

## CHAPTER 1

1.1 Let

$$r_u(k) = E[u(n)u^*(n-k)] \quad (1)$$

$$r_y(k) = E[y(n)y^*(n-k)] \quad (2)$$

We are given that

$$y(n) = u(n+a) - u(n-a) \quad (3)$$

Hence, substituting Eq. (3) into (2), and then using Eq. (1), we get

$$\begin{aligned} r_y(k) &= E[(u(n+a) - u(n-a))(u^*(n+a-k) - u^*(n-a-k))] \\ &= 2r_u(k) - r_u(2a+k) - r_u(-2a+k) \end{aligned}$$

1.2 We know that the correlation matrix  $\mathbf{R}$  is Hermitian; that is

$$\mathbf{R}^H = \mathbf{R}$$

Given that the inverse matrix  $\mathbf{R}^{-1}$  exists, we may write

$$\mathbf{R}^{-1}\mathbf{R}^H = \mathbf{I}$$

where  $\mathbf{I}$  is the identity matrix. Taking the Hermitian transpose of both sides:

$$\mathbf{R}\mathbf{R}^{-H} = \mathbf{I}$$

Hence,

$$\mathbf{R}^{-H} = \mathbf{R}^{-1}$$

That is, the inverse matrix  $\mathbf{R}^{-1}$  is Hermitian.

1.3 For the case of a two-by-two matrix, we may

$$\mathbf{R}_u = \mathbf{R}_s + \mathbf{R}_v$$

$$\begin{aligned}
&= \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} + \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix} \\
&= \begin{bmatrix} r_{11} + \sigma^2 & r_{12} \\ r_{21} & r_{22} + \sigma^2 \end{bmatrix}
\end{aligned}$$

For  $\mathbf{R}_u$  to be nonsingular, we require

$$\det(\mathbf{R}_u) = (r_{11} + \sigma^2)(r_{22} + \sigma^2) - r_{12}r_{21} > 0$$

With  $r_{12} = r_{21}$  for real data, this condition reduces to

$$(r_{11} + \sigma^2)(r_{22} + \sigma^2) - r_{12}r_{21} > 0$$

Since this is quadratic in  $\sigma^2$ , we may impose the following condition on  $\sigma^2$  for nonsingularity of  $\mathbf{R}_u$ :

$$\sigma^2 > \frac{1}{2}(r_{11} + r_{22}) \left( \sqrt{1 - \frac{4\Delta_r}{(r_{11} + r_{22})^2 - 1}} \right)$$

$$\text{where } \Delta_r = r_{11}r_{22} - r_{12}^2$$

1.4 We are given

$$\mathbf{R} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

This matrix is positive definite because

$$a^T \mathbf{R} a = [a_1, a_2] \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$$

$$= a_1^2 + 2a_1a_2 + a_2^2$$

$$= (a_1 + a_2)^2 > 0 \text{ for all nonzero values of } a_1 \text{ and } a_2$$

(Positive definiteness is stronger than nonnegative definiteness.)

But the matrix  $\mathbf{R}$  is singular because

$$\det(\mathbf{R}) = (1)^2 - (1)^2 = 0$$

Hence, it is possible for a matrix to be positive definite and yet it can be singular.

1.5 (a)

$$\mathbf{R}_{M+1} = \begin{bmatrix} r(0) & \mathbf{r}^H \\ \mathbf{r} & \mathbf{R}_M \end{bmatrix} \quad (1)$$

Let

$$\mathbf{R}_{M+1}^{-1} = \begin{bmatrix} a & \mathbf{b}^H \\ \mathbf{b} & \mathbf{C} \end{bmatrix} \quad (2)$$

where  $a$ ,  $\mathbf{b}$  and  $\mathbf{C}$  are to be determined. Multiplying (1) by (2):

$$\mathbf{I}_{M+1} = \begin{bmatrix} r(0) & \mathbf{r}^H \\ \mathbf{r} & \mathbf{R}_M \end{bmatrix} \begin{bmatrix} a & \mathbf{b}^H \\ \mathbf{b} & \mathbf{C} \end{bmatrix}$$

where  $\mathbf{I}_{M+1}$  is the identity matrix. Therefore,

$$r(0)a + \mathbf{r}^H \mathbf{b} = 1 \quad (3)$$

$$\mathbf{r}a + \mathbf{R}_M \mathbf{b} = \mathbf{0} \quad (4)$$

$$\mathbf{r} \mathbf{b}^H + \mathbf{R}_M \mathbf{C} = \mathbf{I}_M \quad (5)$$

$$r(0)\mathbf{b}^H + \mathbf{r}^H \mathbf{C} = \mathbf{0}^T \quad (6)$$

From Eq. (4):

$$\mathbf{b} = -\mathbf{R}_M^{-1} \mathbf{r} a \quad (7)$$

Hence, from (3) and (7):

$$a = \frac{1}{r(0) - \mathbf{r}^H \mathbf{R}_M^{-1} \mathbf{r}} \quad (8)$$

Correspondingly,

$$\mathbf{b} = - \frac{\mathbf{R}_M^{-1} \mathbf{r}}{r(0) - \mathbf{r}^H \mathbf{R}_M^{-1} \mathbf{r}} \quad (9)$$

From (5):

$$\begin{aligned} \mathbf{C} &= \mathbf{R}_M^{-1} - \mathbf{R}_M^{-1} \mathbf{r} \mathbf{b}^H \\ &= \mathbf{R}_M^{-1} + \frac{\mathbf{R}_M^{-1} \mathbf{r} \mathbf{r}^H \mathbf{R}_M^{-1}}{r(0) - \mathbf{r}^H \mathbf{R}_M^{-1} \mathbf{r}} \end{aligned} \quad (10)$$

As a check, the results of Eqs. (9) and (10) should satisfy Eq. (6).

$$\begin{aligned} r(0) \mathbf{b}^H + \mathbf{r}^H \mathbf{C} &= - \frac{r(0) \mathbf{r}^H \mathbf{R}_M^{-1}}{r(0) - \mathbf{r}^H \mathbf{R}_M^{-1} \mathbf{r}} + \mathbf{r}^H \mathbf{R}_M^{-1} + \frac{\mathbf{r}^H \mathbf{R}_M^{-1} \mathbf{r} \mathbf{r}^H \mathbf{R}_M^{-1}}{r(0) - \mathbf{r}^H \mathbf{R}_M^{-1} \mathbf{r}} \\ &= \mathbf{0}^T \end{aligned}$$

We have thus shown that

$$\begin{aligned} \mathbf{R}_{M+1}^{-1} &= \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{R}_M^{-1} \end{bmatrix} + a \begin{bmatrix} 1 & -\mathbf{r}^H \mathbf{R}_M^{-1} \\ \mathbf{R}_M^{-1} \mathbf{r} & \mathbf{R}_M^{-1} \mathbf{r} \mathbf{r}^H \mathbf{R}_M^{-1} \end{bmatrix} \\ &= \begin{bmatrix} 0 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{R}_M^{-1} \end{bmatrix} + a \begin{bmatrix} 1 \\ -\mathbf{R}_M^{-1} \mathbf{r} \end{bmatrix} [1 \ \mathbf{r}^H \mathbf{R}_M^{-1}] \end{aligned}$$

where the scalar  $a$  is defined by Eq. (8):

$$(b) \quad \mathbf{R}_{M+1} = \begin{bmatrix} \mathbf{R}_M & \mathbf{r}^{B*} \\ \mathbf{r}^{BT} & r(0) \end{bmatrix} \quad (11)$$

Let

$$\mathbf{R}_{M+1}^{-1} = \begin{bmatrix} \mathbf{D} & \mathbf{e} \\ \mathbf{e}^H & f \end{bmatrix} \quad (12)$$

where  $\mathbf{D}$ ,  $\mathbf{e}$  and  $f$  are to be determined. Multiplying (11) by (12):

$$\mathbf{I}_{M+1} = \begin{bmatrix} \mathbf{R}_M & \mathbf{r}^{B*} \\ \mathbf{r}^{BT} & r(0) \end{bmatrix} \begin{bmatrix} \mathbf{D} & \mathbf{e} \\ \mathbf{e}^H & f \end{bmatrix}$$

Therefore

$$\mathbf{R}_M \mathbf{D} + \mathbf{r}^{B*} \mathbf{e}^H = \mathbf{I} \quad (13)$$

$$\mathbf{R}_M \mathbf{e} + \mathbf{r}^{B*} f = \mathbf{0} \quad (14)$$

$$\mathbf{r}^{BT} \mathbf{e} + r(0) f = 1 \quad (15)$$

$$\mathbf{r}^{BT} \mathbf{D} + r(0) \mathbf{e}^H = \mathbf{0}^T \quad (16)$$

From (14):

$$\mathbf{e} = -\mathbf{R}_M^{-1} \mathbf{r}^{B*} f \quad (17)$$

Hence, from (15) and (17):

$$f = \frac{1}{r(0) - \mathbf{r}^{BT} \mathbf{R}_M^{-1} \mathbf{r}^{B*}} \quad (18)$$

Correspondingly,

$$\mathbf{e} = - \frac{\mathbf{R}_M^{-1} \mathbf{r}^{B*}}{r(0) - \mathbf{r}^{BT} \mathbf{R}_M^{-1} \mathbf{r}^{B*}} \quad (19)$$

From (13):

$$\begin{aligned} \mathbf{D} &= \mathbf{R}_M^{-1} - \mathbf{R}_M^{-1} \mathbf{r}^{B*} \mathbf{e}^H \\ &= \mathbf{R}_M^{-1} + \frac{\mathbf{R}_M^{-1} \mathbf{r}^{B*} \mathbf{r}^{BT} \mathbf{R}_M^{-1}}{r(0) - \mathbf{r}^{BT} \mathbf{R}_M^{-1} \mathbf{r}^{B*}} \end{aligned} \quad (20)$$

As a check, the results of Eqs. (19) and (20) must satisfy Eq. (16). Thus

$$\begin{aligned} \mathbf{r}^{BT} \mathbf{D} + r(0) \mathbf{e}^H &= \mathbf{r}^{BT} \mathbf{R}_M^{-1} + \frac{\mathbf{r}^{BT} \mathbf{R}_M^{-1} \mathbf{r}^{B*} \mathbf{r}^{BT} \mathbf{R}_M^{-1}}{r(0) - \mathbf{r}^{BT} \mathbf{R}_M^{-1} \mathbf{r}^{B*}} - \frac{r(0) \mathbf{r}^{BT} \mathbf{R}_M^{-1}}{r(0) - \mathbf{r}^{BT} \mathbf{R}_M^{-1} \mathbf{r}^{B*}} \\ &= \mathbf{0}^T \end{aligned}$$

We have thus shown that

$$\begin{aligned} \mathbf{R}_{M+1}^{-1} &= \begin{bmatrix} \mathbf{R}_M^{-1} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + f \begin{bmatrix} \mathbf{R}_M^{-1} \mathbf{r}^{B*} \mathbf{r}^{BT} \mathbf{R}_M^{-1} & -\mathbf{R}_M^{-1} \mathbf{r}^{B*} \\ -\mathbf{r}^{BT} \mathbf{R}_M^{-1} & 1 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{R}_M^{-1} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + f \begin{bmatrix} -\mathbf{R}_M^{-1} \mathbf{r}^{B*} \\ 1 \end{bmatrix} \begin{bmatrix} -\mathbf{r}^{BT} & \mathbf{R}_M^{-1} & 1 \end{bmatrix} \end{aligned}$$

where the scalar  $f$  is defined by Eq. (18).

1.6 (a) We express the difference equation describing the first-order AR process  $u(n)$  as

$$u(n) = v(n) + w_1 u(n-1)$$

where  $w_1 = -a_1$ . Solving this equation by repeated substitution, we get

$$u(n) = v(n) + w_1 v(n-1) + w_1 u(n-2)$$

$$\begin{aligned}
&= \dots \\
&= v(n) + w_1 v(n-1) + w_1^2 v(n-2) + \dots + w_1^{n-1} v(1)
\end{aligned} \tag{1}$$

Here we have used the initial condition

$$u(0) = 0$$

or equivalently

$$u(1) = v(1)$$

Taking the expected value of both sides of Eq. (1) and using

$$E[v(n)] = \mu \quad \text{for all } n,$$

we get the geometric series

$$\begin{aligned}
E[u(n)] &= \mu + w_1 \mu + w_1^2 \mu + \dots + w_1^{n-1} \mu \\
&= \begin{cases} \mu \left( \frac{1 - w_1^n}{1 - w_1} \right), & w_1 \neq 1 \\ \mu n, & w_1 = 1 \end{cases}
\end{aligned}$$

This result shows that if  $\mu \neq 0$ , then  $E[u(n)]$  is a function of time  $n$ . Accordingly, the AR process  $u(n)$  is not stationary. If, however, the AR parameter satisfies the condition:

$$|a_1| < 1 \quad \text{or} \quad |w_1| < 1$$

then

$$E[(n)] \rightarrow \frac{\mu}{1 - w_1} \quad \text{as } n \rightarrow \infty$$

Under this condition, we say that the AR process is asymptotically stationary to order one.

- (b) When the white noise process  $v(n)$  has zero mean, the AR process  $u(n)$  will likewise have zero mean. Then

$$\text{var}[v(n)] = \sigma_v^2$$

$$\text{var}[u(n)] = E[u^2(n)]. \quad (2)$$

Substituting Eq. (1) into (2), and recognizing that for the white noise process

$$E[v(n)v(k)] = \begin{cases} \sigma_v^2 & n = k \\ 0, & n \neq k \end{cases} \quad (3)$$

we get the geometric series

$$\text{var}[u(n)] = \sigma_v^2(1 + w_1^2 + w_1^4 + \cdots + w_1^{2n-2})$$

$$= \begin{cases} \sigma_v^2 \left( \frac{1 - w_1^{2n}}{1 - w_1^2} \right), & w_1 \neq 1 \\ \sigma_v^2 n, & w_1 = 1 \end{cases}$$

When  $|a_1| < 1$  or  $|w_1| < 1$ , then

$$\text{var}[u(n)] \approx \frac{\sigma_v^2}{1 - w_1^2} = \frac{\sigma_v^2}{1 - a_1^2} \text{ for large } n$$

(c) The autocorrelation function of the AR process  $u(n)$  equals  $E[u(n)u(n-k)]$ . Substituting Eq. (1) into this formula, and using Eq. (3), we get

$$E[u(n)u(n-k)] = \sigma_v^2(w_1^k + w_1^{k+2} + \cdots + w_1^{k+2n-2})$$

$$= \begin{cases} \sigma_v^2 w_1^k \left( \frac{1 - w_1^{2n}}{1 - w_1^2} \right), & w_1 \neq 1 \\ \sigma_v^2 n, & w_1 = 1 \end{cases}$$



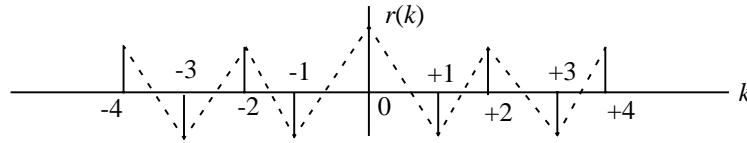
For  $|a_1| < 1$  or  $|w_1| < 1$ , we may therefore express this autocorrelation function as

$$r(k) = E[u(n)u(n-k)]$$

$$\approx \frac{\sigma_v^2 w_1^k}{1 - w_1^2} \text{ for large } n$$

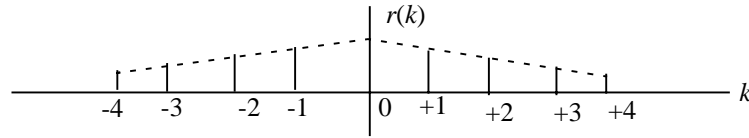
**Case 1:**  $0 < a_1 < 1$

In this case,  $w_1 = -a_1$  is negative, and  $r(k)$  varies with  $k$  as follows:



**Case 2:**  $-1 < a_1 < 0$

In this case,  $w_1 = -a_1$  is positive and  $r(k)$  varies with  $k$  as follows:



1.7 (a) The second-order AR process  $u(n)$  is described by the difference equation:

$$u(n) = u(n-1) - 0.5u(n-2) + v(n)$$

Hence

$$w_1 = 1$$

$$w_2 = -0.5$$

and the AR parameters equal

$$a_1 = -1$$

$$a_2 = 0.5$$

Accordingly, we write the Yule-Walker equations as

$$\begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ -0.5 \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \end{bmatrix}$$

(b) Writing the Yule-Walker equations in expanded form:

$$r(0) - 0.5r(1) = r(1)$$

$$r(1) - 0.5r(0) = r(2)$$

Solving the first relation for  $r(1)$ :

$$r(1) = \frac{2}{3}r(0) \tag{1}$$

Solving the second relation for  $r(2)$ :

$$r(2) = \frac{1}{6}r(0) \tag{2}$$

(c) Since the noise  $v(n)$  has zero mean, so will the AR process  $u(n)$ . Hence,

$$\begin{aligned} \text{var}[u(n)] &= E[(u^2n)] \\ &= r(0). \end{aligned}$$

We know that

$$\begin{aligned} \sigma_v^2 &= \sum_{k=0}^2 a_k r(k) \\ &= r(0) + a_1 r(1) + a_2 r(2) \end{aligned} \tag{3}$$

Substituting (1) and (2) into (3), and solving for  $r(0)$ , we get

$$r(0) = \frac{\sigma_v^2}{1 + \frac{2}{3}a_1\frac{1}{6}a_2} = 1.2$$

1.8 By definition,

$P_0$  = average power of the AR process  $u(n)$

$$\begin{aligned}
&= E[|u(n)|^2] \\
&= r(0)
\end{aligned} \tag{1}$$

where  $r(0)$  is the autocorrelation function of  $u(n)$  for zero lag. We note that

$$\{a_1, a_2, \dots, a_M\} \Leftrightarrow \left\{ \frac{r(1)}{r(0)}, \frac{r(2)}{r(0)}, \dots, \frac{r(M)}{r(0)} \right\}$$

Equivalently, except for the scaling factor  $r(0)$ ,

$$\{a_1, a_2, \dots, a_M\} \Leftrightarrow \{r(1), r(2), \dots, r(M)\} \tag{2}$$

Combining Eqs. (1) and (2):

$$\{P_0, a_1, a_2, \dots, a_M\} \Leftrightarrow \{r(0), r(1), r(2), \dots, r(M)\} \tag{3}$$

1.9 (a) The transfer function of the MA model of Fig. 2.3 is

$$H(z) = 1 + b_1^* z^{-1} + b_2^* z^{-2} + \dots + b_K^* z^{-K}$$

(b) The transfer function of the ARMA model of Fig. 2.4 is

$$H(z) = \frac{b_0^* + b_1^* z^{-1} + b_2^* z^{-2} + \dots + b_K^* z^{-K}}{1 + a_1^* z^{-1} + a_2^* z^{-2} + \dots + a_M^* z^{-M}}$$

(c) The ARMA model reduces to an AR model when

$$b_0 = b_1 = \dots = b_K = 0$$

It reduces to an MA model when

$$a_1 = a_2 = \dots = a_M = 0$$

1.10 We are given

$$x(n) = v(n) + 0.75v(n-1) + 0.25v(n-2)$$

Taking the  $z$ -transforms of both sides:

$$X(z) = (1 + 0.75z^{-1} + 0.25z^{-2})V(z)$$

Hence, the transfer function of the MA model is

$$\begin{aligned} \frac{X(z)}{V(z)} &= 1 + 0.75z^{-1} + 0.25z^{-2} \\ &= \frac{1}{(1 + 0.75z^{-1} + 0.25z^{-2})^{-1}} \end{aligned} \quad (1)$$

Using long division, we may perform the following expansion of the denominator in Eq. (1):

$$\begin{aligned} (1 + 0.75z^{-1} + 0.25z^{-2})^{-1} &= 1 - \frac{3}{4}z^{-1} + \frac{5}{16}z^{-2} - \frac{3}{64}z^{-3} - \frac{11}{256}z^{-4} + \frac{45}{1024}z^{-5} \\ &\quad - \frac{91}{4096}z^{-6} + \frac{93}{16283}z^{-7} + \frac{85}{65536}z^{-8} - \frac{627}{262144}z^{-9} + \frac{1541}{1048576}z^{-10} + \dots \\ &\approx 1 - 0.75z^{-1} + 0.3125z^{-2} - 0.0469z^{-3} - 0.043z^{-4} + 0.0439z^{-5} \\ &\quad - 0.0222z^{-6} + 0.0057z^{-7} + 0.0013z^{-8} - 0.0024z^{-9} + 0.0015z^{-10} \end{aligned} \quad (2)$$

(a)  $M = 2$

Retaining terms in Eq. (2) up to  $z^{-2}$ , we may approximate the MA model with an AR model of order two as follows:

$$\frac{X(z)}{V(z)} \approx \frac{1}{1 - 0.75z^{-1} + 0.3125z^{-2}}$$

(b)  $M = 5$

Retaining terms in Eq. (2) up to  $z^{-5}$ , we obtain the following approximation in the forms of an AR model of order five:

$$\frac{X(z)}{V(z)} \approx \frac{1}{1 - 0.75z^{-1} + 0.3125z^{-2} - 0.0469z^{-3} - 0.043z^{-4} + 0.0439z^{-5}}$$

(c)  $M = 10$

Finally, retaining terms in Eq. (2) up to  $z^{-10}$ , we obtain the following approximation in the form of an AR model of order ten:

$$\frac{X(z)}{V(z)} \approx \frac{1}{D(z)}$$

where  $D(z)$  is given by the polynomial on the right-hand side of Eq. (2).

1.11 (a) The filter output is

$$x(n) = \mathbf{w}^H \mathbf{u}(n)$$

where  $\mathbf{u}(n)$  is the tap-input vector. The average power of the filter output is therefore

$$\begin{aligned} E[|x(n)|^2] &= E[\mathbf{w}^H \mathbf{u}(n) \mathbf{u}^H(n) \mathbf{w}] \\ &= \mathbf{w}^H E[\mathbf{u}(n) \mathbf{u}^H(n)] \mathbf{w} \\ &= \mathbf{w}^H \mathbf{R} \mathbf{w} \end{aligned}$$

(b) If  $\mathbf{u}(n)$  is extracted from a zero mean white noise of variance  $\sigma^2$ , we have

$$\mathbf{R} = \sigma^2 \mathbf{I}$$

where  $\mathbf{I}$  is the identity matrix. Hence,

$$E[|x(n)|^2] = \sigma^2 \mathbf{w}^H \mathbf{w}$$

1.12 (a) The process  $u(n)$  is a linear combination of Gaussian samples. Hence,  $u(n)$  is Gaussian.

(b) From inverse filtering, we recognize that  $v(n)$  may also be expressed as a linear combination of samples represented by  $u(n)$ . Hence, if  $u(n)$  is Gaussian, then  $v(n)$  is also Gaussian.

1.13 (a) From the Gaussian moment factoring theorem:

$$E[(u_1^* u_2)^k] = E[u_1^* \cdots u_1^* u_2 \cdots u_2]$$

$$\begin{aligned}
&= k! E[u_1^* u_2] \cdots E[u_1^* u_2] \\
&= k! (E[u_1^* u_2])^k
\end{aligned} \tag{1}$$

(b) Putting  $u_2 = u_1 = u$ , Eq. (1) reduces to

$$E[|u|^{2k}] = k! (E[|u|^2])^k$$

1.14 It is not permissible to interchange the order of expectation and limiting operations in Eq. (1.113). The reason is that the expectation is a linear operation, whereas the limiting operation with respect to the number of samples  $N$  is nonlinear.

1.15 The filter output is

$$y(n) = \sum_i h(i)u(n-i)$$

Similarly, we may write

$$y(m) = \sum_k h(k)u(m-k)$$

Hence,

$$\begin{aligned}
r_y(n, m) &= E[y(n)y^*(m)] \\
&= E\left[\sum_i h(i)u(n-i) \sum_k h^*(k)u^*(m-k)\right] \\
&= \sum_i \sum_k h(i)h^*(k)E[u(n-i)u^*(m-k)] \\
&= \sum_i \sum_k h(i)h^*(k)r_u(n-i, m-k)
\end{aligned}$$

1.16 The mean-square value of the filter output in response to white noise input is

$$P_o = \frac{2\sigma^2 \Delta\omega}{\pi}$$

The value  $P_o$  is linearly proportional to the filter bandwidth  $\Delta\omega$ . This relation holds irrespective of how small  $\Delta\omega$  is, compared to the mid-band frequency of the filter.

1.17 (a) The variance of the filter output is

$$\sigma_y^2 = \frac{2\sigma^2\Delta\omega}{\pi}$$

We are given

$$\sigma^2 = 0.1 \text{ volt}^2$$

$$\Delta\omega = 2\pi \times 1 \text{ radians/sec.}$$

Hence,

$$\sigma_y^2 = \frac{2 \times 0.1 \times 2}{\pi} = 0.4 \text{ volt}^2$$

(b) The pdf of the filter output  $y$  is

$$\begin{aligned} f(y) &= \frac{1}{\sqrt{2\pi}\sigma_y} e^{-y^2/2\sigma_y^2} \\ &= \frac{1}{0.63\sqrt{2\pi}} e^{-y^2/0.8} \end{aligned}$$

1.18 (a) We are given

$$U_k = \sum_{n=-\infty}^{N-1} u(n) \exp(-jn\omega_k), \quad k = 0, 1, \dots, N-1$$

where  $u(n)$  is real valued and

$$\omega_k = \frac{2\pi}{N}k$$

Hence,

$$\begin{aligned}
E[U_k U_l^*] &= E \left[ \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} u(n) u(m) \exp(-jn\omega_k + jm\omega_l) \right] \\
&= \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} \exp(-jn\omega_k + jm\omega_l) E[u(n)u(m)] \\
&= \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} \exp(-jn\omega_k + jm\omega_l) r(n-m) \\
&= \sum_{m=0}^{N-1} \exp(jm\omega_k) \sum_{n=0}^{N-1} r(n-m) \exp(-jn\omega_k) \tag{1}
\end{aligned}$$

By definition, we also have

$$\sum_{n=0}^{N-1} r(n) \exp(-jn\omega_k) = S_k$$

Moreover, since  $r(n)$  is periodic with period  $N$ , we may invoke the time-shifting property of the discrete Fourier transform to write

$$\sum_{n=0}^{N-1} r(n-m) \exp(-jn\omega_k) = \exp(-jm\omega_k) S_k$$

Thus, recognizing that  $\omega_k = (2\pi/N)k$ , Eq. (1) reduces to

$$\begin{aligned}
E[U_k U_l^*] &= S_k \sum_{m=0}^{N-1} \exp(jm(\omega_l - \omega_k)) \\
&= \begin{cases} S_k, & l = k \\ 0, & \text{otherwise} \end{cases}
\end{aligned}$$

- (b) Part (a) shows that the complex spectral samples  $U_k$  are uncorrelated. If they are Gaussian, then they will also be statistically independent. Hence,



$$f_{\mathbf{U}}(U_0, U_1, \dots, U_{N-1}) = \frac{1}{(2\pi)^N \det(\Lambda)} \exp\left(-\frac{1}{2} \mathbf{U}^H \Lambda \mathbf{U}\right)$$

where

$$\mathbf{U} = [U_0, U_1, \dots, U_{N-1}]^T$$

$$\begin{aligned} \Lambda &= \frac{1}{2} E[\mathbf{U} \mathbf{U}^H] \\ &= \frac{1}{2} \text{diag}(S_0, S_1, \dots, S_{N-1}) \end{aligned}$$

$$\det(\Lambda) = \frac{1}{2^N} \prod_{k=0}^{N-1} S_k$$

Therefore,

$$\begin{aligned} f_{\mathbf{U}}(U_0, U_1, \dots, U_{N-1}) &= \frac{1}{(2\pi)^N 2^{-N} \prod_{k=0}^{N-1} S_k} \exp\left(-\frac{1}{2} \sum_{k=0}^{N-1} \frac{|U_k|^2}{\frac{1}{2} S_k}\right) \\ &= \pi^{-N} \exp\left(\sum_{k=0}^{N-1} -\left(\frac{|U_k|^2}{S_k}\right) - \ln S_k\right) \end{aligned}$$

1.19 The mean square value of the increment process  $dz(\omega)$  is

$$E[|dz(\omega)|^2] = S(\omega) d\omega$$

Hence  $E[|dz(\omega)|^2]$  is measured in watts.

1.20 The third-order cumulant of a process  $u(n)$  is

$$\begin{aligned} c_3(\tau_1, \tau_2) &= E[u(n)u(n + \tau_1)u(n + \tau_2)] \\ &= \text{third-order moment.} \end{aligned}$$

All odd-order moments of a Gaussian process are known to be zero; hence,

$$c_3(\tau_1, \tau_2) = 0$$

The fourth-order cumulant is

$$\begin{aligned} c_4(\tau_1, \tau_2, \tau_3) = & E[u(n)u(n + \tau_1)u(n + \tau_2)u(n + \tau_3)] \\ & - E[u(n)u(n + \tau_1)]E[u(n + \tau_2)u(n + \tau_3)] \\ & - E[u(n)u(n + \tau_2)]E[u(n + \tau_1)u(n + \tau_3)] \\ & - E[u(n)u(n + \tau_3)]E[u(n + \tau_1)u(n + \tau_2)] \end{aligned}$$

For the special case of  $\tau = \tau_1 = \tau_2 = \tau_3$ , the fourth-order moment of a zero-mean Gaussian process of variance  $\sigma^2$  is  $3\sigma^4$ , and its second-order moment of  $\sigma^2$ . Hence, the fourth-order cumulant is zero. Indeed, all cumulants higher than order two are zero.

1.21 The trispectrum is

$$C_4(\omega_1, \omega_2, \omega_3) = \sum_{\tau_1=-\infty}^{\infty} \sum_{\tau_2=-\infty}^{\infty} \sum_{\tau_3=-\infty}^{\infty} c_4(\tau_1, \tau_2, \tau_3) e^{-j(\omega_1\tau_1 + \omega_2\tau_2 + \omega_3\tau_3)}$$

Let the process be passed through a three-dimensional band-pass filter centered on  $\omega_1, \omega_2$ , and  $\omega_3$ . We assume that the bandwidth (along each dimension) is small compared to the respective center frequency. The average power of the filter output is proportional to the trispectrum,  $C_4(\omega_1, \omega_2, \omega_3)$ .

1.22 (a) Starting with the formula

$$c_k(\tau_1, \tau_2, \dots, \tau_{k-1}) = \gamma_k \sum_{i=-\infty}^{\infty} h_i h_{i+\tau_1} \dots h_{i+\tau_{k-1}}$$

the third-order cumulant of the filter output is

$$c_3(\tau_1, \tau_2) = \gamma_3 \sum_{i=-\infty}^{\infty} h_i h_{i+\tau_1} h_{i+\tau_2}$$

where  $\gamma_3$  is the third-order cumulant of the filter input. The bispectrum is

$$\begin{aligned}
C_3(\omega_1, \omega_2) &= \gamma_3 \sum_{\tau_1=-\infty}^{\infty} \sum_{\tau_2=-\infty}^{\infty} c_3(\tau_1, \tau_2) e^{-j(\omega_1 \tau_1 + \omega_2 \tau_2)} \\
&= \gamma_3 \sum_{i=-\infty}^{\infty} \sum_{\tau_1=-\infty}^{\infty} \sum_{\tau_2=-\infty}^{\infty} h_i h_{i+\tau_1} h_{i+\tau_2} e^{-j(\omega_1 \tau_1 + \omega_2 \tau_2)}
\end{aligned}$$

Hence,

$$C_3(\omega_1, \omega_2) = \gamma_3 H(e^{j\omega_1}) H(e^{j\omega_2}) H^*(e^{j(\omega_1 + \omega_2)})$$

(b) From this formula, we immediately deduce that

$$\arg[C_3(\omega_1, \omega_2)] = \arg[H(e^{j\omega_1})] + \arg[H(e^{j\omega_2})] - \arg[H(e^{j(\omega_1 + \omega_2)})]$$

1.23 The output of a filter of impulse response  $h_i$  due to an input  $u(i)$  is given by the convolution sum

$$y(n) = \sum_i h_i u(n-i)$$

The third-order cumulant of the filter output is, for example,

$$\begin{aligned}
C_3(\tau_1, \tau_2) &= E[y(n)y(n+\tau_1)y(n+\tau_2)] \\
&= E\left[\sum_i h_i u(n-i) \sum_k h_k u(n+\tau_1-k) \sum_l h_l u(n+\tau_2-l)\right] \\
&= E\left[\sum_i h_i u(n-i) \sum_k h_{k+\tau_1} u(n-k) \sum_l h_{l+\tau_2} u(n-l)\right] \\
&= \sum_i \sum_k \sum_l h_i h_{k+\tau_1} h_{l+\tau_2} E[u(n-i)u(n-k)u(n-l)]
\end{aligned}$$

For an input sequence of independent and identically distributed random variables, we note that

$$E[u(n-i)u(n-k)u(n-l)] = \begin{cases} \gamma_3 & i = k = l \\ 0, & \text{otherwise} \end{cases}$$

Hence,

$$C_3(\tau_1, \tau_2) = \gamma_3 \sum_{i=-\infty}^{\infty} h_i h_{i+\tau_1} h_{i+\tau_2}$$

In general, we may thus write

$$C_3(\tau_1, \tau_2, \dots, \tau_{k-1}) = \gamma_k \sum_{i=-\infty}^{\infty} h_i h_{i+\tau_1} \dots h_{i+\tau_{k-1}}$$

1.24 By definition:

$$r^{(\alpha)}(k) = \frac{1}{N} \sum_{n=0}^{N-1} E[u(n)u^*(n-k)e^{-j2\pi\alpha n}]e^{j\pi\alpha k}$$

Hence,

$$r^{(\alpha)}(-k) = \frac{1}{N} \sum_{n=0}^{N-1} E[u(n)u^*(n+k)e^{-j2\pi\alpha n}]e^{-j\pi\alpha k}$$

$$r^{(\alpha)*}(k) = \frac{1}{N} \sum_{n=0}^{N-1} E[u^*(n)u(n-k)e^{j2\pi\alpha n}]e^{-j\pi\alpha k}$$

We are told that the process  $u(n)$  is cyclostationary, which means that

$$E[u(n)u^*(n+k)e^{-j2\pi\alpha n}] = E[u^*(n)u(n-k)e^{j2\pi\alpha n}]$$

It follows therefore that

$$r^{(\alpha)}(-k) = r^{(\alpha)*}(k)$$

1.25 For  $\alpha = 0$ , the input to the time-average cross-correlator reduces to the squared amplitude of a narrow-band filter with mid-band frequency  $\omega$ . Correspondingly, the time-average cross-correlator reduces to an average power meter. Thus, for  $\alpha = 0$ , the instrumentation of Fig. 1.16 reduces to that of Fig. 1.13.

## CHAPTER 2

2.1 (a) Let

$$w_k = x + jy$$

$$p(-k) = a + jb$$

We may then write

$$f = w_k p^*(-k)$$

$$= (x + jy)(a - jb)$$

$$= (ax + by) + j(ay - bx)$$

Let

$$f = u + jv$$

with

$$u = ax + by$$

$$v = ay - bx$$

Hence,

$$\frac{\partial u}{\partial x} = a \qquad \frac{\partial u}{\partial y} = b$$

$$\frac{\partial v}{\partial y} = a \qquad \frac{\partial v}{\partial x} = -b$$

From these results we immediately see that

$$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial y}$$

$$\frac{\partial v}{\partial x} = -\frac{\partial u}{\partial y}$$

In other words, the product term  $w_k p^*(-k)$  satisfies the Cauchy-Rieman equations, and so this term is analytic.

(b) Let

$$\begin{aligned} f &= w_k^* p(-k) \\ &= (x - jy)(a + jb) \\ &= (ax + by) + j(bx - ay) \end{aligned}$$

Let

$$f = u + jv$$

with

$$u = ax + by$$

$$v = bx - ay$$

Hence,

$$\frac{\partial u}{\partial x} = a \quad \frac{\partial u}{\partial y} = b$$

$$\frac{\partial v}{\partial x} = b \quad \frac{\partial v}{\partial y} = -a$$

From these results we immediately see that

$$\frac{\partial u}{\partial x} \neq \frac{\partial v}{\partial y}$$

$$\frac{\partial v}{\partial x} = - \frac{\partial u}{\partial y}$$

In other words, the product term  $w_k^* p(-k)$  does not satisfy the Cauchy-Rieman equations, and so this term is *not* analytic.

