{m=0}^{L-1}\cos\left(2m\omega_0 T\right) = 0$ and $\sum_{m=0}^{L-1}\sin\left(2m\omega_0 T\right) = 0$

$$\Rightarrow r_{n,n-i} = r(i) = \frac{A^2}{2}\cos\left(\omega_0 iT\right)$$

$$\Rightarrow R(z) = \frac{A^2}{2}\left[\sum_{i=0}^{\infty}\cos\left(\omega_0 iT\right)z^{-i} - 1\right] = \frac{A^2}{2}\left[\frac{1-\cos\left(\omega_0 T\right)z^{-1}}{1-2\cos\left(\omega_0 T\right)z^{-1}+z^{-2}} - 1\right]$$

$$\frac{E(z)}{D(z)} = \frac{1-2\cos\left(\omega_0 T\right)z^{-1}+z^{-2}}{\left(1-\frac{\alpha LA^2}{2}\right)z^{-2} + \left(\frac{\alpha LA^2}{2}-2\right)\cos\left(\omega_0 T\right)z^{-1}+1}$$

$\Rightarrow$ Notch filter.

<u>For small</u> α,

$$1-\frac{\alpha LA^2}{2} \approx \left(1-\frac{\alpha LA^2}{4}\right)^2 = 1-\frac{\alpha LA^2}{2} + \cancel{\frac{(\alpha LA^2)^2}{16}}$$

$$\therefore \left(1-\frac{\alpha LA^2}{2}\right)z^{-2} \Rightarrow \left(1-\frac{\alpha LA^2}{4}\right)^2 z^{-2}$$

$$\frac{E(z)}{D(z)} = \frac{1-2\cos(\omega_0 T)z^{-1}+z^{-2}}{\left(1-\frac{\alpha LA^2}{4}\right)^2 z^{-2} - 2\left(1-\frac{\alpha LA^2}{4}\right)\left[\frac{1}{2}\left(e^{j\omega_0 T}+e^{-j\omega_0 T}\right)\right]z^{-1}+1}$$

<u>The zeros:</u>

$$1-2\cos\left(\omega_0 T\right)z^{-1}+z^{-2} = 0$$

$$\Rightarrow 1-\left(e^{j\omega_0 T}+e^{-j\omega_0 T}\right)z^{-1}+z^{-2} = 0$$

$$\Rightarrow \left(1-e^{j\omega_0 T}z^{-1}\right)\left(1-e^{-j\omega_0 T}z^{-1}\right) = 0$$

$$\Rightarrow \begin{cases} z_1 = e^{j\omega_0 T} \\ z_2 = e^{-j\omega_0 T} \end{cases}$$

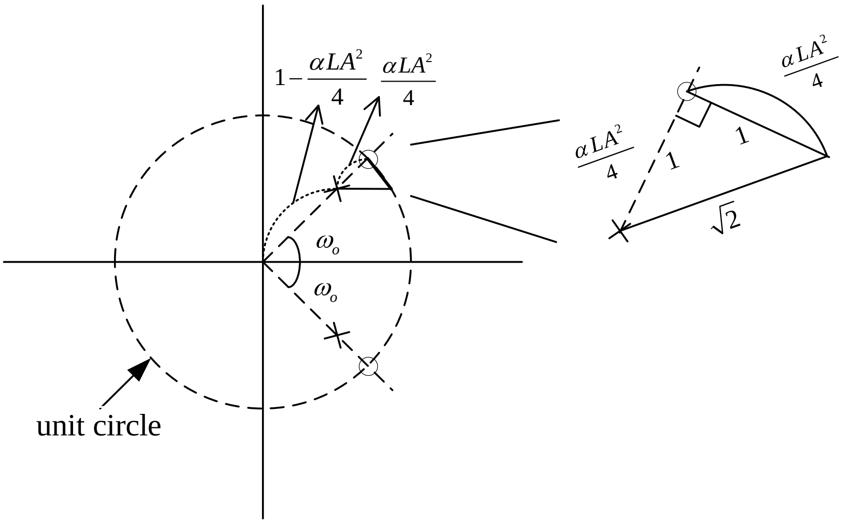
<u>The poles:</u>

$$\left(1-\frac{\alpha LA^2}{4}\right)^2 z^{-2} - 2\left(1-\frac{\alpha LA^2}{4}\right)\left[\frac{1}{2}\left(e^{j\omega_0 T}+e^{-j\omega_0 T}\right)\right]z^{-1}+1 = 0$$

$$\Rightarrow \left[e^{j\omega_0 T}\left(1-\frac{\alpha LA^2}{4}\right)z^{-1}-1\right]\left[e^{-j\omega_0 T}\left(1-\frac{\alpha LA^2}{4}\right)z^{-1}-1\right] = 0$$

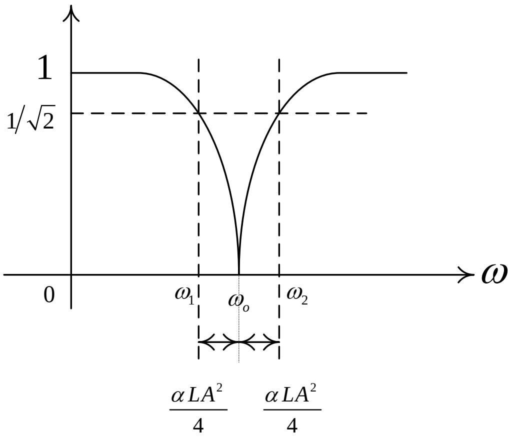
$$\Rightarrow \begin{cases} z_{p1} = \left(1-\frac{\alpha LA^2}{4}\right)e^{j\omega_0 T} \\ z_{p2} = \left(1-\frac{\alpha LA^2}{4}\right)e^{-j\omega_0 T} \end{cases}$$

$\therefore$ When $\omega = \omega_0$, $\frac{E(z)}{D(z)} = 0$.



■ **Figure 4.8** Pole-zero plot for the transfer function using the LMS algorithm [1].

3dB bandwidth = $2 \cdot \frac{\alpha LA^2}{4} = \frac{\alpha LA^2}{2}$.



■ **Figure 4.9** Frequency response for the sinusoidal transfer function using LMS algorithm.

Stability condition:

$$A\cos\left(\omega_o nT + \phi\right) \underset{power}{\Rightarrow} \frac{A^2}{2}$$

To ensure that the poles are inside the unit circle,

$$0 < \frac{\alpha LA^2}{4} < 1 \ \left(\text{inside the unit circle}\right)$$

$$\Rightarrow 0 < \alpha < \frac{4}{LA^2} = \frac{2}{L \cdot \left(\text{input signal power}\right)}$$

■

References

[1] P. M. Clarkson, *Optimal and Adaptive Signal Processing*. Boca Raton, FL: CRC, 1993.