2.2 (a) From the Wiener-Hopf equation, we have

$$\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{p} \tag{1}$$

We are given

$$\mathbf{R} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} 0.5 \\ 0.25 \end{bmatrix}$$

Hence, the inverse matrix  $\mathbf{R}^{-1}$  is

$$\begin{aligned} \mathbf{R}^{-1} &= \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}^{-1} \\ &= \frac{1}{0.75} \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix} \end{aligned}$$

Using Eq. (1), we therefore get

$$\begin{aligned} \mathbf{w}_o &= \frac{1}{0.75} \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.25 \end{bmatrix} \\ &= \frac{1}{3} \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \end{bmatrix} \\ &= \frac{1}{3} \begin{bmatrix} 1.5 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} \end{aligned}$$

(b) The minimum mean-square error is

$$J_{\min} = \sigma_d^2 - \mathbf{p}^H \mathbf{w}_o$$

$$= \sigma_d^2 - [0.5, 0.25] \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}$$

$$= \sigma_d^2 - 0.25$$

(c) The eigenvalues of matrix **R** are roots of the characteristic equation

$$(1 - \lambda)^2 - (0.5)^2 = 0$$

That is, the two roots are

$$\lambda_1 = 0.5 \text{ and } \lambda_2 = 1.5$$

The associated eigenvectors are defined by

$$\mathbf{R} \mathbf{q} = \lambda \mathbf{q}$$

For  $\lambda_1 = 0.5$ , we have

$$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \begin{bmatrix} q_{11} \\ q_{12} \end{bmatrix} = 0.5 \begin{bmatrix} q_{11} \\ q_{12} \end{bmatrix}$$

Expanding

$$q_{11} + 0.5 q_{12} = 0.5 q_{11}$$

$$0.5 q_{11} + q_{12} = 0.5 q_{12}$$

Therefore,

$$q_{11} = -q_{12}$$

Normalizing the eigenvector  $\mathbf{q}_1$  to unit length, we therefore have

$$\mathbf{q}_1 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$



Similarly, for the eigenvalue  $\lambda_2 = 1.5$ , we may show that

$$\mathbf{q}_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Accordingly, we may express the Wiener filter in terms of its eigenvalues and eigenvectors as follows:

$$\begin{aligned} \mathbf{w}_o &= \left( \sum_{i=1}^2 \frac{1}{\lambda_i} \mathbf{q}_i \mathbf{q}_i^H \right) \mathbf{p} \\ &= \left( \begin{bmatrix} 1 \\ -1 \end{bmatrix} [1, -1] + \frac{1}{3} \begin{bmatrix} 1 \\ 1 \end{bmatrix} [1, 1] \right) \begin{bmatrix} 0.5 \\ 0.25 \end{bmatrix} \\ &= \underbrace{\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}}_{\frac{1}{\lambda_1} \mathbf{q}_1 \mathbf{q}_1^H} + \underbrace{\frac{1}{3} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}}_{\frac{1}{\lambda_2} \mathbf{q}_2 \mathbf{q}_2^H} \underbrace{\begin{bmatrix} 0.5 \\ 0.25 \end{bmatrix}}_{\mathbf{p}} \end{aligned}$$

2.3 (a) From the Wiener-Hopf equation we have

$$\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{p} \tag{1}$$

We are given

$$\mathbf{R} = \begin{bmatrix} 1 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 1 \end{bmatrix}$$

and

$$\mathbf{p} = [0.5 \ 0.25 \ 0.125]^T$$

Hence, the use of these values in Eq. (1) yields

$$\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{p}$$

$$= \begin{bmatrix} 1 & 0.5 & 0.25 \\ 0.5 & 1 & 0.5 \\ 0.25 & 0.5 & 1 \end{bmatrix}^{-1} \begin{bmatrix} 0.5 \\ 0.25 \\ 0.125 \end{bmatrix}$$

$$= \begin{bmatrix} 1.33 & -0.67 & 0 \\ -0.67 & 1.67 & -0.67 \\ 0 & -0.67 & 1.33 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0.25 \\ 0.125 \end{bmatrix}$$

$$= \begin{bmatrix} 0.5 & 0 & 0 \end{bmatrix}^T$$

(b) The minimum mean-square error is

$$\begin{aligned} J_{\min} &= \sigma_d^2 - \mathbf{p}^H \mathbf{w}_o \\ &= \sigma_d^2 - \begin{bmatrix} 0.5 & 0.25 & 0.125 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0 \\ 0 \end{bmatrix} \\ &= \sigma_d^2 - 0.25 \end{aligned}$$

(c) The eigenvalues of matrix  $\mathbf{R}$  are

$$\lambda = 0.4069, \quad 0.75, \quad 1.8431$$

The corresponding eigenvectors constitute the orthogonal matrix:

$$\mathbf{Q} = \begin{bmatrix} -0.4544 & -0.7071 & 0.5418 \\ 0.7662 & 0 & 0.6426 \\ -0.4544 & 0.7071 & 0.5418 \end{bmatrix}$$

Accordingly, we may express the Wiener filter in terms of its eigenvalues and eigenvectors as follows:

$$\begin{aligned}
\mathbf{w}_o &= \left( \sum_{i=1}^3 \frac{1}{\lambda_i} \mathbf{q}_i \mathbf{q}_i^H \right) \mathbf{p} \\
&= \left\{ \left[ \frac{1}{0.4069} \begin{bmatrix} -0.4544 \\ 0.7662 \\ -0.4544 \end{bmatrix} \begin{bmatrix} -0.4544 & 0.7662 & -0.4544 \end{bmatrix} \right. \right. \\
&\quad \left. \left. + \frac{1}{0.75} \begin{bmatrix} -0.7071 \\ 0 \\ 0.7071 \end{bmatrix} \begin{bmatrix} -0.7071 & 0 & 0.7071 \end{bmatrix} \right. \right. \\
&\quad \left. \left. \left\{ + \frac{1}{1.8431} \begin{bmatrix} 0.5418 \\ 0.6426 \\ 0.5418 \end{bmatrix} \begin{bmatrix} 0.5418 & 0.6426 & 0.5418 \end{bmatrix} \right\} \times \begin{bmatrix} 0.5 \\ 0.25 \\ 0.125 \end{bmatrix} \right. \right. \\
&= \left\{ \frac{1}{0.4069} \begin{bmatrix} 0.2065 & -0.3482 & 0.2065 \\ -0.3482 & 0.5871 & -0.3482 \\ 0.2065 & -0.3482 & 0.2065 \end{bmatrix} \right. \\
&\quad \left. + \frac{1}{0.75} \begin{bmatrix} 0.5 & 0 & -0.5 \\ 0 & 0 & 0 \\ -0.5 & 0 & 0.5 \end{bmatrix} \right. \\
&\quad \left. + \frac{1}{1.8431} \begin{bmatrix} 0.2935 & 0.3482 & 0.2935 \\ 0.3482 & 0.4129 & 0.3482 \\ 0.2935 & 0.3482 & 0.2935 \end{bmatrix} \right\} \begin{bmatrix} 0.5 \\ 0.25 \\ 0.125 \end{bmatrix}
\end{aligned}$$

2.4 By definition, the correlation matrix

$$\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)]$$

where

$$\mathbf{u}(n) = \begin{bmatrix} u(n) \\ u(n-1) \\ \vdots \\ u(0) \end{bmatrix}$$

Invoking the ergodicity theorem,

$$\mathbf{R}(N) = \frac{1}{N+1} \sum_{n=0}^N \mathbf{u}(n) \mathbf{u}^H(n)$$

Likewise, we may compute the cross-correlation vector

$$\mathbf{p} = E[\mathbf{u}(n) d^*(n)]$$

as the time average

$$\mathbf{p}(N) = \frac{1}{N+1} \sum_{n=0}^N \mathbf{u}(n) d^*(n)$$

The tap-weight vector of the Wiener filter is thus defined by

$$\mathbf{w}_o(N) = \left( \sum_{n=0}^N \mathbf{u}(n) \mathbf{u}^H(n) \right)^{-1} \left( \sum_{n=0}^N \mathbf{u}(n) d^*(n) \right)$$

which is dependent on the length  $(N+1)$  of the time series.

2.5 (a)  $\mathbf{R} = E[\mathbf{u}(n) \mathbf{u}^H(n)]$

$$= E[(\alpha(n)\mathbf{s}(n) + \mathbf{v}(n))(\alpha^*(n)\mathbf{s}^H(n) + \mathbf{v}^H(n))]$$

With  $\alpha(n)$  uncorrelated with  $\mathbf{v}(n)$ , we have

$$\begin{aligned} \mathbf{R} &= E[|\alpha(n)|^2 \mathbf{s}(n) \mathbf{s}^H(n) + E[\mathbf{v}(n) \mathbf{v}^H(n)]] \\ &= \sigma_\alpha^2 \mathbf{s}(n) \mathbf{s}^H(n) + \mathbf{R}_v \end{aligned} \tag{1}$$

where  $\mathbf{R}_v$  is the correlation matrix of  $\mathbf{v}(n)$ .

(b) The cross-correlation vector between the input vector  $\mathbf{u}(n)$  and desired response  $d(n)$  is

$$\mathbf{p} = E[\mathbf{u}(n)d^*(n)] \quad (2)$$

If  $d(n)$  is uncorrelated with  $\mathbf{u}(n)$ , we have

$$\mathbf{p} = \mathbf{0}.$$

Hence, the tap-weight of the Wiener filter is

$$\begin{aligned} \mathbf{w}_o &= \mathbf{R}^{-1}\mathbf{p} \\ &= \mathbf{0} \end{aligned}$$

(c) With  $\sigma_\alpha^2 = 0$ , Eq. (1) reduces to

$$\mathbf{R} = \mathbf{R}_v$$

With the desired response

$$d(n) = v(n-k)$$

Eq. (2) yields

$$\begin{aligned} \mathbf{p} &= E[(\alpha(n)\mathbf{s}(n) + \mathbf{v}(n)v^*(n-k))] \\ &= E[\mathbf{v}(n)v^*(n-k)] \\ &= E\left\{\begin{bmatrix} v(n) \\ v(n-1) \\ \vdots \\ v(n-M+1) \end{bmatrix} (v^*(n-k))\right\} \\ &= \begin{bmatrix} r_v(n) \\ r_v(k-1) \\ \vdots \\ r_v(k-M+1) \end{bmatrix}, \quad 0 \leq k \leq M-1 \end{aligned} \quad (3)$$

where  $r_v(k)$  is the autocorrelation of  $v(n)$  for lag  $k$ . Accordingly, the tap-weight vector of the (optimum) Wiener filter is

$$\mathbf{w}_o = \mathbf{R}^{-1}\mathbf{p}$$

$$= \mathbf{R}_v^{-1} \mathbf{p}$$

where  $\mathbf{p}$  is as defined in Eq. (3).

(d) For a desired response

$$d(n) = \alpha(n)e^{j\omega\tau}$$

the cross-correlation vector  $\mathbf{p}$  is

$$\mathbf{p} = E[\mathbf{u}(n)(d^* n)]$$

$$= E[\alpha(n)\mathbf{s}(n) + \mathbf{v}(n)\alpha^*(n)e^{j\omega\tau}]$$

$$= \mathbf{s}(n)e^{j\omega\tau}E[|\alpha(n)|^2]$$

$$= \sigma_\alpha^2 \mathbf{s}(n)e^{j\omega\tau}$$

$$= \sigma_\alpha^2 \begin{bmatrix} 1 \\ e^{-j\omega} \\ \vdots \\ e^{-j\omega(M-1)} \end{bmatrix} e^{j\omega\tau}$$

$$= \sigma_\alpha^2 \begin{bmatrix} e^{j\omega\tau} \\ e^{j\omega(\tau-1)} \\ \vdots \\ e^{j\omega(\tau-M+1)} \end{bmatrix}$$

The corresponding value of the tap-weight vector of the Wiener filter is

$$\mathbf{w}_o = \sigma_\alpha^2 (\sigma_\alpha^2 \mathbf{s}(n)\mathbf{s}^H(n) + \mathbf{R}_v)^{-1} \begin{bmatrix} e^{j\omega\tau} \\ e^{j\omega(\tau-1)} \\ \vdots \\ e^{j\omega(\tau-M+1)} \end{bmatrix}$$

$$= \left( \mathbf{s}(n)\mathbf{s}^H(n) + \frac{1}{\sigma_\alpha^2} \mathbf{R}_v \right)^{-1} \begin{bmatrix} e^{j\omega\tau} \\ e^{j\omega(\tau-1)} \\ \vdots \\ e^{j\omega(\tau-M+1)} \end{bmatrix}$$

2.6 The optimum filtering solution is defined by the Wiener-Hopf equation

$$\mathbf{R}\mathbf{w}_o = \mathbf{p} \quad (1)$$

for which the minimum mean-square error equals

$$J_{\min} = \sigma_d^2 - \mathbf{p}^H \mathbf{w}_o \quad (2)$$

Combine Eqs. (1) and (2) into a single relation:

$$\begin{bmatrix} \sigma_d^2 & \mathbf{p}^H \\ \mathbf{p} & \mathbf{R} \end{bmatrix} \begin{bmatrix} 1 \\ -\mathbf{w}_o \end{bmatrix} = \begin{bmatrix} J_{\min} \\ \mathbf{0} \end{bmatrix}$$

Define

$$\mathbf{A} = \begin{bmatrix} \sigma_d^2 & \mathbf{p}^H \\ \mathbf{p} & \mathbf{R} \end{bmatrix} \quad (3)$$

Since

$$\sigma_d^2 = E[d(n)d^*(n)],$$

$$\mathbf{p} = E[\mathbf{u}(n)d^*(n)], \text{ and}$$

$$\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)],$$

we may rewrite Eq. (3) as

$$\mathbf{A} = \begin{bmatrix} E[d(n)d^*(n)] & E[d(n)]\mathbf{u}^H(n) \\ E[\mathbf{u}(n)d^*(n)] & E[\mathbf{u}(n)\mathbf{u}^H(n)] \end{bmatrix}$$

$$= E \left\{ \begin{bmatrix} d(n) \\ \mathbf{u}(n) \end{bmatrix} \begin{bmatrix} d^*(n), \mathbf{u}^H(n) \end{bmatrix} \right\}$$

The minimum mean-square error equals

$$J_{\min} = \sigma_d^2 - \mathbf{p}^H \mathbf{w}_o \quad (4)$$

Eliminate  $\sigma_d^2$  between Eqs. (1) and (4):

$$J(\mathbf{w}) = J_{\min} + \mathbf{p}^H \mathbf{w}_o - \mathbf{p}^H \mathbf{w} - \mathbf{w}^H \mathbf{p} + \mathbf{w}^H \mathbf{R} \mathbf{w} \quad (5)$$

Eliminate  $\mathbf{p}$  between (2) and (5):

$$J(\mathbf{w}) = J_{\min} + \mathbf{w}_o^H \mathbf{R} \mathbf{w}_o - \mathbf{w}_o^H \mathbf{R} \mathbf{w} - \mathbf{w}^H \mathbf{R} \mathbf{w}_o + \mathbf{w}^H \mathbf{R} \mathbf{w} \quad (6)$$

where we have used the property  $\mathbf{R}^H = \mathbf{R}$ . We may rewrite Eq. (6) simply as

$$J(\mathbf{w}) = J_{\min} + (\mathbf{w} - \mathbf{w}_o)^H \mathbf{R} (\mathbf{w} - \mathbf{w}_o)$$

which clearly show that  $J(\mathbf{w}_o) = J_{\min}$ .

2.7 The minimum mean-square error equals

$$J_{\min} = \sigma_d^2 - \mathbf{p}^H \mathbf{R}^{-1} \mathbf{p} \quad (1)$$

Using the spectral theorem, we may express the correlation matrix  $\mathbf{R}$  as

$$\begin{aligned} \mathbf{R} &= \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H \\ &= \sum_{k=1}^M \lambda_k \mathbf{q}_k \mathbf{q}_k^H \end{aligned}$$

Hence, the inverse of  $\mathbf{R}$  equals

$$\mathbf{R}^{-1} = \sum_{k=1}^M \frac{1}{\lambda_k} \mathbf{q}_k \mathbf{q}_k^H \quad (2)$$



Substituting Eq. (2) into (1):

$$\begin{aligned}
 J_{\min} &= \sigma_d^2 - \sum_{k=1}^M \frac{1}{\lambda_k} \mathbf{p}^H \mathbf{q}_k \mathbf{q}_k^H \mathbf{p} \\
 &= \sigma_d^2 - \sum_{k=1}^M \frac{1}{\lambda_k} \left| \mathbf{p}^H \mathbf{q}_k \right|^2
 \end{aligned}$$

2.8 When the length of the Wiener filter is greater than the model order  $m$ , the tail end of the tap-weight vector of the Wiener filter is zero; thus,

$$\mathbf{w}_o = \begin{bmatrix} \mathbf{a}_m \\ 0 \end{bmatrix}$$

Therefore, the only possible solution for the case of an over-fitted model is

$$\mathbf{w}_o = \begin{bmatrix} \mathbf{a}_m \\ 0 \end{bmatrix}$$

2.9 (a) The Wiener solution is defined by

$$\mathbf{R}_M \mathbf{a}_M = \mathbf{p}_M$$

$$\begin{bmatrix} \mathbf{R}_M & \mathbf{r}_{M-m} \\ \mathbf{r}_{M-m}^H & \mathbf{R}_{M-m, M-m} \end{bmatrix} \begin{bmatrix} \mathbf{a}_m \\ \mathbf{0}_{M-m} \end{bmatrix} = \begin{bmatrix} \mathbf{p}_m \\ \mathbf{p}_{M-m} \end{bmatrix}$$

$$\mathbf{R}_M \mathbf{a}_m = \mathbf{p}_m$$

$$\mathbf{r}_{M-m}^H \mathbf{a}_m = \mathbf{p}_{M-m}$$

$$\mathbf{p}_{M-m} = \mathbf{r}_{M-m}^H \mathbf{a}_m = \mathbf{r}_{M-m}^H \mathbf{R}_M^{-1} \mathbf{p}_m \quad (1)$$

(b) Applying the condition of Eq. (1) to the example in Section 2.7:

$$\mathbf{r}_{M-m}^H = [-0.05, 0.1, 0.15]$$

$$\mathbf{a}_m = \begin{bmatrix} 0.8719 \\ -0.9129 \\ 0.2444 \end{bmatrix}$$

The last entry in the 4-by-1 vector  $\mathbf{p}$  is therefore

$$\begin{aligned} \mathbf{r}_{M-m}^H \mathbf{a}_m &= -0.0436 - 0.0912 + 0.1222 \\ &= -0.0126 \end{aligned}$$

$$\begin{aligned} 2.10 \quad J_{\min} &= \sigma_d^2 - \mathbf{p}^H \mathbf{w}_o \\ &= \sigma_d^2 - \mathbf{p}^H \mathbf{R}^{-1} \mathbf{p} \end{aligned}$$

When  $m = 0$ ,

$$\begin{aligned} J_{\min} &= \sigma_d^2 \\ &= 1.0 \end{aligned}$$

When  $m = 1$ ,

$$\begin{aligned} J_{\min} &= 1 - 0.5 \times \frac{1}{1.1} \times 0.5 \\ &= 0.9773 \end{aligned}$$

When  $m = 2$ ,

$$\begin{aligned} J_{\min} &= 1 - [0.5 \quad -0.4] \begin{bmatrix} 1.1 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}^{-1} \begin{bmatrix} 0.5 \\ -0.4 \end{bmatrix} \\ &= 1 - 0.6781 \\ &= 0.3219 \end{aligned}$$

When  $m = 3$ ,

$$\begin{aligned}
J_{\min} &= 1 - \begin{bmatrix} 0.5 & -0.4 & -0.2 \end{bmatrix} \begin{bmatrix} 1.1 & 0.5 & 0.1 \\ 0.5 & 1.1 & 0.5 \\ 0.1 & 0.5 & 1.1 \end{bmatrix}^{-1} \begin{bmatrix} 0.5 \\ -0.4 \\ -0.2 \end{bmatrix} \\
&= 1 - 0.6859 \\
&= 0.3141
\end{aligned}$$

When  $m = 4$ ,

$$\begin{aligned}
J_{\min} &= 1 - 0.6859 \\
&= 0.3141
\end{aligned}$$

Thus any further increase in the filter order beyond  $m = 3$  does not produce any meaningful reduction in the minimum mean-square error.

$$2.11 \quad (a) \quad u(n) = x(n) + v_2(n) \quad (1)$$

$$d(n) = -d(n-1) \times 0.8458 + v_1(n) \quad (2)$$

$$x(n) = d(n) + 0.9458x(n-1) \quad (3)$$

$$d(n) = x(n) - 0.9458x(n-1)$$

Using Eqs. (2) and (3):

$$x(n) - 0.9458x(n-1) = 0.8458[-x(n-1) + 0.9458x(n-2)] + v_1(n)$$

Rearranging terms:

$$x(n) = (0.9458 - 0.8458)x(n-1) + 0.8x(n-2) + v_1(n)$$

$$x(n) = 0.1x(n-1) + 0.8x(n-2) + v_1(n)$$

$$(b) \quad u(n) = x(n) + v_2(n)$$

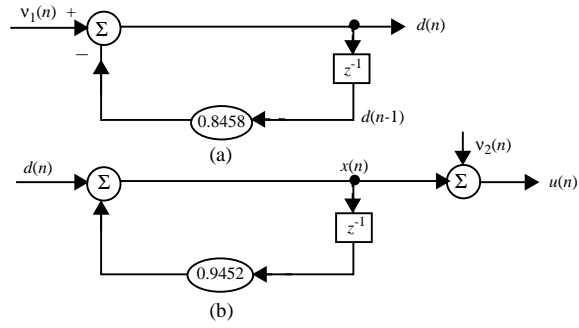
where  $x(n)$  and  $v_2(n)$  are uncorrelated

Therefore,  $\mathbf{R} = \mathbf{R}_x + \mathbf{R}_{v_2}$

$$\mathbf{R}_x = \begin{bmatrix} r_x(0) & r_x(1) \\ r_x(1) & r_x(0) \end{bmatrix}$$

$$r_x(0) = \sigma_x^2$$

$$= \frac{1+a_2}{1-a_2} \cdot \frac{\sigma_1^2}{(1+a_2)^2 - a_1^2} = 1$$



$$r_x(1) = \frac{-a_1}{1+a_2}$$

$$= 0.5$$

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

$$\mathbf{R}_{v_2} = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$$

$$\mathbf{R} = \mathbf{R}_x + \mathbf{R}_2 = \begin{bmatrix} 1.1 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}$$

$$\mathbf{p} = \begin{bmatrix} p(0) \\ p(1) \end{bmatrix}$$

$$p(k) = E[u(n-k) \cdot d(n)], \quad k = 0, 1$$

$$\begin{aligned}
p(0) &= r_x(0) + b_1 r_x(-1) \\
&= 1 - 0.9458 \times 0.5 \\
&= 0.5272
\end{aligned}$$

$$\begin{aligned}
p(1) &= r_x(1) + b_1 r_x(0) \\
&= 0.5 - 0.9458 \\
&= -0.4458
\end{aligned}$$

Therefore,

$$\mathbf{p} = \begin{bmatrix} 0.5272 \\ -0.4458 \end{bmatrix}$$

$$\begin{aligned}
\text{(c) Optimum weight vector } \mathbf{w}_o &= \mathbf{R}^{-1} \mathbf{p} \\
&= \begin{bmatrix} 1.1 & 0.5 \\ 0.5 & 1.1 \end{bmatrix}^{-1} \begin{bmatrix} 0.5272 \\ -0.4458 \end{bmatrix} \\
&= \begin{bmatrix} 0.8363 \\ -0.7853 \end{bmatrix}
\end{aligned}$$

2.12 (a) For  $M = 3$  taps, the correlation matrix of the tap inputs equals

$$\mathbf{R} = \begin{bmatrix} 1.1 & 0.5 & 0.85 \\ 0.5 & 1.1 & 0.5 \\ 0.85 & 0.5 & 1.1 \end{bmatrix}$$

The cross-correlation vector between the tap inputs and the desired response equals

$$\mathbf{p} = \begin{bmatrix} 0.527 \\ -0.446 \\ 0.377 \end{bmatrix}$$

(b) The inverse of the correlation matrix equals

$$\mathbf{R}^{-1} = \begin{bmatrix} 2.234 & -0.304 & -1.666 \\ -0.304 & 1.186 & -0.304 \\ -1.666 & -0.304 & 2.234 \end{bmatrix}$$

Hence, the optimum weight vector equals

$$\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{p} = \begin{bmatrix} 0.738 \\ -0.803 \\ 0.138 \end{bmatrix}$$

The minimum mean-square error equals

$$J_{\min} = 0.15$$

2.13 (a) The correlation matrix  $\mathbf{R}$  is

$$\begin{aligned} \mathbf{R} &= E[\mathbf{u}(n)\mathbf{u}^H(n)] \\ &= E[|A_1|^2] \begin{bmatrix} e^{-j\omega_1 n} \\ e^{-j\omega_1(n-1)} \\ \vdots \\ e^{-j\omega_1(n-M+1)} \end{bmatrix} \begin{bmatrix} e^{+j\omega_1 n}, e^{+j\omega_1(n-1)}, \dots, e^{+j\omega_1(n-M+1)} \end{bmatrix} \\ &= E[|A_1|^2] \mathbf{s}(\omega_1) \mathbf{s}^H(\omega_1) + \mathbf{I} E[|v(n)|^2] \\ &= \sigma_1^2 \mathbf{s}(\omega_1) \mathbf{s}^H(\omega_1) + \sigma_v^2 \mathbf{I} \end{aligned}$$

where  $\mathbf{I}$  is the identity matrix.

(b) The tap-weight vector of the Wiener filter is

$$\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{p}$$

From part (a),

$$\mathbf{R} = \sigma_v^2 \mathbf{I} + \sigma_1^2 \mathbf{s}(\omega_1) \mathbf{s}^H(\omega_1)$$

We are given

$$\mathbf{p} = \sigma_0^2 \mathbf{s}(\omega_o)$$

To invert the matrix  $\mathbf{R}$ , we use the matrix inversion lemma (see Chapter 9), as described here:

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

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

In our case,

$$\mathbf{A} = \sigma_v^2 \mathbf{I}$$

$$\mathbf{B}^{-1} = \sigma_v^2 \mathbf{I}$$

$$\mathbf{D}^{-1} = \sigma_1^2$$

$$\mathbf{C} = \mathbf{s}(\omega_1)$$

Hence,

$$\mathbf{R}^{-1} = \frac{1}{\sigma_v^2} \mathbf{I} - \frac{\frac{1}{\sigma_v^2} \mathbf{s}(\omega_1) \mathbf{s}^H(\omega_1)}{\frac{1}{\sigma_v^2} + \mathbf{s}^H(\omega_1) \mathbf{s}(\omega_1)}$$

The corresponding value of the Wiener tap-weight vector is

$$\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{p}$$

$$= \frac{\sigma_0^2}{\sigma_v^2} \mathbf{s}(\omega_0) - \frac{\frac{\sigma_0^2}{\sigma_v^2} \mathbf{s}(\omega_1) \mathbf{s}^H(\omega_1)}{\frac{1}{\sigma_v^2} + \mathbf{s}^H(\omega_1) \mathbf{s}(\omega_1)} \mathbf{s}(\omega_0)$$

We note that

$$\mathbf{s}^H(\omega_1)\mathbf{s}(\omega_1) = M$$

$$\mathbf{s}^H(\omega_1)\mathbf{s}(\omega_0) = \text{scalar}$$

Hence,

$$\mathbf{w}_o = \frac{\sigma_0^2}{\sigma_v^2} \mathbf{s}(\omega_0) - \left( \frac{\sigma_0^2}{\sigma_v^2} \frac{\mathbf{s}^H(\omega_1)\mathbf{s}(\omega_0)}{\frac{\sigma_v^2}{\sigma_0^2} + M} \mathbf{s}(\omega_1) \right)$$

2.14 The output of the array processor equals

$$e(n) = u(1, n) - wu(2, n)$$

The mean-square error equals

$$\begin{aligned} J(w) &= E[|e(n)|^2] \\ &= E[(u(1, n) - wu(2, n))(u^*(1, n) - w^*u^*(2, n))] \\ &= E[|u(1, n)|^2] + |w|^2 E[|u(2, n)|^2] \\ &\quad - wE[u(2, n)u^*(1, n)] - w^*E[u(1, n)u^*(2, n)] \end{aligned}$$

Differentiate  $J(w)$  with respect to  $w$ :

$$\frac{\partial J}{\partial w} = -2E[u(1, n)u^*(2, n)] + 2wE[|u(2, n)|^2]$$

Putting  $\frac{\partial J(w)}{\partial w} = 0$  and solving for the optimum value of  $w$ :

$$w_o = \frac{E[u(1, n)u^*(2, n)]}{E[|u(2, n)|^2]}$$

2.15 Define the index of performance (i.e., cost function)



$$\begin{aligned}
J(\mathbf{w}) &= E[|e(n)|^2] + \mathbf{c}^H \mathbf{S}^H \mathbf{w} + \mathbf{w}^H \mathbf{S} \mathbf{c} - 2\mathbf{c}^H \mathbf{D}^{1/2} \mathbf{1} \\
&= \mathbf{w}^H \mathbf{R} \mathbf{w} + \mathbf{c}^H \mathbf{S}^H \mathbf{w} + \mathbf{w}^H \mathbf{S} \mathbf{c} - 2\mathbf{c}^H \mathbf{D}^{1/2} \mathbf{1}
\end{aligned}$$

Differentiate  $J(\mathbf{w})$  with respect to  $\mathbf{w}$  and set the result equal to zero:

$$\frac{\partial J}{\partial \mathbf{w}} = 2\mathbf{R}\mathbf{w} + 2\mathbf{S}\mathbf{c} = \mathbf{0}$$

Hence,

$$\mathbf{w}_o = -\mathbf{R}^{-1} \mathbf{S} \mathbf{c}$$

But, we must constrain  $\mathbf{w}_o$  as

$$\mathbf{S}^H \mathbf{w}_o = \mathbf{D}^{1/2} \mathbf{1}$$

This constraint yields

$$-\mathbf{S}^H \mathbf{R}^{-1} \mathbf{S} \mathbf{c} = \mathbf{D}^{1/2} \mathbf{1}$$

Therefore, the vector  $\mathbf{c}$  equals

$$\mathbf{c} = -(\mathbf{S}^H \mathbf{R}^{-1} \mathbf{S})^{-1} \mathbf{D}^{1/2} \mathbf{1}$$

Correspondingly, the optimum weight vector equals

$$\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{S} (\mathbf{S}^H \mathbf{R}^{-1} \mathbf{S})^{-1} \mathbf{D}^{1/2} \mathbf{1}$$

2.16 The weight vector  $\mathbf{w}$  of the beamformer that maximizes the output signal-to-noise ratio

$$(\text{SNR})_o = \frac{\mathbf{w}^H \mathbf{R}_s \mathbf{w}}{\mathbf{w}^H \mathbf{R}_v \mathbf{w}}$$

is derived in part (b) of the solution to Problem 2.18. There it is shown that the optimum weight vector  $\mathbf{w}_{SN}$  so defined is given by

$$\mathbf{w}_{SN} = \mathbf{R}_v^{-1} \mathbf{s} \quad (1)$$

where  $\mathbf{s}$  is the signal component and  $\mathbf{R}_v$  is the correlation matrix of the noise component  $\mathbf{v}(n)$ . On the other hand, the optimum weight vector of the LCMV beamformer is defined by

$$\mathbf{w}_o = g^* \frac{\mathbf{R}^{-1} \mathbf{s}(\phi)}{\mathbf{s}^H(\phi) \mathbf{R}^{-1} \mathbf{s}(\phi)} \quad (2)$$

where  $\mathbf{s}(\phi)$  is the steering vector. In general, the formulas (1) and (2) yield different values for the weight vector of the beamformer.

- 2.17 Let  $\tau_i$  be the propagation delay, measured from the zero-time reference to the  $i$ th element of a nonuniformly spaced array, for a plane wave arriving from a direction defined by angle  $\theta$  with respect to the perpendicular to the array. For a signal of angular frequency  $\omega$ , this delay amounts to a phase shift equal to  $-\omega\tau_i$ . Let the phase shifts for all elements of the array be collected together in a column vector denoted by  $\mathbf{d}(\omega, \theta)$ . The response of a beamformer with weight vector  $\mathbf{w}$  to a signal (with angular frequency  $\omega$ ) originating from angle  $\theta = \mathbf{w}^H \mathbf{d}(\omega, \theta)$ . Hence, constraining the response of the array at  $\omega$  and  $\theta$  to some value  $g$  involves the linear constraint

$$\mathbf{w}^H \mathbf{d}(\omega, \theta) = g$$

Thus, the constraint vector  $\mathbf{d}(\omega, \theta)$  serves the purpose of generalizing the idea of an LCMV beamformer beyond simply the case of a uniformly spaced array. Everything else is the same as before, except for the fact that the correlation matrix of the received signal is no longer Toeplitz for the case of a nonuniformly spaced array.

- 2.18 (a) Under hypothesis  $H_1$ , we have

$$\mathbf{u} = \mathbf{s} + \mathbf{v}$$

The correlation matrix of  $\mathbf{u}$  equals

$$\begin{aligned} \mathbf{R} &= E[\mathbf{u}\mathbf{u}^T] \\ &= \mathbf{s}\mathbf{s}^T + \mathbf{R}_N, \text{ where } \mathbf{R}_N = E[\mathbf{V}\mathbf{V}^T]. \end{aligned}$$

The tap-weight vector  $\mathbf{w}_k$  is chosen so that  $\mathbf{w}_k^T \mathbf{u}$  yields an optimum estimate of the  $k$ th element of  $\mathbf{s}$ . Thus, with  $s(k)$  treated as the desired response, the cross-correlation vector between  $\mathbf{u}$  and  $s(k)$  equals

$$\mathbf{p}_k = E[\mathbf{u}s(k)]$$

$$= \mathbf{s}s(k), \quad k = 1, 2, \dots, M$$

Hence, the Wiener-Hopf equation yields the optimum value of  $\mathbf{w}_k$  as

$$\begin{aligned} \mathbf{w}_{ko} &= \mathbf{R}^{-1} \mathbf{p}_k \\ &= (\mathbf{s}\mathbf{s}^T + \mathbf{R}_N)^{-1} \mathbf{s}s(k), \quad k = 1, 2, \dots, M \end{aligned} \tag{1}$$

To apply the matrix inversion lemma (introduced in Problem 2.13), we let

$$\mathbf{A} = \mathbf{R}$$

$$\mathbf{B}^{-1} = \mathbf{R}_N$$

$$\mathbf{C} = \mathbf{s}$$

$$\mathbf{D} = 1$$

Hence,

$$\mathbf{R}^{-1} = \mathbf{R}_N^{-1} - \frac{\mathbf{R}_N^{-1} \mathbf{s} \mathbf{s}^T \mathbf{R}_N^{-1}}{1 + \mathbf{s}^T \mathbf{R}_N^{-1} \mathbf{s}} \tag{2}$$

Substitute Eq. (2) into (1):

$$\begin{aligned} \mathbf{w}_{ko} &= \left( \mathbf{R}_N^{-1} - \frac{\mathbf{R}_N^{-1} \mathbf{s} \mathbf{s}^T \mathbf{R}_N^{-1}}{1 + \mathbf{s}^T \mathbf{R}_N^{-1} \mathbf{s}} \right) \mathbf{s}s(k) \\ &= \frac{\mathbf{R}_N^{-1} \mathbf{s} (1 + \mathbf{s}^T \mathbf{R}_N^{-1} \mathbf{s}) - \mathbf{R}_N^{-1} \mathbf{s} \mathbf{s}^T \mathbf{R}_N^{-1} \mathbf{s}}{1 + \mathbf{s}^T \mathbf{R}_N^{-1} \mathbf{s}} s(k) \\ &= \frac{s(k)}{1 + \mathbf{s}^T \mathbf{R}_N^{-1} \mathbf{s}} \mathbf{R}_N^{-1} \mathbf{s} \end{aligned}$$

(b) The output signal-to-noise ratio equals

$$\begin{aligned}
 \text{SNR} &= \frac{E[(\mathbf{w}^T \mathbf{s})^2]}{E[(\mathbf{w}^T \mathbf{v})^2]} \\
 &= \frac{\mathbf{w}^T \mathbf{s} \mathbf{s}^T \mathbf{w}}{\mathbf{w}^T E[\mathbf{v} \mathbf{v}^T] \mathbf{w}} \\
 &= \frac{\mathbf{w}^T \mathbf{s} \mathbf{s}^T \mathbf{w}}{\mathbf{w}^T \mathbf{R}_N \mathbf{w}}
 \end{aligned} \tag{3}$$

Since  $\mathbf{R}_N$  is positive definite, we may write

$$\mathbf{R}_N = \mathbf{R}_N^{1/2} \mathbf{R}_N^{1/2}$$

Define the vector

$$\mathbf{a} = \mathbf{R}_N^{1/2} \mathbf{w}$$

or equivalently,

$$\mathbf{w} = \mathbf{R}_N^{-1/2} \mathbf{a} \tag{4}$$

Accordingly, we may rewrite Eq. (3) as follows

$$\text{SNR} = \frac{\mathbf{a}^T \mathbf{R}_N^{1/2} \mathbf{s} \mathbf{s}^T \mathbf{R}_N^{1/2} \mathbf{a}}{\mathbf{a}^T \mathbf{a}} \tag{5}$$

where we have used the symmetric property of  $\mathbf{R}_N$ . Define the normalized vector

$$\bar{\mathbf{a}} = \frac{\mathbf{a}}{\|\mathbf{a}\|}$$

where  $\|\mathbf{a}\|$  is the norm of  $\mathbf{a}$ . Then we may rewrite Eq. (5) as

$$\text{SNR} = \bar{\mathbf{a}}^T \mathbf{R}_N^{1/2} \mathbf{s} \mathbf{s}^T \mathbf{R}_N^{1/2} \bar{\mathbf{a}}$$

$$= \left| \bar{\mathbf{a}}^T \mathbf{R}_N^{1/2} \mathbf{s} \right|^2$$

Thus the output signal-to-noise ratio SNR equals the squared magnitude of the inner product of two vectors  $\bar{\mathbf{a}}$  and  $\mathbf{R}_N^{1/2} \mathbf{s}$ . This inner product is maximized when  $\mathbf{a}$  equals  $\mathbf{R}_N^{-1/2} \mathbf{s}$ . That is,

$$\mathbf{a}_{SN} = \mathbf{R}_N^{-1/2} \mathbf{s} \quad (6)$$

Let  $\mathbf{w}_{SN}$  denote the value of the tap-weight vector that corresponds to Eq. (6). Hence, the use of Eq. (4) in (6) yields

$$\begin{aligned} \mathbf{w}_{SN} &= \mathbf{R}_N^{-1/2} (\mathbf{R}_N^{-1/2} \mathbf{s}) \\ &= \mathbf{R}_N^{-1} \mathbf{s} \end{aligned}$$

(c) Since the noise vector  $\mathbf{v}(n)$  is Gaussian, its joint probability density function equals

$$f_{\mathbf{V}}(\mathbf{v}) = \frac{1}{(2\pi)^{M/2} (\det \mathbf{R}_N)^{1/2}} \exp\left(-\frac{1}{2} \mathbf{v}^T \mathbf{R}_N^{-1} \mathbf{v}\right)$$

Under hypothesis  $H_0$  we have

$$\mathbf{u} = \mathbf{v}$$

and

$$f_{\mathbf{U}}(\mathbf{u}|H_0) = \frac{1}{(2\pi)^{M/2} (\det \mathbf{R}_N)^{1/2}} \exp\left(-\frac{1}{2} \mathbf{u}^T \mathbf{R}_N^{-1} \mathbf{u}\right)$$

Under hypothesis  $H_1$  we have

$$\mathbf{u} = \mathbf{s} + \mathbf{v}$$

and

$$f_{\mathbf{U}}(\mathbf{u}|H_1) = \frac{1}{(2\pi)^{M/2} (\det \mathbf{R}_N)^{1/2}} \exp\left[-\frac{1}{2} (\mathbf{u} - \mathbf{s})^T \mathbf{R}_N^{-1} (\mathbf{u} - \mathbf{s})\right]$$

Hence, the likelihood ratio equals

$$\begin{aligned}\Lambda &= \frac{f_{\mathbf{U}}(\mathbf{u}|H_1)}{f_{\mathbf{U}}(\mathbf{u}|0)} \\ &= \exp\left(-\frac{1}{2}\mathbf{s}^T \mathbf{R}_N^{-1} \mathbf{s} + \mathbf{s}^T \mathbf{R}_N^{-1} \mathbf{u}\right)\end{aligned}$$

The natural logarithm of the likelihood ratio equals

$$\ln \Lambda = -\frac{1}{2}\mathbf{s}^T \mathbf{R}_N^{-1} \mathbf{s} + \mathbf{s}^T \mathbf{R}_N^{-1} \mathbf{u}$$

The first term represents a constant. Hence, testing  $\ln \Lambda$  against a threshold is equivalent to the test

$$\mathbf{s}^T \mathbf{R}_N^{-1} \mathbf{u} \underset{H_0}{\overset{H_1}{\geq}} \lambda$$

where  $\lambda$  is some threshold. Equivalently, we may write

$$\mathbf{w}_{ML} = \mathbf{R}_N^{-1} \mathbf{s}$$

where  $\mathbf{w}_{ML}$  is the maximum likelihood weight vector.

The results of parts (a), (b) and (b) show that the three criteria discussed here yield the same optimum value for the weight vector, except for a scaling factor.

2.19 (a) Assuming the use of a noncausal Wiener filter, we write

$$\sum_{i=-\infty}^{\infty} w_{oi} r(i-k) = p(-k), \quad k = 0, \pm 1, \pm 2, \dots, \pm \infty \quad (1)$$

where the sum now extends from  $i=-\infty$  to  $i=\infty$ . Define the  $z$ -transforms:

$$S(z) = \sum_{k=-\infty}^{\infty} r(k) z^{-k}, \quad H_u(z) = \sum_{k=-\infty}^{\infty} w_{o,k} z^{-k}$$

$$P(z) = \sum_{k=-\infty}^{\infty} p(-k)z^{-k} = P(z^{-1})$$

Hence, applying the  $z$ -transform to Eq. (1):

$$H_u(z)S(z) = P(z^{-1}),$$

which gives

$$H_u(z) = \frac{P(1/z)}{S(z)} \quad (2)$$

$$(b) \ P(z) = \frac{0.36}{\left(1 - \frac{0.2}{z}\right)(1 - 0.2z)}$$

$$P(1/z) = \frac{0.36}{(1 - 0.2z)1 - (0.2/z)}$$

$$S(z) = 1.37 \frac{(1 - 0.146z^{-1})(1 - 0.146z)}{(1 - 0.2z^{-1})(1 - 0.2z)}$$

Thus, applying Eq. (2) yields

$$\begin{aligned} H_u(z) &= \frac{0.36}{1.37(1 - 0.146z^{-1})(1 - 0.146z)} \\ &= \frac{0.36z^{-1}}{1.37(1 - 0.146z^{-1})(z^{-1} - 0.146)} \\ &= \frac{0.2685}{1 - 0.146z^{-1}} + \frac{0.0392}{z^{-1} - 0.146} \end{aligned}$$

Clearly, this system is noncausal. Its impulse response is

$$h(n) = \text{inverse } z\text{-transform of } H_u(z)$$

$$= 0.2685 (0.146)^n u_{\text{step}}(n)$$

$$- \frac{0.0392}{0.146} \left( \frac{1}{0.146} \right)^n u_{\text{step}}(-n)$$

where  $u_{\text{step}}(n)$  is the unit-step function:

$$u_{\text{step}}(n) = \begin{cases} 1 & \text{for } n=0,1,2,\dots \\ 0 & \text{for } n=-1,-2,\dots \end{cases}$$

and  $u_{\text{step}}(-n)$  is its mirror image:

$$u_{\text{step}}(-n) = \begin{cases} 1 & \text{for } n=0,-1,-2,\dots \\ 0 & \text{for } n=1,2,\dots \end{cases}$$

Simplifying,

$$h_u(n) = 0.2685 \times (0.146)^n u_{\text{step}}(+n) - 0.2685 (6.849)^{-n} u_{\text{step}}(-n)$$

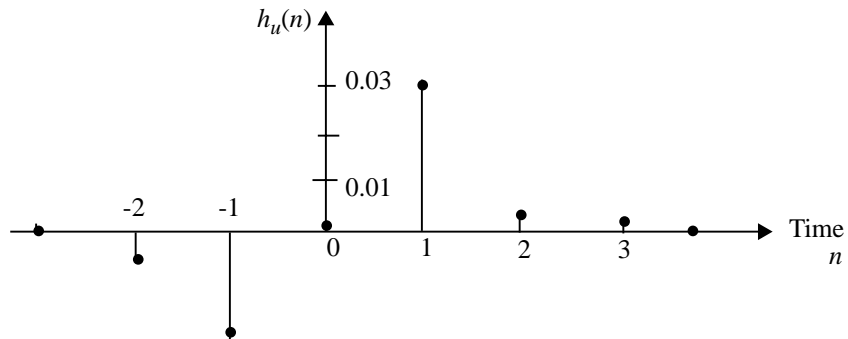
Evaluating  $h_u(n)$  for varying  $n$ :

$$h_u(0) = 0, \text{ and}$$

$$h_u(1) = 0.03, \quad h_u(2) = 0.005, \quad h_u(3) = 0.0008$$

$$h_u(-1) = -0.03, \quad h_u(-2) = -0.005, \quad h_u(-3) = -0.0008$$

These are plotted in the following figure:



- (c) A delay by 3 time units applied to the impulse response will make the system causal and therefore realizable.



## CHAPTER 3

- 3.1 (a) Let  $\mathbf{a}_M$  denote the tap-weight vector of the forward prediction-error filter. With a tap-input vector  $\mathbf{u}_{M+1}(n)$ , the forward prediction error at the filter output equals

$$f_M(n) = \mathbf{a}_M^H \mathbf{u}_{M+1}(n)$$

The mean-square value of  $f_M(n)$  equals

$$\begin{aligned} E[|f_M(n)|^2] &= E[\mathbf{a}_M^H \mathbf{u}_{M+1}(n) \mathbf{u}_{M+1}^H(n) \mathbf{a}_M] \\ &= \mathbf{a}_M^H E[\mathbf{u}_{M+1}(n) \mathbf{u}_{M+1}^H(n)] \mathbf{a}_M \\ &= \mathbf{a}_M^H \mathbf{R}_{M+1} \mathbf{a}_M \end{aligned}$$

where  $\mathbf{R}_{M+1} = E[\mathbf{u}_{M+1}(n) \mathbf{u}_{M+1}^H(n)]$  is the correlation matrix of the tap-input vector.

- (b) The leading element of the vector  $\mathbf{a}$  equals 1. Hence, the constrained cost function to be minimized is

$$J(\mathbf{a}_M) = \mathbf{a}_M^H \mathbf{R}_{M+1} \mathbf{a}_M + \lambda \mathbf{a}_M^H \mathbf{1} + \lambda^* \mathbf{1}^T \mathbf{a}_M$$

where  $\lambda$  is the Lagrange multiplier and  $\mathbf{1}$  is the first unit vector defined by  $\mathbf{1}^T = [1, 0, \dots, 0]$ . Differentiating  $J(\mathbf{a}_M)$  with respect to  $\mathbf{a}_M$ , and setting the result equal to zero yields

$$2\mathbf{R}_{M+1} \mathbf{a}_M + 2\lambda \mathbf{1} = \mathbf{0}$$

Solving for  $\mathbf{a}_M$ :

$$\mathbf{R}_{M+1} \mathbf{a}_M = -\lambda \mathbf{1} \tag{1}$$

However, we may partition  $\mathbf{R}_{M+1}$  as

$$\mathbf{R}_{M+1} = \begin{bmatrix} r(0) & \mathbf{r}^H \\ \mathbf{r} & \mathbf{R}_M \end{bmatrix}$$

Hence,

$$\begin{aligned} -\lambda &= [r(0), \mathbf{r}^H] \mathbf{a}_M \\ &= P_M \end{aligned}$$

where  $P_M$  is the minimum prediction-error power. Accordingly, we may rewrite Eq. (1) as

$$\mathbf{R}_{M+1} \mathbf{a}_M = P_M \mathbf{1} = \begin{bmatrix} P_M \\ \mathbf{0} \end{bmatrix}$$

- 3.2 (a) Let  $\mathbf{a}_M^{B*}$  denote the tap-weight vector of the backward prediction-error filter. With a tap-input vector  $\mathbf{u}_{M+1}(n)$  the backward prediction error equals

$$b_M(n) = \mathbf{a}_M^{BT} \mathbf{u}_{M+1}(n)$$

The mean-square value of  $b_M(n)$  equals

$$\begin{aligned} E[|b_M(n)|^2] &= E[\mathbf{a}_M^{BT} \mathbf{u}_{M+1}(n) \mathbf{u}_{M+1}^H(n) \mathbf{a}_M^{B*}] \\ &= \mathbf{a}_M^{BT} E[\mathbf{u}_{M+1}(n) \mathbf{u}_{M+1}^H(n)] \mathbf{a}_M^{B*} \\ &= \mathbf{a}_M^{BT} \mathbf{R}_{M+1} \mathbf{a}_M^{B*} \end{aligned}$$

- (b) The last element of  $\mathbf{a}_M^{B*}$  equals 1. Hence, the constrained objective function to be minimized is

$$J(\mathbf{a}_M) = \mathbf{a}_M^{BT} \mathbf{R}_{M+1} \mathbf{a}_M^{B*} + \lambda \mathbf{a}_M^{BT} \mathbf{1}^B + \lambda^* \mathbf{1}^{BT} \mathbf{a}_M^{B*}$$

where  $\lambda$  is the Lagrange multiplier, and

$$\mathbf{1}^{BT} = [0, 0, \dots, 1]$$

Differentiating  $J(\mathbf{a}_M)$  with respect to  $\mathbf{a}_M$ ,

$$2\mathbf{R}_{M+1}\mathbf{a}_M^{B*} + 2\lambda\mathbf{1}^B = \mathbf{0}$$

Solving for  $\mathbf{a}_M^{B*}$ , we get

$$\mathbf{R}_{M+1}\mathbf{a}_M^{B*} = -\lambda\mathbf{1}^B \tag{1}$$

However, we may express  $\mathbf{R}_{M+1}$  in the partitioned form:

$$\mathbf{R}_{M+1} = \begin{bmatrix} \mathbf{R}_M & \mathbf{r}^{B*} \\ \mathbf{r}^{BT} & r(0) \end{bmatrix}$$

Therefore,

$$\begin{aligned} -\lambda &= [r^{BT}, \mathbf{r}(0)]\mathbf{a}_M^{B*} \\ &= P_M \end{aligned}$$

where  $P_M$  is the minimum backward prediction-error power. We may thus rewrite Eq. (1) as

$$\mathbf{R}_{M+1}\mathbf{a}_M^{B*} = P_M\mathbf{1}^B = \begin{bmatrix} \mathbf{0} \\ P_M \end{bmatrix}$$

### 3.3 (a) Writing the Wiener-Hopf equation

$$\mathbf{R}\mathbf{g} = \mathbf{r}^{B*}$$

in expanded form, we have

$$\begin{bmatrix} r(0) & r(1) & \dots & r(M-1) \\ r(-1) & r(0) & \dots & r(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(-M+1) & r(-M+2) & \dots & r(0) \end{bmatrix} \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_M \end{bmatrix} = \begin{bmatrix} r(M) \\ r(M-1) \\ \vdots \\ r(1) \end{bmatrix}$$

Equivalently, we may write

$$\sum_{k=1}^M g_k r(k-i) = r(M+1-i), \quad i = 1, 2, \dots, M$$

Let  $k = M-l+1$ , or  $l = M-k+1$ . Then

$$\sum_{l=1}^M g_{M-l+1} r(M-l+1-i) = r(M+1-i), \quad i = 1, 2, \dots, M$$

Next, put  $M+1-i = j$ , or  $i = M+1-j$ . Then

$$\sum_{l=1}^M g_{M-l+1} r(j-l) = r(j), \quad j = 1, 2, \dots, M$$

Putting this relation into matrix form, we write

$$\begin{bmatrix} r(0) & r(-1) & \dots & r(-M+1) \\ r(1) & r(0) & \dots & r(-M+2) \\ \vdots & \vdots & \ddots & \vdots \\ r(M-1) & r(M-2) & \dots & r(0) \end{bmatrix} \begin{bmatrix} g_M \\ g_{M-1} \\ \vdots \\ g_1 \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(M) \end{bmatrix}$$

This, in turn, may be put in the compact form

$$\mathbf{R}^T \mathbf{g}^B = \mathbf{r}^*$$

(b) The product  $\mathbf{r}^{BT} \mathbf{g}$  equals

$$\begin{aligned}
\mathbf{r}^{BT} \mathbf{g} &= [r(-M), r(-M+1), \dots, r(-1)] \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_M \end{bmatrix} \\
&= \sum_{k=1}^M g_k r(k-1-M)
\end{aligned} \tag{1}$$

The product  $\mathbf{r}^T \mathbf{g}^B$  equals

$$\begin{aligned}
\mathbf{r}^T \mathbf{g}^B &= [r(-1), r(-2), \dots, r(-M)] \begin{bmatrix} g_M \\ g_{M-1} \\ \vdots \\ g_1 \end{bmatrix} \\
&= \sum_{k=1}^M g_{M+1-k} r(-k)
\end{aligned}$$

Put  $M+1-k = l$ , or  $k = M+1-l$ . Hence,

$$\mathbf{r}^T \mathbf{g}^B = \sum_{l=1}^M g_l r(l-1-M) \tag{2}$$

From Eqs. (1) and (2):

$$\mathbf{r}^{BT} \mathbf{g} = \mathbf{r}^T \mathbf{g}^B$$

3.4 Starting with the formula

$$r(m) = -\kappa_m^* P_{m-1} - \sum_{k=1}^{m-1} a_{m-1,k}^* r(m-k)$$

and solving for  $\kappa_m$ , we get

$$\kappa_m = \frac{r_m^*}{P_{m-1}} - \frac{1}{P_{m-1}} \sum_{k=1}^{m-1} a_{m-1,k} r^*(m-k) \quad (1)$$

We also note

$$a_{m,k} = a_{m-1,k} + \kappa_m a_{m-1,m-k}^*, \quad k = 0, 1, \dots, m \quad (2)$$

$$P_m = P_{m-1}(1 - |\kappa_m|^2) \quad (3)$$

(a) We are given

$$\begin{aligned} r(0) &= 1 \\ r(1) &= 0.8 \\ r(2) &= 0.6 \\ r(3) &= 0.4 \end{aligned}$$

We also note that

$$P_0 = r(0) = 1$$

Hence, the use of Eq. (1) for  $m = 1$  yields

$$\kappa_1 = -\frac{r^*(1)}{P_0} = -0.8$$

The use of Eq. (3) for  $m = 1$  yields

$$P_1 = P_0(1 - |\kappa_1|^2) = 1 - 0.64 = 0.36$$

We next reapply Eq. (1) for  $m = 2$ :

$$\kappa_2 = -\frac{r^*(2)}{P_1} - \frac{r^*(1)}{P_1} \kappa_1$$

where we have noted that

$$\kappa_1 = a_{1,1}$$

Hence,

$$\kappa_2 = -\frac{0.6}{0.36} + \frac{0.8 \times 0.8}{0.36} = \frac{0.04}{0.36} = \frac{1}{9} = 0.1111$$

The use of Eq. (3) for  $m = 2$  yields

$$\begin{aligned} P_2 &= P_1(1 - |\kappa_2|^2) \\ &= 0.36\left(1 - \frac{1}{81}\right) = \frac{4}{90} = 0.0444 \end{aligned}$$

Next, we reapply Eq. (1) for  $m = 3$ :

$$\kappa_3 = -\frac{r^*(3)}{P_2} - \frac{1}{P_2}(a_{2,1}r^*(2) + a_{2,2}r^*(1))$$

From Eq. (2) we have

$$a_{2,2} = \kappa_2 = \frac{1}{9} = 0.1111$$

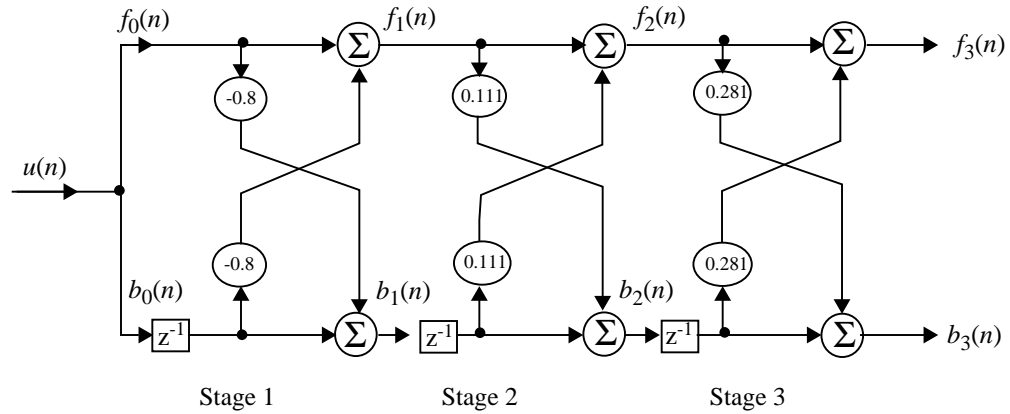
$$\begin{aligned} a_{2,1} &= a_{1,1} + \kappa_2 a_{1,1}^* = \kappa_1 + \kappa_2 \kappa_1^* \\ &= -\frac{4}{5} - \frac{4}{90} \times \frac{4}{5} = -\frac{188}{225} = -0.8356 \end{aligned}$$

Hence,

$$\begin{aligned} \kappa_3 &= -\frac{0.4}{0.0444} - \frac{1}{0.0444}(-0.8356 \times 0.6 + 0.111 \times 0.8) \\ &= -\frac{0.4}{0.0444} + \frac{0.41248}{0.0444} = \frac{0.01248}{0.0444} = 0.281 \end{aligned}$$

Note that all three reflection coefficients have a magnitude less than unity; hence, the lattice-predictor (prediction-error filter) representation of the process is minimum phase.

(b) This lattice-predictor representation is as shown in the following figure:



(c) From the calculations presented in part (a), we have

$$P_0 = 1$$

$$P_1 = 0.36$$

$$P_2 = 0.0444$$

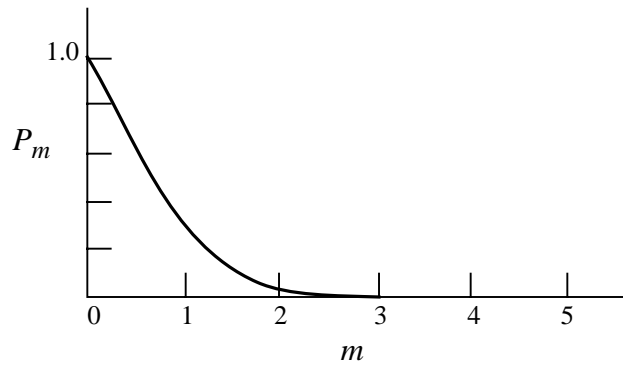
To complete the calculations required, we note that

$$P_3 = P_2(1 - |\kappa_3|^2)$$

$$= 0.0444 (1 - 0.281^2) = 0.0444 (1 - 0.079)$$

$$= 0.0444 \times 0.921 = 0.0409$$

From the power plot



we note that the average power  $P_m$  decreases exponentially with the prediction order  $m$ .



3.5 The estimation error  $e(n)$  is

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

$$\text{where } \mathbf{x}(n) = \mathbf{u}(n - \Delta)$$

$$= [u(n-\Delta), u(n-1-\Delta), \dots, u(n-M-\Delta)]^T$$

The mean-square value of the estimation error is

$$\begin{aligned} J &= E[|e(n)|^2] \\ &= E[(u(n) - \mathbf{w}^H \mathbf{x}(n))(u^*(n) - \mathbf{x}^H(n) \mathbf{w})] \\ &= E[|u(n)|^2] - \mathbf{w}^H E[\mathbf{x}(n) u^*(n)] - E[u(n) \mathbf{x}^H(n)] \mathbf{w} + \mathbf{w}^H E[\mathbf{x}(n) \mathbf{x}^H(n)] \mathbf{w} \\ &= P_0 - \mathbf{w}^H E[\mathbf{u}(n-\Delta) u^*(n)] - E[u(n) \mathbf{u}^H(n-\Delta)] \mathbf{w} + \mathbf{w}^H E[\mathbf{u}(n-\Delta) \mathbf{u}^H(n-\Delta)] \mathbf{w} \quad (1) \end{aligned}$$

We now note the following:

$$E[\mathbf{u}(n-\Delta) u^*(n)] = E \left\{ \begin{bmatrix} u(n-\Delta) \\ u(n-1-\Delta) \\ \vdots \\ u(n-M-\Delta) \end{bmatrix} \right\} u^*(n)$$

$$= \begin{bmatrix} r(-\Delta) \\ r(-\Delta-1) \\ \vdots \\ r(-\Delta-M) \end{bmatrix} = \mathbf{r}_\Delta$$

$$E[u(n) \mathbf{u}^H(n-\Delta)] = \begin{bmatrix} r(-\Delta) \\ r(-\Delta-1) \\ \vdots \\ r(-\Delta-M) \end{bmatrix}^H = \mathbf{r}_\Delta^H$$

$$E[\mathbf{u}(n-\Delta) \mathbf{u}^H(n-\Delta)] = \mathbf{R}$$

We may thus rewrite Eq. (1) as

$$J = P_0 - \mathbf{w}^H \mathbf{r}_\Delta - \mathbf{r}_\Delta^H \mathbf{w} + \mathbf{R}$$

The optimum value of the weight vector is

$$\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{r}_\Delta$$

where  $\mathbf{R}^{-1}$  is the inverse of the correlation matrix  $\mathbf{R}$ .

3.6 We are given the difference equation

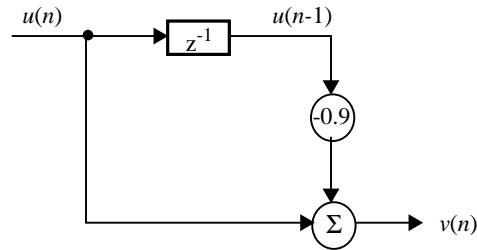
$$u(n) = 0.9u(n-1) + v(n)$$

(a) For a prediction-error filter of order two, we have

$$a_{2,1} = -0.9$$

$$a_{2,2} = 0$$

The prediction-error filter representation of the process is therefore



(b) The corresponding reflection coefficients of the lattice predictor are

$$\kappa_1 = a_{2,1} = -0.9$$

$$\kappa_2 = a_{2,2} = 0$$

We are given a first-order difference equation as the description of the AR process  $u(n)$ . It is therefore natural that we use a first-order predictor for the representation of this process.

3.7 (a) (i) The tap-weight vector of the prediction-error filter of order  $M$  is

$$\mathbf{a}_M = \begin{bmatrix} 1 \\ -\mathbf{w}_o \end{bmatrix} \quad (1)$$

where

$$\mathbf{w}_o = \mathbf{R}_M^{-1} \mathbf{r}_M \quad (2)$$

$$\mathbf{r}_M = E[\mathbf{u}_M(n-1)u^*(n)]$$

$$= \sigma_\alpha^2 \begin{bmatrix} e^{-j\omega} \\ e^{-j2\omega} \\ \vdots \\ e^{-jM\omega} \end{bmatrix} = \sigma_\alpha^2 e^{-j\omega} \mathbf{s}_M(\omega) \quad (3)$$

$$\mathbf{R}_M = \sigma_\alpha^2 \mathbf{s}_M(\omega) \mathbf{s}_M^H(\omega) + \sigma_v^2 \mathbf{I}_M$$

$$\mathbf{s}_M(\omega) = \begin{bmatrix} 1 \\ e^{-j\omega} \\ \vdots \\ e^{-j\omega(M-1)} \end{bmatrix} \quad (4)$$

From the matrix inversion lemma (see Chapter 9), we have:

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

then

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

For our application

$$\mathbf{B}^{-1} = \sigma_v^2 \mathbf{I}_M$$

$$\mathbf{D}^{-1} = \sigma_\alpha^2$$

$$\mathbf{C} = \mathbf{s}_M(\omega)$$

Hence,

$$\begin{aligned}
\mathbf{R}_M^{-1} &= \frac{1}{\sigma_v^2} \mathbf{I}_M - \frac{1/\sigma_v^4}{\frac{1}{\sigma_\alpha^2} + \frac{1}{\sigma_v^2} \mathbf{s}_M^H(\omega) \mathbf{s}_M(\omega)} \mathbf{s}_M(\omega) \mathbf{s}_M^H(\omega) \\
&= \frac{1}{\sigma_v^2} \mathbf{I}_M - \frac{1/\sigma_v^2}{(\sigma_v^2/\sigma_\alpha^2) + \mathbf{s}_M^H(\omega) \mathbf{s}_M(\omega)} \mathbf{s}_M(\omega) \mathbf{s}_M^H(\omega)
\end{aligned}$$

We also note that

$$\mathbf{s}_M^H(\omega) \mathbf{s}_M(\omega) = M$$

Hence,

$$\mathbf{R}_M^{-1} = \frac{1}{\sigma_v^2} \mathbf{I}_M - \frac{1/\sigma_v^2}{(\sigma_v^2/\sigma_\alpha^2) + M} \mathbf{s}_M(\omega) \mathbf{s}_M^H(\omega) \quad (5)$$

Substituting Eqs. (3) and (5) into (2) yields

$$\begin{aligned}
\mathbf{w}_o &= (\sigma_\alpha^2/\sigma_v^2) e^{-j\omega} \mathbf{s}_M(\omega) - \frac{(\sigma_\alpha^2/\sigma_v^2) e^{-j\omega}}{(\sigma_v^2/\sigma_\alpha^2) + M} \mathbf{s}_M(\omega) \mathbf{s}_M^H(\omega) \mathbf{s}_M(\omega) \\
&= (\sigma_\alpha^2/\sigma_v^2) e^{-j\omega} \left[ 1 - \frac{M}{(\sigma_v^2/\sigma_\alpha^2) + M} \right] \mathbf{s}_M(\omega) \\
&= \left( \frac{e^{-j\omega}}{(\sigma_v^2/\sigma_\alpha^2) + M} \right) \mathbf{s}_M(\omega) \quad (6)
\end{aligned}$$

Equations (1) and (6) define the tap-weight vector of the prediction-error filter. Moreover, the final value of the prediction-error power is

$$P_M = r(0) - \mathbf{r}_M^H \mathbf{w}_o$$

$$= r(0) - \left( \frac{\sigma_\alpha^2}{(\sigma_v^2/\sigma_\alpha^2) + M} \right) \mathbf{s}_M^H(\omega) \mathbf{s}_M(\omega) \quad (7)$$

Using (4) and the fact that

$$r(0) = \sigma_\alpha^2 + \sigma_v^2$$

we may rewrite Eq. (7) as

$$\begin{aligned} P_M &= \sigma_\alpha^2 + \sigma_v^2 - \frac{M\sigma_\alpha^2}{(\sigma_v^2/\sigma_\alpha^2) + M} \\ &= \frac{\sigma_v^2[1 + M + (\sigma_v^2/\sigma_\alpha^2)]}{(\sigma_v^2/\sigma_\alpha^2) + M} \end{aligned} \quad (8)$$

(a) (ii) The  $m$ th reflection coefficient is [from Eq. (3.56) of the text]

$$\kappa_m = -\frac{\Delta_{m-1}}{P_{m-1}}$$

where

$$\Delta_{m-1} = \mathbf{r}_m^{BT} \mathbf{a}_{m-1}$$

Hence,

$$\kappa_m = -\frac{\mathbf{r}_m^{BT} \mathbf{a}_{m-1}}{P_{m-1}} \quad (9)$$

From Eq. (3) we deduce that

$$\mathbf{r}_m^B = \sigma_\alpha^2 e^{-j\omega} \mathbf{s}_m^B(\omega)$$

$$= \sigma_{\alpha}^2 \begin{bmatrix} e^{-jm\omega} \\ e^{-j(m-1)\omega} \\ \vdots \\ e^{-j\omega} \end{bmatrix} \quad (10)$$

From Eq. (8):

$$P_{m-1} = \frac{\sigma_v^2 [m + (\sigma_v^2 / \sigma_{\alpha}^2)]}{(\sigma_v^2 / \sigma_{\alpha}^2) + m-1} \quad (11)$$

From Eqs. (1) and (6):

$$\mathbf{a}_{m-1} = \begin{bmatrix} 1 \\ - \left( \frac{e^{-j\omega}}{(\sigma_v^2 / \sigma_{\alpha}^2) + m-1} \right) \mathbf{s}_{m-1}(\omega) \end{bmatrix} \quad (12)$$

Hence, the combined use of Eqs. (9)-(12) yields

$$\begin{aligned} \kappa_m &= - \frac{\sigma_{\alpha}^2 [\sigma_v^2 + (m-1)\sigma_{\alpha}^2] e^{-jm\omega}}{\sigma_v^2 (\sigma_v^2 + m\sigma_{\alpha}^2 - \sigma_{\alpha}^2)} + \frac{\sigma_{\alpha}^4 e^{-jm\omega} (m-1)}{\sigma_v^2 (\sigma_v^2 + m\sigma_{\alpha}^2 - \sigma_{\alpha}^2)} \\ &= - \frac{\sigma_{\alpha}^2 e^{-jm\omega}}{\sigma_v^2 + m\sigma_{\alpha}^2 - \sigma_{\alpha}^2} \end{aligned}$$

(a) (iii) When we let the noise variance  $\sigma_v^2$  approach zero, the  $m$ th reflection coefficient

$\kappa_m$  approaches  $\frac{1}{m-1} e^{-jm\omega}$ , the magnitude of which, in turn, approaches zero as  $m$  approaches infinity.

(b) (i) The tap-weight vector of the prediction-error filter of order  $M$  is

$$\mathbf{a}_M = \begin{bmatrix} 1 \\ \alpha^* e^{-j\omega} \\ \mathbf{0}_{M-1} \end{bmatrix}$$

where  $\mathbf{0}_{M-1}$  is a null vector of dimension  $M-1$ .

(ii) The reflection coefficients of a lattice predictor of order  $M$  are

$$\kappa_1 = \alpha^* e^{-j\omega}$$

$$\kappa_m = 0 \quad \text{for } m = 2, \dots, M.$$

(c) We may consider  $u_1(n) = u_2(n)$  under the limiting conditions:

$$|\alpha| \rightarrow 1$$

and

$$\sigma_v^2 \rightarrow 0$$

where  $\sigma_v^2$  refers to the noise in the AR process.

$$3.8 \quad \mathbf{a}_m = \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix} + \kappa_m \begin{bmatrix} 0 \\ \mathbf{a}_{m-1}^{B*} \end{bmatrix}$$

In expanded form, we have

$$\begin{bmatrix} a_{m,0} \\ a_{m,1} \\ \vdots \\ a_{m,m-1} \\ a_{m,m} \end{bmatrix} = \begin{bmatrix} a_{m-1,0} \\ a_{m-1,1} \\ \vdots \\ a_{m-1,m-1} \\ 0 \end{bmatrix} + \kappa_m \begin{bmatrix} 0 \\ a_{m-1,m-1}^* \\ \vdots \\ a_{m-1,1}^* \\ a_{m-1,0}^* \end{bmatrix}$$

or equivalently,

$$a_{m,k} = a_{m-1,k} + \kappa_m a_{m-1,m-k}^* \quad k = 0, 1, \dots, M$$

Put  $m-k = l$  or  $k = m-l$ . Then,

$$a_{m,m-l} = a_{m-1,m-l} + \kappa_m a_{m-1,l}^*, \quad l = 0, 1, \dots, m$$

Complex conjugate both sides:

$$a_{m,m-l}^* = a_{m-1,m-l}^* + \kappa_m^* a_{m-1,l}, \quad l = 0, 1, \dots, m$$

Rewrite this relation in matrix form:

$$\begin{bmatrix} a_{m,m}^* \\ a_{m,m-1}^* \\ \vdots \\ a_{m-1}^* \\ a_{m-0}^* \end{bmatrix} = \begin{bmatrix} 0 \\ a_{m-1,m-1}^* \\ \vdots \\ a_{m-1,1}^* \\ a_{m-1,0}^* \end{bmatrix} + \kappa_m^* \begin{bmatrix} a_{m-1,0} \\ a_{m-1,1} \\ \vdots \\ a_{m-1,m-1} \\ 0 \end{bmatrix}$$

or equivalently,

$$\mathbf{a}_m^{B*} = \begin{bmatrix} 0 \\ \mathbf{a}_m^{B*} \\ \mathbf{a}_{m-1} \end{bmatrix} + \kappa_m^* \begin{bmatrix} \mathbf{a}_{m-1} \\ 0 \end{bmatrix}$$

Note that (for all  $m$ )

$$a_{m,k} = \begin{cases} 1, & k = 0 \\ 0, & k > m \end{cases}$$

3.9 We start with the formula

$$\Delta_{m-1} = \sum_{k=0}^{m-1} a_{m-1,k} r(k-m) \quad (1)$$

The autocorrelation function  $r(k-m)$  equals (by definition)



$$r(k-m) = E[u(n-m)u^*(n-k)] \quad (2)$$

Hence, substituting Eq. (2) in (1):

$$\begin{aligned} \Delta_{m-1} &= \sum_{k=0}^{m-1} a_{m-1,k} E[u(n-m)u^*(n-k)] \\ &= E \left[ u(n-m) \sum_{k=0}^{m-1} a_{m-1,k} u^*(n-k) \right] \end{aligned} \quad (3)$$

But, by definition,

$$f_{m-1}(n) = \sum_{k=0}^{m-1} a_{m-1,k}^* u(n-k)$$

Hence, we may rewrite Eq. (3) as

$$\Delta_{m-1} = E[u(n-m)f_{m-1}^*(n)] \quad (4)$$

Next we note that

$$u(n-m) = \hat{u}(n-m|U_{n-1}) + b_{m-1}(n-1) \quad (5)$$

where  $U_{n-1}$  is the space spanned by  $u(n-1), \dots, u(n-m+1)$ , and  $b_{m-1}$  is the backward prediction error produced by a predictor of order  $m-1$ . The estimate  $\hat{u}(n-m|U_{n-1})$  equals

$$\hat{u}(n-m|U_{n-1}) = \sum_{k=1}^{m-1} w_{b,k}^* u(n-k) \quad (6)$$

Accordingly, using Eqs. (5) and (6) in (3):

$$\Delta_{m-1} = E[b_{m-1}(n-1)f_{m-1}^*(n)] + E \left[ \sum_{k=1}^{m-1} w_{b,k}^* u(n-k)f_{m-1}^*(n) \right]$$

$$= E[b_{m-1}(n-1)f_{m-1}^*(n)] + \sum_{k=1}^{m-1} w_{b,k}^* E[u(n-k)f_{m-1}^*(n)] \quad (7)$$

But

$$E[f_{m-1}(n)u^*(n-k)] = 0, \quad 1 \leq k \leq m-1$$

Hence, Eq. (7) simplifies to

$$\Delta_{m-1} = E[b_{m-1}(n-1)f_{m-1}^*(n)]$$

3.10 The polynomial  $x(z)$  equals

$$\begin{aligned} x(z) &= a_{M,M}z^M + a_{M,M-1}z^{M-1} + \dots + a_{M,0} \\ &= \sum_{k=0}^M a_{M,k}z^k \\ &= z^M H_{b,M}(z) \end{aligned}$$

where  $H_{b,M}(z)$  is the transfer function of a backward prediction-error filter of order  $M$ . The reciprocal polynomial  $x'(z)$  equals

$$\begin{aligned} x'(z) &= a_{M,M}^* + a_{M,M-1}^*z + \dots + a_{M,0}^*z^M \\ &= z^M H_{f,M}(z) \end{aligned}$$

where  $H_{f,M}(z)$  is the transfer function of a forward prediction-error filter of order  $M$ .

Next, we note, by definition, that

$$\begin{aligned} T[x(z)] &= a_{M,0}^*x(z) - a_{M,M}x'(z) \\ &= \sum_{k=0}^M a_{M,k}z^k - a_{M,M} \sum_{k=0}^M a_{M,M-k}^*z^k \end{aligned}$$

$$= \sum_{k=0}^M (a_{M,k} - a_{M,M}^* a_{M,M-k}^*) z^k$$

But from the inverse recursion:

$$a_{M-1,k} = \frac{a_{M,k} - a_{M,M}^* a_{M,M-k}^*}{1 - |a_{M,M}|^2}, \quad k = 0, 1, \dots, M$$

Therefore,

$$\begin{aligned} T[x(z)] &= (1 - |a_{M,M}|^2) \sum_{k=0}^M a_{M-1,k} z^k \\ &= (1 - |a_{M,M}|^2) [z^{M-1} H_{b,M-1}(z)] \end{aligned}$$

where we have used the fact that  $a_{M-1,M}$  is zero. This shows that  $T[x(z)]$  is of order  $M-1$ , one less than the order of the original polynomial  $x(z)$ .

Similarly, we have

$$T^2[x(z)] = (1 - |a_{M,M}|^2)(1 - |a_{M-1,M-1}|^2) [z^{M-2} H_{b,M-2}(z)]$$

We also note that

$$T^1[x(0)] = 1 - |a_{M,M}|^2$$

$$T^2[x(0)] = (1 - |a_{M,M}|^2)(1 - |a_{M-1,M-1}|^2)$$

Thus

$$T^0[x(z)] = T^0[x(0)] z^M H_{b,M}(z)$$

$$T^1[x(z)] = T^1[x(0)] z^{M-1} H_{b,M-1}(z)$$

$$T^2[x(z)] = T^2[x(0)] z^{M-2} H_{b,M-2}(z)$$

where, in the first line, we have

$$T^0[x(z)] = x(z)$$

and

$$x(0) = 1$$

We may generalize these results by writing

$$T^i[x(z)] = T^i[x(0)]z^{M-i}H_{b, M-i}(z)$$

where

$$T^i[x(0)] = \prod_{j=0}^{i-1} (1 - |a_{M-j, M-j}|^2), \quad 1 \leq i \leq M$$

3.11 (a) The AR parameters equal

$$a_1 = -1$$

$$a_2 = \frac{1}{2}$$

Since  $v(n)$  has zero mean, the average power of  $u(n)$  equals

$$P_0 = r(0)$$

$$= \left( \frac{1 + a_2}{1 - a_2} \right) \frac{\sigma_v^2}{[(1 + a_2) - a_1^2]} = 1.2$$

(b) For prediction order  $M = 2$ , the prediction-error filter coefficients equal the AR parameters:

$$a_{2,1} = a_1 = -1$$

$$a_{2,2} = a_2 = \frac{1}{2}$$

The use of the inverse Levinson-Durbin recursion for real-valued data yields

$$a_{m-1,k} = \frac{a_{m,k} - a_{m,m}a_{m,m-k}}{1 - |a_{m,m}|^2}, \quad k = 0, \dots, M$$

For  $m = 2$ , we have

$$a_{1,k} = \frac{a_{2,k} - a_{2,2}a_{2,2-k}}{1 - |a_{2,2}|^2} \quad k = 0, 1$$

Hence,

$$a_{1,0} = 1$$

$$\begin{aligned} a_{1,1} &= \frac{a_{2,1} - a_{2,2}a_{2,1}}{1 - |a_{2,2}|^2} \\ &= -\frac{2}{3} \end{aligned}$$

The reflection coefficients are thus as follows

$$\kappa_1 = a_{1,1} = -\frac{2}{3}$$

$$\kappa_2 = a_{2,2} = \frac{1}{2}$$

(c) Use of the formula (for real-valued data)

$$P_m = P_{m-1}(1 - \kappa_m^2)$$

yields the following values for the average prediction-error powers:

$$P_1 = P_0(1 - \kappa_1^2) = \frac{2}{3}$$

$$P_2 = P_1(1 - \kappa_2^2) = \frac{1}{2}$$

3.12 For real data, we have

$$r(m) = -\kappa_m P_{m-1} - \sum_{k=1}^{m-1} a_{m-1,k} r(m-k)$$

For  $m = 1$ ,

$$r(1) = -\kappa_1 P_0$$

$$= 0.8$$

For  $m = 2$ ,

$$r(2) = -\kappa_2 P_1 - a_{1,1} r(1)$$

$$= 0.2$$

3.13 (a) The transfer function of the forward prediction-error filter equals

$$H_{f,M}(z) = (1 - z_i z^{-1}) c_i(z)$$

where

$$z_i = \rho_i e^{j\omega_i}$$

The power spectral density of the prediction error  $f_M(n)$  equals

$$S_f(\omega) = |H_{f,M}(e^{j\omega})|^2 S(\omega)$$

where  $S(\omega)$  is the power spectral density of the input process  $u(n)$ . Hence, the mean-square value of the prediction error  $f_M(n)$  equals

$$\begin{aligned} \epsilon &= \int_{-\pi}^{\pi} |H_{f,M}(e^{j\omega})|^2 S(\omega) d(\omega) \\ &= \int_{-\pi}^{\pi} |1 - \rho_i e^{j\omega_i} e^{-j\omega}|^2 |C_i(e^{j\omega})|^2 S(\omega) d(\omega) \end{aligned}$$

$$= \int_{-\pi}^{\pi} [1 - 2\rho_i \cos((\omega - \omega_i) + \rho_i^2)] |C_i(e^{j\omega})|^2 S(\omega) d(\omega)$$

(b) Differentiating  $\varepsilon$  with respect to  $\rho_i$ :

$$\frac{\partial \varepsilon}{\partial \rho_i} = 2 \int_{-\pi}^{\pi} [-\cos(\omega - \omega_i) + \rho_i] |C_i(e^{j\omega})|^2 S(\omega) d(\omega)$$

If  $z_i$  lies outside the unit circle,  $\rho_i > 1$ . We note that (regardless of  $\rho_i$ )

$$-1 \leq \cos(\omega - \omega_i) \leq 1, \quad -\pi \leq (\omega - \omega_i) \leq \pi$$

Hence,

$$-\cos(\omega - \omega_i) + \rho_i > 0, \quad \text{if } \rho_i > 1$$

Since  $|C_i(e^{j\omega})|^2$  and  $S(\omega)$  are both positive, it follows that

$$\frac{\partial \varepsilon}{\partial \rho_i} > 0, \quad \text{if } \rho_i > 1$$

If the prediction-error filter is to be optimum, its parameters (and therefore the  $\rho_i$ ) must be chosen in such a way that  $\frac{\partial \varepsilon}{\partial \rho_i} = 0$ .

Hence it is not possible for  $\rho_i > 1$  and yet satisfy the optimality condition  $\frac{\partial \varepsilon}{\partial \rho_i} = 0$ .

The conclusion to be drawn is that the transfer function of a forward prediction-error filter (that is optimum) cannot have any of its zeros outside the unit circle. In other words, a forward prediction-error filter is necessarily minimum phase.

3.14 An AR process  $u(n)$  of order  $M$  is described by the difference equation

$$u(n) = -a_1^* u(n-1) - \dots - a_M^* u(n-M) + v(n)$$

Equivalently, in the  $z$ -domain we have

$$U(z) = \frac{1}{1 + a_1^* z^{-1} + \dots + a_M^* z^{-M}} V(z)$$

When the process  $u(n)$  is applied to a forward prediction-error filter described by the transfer function

$$H_f(z) = 1 + a_1^* z^{-1} + \dots + a_M^* z^{-M}$$

The  $z$ -transform of the resulting output is

$$H_f(z)U(z) = V(z)$$

In other words, the output consists of a white noise sequence  $v(n)$ .

Suppose next the process  $u(n)$  is applied to a backward prediction-error filter described by the transfer function

$$H_b(z) = a_M + a_{M-1} z^{-1} + \dots + a_1 z^{-M+1} + z^{-M}$$

The  $z$ -transform of the resulting output is

$$H_b(z)U(z) = \frac{a_M + a_{M-1} z^{-1} + \dots + a_1 z^{-M+1} + z^{-M}}{1 + a_1^* z^{-1} + \dots + a_M^* z^{-M}}$$

This rational function is recognized as an all-pass (nonminimum phase) function with unit magnitude but non-zero phase. Equivalently, we may state that the corresponding output sequence is an anticausal realization of white noise.

3.15 Let

$$I = \frac{1}{2\pi j} \oint_C \frac{1}{z} \phi_m(z) \phi_k(z^{-1}) S(z) dz$$

On the unit circle,

$$z = e^{j\omega}$$

$$dz = je^{j\omega} d\omega$$

Hence,

$$I = \frac{1}{2\pi} \int_{-\pi}^{\pi} \phi_m(e^{j\omega}) \phi_k(e^{-j\omega}) S(\omega) d\omega \quad (1)$$



From the definition of  $\phi_m(z)$ , we have

$$\phi_m(e^{j\omega}) = \frac{1}{\sqrt{P_m}} \sum_{i=0}^m a_{m,i} e^{j\omega(m-i)}$$

Hence, we may rewrite Eq. (1) as

$$\begin{aligned} I &= \frac{1}{2\pi\sqrt{P_m P_k}} \int_{-\pi}^{\pi} \sum_{i=0}^m \sum_{l=0}^k a_{m,i} a_{k,l} e^{j\omega(m-i)} e^{-j\omega(k-l)} S(\omega) d\omega \\ &= \frac{1}{2\pi\sqrt{P_m P_k}} \sum_{i=0}^k \sum_{l=0}^k a_{m,i} a_{k,l} \int_{-\pi}^{\pi} S(\omega) e^{j\omega(m-k+l-i)} d\omega \end{aligned} \quad (2)$$

From the Einstein-Wiener-Khintchine relations:

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} S(\omega) e^{j\omega(m-k+l-i)} d\omega = r(m-k+l-i)$$

Accordingly, we may simplify Eq. (2) as

$$I = \frac{1}{\sqrt{P_m P_k}} \sum_{i=0}^m \sum_{l=0}^k a_{m,i} a_{k,l} r(m-k+l-i) \quad (3)$$

From the augmented Wiener-Hopf equation for linear prediction:

$$\sum_{l=0}^k a_{m,i} r(m-k+l-i) = \begin{cases} P_k, & \text{if } m=k \text{ and } l=0 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Substituting Eq. (4) into (3), we get

$$I = \begin{cases} 1, & \text{if } m=k \\ 0, & \text{otherwise} \end{cases}$$

where it is noted that  $a_{k,l} = 1$  for  $l = 0$ . Equivalently,

$$I = \delta_{mk}$$

as required.

3.16 From Eqs. (3.81) and (3.82):

$$\begin{aligned}
 H_{f,m}(z) &= \prod_{i=1}^m (1 - z_i z^{-1}) \\
 H_{b,m}(z) &= \prod_{i=1}^m (z^{-1} - z_i^*) \\
 &= \prod_{i=1}^m \left( \frac{z^{-1} - z_i^*}{1 - z_i z^{-1}} \right) (1 - z_i z^{-1}) \\
 &= H_{f,m}(z) \prod_{i=1}^m \left( \frac{z^{-1} - z_i^*}{1 - z_i z^{-1}} \right)
 \end{aligned}$$

The factor  $(z^{-1} - z_i^*)/(1 - z_i z^{-1})$  represents an all-pass structure with a magnitude response equal to unity for  $z = e^{j\omega}$  and all  $\omega$ . Hence,

$$|H_{b,m}(e^{j\omega})| = |H_{f,m}(e^{j\omega})|$$

Given an input  $u(n)$  of power spectral density  $S_u(\omega)$  applied to both  $H_{f,m}(e^{j\omega})$  and  $H_{b,m}(e^{j\omega})$ , we immediately see that

$$\begin{aligned}
 S_{f,m}(\omega) &= S_u(\omega) |H_{f,m}(e^{j\omega})|^2 \\
 &= S_u(\omega) |H_{b,m}(e^{j\omega})|^2
 \end{aligned}$$

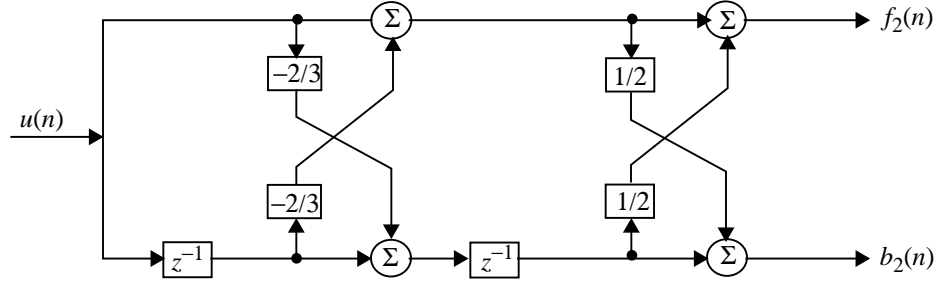
$$S_{b,m}(\omega) \quad \text{for all } m.$$

3.17 (a) The reflection coefficients of the two-stage lattice predictor equal

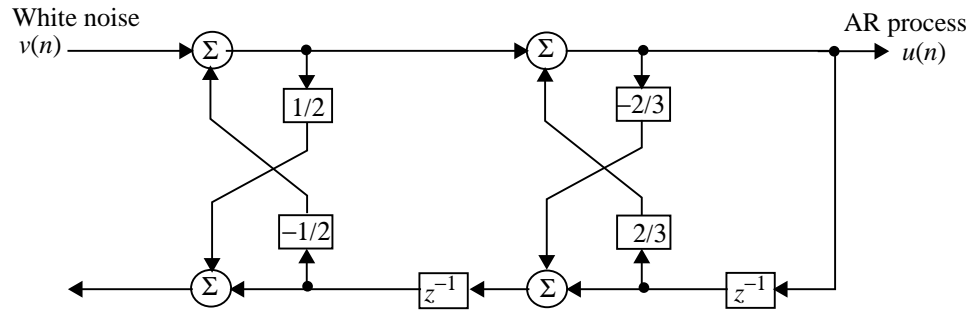
$$\kappa_1 = -\frac{2}{3}$$

$$\kappa_2 = \frac{1}{2}$$

Hence, the structure of this lattice predictor is as follows



(b) The inverse lattice filter for generating the second-order AR process from a white noise process is as follows (see Fig. 3.11):



From this latter structure, we see that

$$u(n) = \frac{2}{3}u(n-1) + \left(-\frac{2}{3}\right)\left(-\frac{1}{2}\right)u(n-1) + \left(-\frac{1}{2}\right)u(n-2) + v(n)$$

That is,

$$u(n) = u(n-1) - 0.5u(n-2) + v(n)$$

which is exactly the same as the difference equation specified in Problem 3.11.

- 3.18 (a) From Fig. 3.10,  $f_M(n)$  is obtained by passing  $u(n)$  through a minimum-phase prediction-error filter of order  $M$ , whereas  $b_M(n)$  is obtained by passing  $u(n)$  through a maximum-phase prediction-error filter of order  $M$ . Hence, going through the steps outlined in Problem 3.16, we readily see that in passing through the path from the input  $f_M(n)$  to the output  $b_M(n)$ , we will have gone through all-pass filter of order  $M$ .

- (b) In going from the input  $f_M(n)$  to the output  $u(n)$ , we will have passed through the inverse of a forward prediction-error filter of order  $M$ . Since such a filter is minimum phase with all its zeros confined to the interior of the unit circle, it follows that its inverse is an all-pole filter with all its poles confined to the interior of the unit circle; hence its physical realizability is assured.

3.19 (a) The  $(M+1)$ -by- $(M+1)$  lower triangular matrix  $\mathbf{L}$  is defined by

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ a_{1,1} & 1 & \dots & 0 \\ a_{2,2} & a_{2,1} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{M,M} & a_{M,M-1} & \dots & 1 \end{bmatrix}$$

Let

$$\mathbf{Y} = \mathbf{L}\mathbf{R}$$

where the  $(M+1)$ -by- $(M+1)$  correlation matrix  $\mathbf{R}$  is defined by

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & \dots & r(M) \\ r(-1) & r(0) & \dots & r(M-1) \\ \vdots & \vdots & \ddots & \vdots \\ r(-M) & r(-M+1) & \dots & r(0) \end{bmatrix}$$

Hence, the  $km$ th element of the matrix product  $\mathbf{L}\mathbf{R}$  equals

$$y_{km} = \sum_{l=0}^k a_{k,k-l} r(m-l), \quad (m, k) = 0, 1, \dots, M \quad (1)$$

For  $k = m$ , we thus have

$$y_{mm} = \sum_{l=0}^m a_{m,m-l} r(m-l), \quad m = 0, 1, \dots, M$$

However, from the augmented Wiener-Hopf equation for backward prediction we have

$$\sum_{l=0}^m a_{m, m-l}^* r(l-i) = \begin{cases} 0, & i = 0, 1, \dots, m-1 \\ P_m, & i = m \end{cases}$$

We therefore find that  $y_{mm}$  is real-valued and that

$$y_{mm} = P_m, \quad m = 0, 1, \dots, M.$$

(b) The  $m$ th column of matrix  $\mathbf{Y}$  equals

$$\begin{bmatrix} y_{0m} \\ y_{1m} \\ \vdots \\ y_{mm} \\ \vdots \\ y_{Mm} \end{bmatrix} = \begin{bmatrix} r(k) \\ \sum_{l=0}^1 a_{1, 1-l} r(m-l) \\ \vdots \\ \sum_{l=0}^m a_{m, m-l} r(m-l) \\ \vdots \\ \mathbf{0} \end{bmatrix}$$

The  $k$ th element of this column vector equals  $y_{km}$  that is defined by Eq. (1). The element  $y_{km}$  is recognized as the output produced by a backward prediction-error with tap weights  $a_{k, k}^*, a_{k, k-1}^*, \dots, a_{k, 0}^*$  and tap inputs  $r(m), r(m-1), \dots, r(m-k)$ , respectively. By summing the inner products of the respective tap weights and tap inputs, we get  $y_{km}$ . Hence, the  $m$ th column of matrix  $\mathbf{Y}$  is obtained by passing the autocorrelation sequence  $\{r(0), r(1), \dots, r(m)\}$  through the sequence of backward prediction-error filters whose transfer functions equal

$$H_{b, 0}(z), H_{b, 1}(z), \dots, H_{b, m}(z).$$

- (c) Apply the autocorrelation sequence  $\{r(0), r(1), \dots, r(m)\}$  to the input of a lattice predictor of order  $m$ . Denote the variables appearing at the various points on the lower part of the predictor as  $x_0, x_1, \dots, x_m$ , as shown here:

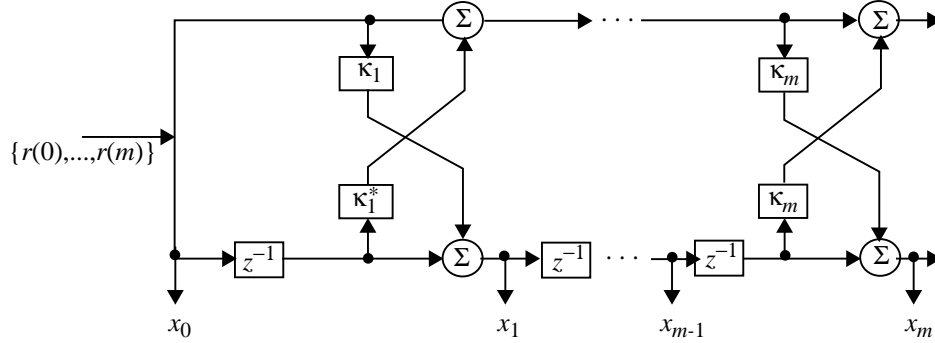


Figure 1

At time  $m$  we may express the resulting values of these outputs as follows:

$$x_0 = r(m)$$

$$x_1 = \text{output of backward prediction-error filter of order 1} \\ \text{and with tap inputs } r(m-1), r(m)$$

$$= \sum_{l=0}^1 a_{1, 1-l} r(m-l)$$

$\vdots$

$$x_m = \text{output of backward prediction-error filter of order } k \\ \text{and with tap inputs } r(0), r(1), \dots, r(m)$$

$$= \sum_{l=0}^m a_{m, m-l} r(m-l)$$

The various sets of prediction-error filter coefficients are related to the reflection coefficients  $\kappa_1, \kappa_2, \dots, \kappa_m$  in accordance with the Levinson-Durbin recursion. Hence, the variables appearing at the various points on the lower line of the lattice predictor in Fig. 1 at time  $m$  equal the elements of the  $m$ th column of matrix  $\mathbf{Y}$ .

- (d) The lower output of stage  $m$  at time  $m$  equals the  $mm$ th element  $y_{mm}$  of matrix  $\mathbf{Y}$ . This output equals  $P_m$ , as shown in part (a). The upper output of stage  $m$  in the lattice predictor is equivalent to the output of a forward prediction-error filter of order  $m$ .

Hence, this output at time  $m+1$ , in response to the autocorrelation sequence  $\{r(1), r(2), \dots, r(m+1)\}$  used as input, equals

$$\sum_{l=0}^m a_{m,l}^* r(m+1-l) = \Delta_m^*$$

Thus, we deduce that (except for a minus sign) the ratio of the upper output of stage  $m$  in the lattice predictor of Fig. 1 at time  $m+1$  to the lower output of this stage at time  $m$  equals the complex conjugate of the reflection coefficient  $\kappa_{m+1}$  for stage  $m+1$  in the lattice predictor.

(e) Using the autocorrelation sequence  $\{r(0), r(1), \dots, r(m)\}$  as input into the lattice predictor of Fig. 1, we may thus compute the corresponding sequence of reflection coefficients of the predictor as follows:

(i) At the input of the lattice predictor (i.e.,  $m=0$ ), the upper output equals  $r(1)$  at time 1, and the lower input equals  $r(0)$  at time 0. Hence, the ratio of these two outputs equals  $[r(1)/r(0)] = -\kappa_1^*$ .

(ii) The upper output of stage 1 at time 2 equals  $\Delta_1^*$ , and the lower output of this stage at time 1 equals  $P_1$ . Hence, the ratio of these two outputs equals  $(\Delta_1^*/P_1) = -\kappa_2^*$ .

(iii) The ratio of the upper output of stage 2 at time 3 to the lower output of this stage at time 2 equals  $(\Delta_2^*/P_2) = -\kappa_3^*$ , and so on for the higher stages of the lattice predictor.

3.20 A lattice filter exhibits some interesting correlation properties between the forward and backward prediction errors developed at the various stages of the filter. Basically, these properties are consequences of the principle of orthogonality, as described below:

**Property 1.** *The forward prediction error  $f_m(n)$  and the input signal  $u(n)$  are orthogonal:*

$$E[f_m(n)u^*(n-k)] = 0, \quad 1 \leq k \leq m \quad (1)$$

*Similarly, the backward prediction error  $b_m(n)$  and the input signal  $u(n)$  are orthogonal:*

$$E[b_m(n)u^*(n-k)] = 0, \quad 1 \leq k \leq m-1 \quad (2)$$

Note the difference between the ranges of the index  $k$  in Eqs. (1) and (2).

Equations (1) and (2) are both restatements of the principle of orthogonality. By definition, the forward prediction error  $f_m(n)$  equals the difference between  $u(n)$  and the prediction of  $u(n)$ , given the tap inputs  $u(n-1), u(n-2), \dots, u(n-m)$ . By the principle of orthogonality, the error  $f_m(n)$  is orthogonal to  $u(n-k)$ ,  $k = 1, 2, \dots, m$ . This proves Eq. (1).

The backward prediction error, by definition, equals the difference between  $u(n-m)$  and the prediction of  $u(n-m)$ , given the tap inputs  $u(n), u(n-1), \dots, u(n-m+1)$ . Here, again, by the principle of orthogonality, the error  $b_m(n)$  is orthogonal to  $u(n-k)$ ,  $k = 0, 1, \dots, m-1$ . This proves Eq. (2).

**Property 2.** *The cross-correlation of the forward prediction error  $f_m(n)$  and the input  $u(n)$  equals the cross-correlation of the backward prediction error  $b_m(n)$  and the time-shifted input  $u(n-m)$ , as shown by*

$$E[f_m(n)u^*(n)] = E[b_m(n)u^*(n-m)] = P_m \quad (3)$$

where  $P_m$  is the corresponding prediction-error power.

To prove the first part of this property, we note that  $u(n)$  equals the forward prediction error  $f_m(n)$  plus the prediction of  $u(n)$ , given the samples  $u(n-1), u(n-2), \dots, u(n-m)$ . Since this prediction is orthogonal to the error  $f_m(n)$  [which is a corollary to the principle of orthogonality], it follows that

$$\begin{aligned} E[f_m(n)u^*(n)] &= E[f_m(n)f_m^*(n)] \\ &= P_m \end{aligned}$$

To prove the second part of the property, we note that  $u(n-m)$  equals the backward prediction error  $b_m(n)$  plus the prediction of  $u(n-m)$ , given the samples  $u(n), u(n-1), \dots, u(n-m+1)$ . Since this prediction is orthogonal to the error  $b_m(n)$ , it follows that

$$\begin{aligned} E[b_m(n)u^*(n-m)] &= E[b_m(n)b_m^*(n)] \\ &= P_m \end{aligned}$$

This completes the proof of Eq. (3).

**Property 3.** *The backward prediction errors are orthogonal to each other, as shown by*

$$E[b_m(n)b_i^*(n)] = \begin{cases} P_m, & m = i \\ 0, & m \neq i \end{cases}$$



The forward prediction errors do not, however, exhibit the same orthogonality property as the backward prediction errors; rather, they are correlated, as shown by

$$E[f_m(n)f_i^*(n)] = P_m \quad m \geq i \quad (4)$$

Without loss of generality, we may assume that  $m \geq i$ . To prove Eq. (4) we express the backward prediction error  $b_i(n)$  in terms of the input  $u(n)$  as the convolution sum

$$b_i(n) = \sum_{k=0}^i a_{i,i-k} u(n-k) \quad (5)$$

where  $a_{i,i-k}^*$ ,  $k = 0, 1, \dots, i$ , are the coefficients of a backward prediction-error filter of order  $i$ . Hence, we may write

$$E[b_m(n)b_i^*(n)] = E\left[b_m(n) \sum_{k=0}^i a_{i,i-k}^* u^*(n-k)\right] \quad (6)$$

Now, by Property 1, we have

$$E[b_m(n)b_i^*(n-k)] = 0 \quad 0 \leq k \leq m-1$$

For the case when  $m \geq i$ , and with  $0 \leq k \leq i$ , we therefore find that all the expectation terms inside the summation on the right side of Eq. (6) are zero. Correspondingly,

$$E[b_m(n)b_i^*(n)] = 0 \quad m \neq i$$

When  $m = i$ , Eq. (6) reduces to

$$\begin{aligned} E[b_m(n)b_i^*(n)] &= E[b_m(n)b_m^*(n)] \\ &= P_m, \quad m = i \end{aligned}$$

This completes the proof of Eq. (3).

To prove Eq. (4), we express the forward prediction error  $f_i(n)$  in terms of the input  $u(n)$  as the convolution sum

$$f_i(n) = \sum_{k=0}^i a_{i,k}^* u(n-k) \quad (7)$$

where  $a_{i,k}$ ,  $k = 0, 1, \dots, i$ , are the coefficients of a forward prediction-error filter of order  $i$ . Hence,

$$\begin{aligned} E[f_m(n) f_i^*(n)] &= E \left[ f_m(n) \sum_{k=0}^i a_{i,k} u^*(n-k) \right] \\ &= E[f_m(n) u^*(n)] + \sum_{k=1}^i a_{i,k} E[f_m(n) u^*(n-k)] \end{aligned} \quad (8)$$

where we have used the fact that  $a_{i,0} = 1$ . However, by Property 1, we have

$$E[f_m(n) u^*(n-k)] = 0 \quad 1 \leq k \leq m$$

Also, by Property 2, we have

$$E[f_m(n) u^*(n)] = P_m$$

Therefore, Eq. (8) reduces to

$$E[f_m(n) f_i^*(n)] = P_m \quad m \geq i$$

This completes the proof of Eq. (4).

**Property 4.** *The time-shifted versions of the forward and backward prediction errors are orthogonal, as shown by, respectively,*

$$E[f_m(n+l) f_i^*(n)] = E[f_m(n-l) f_i^*(n)] = 0 \quad \begin{array}{l} 1 \leq l \leq m-i \\ m > i \end{array} \quad (9)$$

$$E[b_m(n) b_i^*(n-l)] = E[b_m(n+l) b_i^*(n)] = 0 \quad \begin{array}{l} 0 \leq l \leq m-i-1 \\ m > i \end{array} \quad (10)$$

where  $l$  is an integer lag.

To prove Eq. (9), we use Eq. (7) to write

$$\begin{aligned}
E[f_m(n)f_i^*(n-l)] &= E\left[f_m(n)\sum_{k=0}^i a_{i,k}u^*(n-l-k)\right] \\
&= \sum_{k=0}^i a_{i,k}E[f_m(n)u^*(n-l-k)]
\end{aligned} \tag{11}$$

By Property 1, we have

$$E[f_m(n)u^*(n-l-k)] = 0, \quad 1 \leq l+k \leq m \tag{12}$$

In the summation on the right side of Eq. (11) we have  $0 \leq k \leq i$ . For the orthogonality relationship of Eq. (12) to hold for all values of  $k$  inside this range, the lag  $l$  must correspondingly satisfy the condition  $1 \leq l \leq m-i$ . Thus, with the lag  $l$  bounded in this way, and with  $m > i$ , all the expectation terms inside the summation on the right side of Eq. (11) are zero. We therefore have

$$\begin{aligned}
E[f_m(n)f_i^*(n-l)] &= 0 & 1 \leq l \leq m-i \\
&& m > i
\end{aligned}$$

By definition, we have

$$E[f_m(n)f_i^*(n-l)] = E[f_m(n+l)f_i^*(n)]$$

Therefore, if the expectation  $E[f_m(n)f_i^*(n-l)]$  is zero, then so is the expectation  $E[f_m(n+l)f_i^*(n)]$ . This completes the proof of Eq. (9).

To prove Eq. (10), we use Eq. (5) to write

$$\begin{aligned}
E[b_m(n)b_i^*(n-l)] &= E\left[b_m\sum_{k=0}^i a_{i,i-k}^*u^*(n-l-k)\right] \\
&= \sum_{k=0}^i a_{i,i-k}^*E[b_m(n)u^*(n-l-k)]
\end{aligned} \tag{13}$$

By Property 1, we have

$$E[b_m(n)u^*(n-l-k)] = 0, \quad 0 \leq l+k \leq m-1 \quad (14)$$

For the orthogonality relationship to hold for  $0 \leq k \leq i$ , the lag  $l$  must satisfy the condition  $0 \leq l \leq m-i-1$ . Then, with  $m > i$ , we find that all the expectations inside the summation on the right side of Eq. (13) are zero. We therefore have

$$E[b_m(n)b_i^*(n-l)] = 0, \quad \begin{array}{l} 0 \leq l \leq m-i-1 \\ m > i \end{array}$$

By definition, we have

$$E[b_m(n)b_i^*(n-l)] = E[b_m(n+l)b_i^*(n)]$$

Hence, if the expectation  $E[b_m(n)b_i^*(n-l)]$  is zero, then so is the expectation  $E[b_m(n+l)b_i^*(n)]$ . This completes the proof of Eq. (10).

*Property 5. The time-shifted forward prediction errors  $f_m(n+m)$  and  $f_i(n+i)$  are orthogonal, as shown by*

$$E[f_m(n-m)f_i^*(n+i)] = \begin{cases} P_m, & m = i \\ 0, & m \neq i \end{cases} \quad (15)$$

*The corresponding time-shifted backward prediction errors  $b_m(n+m)$  and  $b_i(n+i)$ , on the other hand, are correlated as shown by*

$$E[b_m(n+m)b_i^*(n+i)] = P_m, \quad m \geq i \quad (16)$$

*Equations (15) and (16) are the duals of Eqs. (3) and (4), respectively*

Without loss of generality, we may assume  $m \geq i$ . To prove Eq. (15), we first recognize that

$$E[f_m(n+m)f_i^*(n+i)] = E[f_m(n)f_i^*(n-m+i)]$$

$$= E[f_m(n)f_i^*(n-l)] \quad (17)$$

where

$$l = m - i$$

Therefore, with  $m > i$ , we find from Property 4 that the expectation in Eq. (17) is zero. When, however,  $m = i$ , the lag  $l$  is zero, and this expectation equals  $P_m$ , the mean-square value of  $f_m(n)$ . This completes the proof of Eq. (15).

To prove Eq. (16) we recognize that

$$\begin{aligned} E[b_m(n+m)b_i^*(n+i)] &= E[b_m(n)b_i^*(n-m+i)] \\ &= E[b_m(n)b_i^*(n-l)] \end{aligned} \quad (18)$$

where

$$l = m - i$$

The value of  $l$  lies outside the range for which the expectation in Eq. (18) is zero [see Eq. (10)]. This means that  $b_m(n+m)$  and  $b_i(n+i)$  are correlated. To determine this correlation, we use Eq. (5) to write

$$\begin{aligned} E[b_m(n+m)b_i^*(n+i)] &= E\left[b_m(n+m) \sum_{k=0}^i a_{i,i-k}^* u^*(n+i-k)\right] \\ &= E[b_m(n)u^*(n)] \\ &\quad + \sum_{k=0}^{i-1} a_{i,i-k}^* E[b_m(n+m)u^*(n+i-k)] \end{aligned} \quad (19)$$

where we have used  $a_{i,0} = 1$ . By Property 1, we have

$$\begin{aligned} E[b_m(n+m)u^*(n+i-k)] &= E[b_m(n)u^*(n+i-k-m)] \\ &= 0, \quad 0 \leq k+m-i \leq m-1 \end{aligned} \quad (20)$$

The orthogonality relationship of Eq. (20) holds for  $i-m \leq k \leq i-1$ . The summation on the right side of Eq. (19) applies for  $0 \leq k \leq i-1$ . Hence, with  $m \geq i$ , all the expectation terms inside this summation are zero. Correspondingly, Eq. (19) reduces to

$$\begin{aligned} E[b_m(n+m)b_i^*(n+i)] &= E[b_m(n+m)u^*(n)] \\ &= E[b_m(n)u^*(n-m)] \\ &= P_m, \quad m \geq i \end{aligned}$$

where we have made use of Property 2. This completes the proof of Eq. (16).

**Property 6.** *The forward and backward prediction errors exhibit the following cross-correlation property:*

$$E[f_m(n)b_i^*(n)] = \begin{cases} \kappa_i^* P_m, & m \geq i \\ 0, & m < i \end{cases} \quad (21)$$

To prove this property, we use Eq. (5) to write

$$\begin{aligned} E[f_m(n)b_i^*(n)] &= E\left[f_m(n) \sum_{k=0}^i a_{i,i-k}^* u^*(n-k)\right] \\ &= a_{i,i}^* E[f_m(n)u^*(n)] + \sum_{k=1}^i a_{i,i-k}^* E[f_m(n)u^*(n-k)] \end{aligned} \quad (22)$$

By Property 1, we have

$$E[f_m(n)u^*(n-k)] = 0, \quad 1 \leq k \leq m$$

Assuming that  $m \geq i$ , we therefore find that all the expectation terms in the second term (summation) on the right side of Eq. (22) are zero. Hence,

$$\sum_{k=1}^i a_{i,i-k}^* E[f_m(n)u^*(n-k)] = 0 \quad \text{for } m \geq i$$

By Property 2, we have

$$E[f_m(n)u^*(n)] = P_m$$

Therefore, with  $a_{i,i} = \kappa_i$ , we find that Eq. (22) reduces to

$$E[f_m(n)b_i^*(n)] = \kappa_i^* P_m, \quad m \geq i$$

For the case when  $m < i$ , we adapt Eq. (7) to write

$$\begin{aligned} E[f_m(n)b_i^*(n)] &= E\left[\sum_{k=0}^m a_{m,k}^* u(n-k)b_i^*(n)\right] \\ &= \sum_{k=0}^m a_{m,k}^* u(n-k)b_i^*(n) \end{aligned} \quad (23)$$

By Property 1, we have

$$E[b_i(n)u^*(n-k)] = 0 \quad 0 \leq k \leq i-1$$

Therefore, with  $m < i$ , we find that all the expectation terms inside the summation on the right side of Eq. (23) are zero. Thus,

$$E[f_m(n)b_i^*(n)] = 0 \quad m < i$$

This completes the proof of Property 6.

3.21 The entropy of the input vector  $\mathbf{u}(n)$  is defined by the multiple integral

$$H_u = -\int_{-\infty}^{\infty} f_U(\mathbf{u}) \ln[f_U(\mathbf{u})] d\mathbf{u} \quad (1)$$

But,

$$\mathbf{u}(n) = \mathbf{L}^{-1} \mathbf{b}(n) \quad (2)$$

where  $\mathbf{L}$  is a lower triangular matrix defined by a sequence of backward prediction-error filters. We may therefore express the joint probability density function  $f_B(\mathbf{b})$  of the backward prediction-error vector  $\mathbf{b}(n)$  in terms of  $f_U(\mathbf{u})$  as

$$f_B(\mathbf{b}) = |\det(\mathbf{L}^{-1})| f_U(\mathbf{L}^{-1}\mathbf{b}) \quad (3)$$

where the term inside the absolute value signs is the Jacobian of the transformation described in Eq. (2). We also note

$$d(\mathbf{u}) = |\det(\mathbf{L}^{-1})| d\mathbf{b} \quad (4)$$

Next, we note that the entropy of the backward prediction-error vector  $\mathbf{b}(n)$  is

$$\begin{aligned} H_b &= -\int_{-\infty}^{\infty} f_B(\mathbf{b}) \ln[f_B(\mathbf{b})] d\mathbf{b} \\ &= -\int_{-\infty}^{\infty} |\det(\mathbf{L}^{-1})| f_U(\mathbf{L}^{-1}\mathbf{b}) \ln[|\det(\mathbf{L}^{-1})| f_U(\mathbf{L}^{-1}\mathbf{b})] d\mathbf{b} \\ &= -\int_{-\infty}^{\infty} f_U(u) \ln[|\det(\mathbf{L}^{-1})| f_U(\mathbf{u})] d\mathbf{u} \end{aligned}$$

where in the second line we have used Eq. (3) and in the third line we have used Eqs. (2) and (4). The transformation matrix  $\mathbf{L}$  has unity for all its diagonal elements; hence, the Jacobian of the transformation is unity. Accordingly, we may simplify the entropy  $H_b$  as

$$\begin{aligned} H_b &= -\int_{-\infty}^{\infty} f_u(\mathbf{u}) \ln[f_u(\mathbf{u})] d\mathbf{u} \\ &= H_u \end{aligned}$$

This shows that the backward prediction-error vector  $\mathbf{b}(n)$  has the same entropy, and therefore contains the same amount of information, as the input vector  $\mathbf{u}(n)$ .

3.22 (a) The index of performance to be optimized for stage  $m$  equals

$$J_m = aE[|f_m(n)|^2] + (1-a)E[|b_m(n)|^2] \quad (1)$$

where  $a$  is a constant that determines the mix of contributions from the forward and backward prediction errors to the index of performance. The forward prediction error equals



$$f_m(n) = f_{m-1}(n) + \kappa_m^* b_{m-1}(n-1) \quad (2)$$

The backward prediction error equals

$$b_m(n) = b_{m-1}(n) + \kappa_m f_{m-1}(n) \quad (3)$$

Substituting Eqs. (2) and (3) in (1) yields

$$\begin{aligned} J_m = & aE[|f_{m-1}(n)|^2] + (1-a)E[|b_{m-1}(n-1)|^2] \\ & + |\kappa_m|^2 \left\{ (1-a)E[|f_{m-1}(n)|^2] + aE[|b_{m-1}(n-1)|^2] \right\} \\ & + 2\kappa_m^* E[f_{m-1}^*(n)b_{m-1}(n-1)] + 2\kappa_m E[f_{m-1}(n)b_{m-1}^*(n-1)] \end{aligned}$$

Hence, differentiating  $J_m$  with respect to  $\kappa_m$  and setting the result equal to zero, we find that the optimum value of  $\kappa_m$  equals

$$\kappa_{m,o}(a) = -\frac{E[f_{m-1}^*(n)b_{m-1}(n-1)]}{(1-a)E[|f_{m-1}(n)|^2] + aE[|b_{m-1}(n-1)|^2]}$$

(b) The three special cases of interest are

$$(i) \quad a = 1$$

The index of performance equals

$$J_m = E[|f_m(n)|^2]$$

Correspondingly, the optimum value of the reflection coefficient equals

$$\kappa_{m,o}(1) = -\frac{E[f_{m-1}^*(n)b_{m-1}(n-1)]}{E[|b_{m-1}(n-1)|^2]}$$

We refer to this method of optimization as the *forward method*.

$$(ii) \quad a = 0$$

The index of performance equals

$$J_m = E[|b_{m-1}(n)|^2]$$

Correspondingly, the optimum value of the reflection coefficient equals

$$\kappa_{m,o}(0) = -\frac{E[f_{m-1}^*(n)b_{m-1}(n-1)]}{E[|f_{m-1}(n-1)|^2]}$$

We refer to this method of optimization as the *backward method*.

(iii)  $a = 1/2$

The index of performance equals

$$J_m = \frac{1}{2} \left\{ E[|f_m(n)|^2] + E[|b_m(n)|^2] \right\}$$

Correspondingly, the optimum reflection coefficient yields the Burg formula

$$\kappa_{m,o}\left(\frac{1}{2}\right) = -\frac{2E[f_{m-1}^*(n)b_{m-1}(n-1)]}{E[|f_{m-1}(n)|^2] + E[|b_{m-1}(n-1)|^2]}$$

We refer to this method of optimization as the *forward-backward method*.

3.23 (a) The harmonic mean of the optimum values of the reflection coefficient produced by the forward and backward methods equals

$$\begin{aligned} \frac{1}{2} \left[ \frac{1}{\kappa_{m,o}(1)} + \frac{1}{\kappa_{m,o}(0)} \right] &= -\frac{E[|f_{m-1}(n)|^2] + E[|b_{m-1}(n-1)|^2]}{2E[f_{m-1}^*(n)b_{m-1}(n-1)]} \\ &= \frac{1}{\kappa_{m,o}} \end{aligned}$$

where  $\kappa_{m,o}$  is the optimum value of the reflection coefficient produced by the Burg formula.

- (b) Define the correlation coefficient or the normalized value of the cross-correlation function between the forward prediction error  $f_{m-1}(n)$  and the delayed backward prediction error  $b_{m-1}(n-1)$  as

$$\rho = \frac{E[f_{m-1}^*(n)b_{m-1}(n-1)]}{\sqrt{E[|f_{m-1}(n)|^2] + E[|b_{m-1}(n-1)|^2]}}$$

Hence, we may redefine the optimum values  $\kappa_{m,o}(1)$  and  $\kappa_{m,o}(0)$  for the reflection coefficient that are produced by the forward and backward methods, respectively, as follows

$$\kappa_{m,o}(1) = -\alpha\rho$$

and

$$\kappa_{m,o}(0) = -\frac{\rho}{\alpha}$$

where the parameter  $\alpha$  is itself defined by

$$\alpha = \sqrt{\frac{E[|f_{m-1}(n)|^2]}{E[|b_{m-1}(n-1)|^2]}}$$

Accordingly, using the result of part (a), we may redefine the optimum value of the reflection coefficient produced by the Burg formula as

$$\frac{1}{\kappa_{m,o}} = -\frac{1}{2\rho}\left(\frac{1}{\alpha} + \alpha\right)$$

We note that the parameter  $\alpha$  is a nonnegative real-valued scalar that lies in the range  $0 \leq \alpha \leq \infty$ . Moreover, the factor

$$\frac{1}{\alpha} + \alpha$$

attains its minimum value of 2 at  $\alpha = 1$ . We may therefore write

$$\frac{1}{\alpha} + \alpha \geq 2$$

The correlation coefficient  $\rho$  is likewise a nonnegative scalar that lies in the range  $0 \leq \rho \leq 1$ . That is,  $1 \leq (1/\rho) \leq \infty$ . We conclude therefore that

$$\frac{1}{|\kappa_{m,o}|} \geq 1$$

or, equivalently,

$$|\kappa_{m,o}| \leq 1 \quad \text{for all } m.$$

- (c) The mean-square value of the forward prediction error  $f_m(n)$  equals  $E[|f_m(n)|^2]$ , where

$$f_m(n) = f_{m-1}(n) + \kappa_{m,o}^* b_{m-1}(n-1)$$

where the reflection coefficient is assigned the optimum value  $\kappa_{m,o}$  in accordance with the Burg formula. Hence,

$$\begin{aligned} E[|f_m(n)|^2] &= E[|f_{m-1}(n)|^2] + |\kappa_{m,o}|^2 E[|b_{m-1}(n-1)|^2] \\ &\quad + \kappa_{m,o}^* E[f_{m-1}^*(n) b_{m-1}(n-1)] \\ &\quad + \kappa_{m,o} E[f_{m-1}(n) b_{m-1}^*(n-1)] \end{aligned}$$

Using the Burg formula

$$\kappa_{m,o} = -\frac{2E[f_{m-1}^*(n) b_{m-1}(n-1)]}{E[|f_{m-1}(n)|^2] + E[|b_{m-1}(n-1)|^2]}$$

to eliminate the factor  $E[f_{m-1}^*(n) b_{m-1}(n-1)]$ , we may thus rewrite the mean-square value of  $f_m(n)$  as

$$E[|f_m(n)|^2] = E[|f_{m-1}(n)|^2] + |\kappa_{m,o}|^2 E[|b_{m-1}(n-1)|^2]$$

$$\begin{aligned}
& -|\kappa_{m,o}|^2 \left\{ E[|f_{m-1}(n)|^2] + E[|b_{m-1}(n)|^2] \right\} \\
& = (1 - |\kappa_{m,o}|^2) E[|f_{m-1}(n)|^2]
\end{aligned}$$

Similarly, we may express the mean-square value of the backward prediction error as follows

$$\begin{aligned}
E[|b_{m-1}(n)|^2] &= E[|b_{m-1}(n-1)|^2] + |\kappa_{m,o}|^2 E[|f_{m-1}(n)|^2] \\
&\quad + \kappa_{m,o} E[f_{m-1}(n) b_{m-1}^*(n-1)] \\
&\quad + \kappa_{m,o}^* E[f_{m-1}^*(n) b_{m-1}(n-1)] \\
&= E[|b_{m-1}(n)|^2] + |\kappa_{m,o}|^2 E[|f_{m-1}(n)|^2] \\
&\quad - |\kappa_{m,o}|^2 \left\{ E[|f_{m-1}(n)|^2] + E[|b_{m-1}(n-1)|^2] \right\} \\
&= (1 - |\kappa_{m,o}|^2) E[|b_{m-1}(n-1)|^2]
\end{aligned}$$

3.24 Assuming ergodicity, we may set up the following table for time series of length  $N$ .

Ensemble average	Time-average
$\frac{E[b_{m-1}(n-1)f_{m-1}^*(n)]}{E[ b_{m-1}(n-1) ^2]}$	$\frac{\sum_{n=1}^N b_{m-1}(n-1)f_{m-1}^*(n)}{N}$
$\frac{E[b_{m-1}(n-1)f_{m-1}^*(n)]}{E[ f_{m-1}(n) ^2]}$	$\frac{\sum_{n=1}^N  b_{m-1}(n-1) ^2}{N}$
$\frac{E[b_{m-1}(n-1)f_{m-1}^*(n)]}{E[ f_{m-1}(n) ^2]}$	$\frac{\sum_{n=1}^N  f_{m-1}(n) ^2}{N}$
$\frac{E[b_{m-1}(n-1)f_{m-1}^*(n)]}{\left(E[ f_{m-1}(n) ^2]E[ b_{m-1}(n-1) ^2]\right)^{1/2}}$	$\frac{\sum_{n=1}^N b_{m-1}(n-1)f_{m-1}^*(n)}{\left(\sum_{n=1}^N  f_{m-1}(n) ^2 \sum_{n=1}^N  b_{m-1}(n-1) ^2\right)^{1/2}}$

Applying the Cauchy-Schwartz inequality, we now see that only the last entry assures the condition  $|\kappa_m| \leq 1$  for all  $m$ :

$$\left| \sum_{n=1}^N b_{m-1}(n-1)f_{m-1}^*(n) \right|^2 \leq \left( \sum_{n=1}^N |f_{m-1}(n)|^2 \right) \left( \sum_{n=1}^N |b_{m-1}(n-1)|^2 \right)$$

3.25 (a)  $G(z) = \frac{z^2}{(1 - z/0.4)(1 + z/0.8)}$

$$= \frac{-0.4 \times 0.8}{\left(1 - \frac{0.4}{z}\right)\left(1 + \frac{0.8}{z}\right)}$$

We may therefore set

$$a_{21} = 0.4, \quad a_{22} = -0.32$$

$$\begin{aligned}
a_1 &= \frac{a_{21} - a_{22} \cdot a_{21}}{1 - a_{22}^2} \\
&= \frac{0.4 + 0.4 \times 0.32}{1 - 0.32^2} \\
&= 0.5882
\end{aligned}$$

The reflection coefficients are therefore

$$\kappa_2 = -0.32$$

$$\kappa_1 = 0.5882$$

(b)

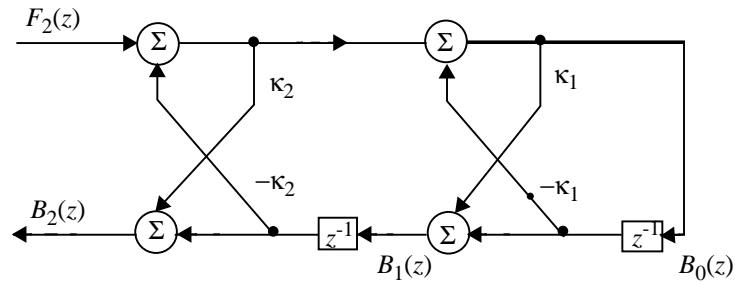


Fig. 1

$$B_2(z) = [-\kappa_2 B_1(z) z^{-1} + F_2(z)] \kappa_2 + B_1(z) z^{-1} \quad (1)$$

$$\frac{B_0(z)}{F_2(z)} = \frac{1}{1 + a_{21} z^{-1} + a_{22} z^{-2}}$$

$$B_1(z) = B_0(z) z^{-1} + B_0(z) \kappa_1$$

$$\frac{B_1(z)}{F_2(z)} = \frac{\kappa_1 + z^{-1}}{1 + a_{21} z^{-1} + a_{22} z^{-2}}$$

$$\frac{B_2(z)}{F_2(z)} = \frac{B_1(z)}{F_2(z)} [z^{-1} - \kappa_2 z^{-1}] + \kappa_2$$

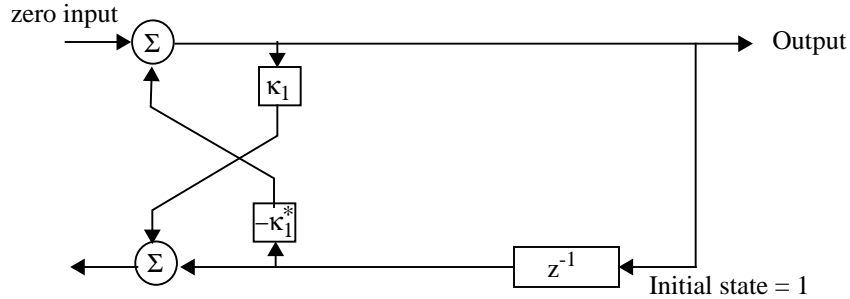
$$\begin{aligned}
&= \frac{[\kappa_1 + z^{-1}][z^{-1} - \kappa_2^* z^{-1}]}{1 + a_{21}z^{-1} + a_{22}z^{-2}} + \kappa_2 \\
&= \frac{\kappa_1 z^{-1} + \kappa_1 \kappa_2^* z^{-1} + \kappa_2 + z^{-2}}{1 + (\kappa_1 + \kappa_1 \kappa_2^*)z^{-1} + \kappa_2 z^{-2}} \\
&= \frac{z^{-2} + 0.4z^{-1} - 0.32}{1 + 0.4z^{-1} - 0.32z^{-2}}
\end{aligned}$$

Therefore, the all pass transfer function realized by the filter of Fig. 1 is

$$\frac{z^{-2} + 0.4z^{-1} - 0.32}{1 + 0.4z^{-1} - 0.32z^{-2}}$$

### 3.26 (a) Prediction order $M=1$

The inverse lattice filter consists of a single stage. By initializing the single stage of the filter to 1 and operating it with zero input, we have the following configuration to consider:

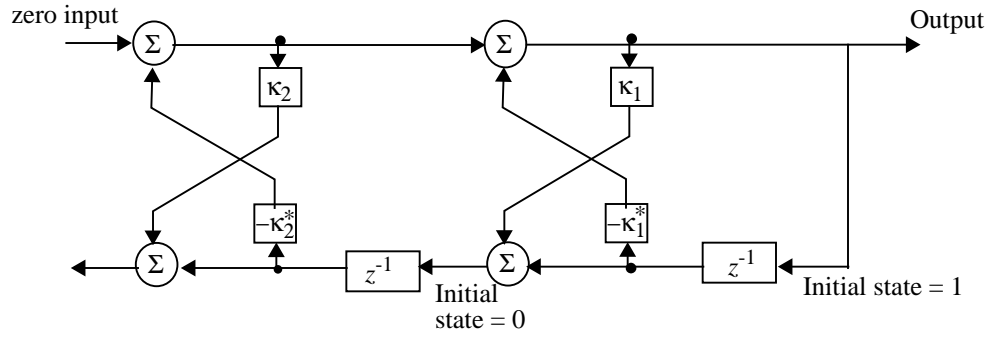


$$\begin{aligned}
\text{Output for lag of one} &= -\kappa_1^* \\
&= \frac{r(1)}{r(0)}
\end{aligned}$$

### (b) Prediction order $M=2$

In this case the inverse lattice filter consists of two stages. By initializing the two states of the filter to 1,0 and operating it with zero input, we have the following configuration to consider:



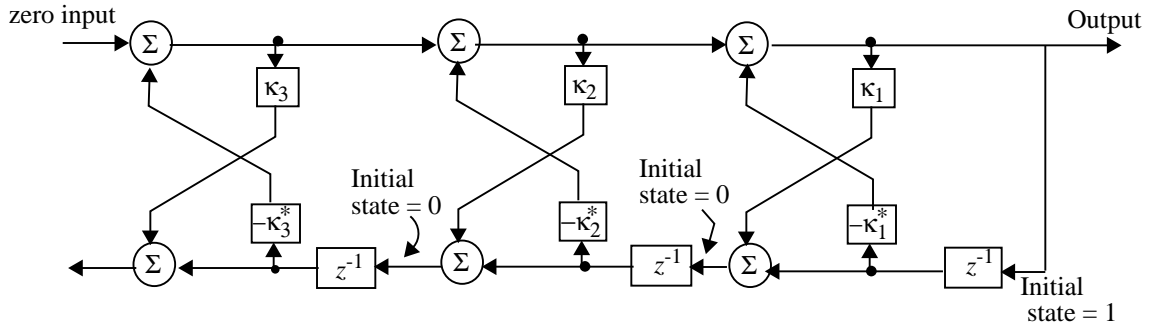


$$\text{Output for lag of one} = -\kappa_1^* = \frac{r(1)}{r(0)}$$

$$\begin{aligned} \text{Output for lag of two} &= (-\kappa_2^*) + (-\kappa_1^*)(\kappa_1)(-\kappa_2^*) + (-\kappa_1^*)(-\kappa_1^*) \\ &= \frac{r(2)}{r(0)} \end{aligned}$$

(c) Prediction order  $M=3$

In this case, the inverse lattice filter consists of three stages. By initializing the three states of the filter to 1,0,0 and operating it with zero input, we have the following configuration to consider



$$\begin{aligned} \text{Output for lag of one} &= -\kappa_1^* \\ &= \frac{r(1)}{r(0)} \end{aligned}$$

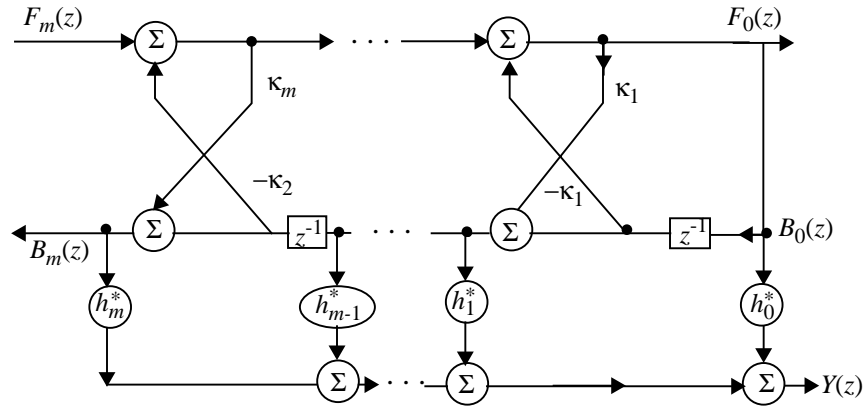
$$\begin{aligned} \text{Output for lag of two} &= (-\kappa_2^*) + (-\kappa_1^*)(\kappa_1)(-\kappa_2^*) + (-\kappa_1^*)(-\kappa_1^*) \\ &= \frac{r(2)}{r(0)} \end{aligned}$$

$$\begin{aligned}
\text{Output for lag of three} &= (-\kappa_3^*) + (-\kappa_2^*)(\kappa_2)(-\kappa_3^*) + (-\kappa_1^*)(\kappa_1)(-\kappa_3^*) \\
&+ (-\kappa_1^*)(\kappa_1)(-\kappa_2^*)(\kappa_2)(-\kappa_3^*) + (-\kappa_1^*)(-\kappa_2^*) \\
&+ (-\kappa_2^*)(\kappa_1^*) + (-\kappa_2^*)(\kappa_1)(-\kappa_2^*) \\
&+ (-\kappa_1^*)(-\kappa_1^*)(\kappa_1)(-\kappa_2^*) \\
&+ (-\kappa_1^*)(\kappa_1)(-\kappa_2^*)(\kappa_1)(-\kappa_2^*) \\
&+ (-\kappa_1^*)(-\kappa_1^*)(-\kappa_1^*) + (-\kappa_2^*)(-\kappa_1^*) \\
&+ (-\kappa_1^*)(-\kappa_1^*)(\kappa_1)(-\kappa_2^*) \\
&= \frac{r(3)}{r(0)}
\end{aligned}$$

Note that a lag of one corresponds to signal flow through a single delay element  $z^{-1}$ , a lag of two corresponds to signal flow through two delay elements, and a lag of three corresponds to signal flow through three delay elements, and so on.

3.27 Backward prediction errors are orthogonal to each other. On the other hand, forward prediction errors are correlated, which follows from Property 3 discussed in Problem 3.20.

3.28



$$\begin{aligned}
\frac{Y(z)}{F_m(z)} &= h_0 \frac{B_0(z)}{F_m(z)} + h_1 \frac{B_1(z)}{F_m(z)} + \dots + h_m \frac{B_m(z)}{F_m(z)} \\
&= \frac{h_0 + h_1[a_{11} + a_{10}z^{-1}] + \dots + h_m[a_{mm} + \dots a_{m0}z^{-m}]}{1 + a_{m1}z^{-1} + \dots + a_{mm}z^{-m}}, \quad a_{m0} = 1
\end{aligned}$$

$$\begin{aligned}
& \frac{(h_0 + h_1 a_{11} + \dots h_m a_{mm}) + (h_1 a_{10} + h_2 a_{21} + \dots h_m a_{mm-1})z^{-1} + \dots + h_m z^{-m}}{1 + a_{m1} z^{-1} + \dots + a_{mm} z^{-m}} \\
&= \frac{P_0 + P_1 z^{-1} + \dots P_m z^{-m}}{1 + d_1 z^{-1} + \dots d_m z^{-m}}, \quad d_i = a_{mi} \\
&= P_0 \frac{[1 + P_1/P_0 z^{-1} + \dots + P_m/P_0 z^{-m}]}{1 + d_1 z^{-1} + \dots d_m z^{-m}} \tag{1}
\end{aligned}$$

$$\begin{aligned}
&= G_0 \frac{\prod_{i=1}^m (1 - z/z_i)}{\prod_{i=1}^m (1 - z/p_i)} \\
&= \frac{G_0 p_1 \dots p_m}{z_1 \dots z_m} \frac{\prod_{i=1}^m (1 - z_i/z)}{\prod_{i=1}^m (1 - p_i/z)} \tag{2}
\end{aligned}$$

$$= \frac{G_0 p_1 \dots p_m}{z_1 \dots z_m} \left[ \frac{1 + (-1)^1 \sum_{i=1}^m z_i z^{-1} + (-1)^2 \sum_{i=1}^m \sum_{\substack{k=1 \\ i \neq k}}^m z_i z_k z^{-2} + \dots + z^{-m} (-1)^m z_1 \dots z_m}{1 + (-1)^1 \sum_{i=1}^m p_i z^{-1} + (-1)^2 \sum_{i=1}^m \sum_{\substack{k=1 \\ i \neq k}}^m p_i p_k z^{-2} + \dots + (-1)^m p_1 \dots p_m z^{-m}} \right]$$

(a) By equating the denominators of Eqs. (1) and (2):

$$1 + a_{m1} z^{-1} + \dots a_{mm} z^{-m} = \prod_{i=1}^m (1 - p_i/z)$$

$$= 1 + (-1)^1 \sum_{i=1}^m p_i z^{-1} + \sum_{i=1}^m \sum_{\substack{k=1 \\ i \neq k}}^m p_i p_k z^{-2} + \dots + (-1)^m p_1 \dots p_m z^{-m}$$

By equating the coefficients of  $z^{-i}$ ,  $a_{m,i}$  can be calculated for  $i = 1 \dots m$ . Then by using the equation

$$a_{m-1,k} = \frac{a_{mk} - a_{mm} a_{m,m-k}^*}{1 - |a_{mm}|^2} \quad k = 0, 1, \dots, m$$

$a_{i,i}$   $i = 1, \dots, m$  can be calculated.

Reflection coefficient  $k_i - a_{i,i}$   $i = 1, \dots, m$

(b) By equating the numerator of Eqs. (1) and (2):

$$\begin{aligned} h_0 + h_1 a_{11} + \dots h_m a_{mm} + \dots + h_m z^{-m} &= \frac{G_0 p_1 \dots p_m}{z_1 \dots z_m} \prod_{i=1}^m (1 - z_i/z) \\ &= \frac{G_0 p_1 \dots p_m}{z_1 \dots z_m} \left[ 1 + (+1)^1 \sum_{i=1}^m z_i z^{-1} + \dots (-1)^m z_1 \dots z_m z^{-m} \right] \end{aligned}$$

By equating the coefficients of  $z^{-i}$ , from  $z^{-m}$  to  $z^{-1}$ , the regression coefficients  $h_m, \dots, h_0$  can be calculated

$$(h_0 + h_1 a_{11} + \dots h_m a_{m1}) = \frac{G_0 p_1 \dots p_m}{z_1 \dots z_m} \quad (1)$$

Coefficient of  $z^{-m}$  is

$$\begin{aligned} h_m &= \frac{G_0 p_1 \dots p_m}{z_1 \dots z_m} (-1)^m z_1 \dots z_m \\ &= (-1)^m G_0 p_1 \dots p_m \end{aligned}$$

By equating the coefficients of  $z^{-m+1}$

$$h_{m-1}a_{m-1,0} + h_{m-1}a_{m-1} = \frac{G_0 p_1 \cdots p_m}{z_1 \cdots z_m} \{\text{coefficients of } z^{-m+1}\}$$

By back substitution,  $h_{m-1}$  can be calculated in the same way as  $h_{m-2}, \dots, h_1, h_0$  can be calculated.

$$\begin{aligned} 3.29 \quad (i) \quad G(z) &= 10 \frac{(1+z/0.1)(1+z/0.6)}{(1-z/0.4)(1+z/0.8)} \\ &= \frac{10 \times 0.4 \times 0.8}{0.1 \times 0.6} \frac{\left(\frac{0.1}{z} + 1\right)\left(\frac{0.6}{z} + 1\right)}{\left(\frac{0.4}{z} - 1\right)\left(1 + \frac{0.2}{z}\right)} \\ &= -\frac{32}{6} \times 10 \frac{\left(1 + \frac{0.1}{z}\right)\left(1 + \frac{0.6}{z}\right)}{\left(1 - \frac{0.4}{z}\right)\left(1 + \frac{0.8}{z}\right)} \\ &= -\frac{32}{6} \times 10 \frac{[1 + 0.7z^{-1} + 0.06z^{-2}]}{1 + 0.4z^{-1} - 0.32z^{-2}} \end{aligned}$$

Therefore,  $a_{21} = 0.4$ ,  $a_{22} = -0.32$

$$\begin{aligned} a_{11} &= \frac{a_{21} - a_{22}a_{21}}{1 - a_{22}^2} \\ &= \frac{0.4 + 0.32 \times 0.4}{1 - 0.32^2} \\ &= \frac{0.528}{0.8976} \\ &= 0.5882 \end{aligned}$$

Hence, the reflection coefficients are

$$\kappa_1 = 0.5882 \text{ and}$$

$$\kappa_2 = -0.32$$

The regression coefficients are calculated as follows:

$$\begin{aligned} h_2 &= \frac{-32}{6} \times 10 \times 0.06 \\ &= -3.2 \end{aligned}$$

$$h_1 a_{10} + h_2 a_{21} = 0.7 \times \frac{-32}{6} \times 10$$

$$\text{Therefore } h_1 = -34.13$$

$$h_0 + h_1 a_{11} + h_2 a_{22} = \frac{-32}{6} \times 10$$

$$\text{Therefore } h_0 = -34.28$$

$$\begin{aligned} \text{(ii) } G(z) &= \frac{10(1+z+z^2)}{(1-z/0.4)(1+z/0.8)} \\ &= -(10 \times 0.4 \times 0.8) \frac{[1+z^{-1}+z^{-2}]}{\left(1-\frac{0.4}{z}\right)(1+0.8/z)} \end{aligned}$$

From part (i)

$$a_{21} = 0.4, \quad a_{22} = -0.32$$

$$\begin{aligned} a_{11} &= 0.5882 & \kappa_1 &= 0.5882 \\ & & \kappa_2 &= -0.32 \end{aligned}$$

$$\begin{aligned} h_2 &= -10 \times 0.4 \times 0.8 \\ &= -3.2 \end{aligned}$$

$$h_1 a_{10} + h_2 a_{21} = -10 \times 0.4 \times 0.8$$

$$h_1 = -1.92$$

$$h_0 + h_1 a_{11} + h_2 a_{22} = -10 \times 0.4 \times 0.8$$

$$h_0 = -3.095$$

$$\begin{aligned} \text{(iii)} \quad G(z) &= \frac{10(1 + z/0.6)}{(1 - z/0.4)(1 + z/0.8)} \\ &= -\frac{10 \times 0.4 \times 0.8}{0.6 \times 1.5} \frac{\left(\frac{0.6}{z} + 1\right)(1.5/(z + 1))}{(1 - 0.4/z)(1 + 0.3/z)} \end{aligned}$$

From part (i),  $a_{21} = 0.4$ ,  $a_{22} = -0.32$

$$\begin{aligned} a_{11} &= 0.5882 & \kappa_2 &= -0.32 \\ & & \kappa_1 &= 0.5882 \end{aligned}$$

$$h_2 = \frac{-10 \times 32}{90} \times 0.6 \times 1.5 = -3.2$$

$$h_1 a_{10} + h_2 a_{21} = 2.1 \times \left( \frac{-10 \times 32}{80} \right)$$

Therefore  $h_1 = -6.19$

$$h_0 + h_1 a_{11} + h_2 a_{22} = \frac{-10 \times 32}{90}$$

$$h_0 = -0.938$$

$$\begin{aligned} \text{(iv)} \quad & \frac{10(1 + z + 0.5z^2)}{(1 - z/0.4)(1 + z/0.8)} \\ &= -10 \times 0.4 \times 0.8 \frac{z^{-2} + z^{-1} + 0.5}{\left(1 - \frac{0.4}{z}\right)\left(1 + \frac{0.8}{z}\right)} \end{aligned}$$

From part (i),  $a_{21} = 0.4$ ,  $a_{22} = -0.32$

$$a_{11} = 0.5882$$

$$\kappa_1 = 0.5882$$

$$\kappa_2 = -0.32$$

$$h_2 = 10 \times 0.4 \times 0.8 = -3.2$$

$$h_1 a_{10} + h_2 a_{21} = -10 \times 0.4 \times 0.8$$

$$h_1 = -1.92$$

$$h_0 + h_1 a_{11} + h_2 a_{22} = -0.5 \times 10 \times 0.4 \times 0.8$$

$$h_0 = -1.495$$

$$3.30 \quad (a) \quad A(e^{j\omega}) = \sum_{k=0}^M a_k e^{-j\omega_k}$$

$$\hat{H}(e^{j\omega}) = \sum_{k=0}^N \hat{h}(i) e^{-j\omega_i}$$

$$\hat{H}\left(e^{j\omega_m}\right)A\left(e^{j\omega_m}\right)=1 \tag{1}$$

$$\hat{p}(\omega) = |\hat{H}(\omega)|^2$$

$$\hat{H}^*\left(e^{j\omega_m}\right)\hat{H}\left(e^{j\omega_m}\right)A\left(e^{j\omega_m}\right)=\hat{H}^*\left(e^{j\omega_m}\right)$$

$$\hat{p}(\omega_m)A\left(e^{j\omega_m}\right)=\hat{H}^*\left(e^{j\omega_m}\right)$$

$$\sum_{k=0}^p \hat{p}\left(e^{j\omega_m}\right)a_k e^{-j\omega_m k} = \hat{H}^*\left(e^{j\omega_m}\right)$$

$$\sum_{k=1}^p \sum_{m=1}^N \hat{p}\left(e^{j\omega_m}\right)a_k e^{-j\omega_m k} e^{j\omega_m i} = \sum_{m=1}^N \hat{H}^*\left(e^{j\omega_m}\right) e^{j\omega_m i}$$



From definition

$$\hat{R}(i) = \frac{1}{N} \sum_{m=1}^N \hat{p}(\omega_m) \cos \omega_{mi}$$

and averaging over  $\omega_m \in \Omega$

$$\sum_{k=0}^p a_k \hat{R}(i-k) = \frac{1}{N} \sum_{m=1}^N \hat{H}(\omega_m) e^{-j\omega_{mi}}$$

$$= \hat{h}(-1) \text{ for all } i$$

$$= \frac{1}{N} \sum_{m=1}^N \left[ \frac{\exp(-j\omega_{mi})}{\sum_{k=0}^M a_k \exp(-j\omega_{mk})} \right]$$

$$(b) \ D(w) = \sum_{k=0}^M d_k \cos \omega_k$$

$$D_{IS} = \sum_{k=0}^{N-1} \left[ \frac{|U_k|^2}{S_k(a)} - \ln \frac{|U_k|^2}{S_k(a)} - 1 \right]$$

$$= \sum_{k=0}^{N-1} \left[ |U_k|^2 \sum_{k=0}^M \cos \omega_k - \ln \left[ |U_k|^2 \sum_{k=0}^M d_k \cos \omega_k - 1 \right] \right]$$

$$\frac{\partial D_{JS}}{\partial d_i} = 0$$

$$= |U_i|^2 \cos \omega_i - \frac{|U_i|^2 \cos \omega_i}{\sum_{k=0}^M d_k \cos \omega_k}$$

Therefore  $|U_i|^2 \sum_{k=0}^M d_k \cos \omega_k = 1 \quad \text{for } j = 0 \dots N-1$

$$|U_i|^2 = |\hat{H}(e^{j\omega})|^2$$

$$p(w_m)A(e^{j\omega_m}) = \hat{H}^*(e^{j\omega_m})$$

As in part (a), it can be shown:

$$\sum_{k=0}^p a_k R(i-k) = \hat{h}(-i)$$

where  $R(i) = \frac{1}{N} \sum_{m=1}^N p(w_m) \cos \omega_{mi}$

By comparing with part (a) the results are:

$$\sum_{k=0}^p a_k \hat{R}(j-k) = \sum_{k=0}^p a_k R(j-k) \quad \text{for } 0 \leq i \leq p$$

Therefore  $\hat{R}(i) = \hat{R}(i)$

(c)  $S_k(a) = \frac{1}{\left| \sum_{k=0}^m a_k e^{-j\omega_k} \right|^2}$

$$D_{IS}(a) = \sum_{k=0}^{N-1} \left[ \frac{|U_k|^2}{S_k(a)} - \ln \frac{|U_k|^2}{S_k(a)} - 1 \right]$$

$$\frac{\partial D_{IS}}{\partial a_i} = 0 \quad \text{for all } i = 0, \dots, m$$

$$D_{IS}(a) = \sum_{k=0}^{N-1} \left[ |U_k|^2 \left| \sum_{k=0}^m a_k e^{-j\omega_k} \right|^2 - \ln \left[ |U_k|^2 \left| \sum_{k=0}^m a_k e^{-j\omega_k} \right|^2 - 1 \right] \right]$$

$$\frac{\partial D_{IS}(a)}{\partial a_i} = 0$$

$$= |U_i|^2 |e^{-j\omega_i}|^2 - \frac{|U_i|^2 |e^{-j\omega_i}|^2}{|U_i|^2 \left| \sum_{k=0}^m a_k e^{-j\omega_k} \right|^2}$$

$$= 0$$

$$|U_i|^2 \left| \sum_{k=0}^m a_k e^{-j\omega_k} \right|^2 = 1 \quad \text{for all } i = 0, \dots, m$$

$$|U_i|^2 = \left| \hat{H}(e^{j\omega_m}) \right|^2$$

From part (a) it can be shown that

$$\sum_{k=0}^m a_k R(i-k) = \hat{h}(-i) \quad 0 \leq i \leq m$$

## CHAPTER 4

4.1 (a) For convergence of the steepest-descent algorithm:

$$0 < \mu < \frac{2}{\lambda_{\max}}$$

where  $\lambda_{\max}$  is the largest eigenvalue of the correlation matrix  $\mathbf{R}$ . We are given

$$\mathbf{R} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

The two eigenvalues of  $\mathbf{R}$  are

$$\lambda_1 = 0.5$$

$$\lambda_2 = 1.5$$

Hence  $\lambda_{\max} = 1.5$ . The step-size parameter  $\mu$  must therefore satisfy the condition

$$0 < \mu < \frac{2}{1.5} = 1.334$$

We may thus choose  $\mu = 1.0$ .

(b) From Eq. 4.9 of the text,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)]$$

With  $\mu = 1$  and

$$\mathbf{p} = \begin{bmatrix} 0.5 \\ 0.25 \end{bmatrix}$$

we therefore have

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \left\{ \begin{bmatrix} 0.5 \\ 0.25 \end{bmatrix} - \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \mathbf{w}(n) \right\}$$

$$\begin{aligned}
&= \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \right\} \mathbf{w}(n) - \begin{bmatrix} 0.5 \\ 0.25 \end{bmatrix} \\
&= \begin{bmatrix} 0 & -0.5 \\ -0.5 & 0 \end{bmatrix} \mathbf{w}(n) - \begin{bmatrix} 0.5 \\ 0.25 \end{bmatrix}
\end{aligned}$$

That is,

$$\begin{bmatrix} w_1(n+1) \\ w_2(n+1) \end{bmatrix} = \begin{bmatrix} 0 & -0.5 \\ -0.5 & 1 \end{bmatrix} \begin{bmatrix} w_1(n) \\ w_2(n) \end{bmatrix} - \begin{bmatrix} 0.5 \\ 0.25 \end{bmatrix}$$

Equivalently, we may write

$$w_1(n+1) = -0.5w_2(n) - 0.5$$

$$w_2(n+1) = -0.5w_1(n) - 0.25$$

- (c) To investigate the effect of varying the step-size parameter  $\mu$  on the trajectory, we find it convenient to work with  $v(n)$  rather than with  $w(n)$ . The  $k$ th natural mode of the steepest descent algorithm is described by

$$v_k(n+1) = (1 - \mu\lambda_k)v_k(n), \quad k = 1, 2$$

Specifically,

$$v_1(n+1) = (1 - 0.5\mu)v_1(n)$$

$$v_2(n+1) = (1 - 1.5\mu)v_2(n)$$

For the initial condition, we have

$$\begin{aligned}
\mathbf{v}(0) &= \begin{bmatrix} v_1(0) \\ v_2(0) \end{bmatrix} \\
&= \mathbf{Q}^H \mathbf{w}_o
\end{aligned}$$

From the solution to Problem 2.2:

$$\mathbf{w}_o = \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}$$

$$\mathbf{Q} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$$

Hence,

$$\begin{aligned} \mathbf{v}(0) &= \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0.5 \\ 0 \end{bmatrix} \\ &= \frac{1}{\sqrt{2}} \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \end{aligned}$$

That is,

$$v_1(0) = v_2(0) = \frac{0.5}{\sqrt{2}}$$

For  $n \geq 0$ , we have

$$v_k(n) = (1 - \mu\lambda_k)^n v_k(0), \quad k = 1, 2$$

Hence,

$$v_1(n) = (1 - 0.5\mu)^n v_1(0)$$

$$v_2(n) = (1 - 1.5\mu)^n v_2(0)$$

$$\boxed{\mu = 1}$$

$$v_1(n) = (0.5)^n v_1(0)$$

$$v_2(n) = (-0.5)^n v_2(0)$$

**Solution -** This represents an oscillatory trajectory.

$$\mu = 0.1$$

$$v_1(n) = (0.59)^n v_1(0)$$

$$v_2(n) = (0.85)^n v_2(0)$$

This second solution represents a damped trajectory.

The transition from a damped to an oscillatory trajectory occurs at

$$\mu = \frac{1}{1.5} = 0.667$$

Specifically, for  $0 < \mu < 0.667$ , the trajectory is damped. On the other hand, for  $0.667 < \mu < 1.334$  the trajectory is oscillatory.

4.2 We are given

$$J(w) = J_{\min} + r(0)(w - w_o)^2$$

(a) The correlation matrix  $\mathbf{R} = r(0)$ .

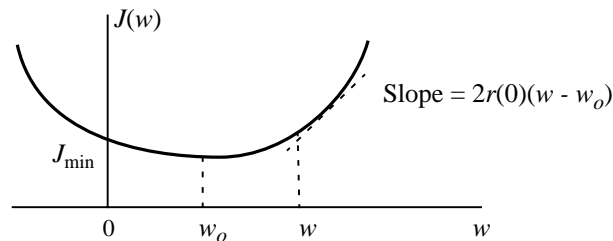
Hence,  $\lambda_{\max} = r(0)$

$$\text{Correspondingly, } \mu_{\max} = \frac{2}{\lambda_{\max}} = \frac{2}{r(0)}$$

(b) Time constant of the filter is

$$\tau_1 \approx \frac{1}{\mu \lambda_1} = \frac{1}{\mu r(0)}$$

(c)



$$\frac{\partial J}{\partial w} = 2r(0)(w - w_o)$$

4.3 (a) There is a single mode with eigenvalue  $\lambda_1 = r(0)$ , and  $q_1 = 1$ , Hence,

$$J(n) = J_{\min} + \lambda_1 |v_1(n)|^2$$

$$\text{where } v_1(n) = q_1(w_o - w(n)) = (w_o - w(n))$$

$$(b) \frac{\partial J(n)}{\partial \lambda_1} = |v_1(n)|^2 = (w_o - w(n))^2$$

4.4 The estimation error  $e(n)$  equals

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

where  $d(n)$  is the desired response,  $\mathbf{w}(n)$  is the tap-weight vector, and  $\mathbf{u}(n)$  is the tap-input vector. Hence, the gradient of the instantaneous squared error equals

$$\begin{aligned} \hat{\nabla} J(n) &= \frac{\partial}{\partial \mathbf{w}} [|e(n)|^2] = \frac{\partial}{\partial \mathbf{w}} [e(n)e^*(n)] \\ &= e(n) \frac{\partial e^*(n)}{\partial \mathbf{w}} + e^*(n) \frac{\partial e(n)}{\partial \mathbf{w}} \\ &= -2e^*(n)\mathbf{u}(n) = 2\mathbf{u}(n)d^*(n) + 2\mathbf{u}(n)\mathbf{u}^H(n)\mathbf{w}(n) \end{aligned}$$

4.5 Consider the approximation to the inverse of the correlation matrix:

$$\mathbf{R}^{-1}(n+1) \approx \mu \sum_{k=0}^n (\mathbf{I} - \mu \mathbf{R})^k$$

where  $\mu$  is a positive constant bounded in value as

$$0 < \mu < \frac{2}{\lambda_{\max}}$$

where  $\lambda_{\max}$  is the largest eigenvalue of  $\mathbf{R}$ . Note that according to this approximation, we have  $\mathbf{R}^{-1}(1) = \mu \mathbf{I}$ . Correspondingly, we may approximate the optimum Wiener solution as



$$\begin{aligned}
\mathbf{w}(n+1) &= \mathbf{R}^{-1}(n+1)\mathbf{p} \\
&\approx \mu \sum_{k=0}^n (\mathbf{I} - \mu\mathbf{R})^k \mathbf{p} \\
&= \mu\mathbf{p} + \mu \sum_{k=1}^n (\mathbf{I} - \mu\mathbf{R})^k \mathbf{p}
\end{aligned}$$

In the second term, put  $k = i+1$  or  $i = k-1$ :

$$\begin{aligned}
\mathbf{w}(n+1) &= \mu\mathbf{p} + \mu(\mathbf{I} - \mu\mathbf{R}) \sum_{k=1}^n (\mathbf{I} - \mu\mathbf{R})^i \mathbf{p} \\
&= \mu\mathbf{p} + (\mathbf{I} - \mu\mathbf{R})\mathbf{w}(n)
\end{aligned} \tag{1}$$

where, in the second line, we have used the fact that

$$\mu \sum_{i=0}^n (\mathbf{I} - \mu\mathbf{R})^i \mathbf{p} = \mathbf{w}(n)$$

Hence, rearranging Eq. (1):

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)]$$

which is the standard formula for the steepest descent algorithm.

$$4.6 \quad J(\mathbf{w}(n+1)) = J(\mathbf{w}(n)) - \frac{1}{2}\mu\|\mathbf{g}(n)\|^2$$

For stability of the steepest-descent algorithm, we therefore require

$$J(\mathbf{w}(n+1)) < J(\mathbf{w}(n))$$

To satisfy this requirement, the step-size parameter  $\mu$  should be positive, since  $\mu\|\mathbf{g}(n)\|^2 > 0$ .

Hence, the steepest-descent algorithm becomes unstable when the step-size parameter is negative.

4.7 Let

$$\begin{aligned}\delta \mathbf{w}(n+1) &= \mathbf{w}(n+1) - \mathbf{w}(n) \\ &= \mu[\mathbf{p} - \mathbf{R}\mathbf{w}(n)]\end{aligned}$$

By definition,

$$\mathbf{p} = E[\mathbf{u}(n)d^*(n)]$$

$$\mathbf{R} = E[\mathbf{u}(n)\mathbf{u}^H(n)]$$

Hence,

$$\begin{aligned}\delta \mathbf{w}(n+1) &= \{\mu E[\mathbf{u}(n)d^*(n)] - E[\mathbf{u}(n)\mathbf{u}^H(n)]\mathbf{w}(n)\} \\ &= \mu E[\mathbf{u}(n)\{d^*(n) - \mathbf{u}^H(n)\mathbf{w}(n)\}]\end{aligned}$$

But, the estimation error is

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

Therefore,

$$\delta \mathbf{w}(n+1) = \mu E[\mathbf{u}(n)e^*(n)]$$

At the minimum point of the error-performance surface, the correction  $\delta \mathbf{w}(n+1)$  is zero. Hence, at this point, we have

$$E[\mathbf{u}(n)e^*(n)] = 0$$

which is a restatement of the principle of orthogonality.

4.8 We are given

$$J_{\text{approx}}(n) = [J(0) - J(\infty)]e^{-n/\tau} + J(\infty)$$

For  $n = 1$ , we have

$$\begin{aligned}
J_{\text{approx}}(1) &= J(1) \\
&= [J(0) - J(\infty)]e^{-1/\tau} + J(\infty)
\end{aligned}$$

Solving for  $e^{1/\tau}$ , we get

$$e^{1/\tau} = \frac{J(0) - J(\infty)}{J(1) - J(\infty)}$$

Taking natural logarithms:

$$\frac{1}{\tau} = \ln \left[ \frac{J(0) - J(\infty)}{J(1) - J(\infty)} \right]$$

or

$$\tau = \ln \left[ \frac{J(1) - J(\infty)}{J(0) - J(\infty)} \right]$$

- 4.9 (a) The cross-correlation between the “desired response”  $u(n)$  and the tap input  $u(n-1)$  is

$$p = E[u(n)u(n-1)] = r(1)$$

The mean-square value of the tap input  $u(n-1)$  is

$$E[u^2(n-1)] = r(0)$$

Hence,

$$w(n+1) = w(n) + \mu(r(1) - r(0)w(n))$$

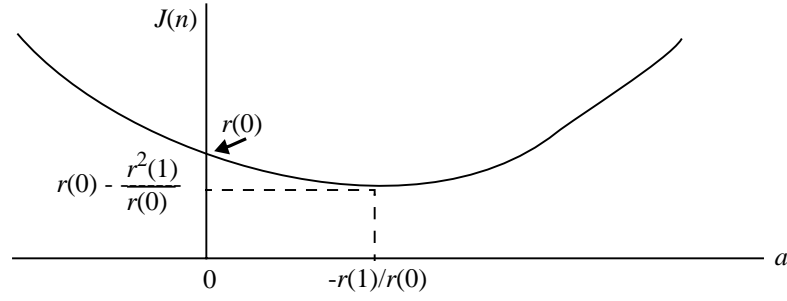
For the recursive computation of the parameter  $a$ , we note that  $a(n) = -w(n)$ ; hence

$$a(n+1) = a(n) - \mu[r(1) + r(0)a(n)]$$

- (b) The error-performance surface is defined by

$$\begin{aligned}
J(n) &= r(0) - 2r(1)w(n) + r(0)w^2(n) \\
&= r(0) + 2r(1)a(n) + r(0)a^2(n)
\end{aligned}$$

The corresponding plot of the error performance surface is therefore



(c) The condition on the step-size parameter is

$$0 < \mu < \frac{2}{r(0)}$$

4.10 The second-order AR process  $u(n)$  is described by the difference equation

$$u(n) = -0.5u(n-1) + u(n-2) + v(n) \quad (1)$$

Hence,

$$w_1 = -0.5, \quad w_2 = 1$$

and the AR parameters equal

$$a_1 = 0.5, \quad a_2 = -1$$

Accordingly, we write the Yule-Walker equations as

$$\begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} -0.5 \\ 1 \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \end{bmatrix} \quad (2)$$

$$\sigma_v^2 = \sum_{k=0}^2 a_k r(k)$$

$$1 = a_0 r(0) + a_1 r(1) + a_2 r(2)$$

$$= r(0) + 0.5r(1) - r(2) \quad (3)$$

Eqs. (1), (2) and (3) yield

$$\begin{aligned}r(0) &= 0 \\r(1) &= 1 \\r(2) &= -1/2\end{aligned}$$

Hence,

$$\mathbf{R} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

The eigenvalues of  $\mathbf{R}$  are -1, +1. For convergence of the steepest descent algorithm:

$$0 < \mu < \frac{2}{\lambda_{\max}}$$

where  $\lambda_{\max}$  is the largest eigenvalue of the correlation matrix  $\mathbf{R}$ . Hence, with  $\lambda_{\max} = 1$ .

$$0 < \mu < 2$$

$$4.11 \quad u(n-2) = 0.5u(n-1) + u(n) - v(n) \quad (1)$$

Hence,

$$w_1 = 1 \quad w_2 = 0.5$$

Accordingly, we may write  $\mathbf{R}\mathbf{w}_b = \mathbf{r}^{B*}$  as

$$\begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} = \begin{bmatrix} r(2) \\ r(1) \end{bmatrix}$$

$$\sigma_v^2 = \sum_{k=0}^2 a_k r(k)$$

$$1 = r(0) - r(1) - 0.5r(2) \quad (2)$$

$$\begin{aligned}r(0) &= 0 \\r(1) &= -2/3\end{aligned}$$

$$\text{Therefore, } \mathbf{R} = \begin{bmatrix} 0 & -2/3 \\ -2/3 & 0 \end{bmatrix}$$

The eigenvalues of  $\mathbf{R}$  are  $\lambda = -2/3, 2/3$

For convergence of the steepest descent algorithm:

$$0 < \mu < \frac{2}{\lambda_{\max}}$$

$$0 < \mu < \frac{2}{2/3}$$

$$0 < \mu < 3$$

4.12 The third-order AR process  $u(n)$  is described by the difference equation

$$u(n) = -0.5u(n-1) - 0.5u(n-2) + 0.5u(n-3) + v(n)$$

Hence

$$w_1 = -0.5, \quad w_2 = -0.5, \quad w_3 = 0.5$$

and the AR parameters equal

$$a_1 = 0.5, \quad a_2 = 0.5, \quad a_3 = -0.5$$

Accordingly, we write the Yule-Walker equations as

$$\begin{bmatrix} r(0) & r(1) & r(2) \\ r(1) & r(0) & r(1) \\ r(2) & r(1) & r(0) \end{bmatrix} \begin{bmatrix} -0.5 \\ -0.5 \\ 0.5 \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ r(3) \end{bmatrix}$$

$$\sigma_v^2 = \sum_{k=0}^3 a_k r(k)$$

$$1 = r(0) + a_1 r(1) + a_2 r(2) + a_3 r(3) \quad (3)$$

Equations (1), (2) and (3) yield

$$\begin{aligned} r(0) &= 1 \\ r(1) &= -1/2 \\ r(2) &= -1/2 \\ r(3) &= +1 \end{aligned}$$

(a) The correlation matrix is

$$\mathbf{R} = \begin{bmatrix} 1 & -1/2 & -1/2 \\ -1/2 & 1 & -1/2 \\ -1/2 & -1/2 & 1 \end{bmatrix}$$

(b) The eigenvalues of  $\mathbf{R}$  are 0, 1.5, 1.5

(c) For convergence of the steepest descent algorithm, we require

$$0 < \mu < \frac{2}{\lambda_{\max}}$$

Hence, with  $\lambda_{\max} = 1.5$ , we have

$$0 < \mu < \frac{2}{1.5}$$

4.13 (a) The first-order MA process is described by the difference equation

$$u(n) = v(n) - 0.2v(n-1)$$

Taking the  $z$ -transform of both sides:

$$U(z) = (1 - 0.2z^{-1})V(z)$$

$$\begin{aligned} \frac{U(z)}{V(z)} &= \frac{1}{(1 - 0.2z^{-1})} \\ &\approx \frac{1}{1 + (0.2)z^{-1} + 0.04z^{-2}} \end{aligned}$$

Accordingly,

$$u(n) + 0.2u(n-1) + 0.04u(n-2) = v(n)$$

$$u(n) = -0.2u(n-1) - 0.04u(n-2) + v(n)$$

for which we infer

$$w_1 = -0.2, \quad w_2 = -0.04$$

Correspondingly, the AR parameters equal

$$a_1 = 0.2, \quad a_2 = 0.04$$

Yule-Walker equations are

$$\begin{bmatrix} r(0) & r(1) \\ r(1) & r(0) \end{bmatrix} \begin{bmatrix} -0.2 \\ -0.04 \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \end{bmatrix} \quad (1)$$

$$\sigma_v^2 = \sum_{k=0}^3 a_k r(k) \quad (2)$$

$$1 = r(0) + a_1 r(1) + a_2 r(2) \quad (3)$$

Equations (1), (2) and (3) yield

$$r(0) = 1.0401$$

$$r(1) = -0.2000$$

$$r(2) = -0.0016$$

The correlation matrix is

$$R = \begin{bmatrix} r(0) & r(1) \\ R(1) & R(0) \end{bmatrix} = \begin{bmatrix} 1.0401 & -0.2000 \\ -0.2000 & 1.0401 \end{bmatrix}$$

The eigenvalues of  $\mathbf{R}$  are  $= 1.2401, 0.8401, \lambda_{\max} = 1.2401$

For convergence of the steepest descent algorithm:

$$0 < \mu < \frac{2}{1.2401}$$

That is,

$$0 < \mu < 1.6127$$

$$(b) \frac{U(z)}{V(z)} = \frac{1}{(1 - 0.2z^{-1})^{-1}}$$



$$\approx \frac{1}{1 + 0.2z^{-1} + 0.04z^{-2} + 0.008z^{-3}}$$

Accordingly,

$$u(n) + 0.2u(n-1) + 0.04u(n-2) + 0.008u(n-3) = v(n)$$

$$w_1 = -0.2, \quad w_2 = -0.04, \quad w_3 = -0.008$$

The AR parameters are

$$a_1 = 0.2, \quad a_2 = 0.04, \quad a_3 = 0.008$$

The Yule-Walker equations are

$$\begin{bmatrix} r(0) & r(1) & r(2) \\ r(1) & r(0) & r(1) \\ r(2) & r(1) & r(0) \end{bmatrix} \begin{bmatrix} -0.2 \\ -0.04 \\ 0.008 \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ r(3) \end{bmatrix} \quad (4)$$

$$\sigma_v^2 = \sum_{k=0}^3 a_k r(k) \quad (5)$$

$$1 = r(0) + a_1 r(1) + a_2 r(2) + a_3 r(3) \quad (6)$$

Equations (4), (5) and (6) yield

$$\begin{aligned} r(0) &= 1.04 \\ r(1) &= -0.2 \\ r(2) &= 0.0 \\ r(3) &= -0.003 \end{aligned}$$

$$\text{The correlation matrix } \mathbf{R} = \begin{bmatrix} 1.04 & -0.2 & 0 \\ -0.2 & 1.04 & -0.2 \\ 0 & -0.2 & 1.04 \end{bmatrix}$$

and its eigenvalues are

$$\lambda = 0.7572, \quad 1.04 \quad 1.3228, \quad \lambda_{\max} = 1.3228$$

For convergence of the steepest descent algorithm:

$$0 < \mu < \frac{2}{1.3228}$$

$$0 < \mu < 1.5119$$

(c) When the system is approximated by a second-order AR model,

$$0 < \mu < 1.6127$$

When it is approximated by a third-order AR model,

$$0 < \mu < 1.5119$$

These results show that the upperbound on the step-size parameters becomes tighter as the approximating model order is increased.

4.14 (a)  $u(n) = -0.5u(n-1) + v(n) - 0.2v(n-1)$

Take  $z$ -transforms of both sides:

$$U(z) = -0.5z^{-1}U(z) + V(z) - 0.2z^{-1}V(z)$$

$$\frac{U(z)}{V(z)} = \frac{1 - 0.2z^{-1}}{1 + 0.5z^{-1}}$$

$$= \frac{1}{(1 + 0.5z^{-1})(1 - 0.2z^{-1})^{-1}}$$

Approximation to third-order yields

$$\frac{U(z)}{V(z)} = \frac{1}{(1 + 0.5z^{-1})(1 + 0.2z^{-1} + 0.04z^{-2} + 0.08z^{-3})}$$

$$\approx \frac{1}{1 + 0.7z^{-1} + 0.14z^{-2} + 0.028z^{-3}}$$

Accordingly,

$$u(n) + 0.7u(n-1) + 0.14u(n-2) + 0.028u(n-3) = v(n)$$

$$u(n) = -0.7u(n-1) - 0.14u(n-2) - 0.028u(n-3) + v(n)$$

$$w_1 = -0.7, \quad w_2 = -0.14, \quad w_3 = -0.028$$

The AR parameters are

$$a_1 = 0.7, \quad a_2 = 0.14, \quad a_3 = 0.028$$

The Yule-Walker equations are

$$\begin{bmatrix} r(0) & r(1) & r(2) \\ r(1) & r(0) & r(1) \\ r(2) & r(1) & r(0) \end{bmatrix} \begin{bmatrix} -0.7 \\ -0.14 \\ 0.028 \end{bmatrix} = \begin{bmatrix} r(1) \\ r(2) \\ r(3) \end{bmatrix} \quad (1)$$

$$\sigma_v^2 = \sum_{k=0}^3 a_k r(k) \quad (2)$$

$$r(0) + a_1 r(1) + a_2 r(2) + a_3 r(3) = 1 \quad (3)$$

Equations (1), (2), and (3) yield

$$\begin{aligned} r(0) &= 1.6554 \\ r(1) &= -1.0292 \\ r(2) &= 0.5175 \\ r(3) &= -0.2646 \end{aligned}$$

$$\text{The correlation matrix } \mathbf{R} = \begin{bmatrix} r(0) & r(1) & r(2) \\ r(1) & r(0) & r(1) \\ r(2) & r(1) & r(0) \end{bmatrix} = \begin{bmatrix} 1.6554 & -1.0292 & 0.5175 \\ -1.0292 & 1.6554 & -1.0292 \\ 0.5175 & -1.0292 & 1.6554 \end{bmatrix}$$

(b) The eigenvalues of  $\mathbf{R}$  are 0.4358, 1.1379, 3.3925,  $\lambda_{\max} = 3.3925$

(c) For convergence of the steepest descent algorithm we require

$$0 < \mu < \frac{2}{\lambda_{\max}}, \quad \lambda_{\max} = 3.3925$$

$$0 < \mu < \frac{2}{3.3925}$$

$$0 < \mu < 0.589$$

$$4.15 \quad (a) \quad \mathbf{w}(n+1) = (1 - \mu)\mathbf{w}(n) + \mu \mathbf{R}^{-1} \mathbf{p}$$

$$= (1 - \mu)\mathbf{w}(n) + \mu \mathbf{w}_o, \quad (\mathbf{R}^{-1} \mathbf{p} = \mathbf{w}_o)$$

Subtract  $\mathbf{w}_o$  from both sides

$$\mathbf{w}(n+1) - \mathbf{w}_o = (1 - \mu)\mathbf{w}(n) + \mu \mathbf{w}_o - \mathbf{w}_o$$

$$= (1 - \mu)(\mathbf{w}(n) - \mathbf{w}_o)$$

Starting with the initial value  $\mathbf{w}_o$  and iterating:

$$\mathbf{w}(n) - \mathbf{w}_o = (1 - \mu)^n (\mathbf{w}(0) - \mathbf{w}_o)$$

From Eq. (2.52) of the text:

$$J(n) = J_{\min} + (\mathbf{w} - \mathbf{w}_o)^H \mathbf{R} (\mathbf{w} - \mathbf{w}_o) \quad (1)$$

$$= J_{\min} + (1 - \mu)^n [\mathbf{w}(0) - \mathbf{w}_o]^H \mathbf{R} (1 - \mu)^n [\mathbf{w}(0) - \mathbf{w}_o]$$

$$= J_{\min} + (1 - \mu)^{2n} [\mathbf{w}(0) - \mathbf{w}_o]^H \mathbf{R} [\mathbf{w}(0) - \mathbf{w}_o]$$

$$= J_{\min} + (1 - \mu)^{2n} (J(0) - J_{\min})$$

The transient behavior of Newton's algorithm is thus characterized by a single exponential whose corresponding time constant is

$$(1 - \mu)^{2k} = e^{-k/\tau} \quad (2)$$

where  $\mu \ll 1$ .

(b) Taking the logarithm of both sides of Eq. (2):

$$2k \ln(1 - \mu) = -k/\tau$$

When  $\mu \ll 1$ ,  $\ln(1 - \mu) \approx -\mu$ . Hence

$$-2k\mu \approx -k/\tau$$

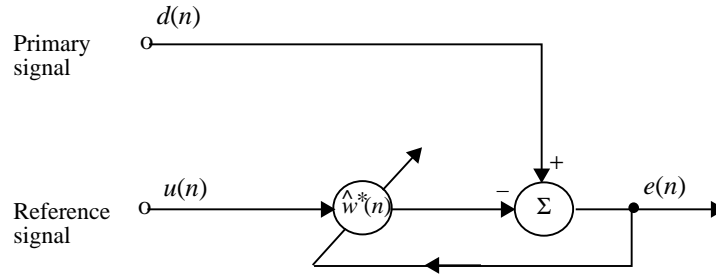
from which we readily deduce the time constant

$$\tau = \frac{1}{2\mu}, \quad \mu \ll 1.$$

## CHAPTER 5

5.1 From Fig. 5.2 of the text we see that the LMS algorithm requires  $2M+1$  complex multiplications and  $2M$  complex additions per iteration, where  $M$  is the number of tap weights used in the adaptive transversal filter. Therefore, the computational complexity of the LMS algorithm is  $O(M)$ .

5.2



$$e(n) = d(n) - \hat{w}_1^*(n)u(n)$$

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

5.3 For backward prediction, we have (assuming real data)

$$\hat{u}(n - n_0) = \hat{\mathbf{w}}^T(n) \mathbf{u}(n)$$

$$y(n) = u(n) - \hat{u}(n)$$

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu [u(n - n_0) - \hat{u}(n - n_0)] \mathbf{u}(n)$$

5.4 The adaptive line enhancer minimizes the mean-square error,  $E[|e(n)|^2]$ . For the problem at hand, the cost function  $J = E[|e(n)|^2]$  consists of the sum of three components:

- (1) The average power of the primary input noise, denoted by  $\sigma_v^2$
- (2) The average power of the noise at the filter output
- (3) The average power of contribution produced by the sinusoidal components at the input and output of the filter

Let the peak value of the transfer function be denoted by  $a$ . We may then approximate the peak value of the weights as  $2a/M$ , where  $M$  is the length of the filter. On this basis, we may approximate the average power of the noise at the filter output as  $(2a^2/M) \sigma_v^2$ , which takes care of term (2). For term (3), we assume that the input and output sinusoidal

components subtract coherently, thereby yielding the average power  $(A^2/2)(1-a)^2$ . Hence, we may express the cost function  $J$  as

$$J \approx \sigma_v^2 + \left(\frac{2a^2}{M}\right)\sigma_v^2 + \frac{A^2}{2}(1-a)^2$$

Differentiating  $J$  with respect to  $a$  and setting the result equal to zero yields the optimum scale factor

$$\begin{aligned} a_{\text{opt}} &= \frac{A^2}{A^2 + 4(\sigma_v^2/M)} \\ &= \frac{(A^2/2\sigma_v^2)(M/2)}{1 + (A^2/2\sigma_v^2)(M/2)} \\ &= \frac{(M/2)\text{SNR}}{1 + (M/2)\text{SNR}} \end{aligned}$$

$$\text{where SNR} = A^2/2\sigma_v^2$$

5.5 The index of performance equals

$$J(\mathbf{w}, K) = E[e^{2K}(n)], \quad K = 1, 2, 3, \dots$$

The estimation error  $e(n)$  equals

$$e(n) = d(n) - \mathbf{w}^T(n)\mathbf{u}(n) \tag{1}$$

where  $d(n)$  is the desired response,  $\mathbf{w}(n)$  is the tap-weight vector of the transversal filter, and  $\mathbf{u}(n)$  is the tap-input vector. In accordance with the multiple linear regression model for  $d(n)$ , we have

$$d(n) = \mathbf{w}_o^T(n)\mathbf{u}(n) \tag{2}$$

where  $\mathbf{w}_o$  is the parameter vector, and  $v(n)$  is a white-noise process of zero mean and variance  $\sigma_v^2$ .

(a) The instantaneous gradient vector equals

$$\begin{aligned}
\hat{\nabla}(n, K) &= \frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}, K) \\
&= \frac{\partial}{\partial \mathbf{w}} [e^{2K}(n)] \\
&= 2K e^{2K-1}(n) \frac{\partial e(n)}{\partial \mathbf{w}} \\
&= -2K e^{2K-1}(n) \mathbf{u}(n)
\end{aligned}$$

Hence, we may express the new adaptation rule for the estimate of the tap-weight vector as

$$\begin{aligned}
\hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n) - \frac{1}{2} \mu (-\hat{\nabla}(n, K)) \\
&= \hat{\mathbf{w}}(n) + \mu K \mathbf{u}(n) e^{2K-1}(n)
\end{aligned} \tag{3}$$

(b) Eliminate  $d(n)$  between Eqs. (1) and (2), with the estimate  $\hat{\mathbf{w}}(n)$  used in place of  $\mathbf{w}(n)$ :

$$\begin{aligned}
e(n) &= (\mathbf{w}_o - \hat{\mathbf{w}}(n))^T \mathbf{u}(n) + v(n) \\
&= \epsilon^T(n) \mathbf{u}(n) + v(n) \\
&= \mathbf{u}^T(n) \epsilon(n) + v(n)
\end{aligned} \tag{4}$$

Subtract  $\mathbf{w}_o$  from both sides of Eq. (3):

$$\epsilon(n+1) = \epsilon(n) - \mu K \mathbf{u}(n) e^{2K-1}(n) \tag{5}$$

For the case when  $\epsilon(n)$  is close to zero (i.e.,  $\hat{\mathbf{w}}(n)$  is close to  $\mathbf{w}_o$ ), we may use Eq. (4) to write

$$\begin{aligned}
e^{2K-1}(n) &= [\mathbf{u}^T(n) \epsilon(n) + v(n)]^{2K-1} \\
&= v^{2K-1}(n) \left[ 1 + \frac{\mathbf{u}^T(n) \epsilon(n)}{v(n)} \right]^{2K-1}
\end{aligned}$$



$$\begin{aligned}
&\approx v^{2K-1}(n) \left[ 1 + (2K-1) \frac{\mathbf{u}^T(n) \in (n)}{v(n)} \right] \\
&= v^{2K-1}(n) + (2K-1) \mathbf{u}^T(n) \in (n) v^{2(K-1)}(n)
\end{aligned} \tag{6}$$

Substitute Eq. (6) into (5):

$$\in (n+1) \approx [\mathbf{I} - \mu K (2K-1) v^{2(K-1)}(n) \mathbf{u}(n) \mathbf{u}^T(n)] \in (n) - \mu K v^{2K-1} \mathbf{u}(n)$$

Taking the expectation of both sides of this relation and recognizing that (1)  $\in (n)$  is independent of  $\mathbf{u}(n)$  by low-pass filtering action of the filter, (2)  $\mathbf{u}(n)$  is independent of  $v(n)$  by assumption, and (3)  $\mathbf{u}(n)$  has zero mean, we get

$$E[\in (n+1)] = \{\mathbf{I} - \mu K (2K-1) E[v^{2(K-1)}(n)] \mathbf{R}\} E[\in (n)] \tag{7}$$

where

$$\mathbf{R} = E[\mathbf{u}(n) \mathbf{u}^T(n)]$$

(c) Let

$$\mathbf{R} = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T \tag{8}$$

where  $\mathbf{\Lambda}$  is the diagonal matrix of eigenvalues of  $\mathbf{R}$ , and  $\mathbf{Q}$  is a matrix whose column vectors equal the associated eigenvectors. Hence, substituting Eq. (8) in (7) and using

$$\mathbf{v}(n) = \mathbf{Q}^T E[\in (n)]$$

we get

$$\mathbf{v}(n+1) = \{\mathbf{I} - \mu K (2K-1) E[v^{2(K-1)}(n)] \mathbf{\Lambda}\} \mathbf{v}(n)$$

That is, the  $i$ th element of this equation is

$$v_i(n+1) = \left\{ 1 - \mu K (2K-1) E[v^{2(K-1)}(n)] \lambda_i \right\} v_i(n) \tag{9}$$

where  $v_i(n)$  is the  $i$ th element of  $\mathbf{v}(n)$ , and  $i = 1, 2, \dots, M$ . Solving the first-order difference equation (9):

$$v_i(n) = \left\{ 1 - \mu K(2K-1)E[v^{2(K-1)}(n)]\lambda_i \right\}^{n-1} v_i(0)$$

where  $v_i(0)$  is the initial value of  $v_i(n)$ . Hence, for  $v_i(n)$  to converge, we require that

$$\left| 1 - \mu K(2K-1)E[v^{2(K-1)}(n)]\lambda_{\max} \right| < 1$$

where  $\lambda_{\max}$  is the largest eigenvalue of  $\mathbf{R}$ . This condition on  $\mu$  may be rewritten as

$$0 < \mu < \frac{2}{K(2K-1)\lambda_{\max}E[v^{2(K-1)}(n)]} \quad (10)$$

When this condition is satisfied, we find that

$$v_i(\infty) \rightarrow 0 \quad \text{for all } i$$

That is,  $\epsilon(\infty) \rightarrow 0$  and, correspondingly,  $\hat{\mathbf{w}}(\infty) \rightarrow \mathbf{w}_o$ .

- (d) For  $K = 1$ , the results described in Eqs. (3), (7) and (10) reduce as follows, respectively,

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu \mathbf{u}(n)e(n)$$

$$E[\epsilon(n+1)] = (\mathbf{I} - \mu \mathbf{R})E[\epsilon(n)]$$

$$0 < \mu < \frac{2}{\lambda_{\max}}$$

These results are recognized to be the same as those for the conventional LMS algorithm for real-valued data.

5.6 (a) We start with the equation

$$E[\epsilon(n+1)] = (\mathbf{I} - \mu \mathbf{R})E[\epsilon(n)] \quad (1)$$

where

$$\epsilon(n) = \mathbf{w}_o - \hat{\mathbf{w}}(n)$$

We note that

$$\mathbf{m}(n) = E[\hat{\mathbf{w}}(n)]$$

$$\mathbf{w}_o = \mathbf{m}(\infty)$$

Hence, we may rewrite Eq. (1) as

$$\mathbf{m}(n+1) - \mathbf{m}(\infty) = (\mathbf{I} - \mu \mathbf{R})(\mathbf{m}(n) - \mathbf{m}(\infty))$$

This is a first-order difference equation in  $\mathbf{m}(n) - \mathbf{m}(\infty)$ . Solving it, we get

$$\mathbf{m}(n) - \mathbf{m}(\infty) = (\mathbf{I} - \mu \mathbf{R})^n (\mathbf{m}(0) - \mathbf{m}(\infty))$$

where  $\mathbf{m}(0)$  is the initial value of  $\mathbf{m}(n)$ . Hence,

$$\mathbf{m}(n) = \mathbf{m}(\infty) + (\mathbf{I} - \mu \mathbf{R})^n (\mathbf{m}(0) - \mathbf{m}(\infty)) \quad (2)$$

(b) Using the expansion

$$\mathbf{R} = \mathbf{Q}^H \mathbf{\Lambda} \mathbf{Q}$$

and the fact that

$$\mathbf{Q}^H \mathbf{Q} = \mathbf{I}$$

we may rewrite Eq. (2) as

$$\mathbf{m}(n) = \mathbf{m}(\infty) + (\mathbf{Q}^H \mathbf{Q} - \mu \mathbf{Q}^H \mathbf{\Lambda} \mathbf{Q})(\mathbf{m}(0) - \mathbf{m}(\infty)) \quad (3)$$

Let

$$\mathbf{v}(n) = \mathbf{Q}(\mathbf{m}(n) - \mathbf{m}(\infty))$$

$$\mathbf{v}(0) = \mathbf{Q}(\mathbf{m}(0) - \mathbf{m}(\infty))$$

We may then rewrite Eq. (3) as

$$\mathbf{v}(n) = (\mathbf{I} - \mu \mathbf{\Lambda})^n \mathbf{v}(0)$$

For an arbitrary  $\mathbf{v}(0)$ ,  $\mathbf{v}(n) \rightarrow \mathbf{0}$  and the LMS algorithm is therefore convergent in the mean value if and only if

$$|1 - \mu\lambda_{\max}| < 1$$

that is

$$\mu < \frac{2}{\lambda_{\max}}$$

- 5.7 (a) For convergence of the LMS algorithm in the mean value, we require that the step-size parameter  $\mu$  satisfy the following condition:

$$0 < \mu < \frac{2}{\lambda_{\max}} \quad (1)$$

where  $\lambda_{\max}$  is the largest eigenvalue of the correlation matrix of the input vector  $\mathbf{u}(n)$ . For a white noise input of zero mean and variance  $\sigma^2$ , we have

$$\mathbf{R} = \sigma^2 \mathbf{I}$$

For such an input, all the eigenvalues assume the common value  $\sigma^2$ . Correspondingly, Eq. (1) takes on the form

$$0 < \mu < \frac{2}{\sigma^2}$$

- (b) The excess mean-square error produced by the LMS algorithm is

$$\begin{aligned} J_{\text{ex}} &= J(\infty) - J_{\min} \\ &= \frac{2}{M \sum_{i=1} \lambda_i} \end{aligned} \quad (2)$$

With

$$\lambda_i = \sigma^2 \text{ for all } i,$$

Eq. (2) takes on the form

$$J_{\text{ex}} = \frac{2}{M\sigma^2}$$

where  $M$  is the number of taps in the transversal filter.

5.8 (a) We are given

$$\begin{aligned} J(n) &= |e(n)|^2 + \alpha \|\mathbf{w}(n)\|^2 \\ &= e(n)e^*(n) + \alpha \mathbf{w}^H(n)\mathbf{w}(n) \end{aligned}$$

Hence, using the material described in Appendix B:

$$\begin{aligned} \nabla J(n) &= \frac{\partial J(n)}{\partial \mathbf{w}^*(n)} \\ &= \frac{\partial e(n)}{\partial \mathbf{w}^*(n)} e^*(n) + \frac{\partial e^*(n)}{\partial \mathbf{w}^*(n)} + 2\alpha \mathbf{w}(n) \end{aligned} \quad (1)$$

By definition,

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

$$e^*(n) = d^*(n) - \mathbf{u}^H(n)\mathbf{w}(n)$$

Therefore,

$$\frac{\partial e(n)}{\partial \mathbf{w}^*(n)} = -\mathbf{u}(n) \quad \text{and} \quad \frac{\partial e^*(n)}{\partial \mathbf{w}^*(n)} = \mathbf{0}$$

Thus, Eq. (1) reduces to

$$\nabla J(n) = -2\mathbf{u}(n)e^*(n) + 2\alpha \mathbf{w}$$

The update for the tap-weight vector in the leaky LMS algorithm is therefore as follows:

$$\begin{aligned} \hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n) - \frac{1}{2}\mu \nabla J(n) \\ &= \hat{\mathbf{w}}(n) + \mu(\mathbf{u}(n)e^*(n) - \alpha \mathbf{w}(n)) \end{aligned}$$

$$= (1 - \mu\alpha)\hat{\mathbf{w}}(n) + \mu\mathbf{u}(n)e^*(n) \quad (2)$$

(b) Taking expectations of both sides of Eq. (2):

$$E[\hat{\mathbf{w}}(n+1)] = (1 - \mu\alpha)E[\hat{\mathbf{w}}(n)] + \mu E[\mathbf{u}(n)]$$

But

$$e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n)$$

Hence,

$$\begin{aligned} E[\hat{\mathbf{w}}(n+1)] &= (1 - \mu\alpha)E[\hat{\mathbf{w}}(n)] + E[\mathbf{u}(n)d^*(n)] \\ &\quad - \mu E[\mathbf{u}(n)\mathbf{u}^H(n)\hat{\mathbf{w}}(n)] \end{aligned} \quad (3)$$

Invoking the fact that for small  $\mu$ ,  $\hat{\mathbf{w}}(n)$  is essentially independent of  $\mathbf{u}(n)$ , we may write

$$\begin{aligned} E[\mathbf{u}(n)\mathbf{u}^H(n)\hat{\mathbf{w}}(n)] &= E[\mathbf{u}(n)\mathbf{u}^H(n)]E[\hat{\mathbf{w}}(n)] \\ &= \mathbf{R}E[\hat{\mathbf{w}}(n)] \end{aligned}$$

We also recognize that

$$E[\mathbf{u}(n)d^*(n)] = \mathbf{p}$$

We may therefore rewrite Eq. (3) as

$$E[\hat{\mathbf{w}}(n+1)] = (1 - \mu\alpha)E[\hat{\mathbf{w}}(n)] + \mu\mathbf{p} - \mu\mathbf{R}E[\hat{\mathbf{w}}(n)] \quad (4)$$

As  $n \rightarrow \infty$ ,  $E[\hat{\mathbf{w}}(n+1)]$  and  $E[\hat{\mathbf{w}}(n)]$  become indistinguishable, in which case we may simplify Eq. (4) to

$$0 = -\mu\alpha E[\hat{\mathbf{w}}(n)] + \mu\mathbf{p} - \mu\mathbf{R}E[\hat{\mathbf{w}}(n)] \quad \text{for } n \rightarrow \infty$$

Equivalently, we may write

$$\lim_{n \rightarrow \infty} E[\hat{\mathbf{w}}(n)] = (\mathbf{R} + \alpha\mathbf{I})^{-1}\mathbf{p}$$

(c) Let the input vector applied to the standard LMS algorithm be written as

$$\mathbf{u}_{\text{eq}}(n) = \mathbf{u}(n) + \mathbf{v}(n) \quad (5)$$

where  $\mathbf{v}(n)$  is a white noise process of zero mean and variance  $\alpha$ . The correlation matrix of  $\mathbf{u}_{\text{eq}}(n)$  is therefore

$$\mathbf{R}_{\text{eq}} = \mathbf{R} + \alpha \mathbf{I}$$

Correspondingly, we may write

$$\begin{aligned} \lim_{n \rightarrow \infty} \hat{\mathbf{w}}(n) &= \mathbf{R}_{\text{eq}}^{-1} \mathbf{p} \\ &= (\mathbf{R} + \alpha \mathbf{I})^{-1} \mathbf{p} \end{aligned}$$

Thus, the standard LMS algorithm is equivalent to a leaky LMS algorithm with its input vector  $\mathbf{u}_{\text{eq}}(n)$  related to the input vector  $\mathbf{u}(n)$  of the leaky LMS algorithm in the manner described in Eq. (5).

5.9 We start with

$$\mathbf{K}(n+1) = (\mathbf{I} - \mu \mathbf{R}) \mathbf{K}(n) (\mathbf{I} - \mu \mathbf{R}) + \mu^2 J_{\min} \mathbf{R} \quad (1)$$

We are given

$$\mathbf{R} = \sigma_v^2 \mathbf{I} \quad (2)$$

Hence, Eq. (1) reduces to

$$\mathbf{K}(n+1) = (1 - \mu \sigma_v^2)^2 \mathbf{K}(n) + \mu^2 \sigma_v^2 J_{\min} \mathbf{I}$$

For  $n \rightarrow \infty$ , we thus have

$$\mathbf{K}(\infty) = (1 - \mu \sigma_v^2)^2 \mathbf{K}(\infty) + \mu^2 \sigma_v^2 J_{\min} \mathbf{I}$$

That is,

$$(2 - \mu \sigma_v^2) \mathbf{K}(\infty) = \mu J_{\min} \mathbf{I} \quad (3)$$

For convergence in the mean square, we choose the step-size parameter  $\mu$  to satisfy the condition

$$0 < \mu < \frac{2}{\sum_{i=1}^M \lambda_i}$$

For an input process  $\mathbf{u}(n)$  whose correlation matrix  $\mathbf{R}$  is described in Eq. (2), we have

$$\lambda_i = \sigma_v^2 \text{ for } i = 1, 2, \dots, M$$

where  $M$  is the number of taps in the transversal filter. Hence,

$$0 < \mu < \frac{2}{M\sigma_v^2}$$

For large  $M$ , it follows that  $(2 - \mu\sigma_v^2)$  is closely approximated by 2, in which case Eq. (3) takes on the approximate form

$$\mathbf{K}(\infty) \approx \frac{\mu}{2} J_{\min} \mathbf{I}$$

5.10 We start with Eq. (5.58) in the text

$$\epsilon_0(n+1) = (\mathbf{I} - \mu\mathbf{R}) \epsilon_0(n) + \mu\mathbf{u}(n)e_0^*(n) \quad (1)$$

This difference equation has a solution in convolutional form

$$\epsilon_0(n) = \mathbf{H}(n) \star [\mu\mathbf{u}(n)e_0^*(n)] = \sum_{m=-\infty}^{\infty} \mathbf{H}(m) [\mu\mathbf{u}(n-m)e_0^*(n-m)],$$

where  $\star$  denotes convolution, and the matrix

$$\mathbf{H}(n) = (\mathbf{I} - \mu\mathbf{R})^{n-1} \text{ for } n > 0 \text{ and } \mathbf{H}(n) = \mathbf{0} \text{ for } n \leq 0 \quad (2)$$

represents the causal symmetric  $M \times M$  matrix impulse response of the system.



We are interested in the transmission of stationary stochastic signals, particularly the transformation of the input autocorrelation (use is made of the statistical independence of  $\mathbf{u}(n)$  and  $e_0^*(n)$ )

$$\begin{aligned}\mathbf{F}^{(l)} &= E\left\{\mu\mathbf{u}(n)e_0(n)\mu\mathbf{u}^H(n-l)e_0^*(n-l)\right\} \\ &= E\left\{e_0(n)e_0^*(n-l)\right\}E\{\mu\mathbf{u}(n)\mu\mathbf{u}^H(n-l)\} = \mu^2 J_{\min}^{(l)} \mathbf{R}^{(l)}\end{aligned}\quad (3)$$

into the weight-error correlation matrix  $\mathbf{K}_0 = E\{\epsilon_0(n)\epsilon_0^H(n)\}$ . Using standard results of linear system theory, this is found as

$$\mathbf{K}_0 = \sum_m \sum_l \mathbf{H}(m) \mathbf{F}^{(l)} \mathbf{H}^H(m+l) = \sum_l \mathbf{T}^{(l)}, \quad \text{where } \mathbf{T}^{(l)} = \sum_m \mathbf{H}(m) \mathbf{F}^{(l)} \mathbf{H}^H(m+l) \quad (4)$$

In an attempt to evaluate the last sum in Eq. (4), we encounter the difficulty that the  $\mathbf{H}(m)$  and  $\mathbf{F}^{(l)}$ , in general, do not commute, thus prohibiting the extraction of  $\mathbf{F}^{(l)}$  from the sum. In fact, the last sum does not admit an explicit summation. Instead we show that for  $\mu \rightarrow 0$ ,  $\mathbf{T}^{(l)}$  satisfies a Lyapunov equation. To this end, we write the impulse response Eq. (2) in the recursive form [where the Dirac function  $\delta(m)$  equals unity for  $m = 0$  and zero elsewhere]

$$\mathbf{H}(m+1) = (\mathbf{I} - \mu\mathbf{R})\mathbf{H}(m) + \delta(m)\mathbf{I} = \mathbf{H}(m)(\mathbf{I} - \mu\mathbf{R}) + \delta(m)\mathbf{I} \quad (5)$$

which yields

$$\begin{aligned}\mathbf{T}^{(l)} &= \sum_m \mathbf{H}(m+1) \mathbf{F}^{(l)} \mathbf{H}^H(m+l+1) \\ &= \sum_m [(\mathbf{I} - \mu\mathbf{R})\mathbf{H}(m) + \delta(m)\mathbf{I}] \mathbf{F}^{(l)} [\mathbf{H}^H(m+l)(\mathbf{I} - \mu\mathbf{R}) + \delta(m+l)\mathbf{I}] \\ &= (\mathbf{I} - \mu\mathbf{R}) \mathbf{T}^{(l)} (\mathbf{I} - \mu\mathbf{R}) + \mathbf{F}^{(l)} \mathbf{H}(l) (\mathbf{I} - \mu\mathbf{R}) + (\mathbf{I} - \mu\mathbf{R}) \mathbf{H}^H(-l) \mathbf{F}^{(l)} + \delta(l) \mathbf{F}^{(l)}\end{aligned}\quad (6)$$

For small  $\mu$  we may simplify Eq. (6) by neglecting the terms  $\mu \mathbf{R} \mathbf{T}^{(l)} \mu \mathbf{R}$ ,  $-\mathbf{F}^{(l)} \mathbf{H}(l) \mu \mathbf{R}$ ,  $-\mu \mathbf{R} \mathbf{H}(-l) \mathbf{F}^{(l)}$ , and approximate  $\mathbf{F}^{(l)} \mathbf{H}(l) + \mathbf{H}(-l) \mathbf{F}^{(l)} + \delta(l) \mathbf{F}^{(l)}$  by  $\mathbf{F}^{(l)}$ . Thus we arrive at the Lyapunov equation

$$\mathbf{R} \mathbf{T}^{(l)} + \mathbf{T}^{(l)} \mathbf{R} = \mu^{-1} \mathbf{F}^{(l)} \quad (7)$$

and, after summation over  $l$ , using  $\mathbf{K}_0 = \sum_l \mathbf{T}^{(l)}$ , we get

$$\mathbf{R} \mathbf{K}_0 + \mathbf{K}_0 \mathbf{R} = \sum_l \mu^{-1} \mathbf{F}^{(l)} = \mu \sum_l J_{\min}^{(l)} \mathbf{R}^{(l)}. \quad (8)$$

which is the desired result.

5.11 (a) We start with Eq. (5.80) of the text:

$$v_k(n) = (1 - \mu \lambda_k)^n v_k(0) + \sum_{i=0}^{n-1} (1 - \mu \lambda_k)^{n-1-i} \phi_k(i)$$

Applying the expectation operator and noting that

$$E[\phi_k(n)] = 0 \quad \text{for all } k$$

we get

$$E[v_k(n)] = (1 - \mu \lambda_k)^n v_k(0), \quad k = 1, 2, \dots, M$$

(b) The mean-square value of  $v_k(n)$  is

$$\begin{aligned} E[|v_k(n)|^2] &= (1 - \mu \lambda_k)^{2n} |v_k(0)|^2 \\ &\quad + \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (1 - \mu \lambda_k)^{n-1-i} (1 - \mu \lambda_k)^{n-1-j} E[\phi_k(i) \phi_k^*(j)] \end{aligned}$$

where we have ignored the cross-product terms as the initial value  $v_k(0)$  is independent of  $\phi_k(i)$ . Next we note

$$E[\phi_k(i)\phi_k^*(j)] = \begin{cases} \mu^2 J_{\min} \lambda_k, & i = j \\ 0, & \text{otherwise} \end{cases}$$

Hence,

$$\begin{aligned} E[|v_k(n)|^2] &= (1 - \mu \lambda_k)^{2n} |v_k(0)|^2 \\ &\quad + \sum_{i=0}^{n-1} (1 - \mu \lambda_k)^{2n-2-2i} \mu^2 J_{\min} \lambda_k \\ &= (1 - \mu \lambda_k)^{2n} |v_k(0)|^2 \\ &\quad + \mu^2 J_{\min} \lambda_k (1 - \mu \lambda_k)^{2n-2} \sum_{i=0}^{n-1} (1 - \mu \lambda_k)^{-2i} \end{aligned} \tag{1}$$

The sum

$$\sum_{i=0}^{n-1} (1 - \mu \lambda_k)^{-2i}$$

represents a geometric series with a first term equal to 1, a geometric ratio of  $(1 - \mu \lambda_k)^{-2}$ , and a total number of terms equal to  $n$ . We thus have

$$\begin{aligned} \sum_{i=0}^{n-1} (1 - \mu \lambda_k)^{-2i} &= \frac{1 - (1 - \mu \lambda_k)^{-2n}}{1 - (1 - \mu \lambda_k)^{-2}} \\ &= -\frac{(1 - \mu \lambda_k)^2 - (1 - \mu \lambda_k)^{-2n+2}}{\mu \lambda_k (2 - \mu \lambda_k)} \end{aligned}$$

Hence, Eq. (1) reduces to

$$E[|v_k(n)|^2] = (1 - \mu \lambda_k)^{2n} |v_k(0)|^2$$

$$\begin{aligned}
& -\mu J_{\min}(1-\mu\lambda_k)^{2n-2} \left( \frac{(1-\mu\lambda_k)^2 - (1-\mu\lambda_k)^{-2n+2}}{2-\mu\lambda_k} \right) \\
& = \frac{\mu}{2-\mu\lambda_k} J_{\min} + (1-\mu\lambda_k)^{2n} \left( |v_k(0)|^2 - \frac{\mu J_{\min}}{2-\mu\lambda_k} \right)
\end{aligned}$$

5.12 (a) The mean-square deviation is defined by

$$\begin{aligned}
D(n) &= E[\| \mathbf{e}(n) \|^2] \\
&\approx E[\| \mathbf{e}(n) \|^2] \text{ for small step-size } \mu \\
&= E[\| \mathbf{Q}\mathbf{V}(n) \|^2], \quad \mathbf{Q} = \text{orthonormal matrix} \\
&= E[\| \mathbf{V}(n) \|^2]
\end{aligned}$$

For  $n \rightarrow \infty$ , we thus have

$$\begin{aligned}
D(\infty) &\approx E[\| \mathbf{V}(\infty) \|^2] \\
&= E \left[ \sum_{k=1}^M |v_k(\infty)|^2 \right] \\
&= \sum_{k=1}^M E[|v_k(\infty)|^2]
\end{aligned}$$

From Eq. (5.82),

$$E[|v_k(n)|^2] = \frac{\mu J_{\min}}{2-\mu\lambda_k} + (1-\mu\lambda_k)^{2n} \left( |v_k(\infty)|^2 - \frac{\mu J_{\min}}{2-\mu\lambda_k} \right)$$

Hence, with  $\mu$  small compared to  $2/\lambda_{\max}$ ,

$$E[|v_k(\infty)|^2] = \frac{\mu J_{\min}}{2-\mu\lambda_k}$$

$$\approx \frac{\mu J_{\min}}{2} \quad \text{for small } \mu.$$

(b) From the Lyapunov equation derived in Problem 5.10, we have

$$\mathbf{R}\mathbf{K}_0(n) + \mathbf{K}_0(n)\mathbf{R} \approx \mu J_{\min} \mathbf{R}, \quad \mu \text{ small}$$

where only the first term of the summation in the right-hand side of Eq. 8 in the solution to Problem 5.10 is retained. Taking the trace of both sides of this equation, and recognizing that

$$\text{tr}[\mathbf{R}\mathbf{K}_0(n)] = \text{tr}[\mathbf{K}_0(n)\mathbf{R}]$$

we get for  $n \rightarrow \infty$ :

$$2\text{tr}[\mathbf{R}\mathbf{K}_0(\infty)] \approx \mu J_{\min} \text{tr}[\mathbf{R}]$$

From Eq. (5.90) of the text,

$$J_{\text{ex}}(\infty) = \text{tr}[\mathbf{R}\mathbf{K}_0(\infty)] = \frac{\mu}{2} J_{\min} \text{tr}[\mathbf{R}]$$

Hence, the misadjustment is

$$\begin{aligned} M &\approx \frac{J_{\text{ex}}(\infty)}{J_{\min}} \\ &= \frac{\mu}{2} \text{tr}[\mathbf{R}] \end{aligned}$$

5.13 The error correlation matrix  $\mathbf{K}(n)$  equals

$$\mathbf{K}(n) = E[\mathbf{e}(n)\mathbf{e}^H(n)]$$

The trace of  $\mathbf{K}(n)$  equals

$$\begin{aligned} \text{tr}[\mathbf{K}(n)] &= \text{tr}\{E[\mathbf{e}(n)\mathbf{e}^H(n)]\} \\ &= E\{\text{tr}[\mathbf{e}(n)\mathbf{e}^H(n)]\} \end{aligned}$$

Since

$$\text{tr}[\epsilon(n) \epsilon^H(n)] = \text{tr}[\epsilon^H(n) \epsilon(n)]$$

we may express the trace of  $\mathbf{K}(n)$  as

$$\begin{aligned} \text{tr}[\mathbf{K}(n)] &= E\{\text{tr}[\epsilon^H(n) \epsilon(n)]\} \\ &= \text{tr}\{E[\epsilon^H(n) \epsilon(n)]\} \end{aligned}$$

The inner product  $\epsilon^H(n)\epsilon(n)$  equals the squared norm of  $\epsilon(n)$  which is a scalar. Hence

$$\text{tr}[\mathbf{K}(n)] = E[\|\epsilon(n)\|^2] \quad (1)$$

From convergence analysis of the LMS algorithm, we have

$$\mathbf{K}(n+1) = (\mathbf{I} - \mu\mathbf{R})\mathbf{K}(n)(\mathbf{I} - \mu\mathbf{R}) + \mu^2 J_{\min} \mathbf{R} \quad (2)$$

Initially,  $\|\epsilon(n)\|$  is so large that we may justifiably ignore the term  $\mu^2 J_{\min} \mathbf{R}$ , in which case Eq. (2) may be approximated as

$$\mathbf{K}(n+1) = (\mathbf{I} - \mu\mathbf{R})\mathbf{K}(n)(\mathbf{I} - \mu\mathbf{R}), \quad n \text{ small} \quad (3)$$

Assuming that

$$\mathbf{R} = \sigma_u^2 \mathbf{I}$$

we may further reduce Eq. (3) to

$$\mathbf{K}(n+1) \approx (1 - \mu\sigma_u^2)^2 \mathbf{K}(n)$$

Thus, in light of Eq. (1) we may write

$$E[\|\epsilon(n+1)\|^2] \approx (1 - \mu\sigma_u^2)^2 E[\|\epsilon(n)\|^2], \quad n \text{ small}$$

The convergence ratio is therefore approximately

$$c(n) = \frac{E[\|\mathbf{e}(n+1)\|^2]}{E[\|\mathbf{e}(n)\|^2]}$$

$$\approx (1 - \mu\sigma_u^2)^2$$

5.14 (a) The tap-weight update in the LMS algorithm is defined by

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

where  $\mathbf{u}(n)$  is the tap-input vector, and  $e(n)$  is the estimation error (i.e., error signal). There are two different cases to be considered:

1. The process from which  $\mathbf{u}(n)$  is drawn is deterministic, and so is the desired response  $d(n)$ : under these conditions, the estimation error  $e(n)$  is likewise deterministic. Correspondingly, the correction term  $\mu \mathbf{u}(n) e^*(n)$  computed for each iteration of the algorithm is a deterministic quantity. Hence, starting from a known initial value  $\hat{\mathbf{w}}(0)$ , the trajectory traced out by the evolution of  $\hat{\mathbf{w}}(n)$  with increasing number of iterations,  $n$ , can be determined exactly. In other words, when the LMS algorithm operates in a completely deterministic environment, exemplified by the sinusoidal example presented in Section 5.9, the algorithm exhibits a deterministic behavior.
  2. The environment in which the LMS algorithm is stochastic: In this second case, the correction term  $\mu \mathbf{u}(n) e^*(n)$  assumes a stochastic form. Remembering that the product  $\mathbf{u}(n) e^*(n)$  represents an estimate of the gradient vector, we may (in this case) view the LMS algorithm as a stochastic gradient algorithm.
- (b) When white noise is added to the sinusoidal process of Example 7 in Section 5.9, the input vector  $\mathbf{u}(n)$  takes on a stochastic form. Specifically, the correlation matrix  $\mathbf{R}$  of  $\mathbf{u}(n)$  now consists of the expression described in Eq. (5.135) of the text, plus  $\sigma_v^2 \mathbf{I}$ . In this new situation the evolution of  $\hat{\mathbf{w}}(n)$  with increasing  $n$  follows a stochastic (random) trajectory. The ensemble-averaged value of this trajectory follows a path defined in terms of  $\mathbf{R}$ .

5.15 (a) The update equations for the tandem configuration of the LMS filters are formulated as follows:

$$e_1(n) = d(n) - \hat{\mathbf{w}}_1^H(n) \mathbf{u}(n)$$

$$\hat{\mathbf{w}}_1^H(n+1) = \hat{\mathbf{w}}_1^H(n) + \mu \mathbf{u}(n) e_1(n)$$

$$e_2(n) = e_1(n) - \hat{\mathbf{w}}_2^H(n) \mathbf{u}(n)$$

$$\hat{\mathbf{w}}_2(n+1) = \hat{\mathbf{w}}_2(n) + \mu \mathbf{u}(n) e_2(n)$$

This assumes that both filters are updated with the same step-size  $\mu$ .

- (b) As can be seen from Fig. P5.2 as well as from the equations in part (a) of this problem, the adaptation of  $\hat{\mathbf{w}}_1$  follows the exponential convergence of the LMS algorithm and its behavior is unaffected by  $\hat{\mathbf{w}}_2$ . The adaptation of  $\hat{\mathbf{w}}_2$  is, however, dependent on  $\hat{\mathbf{w}}_1$ . Let  $\mathbf{R} = [\mathbf{u}(n), \mathbf{u}^H(n)] = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^H$ , where  $\mathbf{R}$  is the correlation matrix of the input vector. We use the eigendecomposition of  $\mathbf{R}$ . If we consider rotated representations of the weight vectors

$$\hat{\mathbf{v}}_1^1(n) = \mathbf{Q}^H \hat{\mathbf{w}}_1$$

$$\hat{\mathbf{v}}_2^1(n) = \mathbf{Q}^H \hat{\mathbf{w}}_2$$

and keep in mind that the signal is modelled by  $d(n) = \mathbf{w}_o^H \mathbf{u}(n) + v(n)$  where  $v(n)$  is a zero mean random noise, we may show that the total  $\text{MSE} = \text{MSE}_{11} + \text{MSE}_{12} + \text{MSE}_{22}$ .

Hence,  $\text{MSE}_{ij} = E[\epsilon_i(n) \epsilon_j^H(n)]$ , where  $(i, j) = 1, 2$  is the ordered number of the LMS filters, and  $\epsilon_i(n)$  = weight-error vector of LMS filter  $i$ . This result clearly indicates that the tandem system converges in the mean-square sense if  $\text{MSE}_{11}$ ,  $\text{MSE}_{12}$ , and  $\text{MSE}_{22}$  converge in the mean-square.

- 5.16 (a) The cost function is defined by

$$J(n) = \frac{1}{2} |e(n)|^2$$

The derivative of  $J(n)$  with respect to the step-size parameter  $\mu$  is

$$\frac{\partial J(n)}{\partial \mu} = \frac{1}{2} \left[ e(n) \frac{\partial e^*(n)}{\partial \mu} + e^*(n) \frac{\partial e(n)}{\partial \mu} \right]$$

But



$$e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n)$$

Hence

$$\frac{\partial e(n)}{\partial \mu} = -\frac{\partial \hat{\mathbf{w}}^H(n)}{\partial \mu}\mathbf{u}(n) = -\gamma^H(n)\mathbf{u}(n)$$

$$\begin{aligned}\Delta \mu(n) &= -\rho \frac{\partial J(n)}{\partial \mu} \\ &= \frac{\rho}{2} [e(n)\mathbf{u}^H(n)\gamma(n) + e^*(n)\gamma^H(n)\mathbf{u}(n)]\end{aligned}$$

where  $\rho$  is the learning-rate parameter for adjusting the step-size in an iterative manner.

From the LMS algorithm, we have (assuming a variable step size)

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

Hence

$$\frac{\partial \hat{\mathbf{w}}(n+1)}{\partial \mu} = \frac{\partial \hat{\mathbf{w}}(n)}{\partial \mu} + \mathbf{u}(n)e^*(n)\mu(n+1)\mathbf{u}(n)\frac{\partial e^*(n)}{\partial \mu}$$

That is,

$$\begin{aligned}\gamma(n+1) &= \gamma(n) + \mathbf{u}(n)e^*(n) - \mu(n+1)\mathbf{u}(n)\mathbf{u}^H(n)\gamma(n) \\ &= [\mathbf{I} - \mu(n+1)\mathbf{u}(n)\mathbf{u}^H(n)]\gamma(n) + \mathbf{u}(n)e^*(n)\end{aligned}$$

Finally, stepping back by one iteration, we get the desired result

$$\gamma(n) = [\mathbf{I} - \mu(n)\mathbf{u}(n-1)\mathbf{u}^H(n-1)]\gamma(n-1) + \mathbf{u}(n-1)e^*(n-1)$$

- (b) We may now formulate the two recursions for an LMS algorithm with adjustable step-size as follows:

$$\mu(n) = \mu(n-1) + \frac{1}{2}\rho [e(n)\mathbf{u}^H(n)\gamma(n) + e^*(n)\gamma^H(n)\mathbf{u}(n)]$$

$$= \mu(n-1) + \text{Re}[e^*(n)\gamma^H(n)\mathbf{u}(n)]$$

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

5.17 The tightening of the inequality in ERq. (5.145) is justified by invoking two relations:

- The update equation for the LMS algorithm:

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

or, equivalently,

$$\epsilon(n+1) = \epsilon(n) - \mu\mathbf{u}(n)e^*(n)$$

$$= \epsilon(n) - \mu\mathbf{u}(n)[d^*(n) - \mathbf{u}^H(n)\hat{\mathbf{w}}(n)]$$

- The multiple regression model

$$d(n) = \mathbf{w}^H\mathbf{u}(n) + v(n)$$

where  $\mathbf{w}$  is the parameter vector of the model.

Here it is assumed that the step-size parameter  $\mu$  is maintained smaller than  $1/\|\mathbf{u}(n)\|^2$  in accordance with Eq. (5.146).

5.18 The input signal is the sum of two uncorrelated sinusoidal inputs:

$$u(n) = A_0\cos(w_0n) + A_1\cos(w_1n + \phi) \quad (1)$$

where the phase difference  $\phi$  is a random variable. [In the first printing of the book, the phase  $\phi$  was unfortunately missed out.] Hence, the autocorrelation function of the input  $u(n)$  is the sum of the autocorrelation functions of the two sinusoidal inputs.

The transfer function approach works on the premise that the LMS filter length  $M$  is large. Under this condition, the nonlinear LMS filter may be approximated as a linear discrete-time system with an excitation

$$x(n) = d(n) - \hat{\mathbf{w}}^H(0)\mathbf{u}(n), \quad d(n) = \text{desired response}$$

$\hat{\mathbf{w}}(0) = \text{initial tap-weight vector}$

and the error

$$e(n) = d(n) - \hat{\mathbf{w}}^H(n)u(n)$$

being the resulting response.

Whenever we have a linear system, the principle of superposition applies. Invoking this principle, we may then state that, under the above-mentioned conditions, the LMS filter, with its input defined by Eq. (1), produces an output equal to the sum of two components, one produced by the sinusoidal input of frequency  $\omega_0$  and the other by the second sinusoidal input of frequency  $\omega_1$ .

Now, from the material on adaptive noise cancellation presented in Section 5.3 of the text, we know that the LMS filter with a sinusoidal input of frequency  $\omega_0$  acts as a notch filter centered on  $\omega_0$ ; see Fig. 5.8(b) of the text. Accordingly, we conclude that the LMS filter acting on the composite sinusoidal input of Eq. (1) acts as the sum of two notch filters, one centered on frequency  $\omega_0$  and the other on  $\omega_1$ .

5.19 From Eq. (2.77) of the text, the cost function for a constrained MVDR beamformer is

$$J = \sum_{k=0}^{m-1} \sum_{i=0}^{m-1} w_k^* w_i r(i-k) + \text{Re} \left[ \lambda^* \left( \sum_{k=0}^{m-1} w_k^* e^{-j\theta_0 k} - g \right) \right] \quad (1)$$

where  $\lambda$  is the Lagrange multiplier defined in Eq. (2.84)

$$\lambda = -\frac{2g}{\mathbf{s}^H(\theta_0) \mathbf{R}^{-1} \mathbf{s}(\theta_0)} \quad (2)$$

where  $g$  is a specified complex-valued gain,  $\mathbf{R}$  is the correlation matrix of the input vector  $\mathbf{u}(n)$ , and  $\mathbf{s}(\theta_0)$  is the steering vector directed at the target of interest. Combining Eqs. (1) and (2) and rewriting it in matrix form:

$$J = \mathbf{w}^H \mathbf{R} \mathbf{w} - (k^* \mathbf{w}^H \mathbf{s}(\theta_0) + k \mathbf{s}^H(\theta_0) \mathbf{w} + \text{Re}[g]) \quad (3)$$

where

$$k = \frac{g}{\mathbf{s}^H(\theta_0) \mathbf{R}^{-1} \mathbf{s}(\theta_0)} \quad (4)$$

which acts merely as a scaling factor ensuring that the gain of the beamformer along the direction  $\theta_0$  is equal to  $g$ . Differentiating the cost function  $J$  with respect to the weight vector  $\mathbf{w}$ :

$$\frac{\partial J}{\partial \mathbf{w}} = 2\mathbf{R}\mathbf{w} - 2\mathbf{s}(\theta_0)k^* \quad (5)$$

Substituting  $\mathbf{R} = \mathbf{u}(n)\mathbf{u}^H(n)$  into Eq. (5), where  $\mathbf{u}(n)$  is the input vector:

$$\frac{\partial J(n)}{\partial \mathbf{w}} = 2\mathbf{u}(n)\mathbf{u}^H(n)\hat{\mathbf{w}}(n) - 2\mathbf{s}(\theta_0)k^* \quad (6)$$

Hence, the LMS algorithm for the MVDR beamformer is

$$\begin{aligned} \hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n) - \frac{1}{2}\mu \frac{\partial J}{\partial \mathbf{w}} \\ &= [\mathbf{I} - \mu\mathbf{u}(n)\mathbf{u}^H(n)]\hat{\mathbf{w}}(n) + \mathbf{s}(\theta_0)k^* \end{aligned} \quad (7)$$

where  $\mu$  is the step-size parameter.

Comparing this algorithm with the LMS algorithm for temporal processing, we see that  $\mathbf{s}(\theta_0)k^*$  plays the product role of desired response and input vector.

As a check on the algorithm described in Eq. (7), suppose the step-size parameter  $\mu$  is small enough to justify using the direct method. We may then replace  $\mathbf{u}(n)\mathbf{u}^H(n)$  by  $\mathbf{R}$  and write

$$\hat{\mathbf{w}}(n+1) \approx [\mathbf{I} - \mu\mathbf{R}]\hat{\mathbf{w}}(n) + \mu k^* \mathbf{s}(\theta_0) \quad (8)$$

For  $n \rightarrow \infty$ ,  $\hat{\mathbf{w}}(n) \approx \hat{\mathbf{w}}(n+1)$ , in which case Eq. (4) reduces to

$$\mathbf{R}\hat{\mathbf{w}}(n) \approx k^* \mathbf{s}(\theta_0), \quad n \text{ large}$$

or

$$\hat{\mathbf{w}}(n) \approx k^* \mathbf{R}^{-1} \mathbf{s}(\theta_0), \quad n \text{ large}$$

The corresponding beamformer output along  $\theta_0$  is, as desired,

$$\begin{aligned}\hat{\mathbf{w}}^H(n)\mathbf{s}(\theta_0) &= k^* \mathbf{s}^H(\theta_0) \mathbf{R}^{-1} \mathbf{s}(\theta_0) \\ &= g^*, \quad n \text{ large}\end{aligned}$$

## CHAPTER 6

6.1 (a) We note that

$$J(n+1) = E[|e(n+1)|^2]$$

where

$$e(n+1) = d(n+1) - \mathbf{w}^H(n+1)\mathbf{u}(n+1)$$

Hence,

$$\begin{aligned} J(n+1) &= E[|d(n+1) - \mathbf{w}^H(n+1)\mathbf{u}(n+1)|^2] \\ &= \sigma_d^2 - \left[ \mathbf{w}(n) - \frac{1}{2}\mu(n)\nabla(n) \right]^H \mathbf{p} - \mathbf{p}^H \left[ \mathbf{w}(n) - \frac{1}{2}\mu(n)\nabla(n) \right] \\ &\quad + \left[ \mathbf{w}(n) - \frac{1}{2}\mu(n)\nabla(n) \right]^H \mathbf{R} \left[ \mathbf{w}(n) - \frac{1}{2}\mu(n)\nabla(n) \right] \end{aligned}$$

Dropping the terms that are *not* a function of  $\mu(n)$ , and differentiating  $J(n+1)$  with respect to  $\mu(n)$ , we get

$$\begin{aligned} \frac{\partial}{\partial \mu(n)} J(n+1) &= \frac{\partial}{\partial \mu(n)} \left\{ \frac{1}{2}\mu(n)\nabla^H(n)\mathbf{p} + \frac{1}{2}\mu(n)\mathbf{p}^H\nabla(n) \right. \\ &\quad \left. - \frac{1}{2}\mu(n)\nabla^H(n)\mathbf{R} \left[ \mathbf{w}(n) - \frac{1}{2}\mu(n)\nabla(n) \right] \right. \\ &\quad \left. - \frac{1}{2}\mu(n) \left( \mathbf{w}(n) - \frac{1}{2}\mu(n)\nabla(n) \right)^H \mathbf{R} \nabla(n) \right\} \\ &= \frac{1}{2}(\nabla^H(n)\mathbf{p}^H\nabla(n)) \\ &\quad - \frac{1}{2}(\nabla^H(n)\mathbf{R}\mathbf{w}(n) + \mathbf{w}^H(n)\mathbf{R}\nabla(n)) + \mu(n)\nabla^H(n)\mathbf{R}\nabla(n) \end{aligned}$$

Setting the result equal to zero, and solving for  $\mu(n)$ :

$$\mu_o(n) = \frac{\frac{1}{2}(\nabla^H(n)\mathbf{R}\mathbf{w}(n) + \mathbf{w}^H(n)\mathbf{R}\nabla(n) - \nabla^H(n)\mathbf{p} - \mathbf{p}^H\nabla(n))}{\nabla^H(n)\mathbf{R}\nabla(n)} \quad (1)$$

(b) We are given that

$$\nabla(n) = 2(\mathbf{R}\mathbf{w}(n) - \mathbf{p})$$

Hence, Eq. (1) simplifies to

$$\mu_o(n) = \frac{\nabla^H(n)\nabla(n)}{\nabla^H(n)\mathbf{R}\nabla(n)}$$

Using instantaneous estimates for  $\mathbf{R}$  and  $\nabla(n)$ :

$$\hat{\mathbf{R}} = \mathbf{u}(n)\mathbf{u}^H(n)$$

$$\nabla(n) = 2[\mathbf{u}(n)\mathbf{u}^H(n)\hat{\mathbf{w}}(n) - \mathbf{u}(n)d^*(n)]$$

$$= -2\mathbf{u}(n)e^*(n)$$

we find that the corresponding value of  $\mu_o(n)$  is

$$\begin{aligned} \hat{\mu}_o(n) &= \frac{\mathbf{u}^H(n)\mathbf{u}(n)|e(n)|^2}{(\mathbf{u}^H(n)\mathbf{u}(n))^2|e(n)|^2} \\ &= \frac{1}{\mathbf{u}^H(n)\mathbf{u}(n)} = \frac{1}{\|\mathbf{u}(n)\|^2} \end{aligned}$$

Correspondingly, we have

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

which is recognized as the normalized LMS algorithm.

$$\begin{aligned}
6.2 \quad \delta \hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n) \\
&= \frac{1}{\|\mathbf{u}(n)\|^2} \mathbf{u}^H(n) \mathbf{u}(n) [\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)] \\
&= \frac{1}{\|\mathbf{u}(n)\|^2} \mathbf{u}(n) [\mathbf{u}^H(n) \hat{\mathbf{w}}(n+1) - \mathbf{u}^H(n) \hat{\mathbf{w}}(n)] \\
&= \frac{1}{\|\mathbf{u}(n)\|^2} \mathbf{u}(n) (d^*(n) - \mathbf{u}^H(n) \hat{\mathbf{w}}(n)) \\
&= \frac{1}{\|\mathbf{u}(n)\|^2} \mathbf{u}(n) e^*(n)
\end{aligned}$$

6.3 The second statement is the correct one. The justification is obvious from the solution to Problem 2; see also Eq. (6.10) of the text. By definition,

$$\begin{aligned}
\text{Since } \mathbf{u}(n) &= [u(n), u(n-1) \dots u(n-M+1)]^T \\
\hat{\mathbf{w}}(n) &= [w_0(n), w_1(n) \dots w_{M-1}(n)]^T \\
\hat{\mathbf{w}}(n+1) &= [w_0(n+1), w_1(n+1) \dots w_{M-1}(n+1)]
\end{aligned}$$

Replace all of these terms into the NLMS formula

$$\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n) = \frac{\tilde{u}}{\|\mathbf{u}(n)\|^2} \mathbf{u}(n) e^*(n)$$

and look at each element in the vector. We then find that the correct answer is

$$\hat{w}_k(n+1) - \hat{w}_k(n) = \frac{\tilde{u}}{\|\mathbf{u}(n)\|^2} u(n-k) e^*(n) \quad k = 0, 1, \dots, M-1$$

$$\begin{aligned}
6.4 \quad \delta \hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n) \\
&= \mathbf{A}^{-1}(n) \mathbf{A}(n) [\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)] \\
&= \mathbf{A}^{-1}(n) [\mathbf{A}(n) \hat{\mathbf{w}}(n+1) - \mathbf{A}(n) \hat{\mathbf{w}}(n)]
\end{aligned}$$



$$\begin{aligned}
&= \mathbf{A}^{-1}(n)[\mathbf{d}(n) - \mathbf{A}(n)\hat{\mathbf{w}}(n)] \\
&= \mathbf{A}^{-1}(n)\left[\frac{1}{2}\mathbf{A}(n)\mathbf{A}^H(n)\lambda\right], \quad \text{from Eq. (6.44) of the text} \\
&= \frac{1}{2}\mathbf{A}^{-1}(n)\mathbf{A}(n)\mathbf{A}^H(n)\lambda \\
&= \frac{1}{2}\mathbf{A}^H(n)2(\mathbf{A}(n)\mathbf{A}^H(n))^{-1}\mathbf{e}(n) \\
&= \mathbf{A}^H(n)(\mathbf{A}(n)\mathbf{A}^H(n))^{-1}\mathbf{e}(n)
\end{aligned}$$

6.5

	Virtues	Limitations
LMS	Simple. Stable $H^\infty$ robust Model-independent	Slow convergence. Learning rate must have the dimension of inverse power.
NLMS	Convergent in the mean sense. Stable in mean-square sense. $H^\infty$ robust. Invariant to scaling factor of input. Dimensionless step-size. (special case of APAF).	Little increase in computational complexity. (compared to LMS)
APAF	More accurate due to the use of more information. Semi-batch learning. Faster convergence.	Increased computational complexity.

6.6 Provided we show, under different scaling situations, that the learning rules (weight-update formula) are the same, then we can say the final solutions are the same under the same initial condition.

(1) NLMS, unscaled:

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

NLMS, scaled:

$$\begin{aligned}
\hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n) + \frac{\tilde{\mu}}{\|a\mathbf{u}(n)\|^2} a\mathbf{u}(n) [\mathbf{w}^H(n+1)a\mathbf{u}(n) - a\hat{\mathbf{w}}^H(n)\mathbf{u}(n)] \\
&= \hat{\mathbf{w}}(n) + \frac{\tilde{\mu}}{a^2\|\mathbf{u}(n)\|^2} a^2\mathbf{u}(n) [\mathbf{w}^H(n+1)\mathbf{u}(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n)] \\
&= \hat{\mathbf{w}}(n) + \frac{\tilde{\mu}}{\|\mathbf{u}(n)\|^2} \mathbf{u}(n) [d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n)] \\
&= \hat{\mathbf{w}}(n) + \frac{\tilde{\mu}}{\|\mathbf{u}(n)\|^2} \mathbf{u}(n) e^*(n)
\end{aligned}$$

which is the same as the unscaled NLMS.

(2) APAF

Denote  $\mathbf{A}_{\text{scaled}} = [a\mathbf{u}(n), a\mathbf{u}(n-1), \dots, a\mathbf{u}(n-N+1)]$

$$\mathbf{d}_{\text{scaled}} = \mathbf{A}_{\text{scaled}}(n) \hat{\mathbf{w}}(n+1)$$

Unscaled case:

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mathbf{A}^H(n)(\mathbf{A}(n)\mathbf{A}^H(n))^{-1} \mathbf{e}(n)$$

Scaled case:

$$\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)$$

$$\begin{aligned}
&= \mathbf{A}_{\text{scaled}}^{-1}(n) \mathbf{A}_{\text{scaled}}(n) [\hat{\mathbf{w}}(n+1) - \hat{\mathbf{w}}(n)] \\
&= \mathbf{A}_{\text{scaled}}^{-1}(n) [\mathbf{A}_{\text{scaled}}(n) \hat{\mathbf{w}}(n+1) - \mathbf{A}_{\text{scaled}}(n) \hat{\mathbf{w}}(n)] \\
&= \mathbf{A}_{\text{scaled}}^{-1}(n) [\mathbf{d}_{\text{scaled}}(n) - \mathbf{A}_{\text{scaled}}(n) \mathbf{u}(n)] \\
&= \mathbf{A}_{\text{scaled}}^{-1}(n) \left[ \frac{1}{2} \mathbf{A}_{\text{scaled}}(n) \mathbf{A}_{\text{scaled}}^H(n) \lambda \right]
\end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2} \mathbf{A}_{\text{scaled}}^H(n) 2 (\mathbf{A}_{\text{scaled}}(n) \mathbf{A}_{\text{scaled}}^H(n))^{-1} \mathbf{e}_{\text{scaled}}(n) \\
&= a \mathbf{A}^H(n) (a^2 \mathbf{A}(n) \mathbf{A}^H(n))^{-1} a \mathbf{e}(n) \\
&= \mathbf{A}^H(n) (\mathbf{A}(n) \mathbf{A}^H(n))^{-1} \mathbf{e}(n)
\end{aligned}$$

which is the same as the unscaled APAF.

$$6.7 \quad u(n) = \sum_{k=1}^{N-1} w_k^* u(n-k) + v(n)$$

(a) The algorithms can be APAF formulated as follows:

Step (1): Initialize AR coefficients  $w_1, w_2, \dots, w_{N-1}$  as random values

Step (2): Suppose  $\mathbf{w}(n) = [w_1(n), \dots, w_{N-1}(n)]^H$   
 $\mathbf{u}(n) = [u(n-1), \dots, u(n-N+1)]^H$

Hence, we have  $u(n) = \hat{\mathbf{w}}^H(n) \mathbf{u}(n-1) + v(n)$

Suppose  $\mathbf{A}^H(n) = [\mathbf{u}(n-1), \dots, \mathbf{u}(n-N+1)]$

Calculate  $\mathbf{e}(n) = \mathbf{u}(n) - \mathbf{A}^H(n) \hat{\mathbf{w}}(n)$

If  $\|\mathbf{e}(n)\|^2 \leq \alpha$  ( $\alpha$  is predefined small positive value)

go to end

else go to step (3)

Step (3): Update  $\hat{\mathbf{w}}(n)$

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \tilde{\mu} \mathbf{A}^H(n) (\mathbf{A}(n) \mathbf{A}^H(n))^{-1} \mathbf{e}(n)$$

go to Step (2).

$$\begin{aligned}
\text{(b) } \phi(n) &= [\mathbf{I} - \mathbf{A}^H(n)(\mathbf{A}^H(n)\mathbf{A}^H(n))^{-1}\mathbf{A}(n)]\mathbf{u}(n) \\
&= \mathbf{u}(n) - \mathbf{A}^H(n)(\mathbf{A}(n)\mathbf{A}^H(n))^{-1}\mathbf{A}^H(n)\mathbf{u}(n) \\
&= \mathbf{u}(n) - \mathbf{P}\mathbf{u}(n)
\end{aligned}$$

where  $\mathbf{P}$  is the projection operator.

Geometrically, the distance (difference) between  $\mathbf{u}(n)$  and its projected vector  $\mathbf{P}\mathbf{u}(n)$  is the error (noise vector). Hence, with  $\mathbf{v}(n)$  assumed to be white Gaussian of zero mean, it follows that the elements of the vector  $\phi(n)$  are themselves zero-mean white Gaussian processes.

6.8 (a) From Eq. (6.19),

$$\tilde{\mu}_{\text{opt}} = \frac{\text{Re}\left\{E[\xi_u(n)e^*((n)/\|\mathbf{u}(n)\|^2)]\right\}}{E[|e(n)|^2/\|\mathbf{u}(n)\|^2]}$$

Assuming that the undisturbed estimation error  $\xi_u(n)$  is equal to the disturbed estimation error (i.e., normal error signal)  $e(n)$ , then we may put

$$\tilde{\mu}_{\text{opt}} \approx 1$$

in which case Eq. (6.18) gives the bounds on  $\tilde{\mu}$  as

$$0 < \tilde{\mu} < 2$$

(b) For the APAF, from Eq. (6.56) we know

$$0 < \tilde{u} < \frac{2E\left\{\text{Re}\left[\xi_u^H(n)(\mathbf{A}(n)\mathbf{A}^H(n))^{-1}\mathbf{e}(n)\right]\right\}}{E[\mathbf{e}^H(n)(\mathbf{A}(n)\mathbf{A}^H(n))^{-1}\mathbf{e}(n)]}$$

where  $\xi_u(n) = \mathbf{A}(n)\mathbf{w} - \mathbf{A}(n)\hat{\mathbf{w}}(n)$  is the undisturbed error vector, and

$\mathbf{e}(n) = \mathbf{A}(n)\hat{\mathbf{w}}(n+1) - \mathbf{A}(n)\hat{\mathbf{w}}(n)$  is the disturbed error vector. Here again, if these two error vectors are assumed to be equal, then the bounds on  $\tilde{\mu}$  given in Eq. (6.56) reduce to  $0 < \tilde{\mu} < 2$ .

- 6.9 (a) For the NLMS filter, suppose  $M$  is the length of the filter. Examine the weight-update formula

$$\begin{cases} \hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n) + \tilde{\mu} \frac{\mathbf{u}(n)\mathbf{e}^*(n)}{\|\mathbf{u}(n)\|^2} \\ e(n) &= d(n) - \hat{\mathbf{w}}^H(n)\mathbf{u}(n) \end{cases}$$

The computation involving these calculations involves  $5M$  multiplications (or division) and  $2M$  additions. Hence, the computation complexity is  $O(M)$ .

- (b) For the APAF, suppose  $N$  is the order of the filter. Examine the weight-update formula

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \tilde{\mu} \mathbf{A}^H(n)(\mathbf{A}(n)\mathbf{A}^H(n))^{-1} \mathbf{e}(n)$$

$$\mathbf{A}^H(n) = [\mathbf{u}(n), \mathbf{u}(n-1), \dots, \mathbf{u}(n-N+1)]$$

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{A}(n)\hat{\mathbf{w}}(n)$$

$$\mathbf{d}(n) = [d(n), d(n-1), \dots, d(n-N+1)]$$

Here we see that the computation involved in APAF is about  $N$  times as that of NLMS. Hence, the computational complexity of APAF is  $O(MN)$ .

## CHAPTER 7

7.1 When  $M = 6, L = 4$ , we have for the  $k$ th block

$$\begin{bmatrix} u(4k) & u(4k-1) & u(4k-2) & u(4k-3) & u(4k-4) & u(4k-5) \\ u(4k+1) & u(4k) & u(4k-1) & u(4k-2) & u(4k-3) & u(4k-4) \\ u(4k+2) & u(4k+1) & u(4k) & u(4k-1) & u(4k-2) & u(4k-3) \\ u(4k+3) & u(4k+2) & u(4k+1) & u(4k) & u(4k-1) & u(4k-2) \end{bmatrix} \begin{bmatrix} w_0(k) \\ w_1(k) \\ w_2(k) \\ w_3(k) \\ w_4(k) \\ w_5(k) \end{bmatrix} = \begin{bmatrix} y(4k) \\ y(4k+1) \\ y(4k+2) \\ y(4k+3) \end{bmatrix}$$

For the  $(k+j)$ th, with  $j = \pm 1, \pm 2, \dots$ , we have

$$\begin{bmatrix} u(uk+4j) & u(uk+4j-1) & u(uk+4j-2) & u(uk+4j-3) & u(uk+4j-4) & u(uk+4j-5) \\ u(uk+4j+1) & u(uk+4j) & u(uk+4j-1) & u(uk+4j-2) & u(uk+4j-3) & u(uk+4j-4) \\ u(uk+4j+2) & u(uk+4j+1) & u(uk+4j) & u(uk+4j-1) & u(uk+4j-2) & u(uk+4j-3) \\ u(uk+4j+3) & u(uk+4j+2) & u(uk+4j+1) & u(uk+4j) & u(uk+4j-1) & u(uk+4j-2) \end{bmatrix} \begin{bmatrix} w_0(k) \\ w_1(k) \\ w_2(k) \\ w_3(k) \\ w_4(k) \\ w_5(k) \end{bmatrix} = \begin{bmatrix} y(4k+4j) \\ y(4k+4j+1) \\ y(4k+4j+2) \\ y(4k+4j+3) \end{bmatrix}$$

7.2 (a) Using the  $2M$ -by- $2M$  constraint matrix  $\mathbf{G}$ , we may rewrite the cross-correlation vector  $\phi(k)$  as follows:

$$[\phi^T(k), 0, \dots, 0]^T = \mathbf{G}_1 \mathbf{F}^{-1} \mathbf{U}^H(k) \mathbf{E}(k)$$

Accordingly, the weight-update is rewritten in the compact form:

$$\hat{\mathbf{W}}(k+1) = \hat{\mathbf{W}}(k) + \mu \mathbf{F} \mathbf{G}_1 \mathbf{F}^{-1} \mathbf{U}^H(k) \mathbf{E}(k)$$

$$= \hat{\mathbf{W}}(k) + \mu \mathbf{G} \mathbf{U}^H(k) \mathbf{E}(k)$$

where

$$\mathbf{G} = \mathbf{F} \mathbf{G}_1 \mathbf{F}^{-1}$$

The matrix operator  $\mathbf{F}$  denotes discrete Fourier transformation, and  $\mathbf{F}^{-1}$  denotes inverse discrete Fourier transformation.

(b) With

$$\mathbf{G}_2 = [\mathbf{0}, \mathbf{I}]$$

the error vector  $\mathbf{E}(k)$  is readily rewritten as follows:

$$\begin{aligned}\mathbf{E}(k) &= \mathbf{F}[0, \dots, 0, \mathbf{e}^T(k)]^T \\ &= \mathbf{F}\mathbf{G}_2^T \mathbf{e}^T(k)\end{aligned}$$

(c) Using these matrix notations, the fast LMS algorithm may be reformulated as follows:

Initialization:

$$\hat{\mathbf{W}}(0) = [0, 0, \dots, 0]^T$$

$$P_i(0) = \delta_i, \quad i = 0, 1, \dots, 2M-1$$

where  $M$  is the number of taps, and  $\delta_i$  is a small positive number.

Computation: For each new block of  $M$  input samples, compute the following:

$$\mathbf{U}(k) = \text{diag}(\mathbf{F}[u(kM-M), \dots, u(kM-1), u(kM), \dots, u(kM+M-1)]^T)$$

$$\mathbf{Y}(k) = \mathbf{U}(k)\mathbf{W}(k)$$

$$\mathbf{y}(k) = \mathbf{G}_2\mathbf{F}^{-1}\mathbf{Y}(k)$$

$$\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{y}(k)$$

$$\mathbf{E}(k) = \mathbf{F}\mathbf{G}_2^T \mathbf{e}(k)$$

$$P_i(k) = \gamma P_i(k-1) + (1 - \gamma) |U_i(k)|^2, \quad i = 0, 1, \dots, 2M-1$$

$$\mu(k) = \text{diag}(P_0^{-1}(k), P_1^{-1}(k), \dots, P_{2M-1}^{-1}(k))$$

$$\hat{\mathbf{W}}(k+1) = \hat{\mathbf{W}}(k) + \mathbf{F}\mathbf{G}_1\mathbf{F}^{-1}\tilde{\mu}(k)\mathbf{U}(k)\mathbf{E}(k)$$

(d) When the constraint gradient  $\mathbf{G}$  is equal to the identity matrix, the fast LMS algorithm reduces to its unconstrained frequency-domain form.

7.3 Removing the gradient constraint shown in Fig. 7.3 permits the adaptive filter to perform circular convolution instead of linear convolution. (Circular convolution is the form of convolution performed by the discrete Fourier transform.) This modification may be acceptable in those applications where there is no particular concern as to how the input signal is used to minimize the mean-square value of the error signal (estimation error). One such application is that of noise (interference) cancellation.

7.4 (a) From Fig. 7.1, we note that the  $z$ -transform of  $x_k(n)$  is related to the  $z$ -transform of the input  $u(n)$  as follows:

$$z[x_k(n)] = \frac{1 - z^{-M}}{1 - e^{-\frac{j2\pi}{M}} z^{-1}} z[u(n)], \quad k = 0, 1, \dots, M-1$$

Cross-multiplying and rearranging terms:

$$z[x_k(n)] = e^{-\frac{j2\pi}{M}} z^{-1} \cdot z[x_k(n)] + z[u(n)] - z^{-M} \cdot z[u(n)]$$

Taking inverse  $z$ -transforms:

$$x_k(n) = e^{-\frac{j2\pi}{M}k} x_k(n-1) + u(n) - u(n-M), \quad k = 0, 1, \dots, M-1$$

(b) We are given that

$$\mathbf{h}(n+1) = \mathbf{h}(n) + \tilde{\mu}\mathbf{D}^{-1}\mathbf{x}(n)e^*(n) \quad (1)$$

The error signal is

$$\begin{aligned} e(n) &= d(n) - \mathbf{h}^H(n)\mathbf{x}(n) \\ &= d(n) - \mathbf{x}^T(n)\mathbf{h}^*(n) \end{aligned} \quad (2)$$

Hence, substituting Eq. (2) into (1), and rearranging:



$$\mathbf{h}(n+1) = [\mathbf{I} - \tilde{\mu} \mathbf{D}^{-1} \mathbf{x}(n) \mathbf{x}^H(n)] \mathbf{h}(n) + \tilde{\mu} \mathbf{D}^{-1} \mathbf{x}(n) d^*(n)$$

Taking the expectation of both sides of this equation, and invoking the properties of LMS filters with a small step size:

$$E[\mathbf{h}(n+1)] = [\mathbf{I} - \tilde{\mu} \mathbf{D}^{-1} \mathbf{R}_x] E[\mathbf{h}(n)] + \tilde{\mu} \mathbf{D}^{-1} \mathbf{p}_x \quad (3)$$

where  $\mathbf{R}_x$  is the correlation matrix of the input vector  $\mathbf{x}(n)$  and  $\mathbf{p}_x$  is the cross-correlation vector between  $\mathbf{x}(n)$  and  $d(n)$ . Assuming convergence in the mean,

$$E[\mathbf{h}(n+1)] = E[\mathbf{h}(n)] = \mathbf{h}_o \text{ as } n \rightarrow \infty$$

Hence, from Eq. (3) we deduce that

$$\mathbf{R}_x \mathbf{h}_o = \mathbf{p}_x \quad (4)$$

But  $x_k(n)$  is related to the input vector  $\mathbf{u}(n)$  by the DFT formula:

$$\begin{aligned} x_k(n) &= \sum_{i=0}^{M-1} u(n-i) e^{-\frac{j2\pi}{M} ik} \\ &= \mathbf{u}^T(n) \boldsymbol{\varepsilon}_k^* = \boldsymbol{\varepsilon}_k^H \mathbf{u}(n) \end{aligned}$$

$$\text{where } \boldsymbol{\varepsilon}_k = \left[ 1, e^{\frac{j2\pi}{M} k}, \dots, e^{\frac{j2\pi}{M} (M-1)k} \right]^T$$

Hence, the vector  $\mathbf{x}(n)$  is related to  $\mathbf{u}(n)$  by

$$\begin{aligned} \mathbf{x}(n) &= \begin{bmatrix} \boldsymbol{\varepsilon}_0^H \\ \boldsymbol{\varepsilon}_1^H \\ \vdots \\ \boldsymbol{\varepsilon}_{M-1}^H \end{bmatrix} \mathbf{u}(n) \\ &= \sqrt{M} \mathbf{Q} \mathbf{u}(n) \end{aligned}$$

where  $\mathbf{Q}$  is the  $M$ -by- $M$  unitary matrix

$$\mathbf{Q} = \frac{1}{\sqrt{M}} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & e^{-\frac{j2\pi}{M}} & \cdots & e^{-\frac{j2\pi}{M}(M-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-\frac{j2\pi}{M}(M-1)} & \cdots & e^{-\frac{j2\pi}{M}(M-1)^2} \end{bmatrix} \quad (5)$$

Accordingly, we have

$$\begin{aligned} \mathbf{R}_x &= E[\mathbf{x}(n)\mathbf{x}^H(n)] \\ &= E[\mathbf{Q}\mathbf{u}(n)\mathbf{u}^H(n)\mathbf{Q}^H] \\ &= \mathbf{Q}\mathbf{R}\mathbf{Q}^H \end{aligned} \quad (6)$$

where  $\mathbf{R}$  is the correlation matrix of the input vector  $\mathbf{u}(n)$ . Similarly, we may write

$$\begin{aligned} \mathbf{p}_x &= E[\mathbf{x}(n)d^*(n)] \\ &= E[\mathbf{Q}\mathbf{u}(n)d^*(n)] \\ &= \mathbf{Q}\mathbf{p} \end{aligned} \quad (7)$$

where  $\mathbf{p}$  is the cross-correlation vector between  $\mathbf{u}(n)$  and  $d(n)$ . Next, substituting Eqs. (6) and (7) into (4):

$$\mathbf{Q}\mathbf{R}\mathbf{Q}^H \mathbf{h}_o = \mathbf{Q}\mathbf{p}$$

The matrix  $\mathbf{Q}$  is nonsingular, and its inverse matrix therefore exists. Hence,

$$\mathbf{R}\mathbf{Q}^H \mathbf{h}_o = \mathbf{p}$$

We recognize the Wiener solution as

$$\mathbf{w}_o = \mathbf{R}^{-1} \mathbf{p}$$

It follows therefore that

$$\mathbf{Q}^H \mathbf{h}_o = \mathbf{w}_o$$

The matrix  $\mathbf{Q}$  is in actual fact a unitary matrix; we may thus also write

$$\mathbf{h}_o = \mathbf{Q} \mathbf{w}_o$$

The unitary matrix  $\mathbf{Q}$  is related to the DFT by Eq. (5).

(c) (i) From part (b), we have

$$\begin{aligned} \mathbf{h}_o &= \mathbf{Q} \mathbf{w}_o \\ &= \mathbf{Q} \mathbf{R}^{-1} \mathbf{p} \\ &= \mathbf{Q} \mathbf{R}^{-1} \mathbf{Q}^H \mathbf{Q} \mathbf{p} \end{aligned}$$

We recognize  $\mathbf{Q} \mathbf{R}^{-1} \mathbf{Q}^H$  as the correlation matrix of the tap-input vector  $\mathbf{x}(n)$ , and  $\mathbf{Q} \mathbf{p}$  as the cross-correlation vector between  $\mathbf{x}(n)$  and the desired response  $d(n)$ . From the unitary similarity transformation,  $\mathbf{Q} \mathbf{R}^{-1} \mathbf{Q}^H$  equals a diagonal matrix. This is functionally equivalent to prewhitening the original input vector  $\mathbf{u}(n)$  characterized by the correlation matrix  $\mathbf{R}$ .

- (ii) The eigenvalue spread  $\chi(\mathbf{R})$  of the original input vector  $\mathbf{u}(n)$  is, in general, greater than unity. On the other hand, the eigenvalue spread  $\chi(\mathbf{R})$  of the DFT output vector  $\mathbf{x}(n)$ , which is prewhitened by the DFT, is closer to unity. In general, we therefore find that the eigenvalue spread of  $\mathbf{u}(n)$  is larger than that of  $\mathbf{x}(n)$ .
- (iii) The frequency-domain LMS algorithm has a faster rate of convergence than the conventional LMS algorithm, because of the result reported in sub-part (ii) of (c).

7.5 (a) We start with

$$\begin{aligned} C_m(n) &= \frac{1}{2} k_m (-1)^m W_{2M}^{m/2} A_m(n) \\ &= \frac{1}{2} k_m (-1)^m W_{2M}^{m/2} (A_m^{(1)}(n) + A_m^{(2)}(n)) \end{aligned}$$

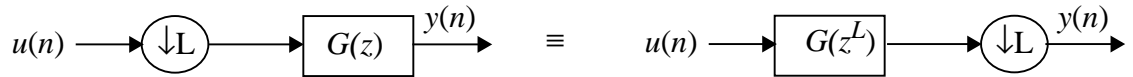
$$= \frac{1}{2}k_m[C_m^{(1)}(n) + C_m^{(2)}(n)], \quad m = 0, 1, \dots, M-1$$

where

$$\begin{aligned} C_m^{(1)}(n) &= (-1)^m W_{2M}^{m/2} A_m^{(1)}(n) \\ &= (-1)^m W_{2M}^{m/2} [W_{2M}^m A_m^{(1)}(n-1) + u(n) - (-1)^m u(n-M)] \\ &= W_{2M}^{m/2} [W_{2M}^{m/2} C_m^{(1)}(n-1) + (-1)^m u(n) - u(n-M)] \\ C_m^{(2)}(n) &= (-1)^m W_{2M}^{m/2} A_m^{(2)}(n) \\ &= (-1)^m W_{2M}^{m/2} [W_{2M}^{-m} A_m^{(2)}(n-1) + W_{2M}^{-m} (u(n) - (-1)^m u(n-M))] \\ &= W_{2M}^{-m/2} [W_{2M}^{-m/2} C_m^{(2)}(n-1) + (-1)^m u(n) - u(n-M)] \end{aligned}$$

(b) Basically, the terms  $W_{2M}^{m/2}$ ,  $C_m^{(1)}(n-1)$  and  $u(n-m)$  involved in the computation of  $C_m^{(1)}(n)$ , and likewise the terms  $W_{2M}^{-m/2}$ ,  $C_m^{(2)}(n-1)$  and  $u(n-m)$  involved in the computation of  $C_m^{(2)}(n)$  are all multiplied by the constant  $\beta$ , to ensure stability of the algorithm.

## 7.6 Using noble identity 1



Now put  $L = 2$ . Noting that in the  $z$ -domain, the 2-band down-sampling operator

$$x(n) \longrightarrow \downarrow 2 \longrightarrow y(n)$$

can be expressed by

$$Y(E) = \frac{1}{2}[X(z^{1/2}) + X(-z^{1/2})] \quad (1)$$

Observing the bottom part of Fig. 7.13 in the text, we may formulate the synthesis filter output as

$$\text{upper} \quad \frac{1}{2}[U(z^{1/2})\hat{W}(z^{1/2})H_1(z^{1/2}) + U(-z^{1/2})\hat{W}(-z^{1/2})H_1(-z^{1/2})] \quad (2)$$

$$\text{lower} \quad \frac{1}{2}[U(z^{1/2})\hat{W}(z^{1/2})H_0(z^{1/2}) + U(-z^{1/2})\hat{W}(-z^{1/2})H_1(-z^{1/2})] \quad (3)$$

By polyphase decomposition of  $\hat{W}(z)$  [see Eq. (7.79) of the text]

$$\hat{W}(z) = \hat{W}_1(z^2) + z^{-1}\hat{W}_2(z^2)$$

it follows that

$$\hat{W}(z^{1/2}) = \hat{W}_1(z) + z^{-1/2}\hat{W}_2(z) \quad (4)$$

Substituting Eq. (4) into (2) and rearranging the terms, we have

$$\begin{aligned} & \frac{1}{2}[U(z^{1/2})H_1(z^{1/2}) + U(-z^{1/2})H_1(-z^{1/2})]\hat{W}_1(z) \\ & + \frac{1}{2}[U(z^{1/2})H_1(z^{1/2})z^{1/2} + U(-z^{1/2})H_1(-z^{1/2})z^{1/2}]\hat{W}_2(z) \end{aligned} \quad (5)$$

Similarly, substituting Eq. (4) into (3), we have

$$\begin{aligned} & \frac{1}{2}[U(z^{1/2})H_0(z^{1/2}) + U(-z^{1/2})H_0(-z^{1/2})]\hat{W}_1(z) \\ & + \frac{1}{2}[U(z^{1/2})H_0(z^{1/2})z^{-1/2} + U(-z^{1/2})H_0(-z^{1/2})z^{-1/2}]\hat{W}_2(z) \end{aligned} \quad (6)$$

Equation (5) corresponds to the description of the upper-bottom part of Fig. 7.15 (following from Eq. (1)). Equation (6) corresponds to the description of the lower-bottom part of Fig. 7.15. The equivalence between the two figures, 7.13 and 7.15, is established.

7.7 Consider the case when the input  $u(n)$  is complex-valued. Suppose

$$u(n) = u_I(n) + ju_Q(n)$$

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}_I(n) + j\hat{\mathbf{w}}_Q(n)$$

$$\begin{aligned} e(n) &= [e_{1,I}(n) + je_{1,Q}(n), e_{2,I}(n) + je_{2,Q}(n)] \\ &= ([e_{1,I}(n), e_{2,I}(n)] + j[e_{1,Q}(n), e_{2,Q}(n)]) \end{aligned}$$

$$\mathbf{P} = \begin{bmatrix} \alpha_1 & 0 \\ 0 & \alpha_2 \end{bmatrix}$$

Define the cost function

$$\begin{aligned} J(n) &= \frac{1}{2} \mathbf{e}^H(n) \mathbf{P} \mathbf{e}^*(n) \\ &= \frac{1}{2} [(\alpha_1 e_{1,I}^2 + e_{1,Q}^2) + \alpha_2 (e_{2,I}^2 + e_{2,Q}^2)] \end{aligned} \quad (1)$$

Suppose

$$\hat{w}_{1,k} = \hat{w}_{1,k}^{(I)} + j\hat{w}_{1,k}^{(Q)}$$

$$\hat{w}_{2,k} = \hat{w}_{2,k}^{(I)} + j\hat{w}_{2,k}^{(Q)}$$

Similar to the real-valued case, we may derive

$$\hat{w}_{1,k}^{(I)}(n+1) = \hat{w}_{1,k}^{(I)}(n) + \mu \frac{\partial J(n)}{\partial \hat{w}_{1,k}^{(I)}}$$

$$\hat{w}_{2,k}^{(I)}(n+1) = \hat{w}_{2,k}^{(I)}(n) + \mu \frac{\partial J(n)}{\partial \hat{w}_{2,k}^{(I)}}$$

$$\hat{w}_{1,k}^{(Q)}(n+1) = \hat{w}_{1,k}^{(Q)}(n) + \mu \frac{\partial J(n)}{\partial \hat{w}_{1,k}^{(Q)}}$$

$$\hat{w}_{2,k}^{(Q)}(n+1) = \hat{w}_{2,k}^{(Q)}(n) + \mu \frac{\partial J(n)}{\partial \hat{w}_{2,k}^{(Q)}}$$

From Eq. (1), we also have

$$\begin{aligned} \frac{\partial J(n)}{\partial \hat{W}_{1,k}^{(I)}} = & \alpha_1 \left[ e_{1,I}(n) \frac{\partial e_{1,I}(n)}{\partial \hat{W}_{1,k}^{(I)}} + e_{1,Q}(n) \frac{\partial e_{1,Q}(n)}{\partial \hat{W}_{1,k}^{(I)}} \right] \\ & + \alpha_2 \left[ e_{2,I}(n) \frac{\partial e_{2,I}(n)}{\partial \hat{W}_{1,k}^{(I)}} + e_{2,Q}(n) \frac{\partial e_{2,Q}(n)}{\partial \hat{W}_{1,k}^{(I)}} \right] \end{aligned} \quad (2)$$

$$\begin{aligned} \frac{\partial J(n)}{\partial \hat{W}_{2,k}^{(I)}} = & \alpha_1 \left[ e_{1,I}(n) \frac{\partial e_{1,I}(n)}{\partial \hat{W}_{2,k}^{(I)}} + e_{1,Q}(n) \frac{\partial e_{1,Q}(n)}{\partial \hat{W}_{2,k}^{(I)}} \right] \\ & + \alpha_2 \left[ e_{2,I}(n) \frac{\partial e_{2,I}(n)}{\partial \hat{W}_{2,k}^{(I)}} + e_{2,Q}(n) \frac{\partial e_{2,Q}(n)}{\partial \hat{W}_{2,k}^{(I)}} \right] \end{aligned} \quad (3)$$

$$\begin{aligned} \frac{\partial J(n)}{\partial \hat{W}_{1,k}^{(Q)}} = & \alpha_1 \left[ e_{1,I}(n) \frac{\partial e_{1,I}(n)}{\partial \hat{W}_{1,k}^{(Q)}} + e_{1,Q}(n) \frac{\partial e_{1,Q}(n)}{\partial \hat{W}_{1,k}^{(Q)}} \right] \\ & + \alpha_2 \left[ e_{2,I}(n) \frac{\partial e_{2,I}(n)}{\partial \hat{W}_{1,k}^{(Q)}} + e_{2,Q}(n) \frac{\partial e_{2,Q}(n)}{\partial \hat{W}_{1,k}^{(Q)}} \right] \end{aligned} \quad (4)$$

$$\begin{aligned} \frac{\partial J(n)}{\partial \hat{W}_{2,k}^{(Q)}} = & \alpha_1 \left[ e_{1,I}(n) \frac{\partial e_{1,I}(n)}{\partial \hat{W}_{2,k}^{(Q)}} + e_{1,Q}(n) \frac{\partial e_{1,Q}(n)}{\partial \hat{W}_{2,k}^{(Q)}} \right] \\ & + \alpha_2 \left[ e_{2,I}(n) \frac{\partial e_{2,I}(n)}{\partial \hat{W}_{2,k}^{(Q)}} + e_{2,Q}(n) \frac{\partial e_{2,Q}(n)}{\partial \hat{W}_{2,k}^{(Q)}} \right] \end{aligned} \quad (5)$$

Similarly, we obtain for  $i = 1, 2$  and  $j = 1, 2$

$$\frac{\partial e_{i,I}(n)}{\partial \hat{W}_{j,k}^{(I)}} = -X_{ij}^{(I)}(n-k) \quad \frac{\partial e_{i,Q}(n)}{\partial \hat{W}_{j,k}^{(I)}} = -X_{ij}^{(Q)}(n-k) \quad (6)$$

$$\frac{\partial e_{i,Q}(n)}{\partial \hat{W}_{j,K}^{(Q)}} = -X_{ij}^{(Q)}(n-k) \quad \frac{\partial e_{i,I}(n)}{\partial \hat{W}_{j,K}^{(Q)}} = -X_{ij}^{(I)}(n-k) \quad (7)$$

Substituting Eqs. (6) and (7) into (2)-(5), we finally get

$$\hat{W}_{1,k}^{(I)}(n+1) = \hat{W}_{1,k}^{(I)}(n) + \mu [\alpha_1 e_{1,I}(n) X_{11}^{(I)}(n-k) - \alpha_1 e_{1,Q}(n) X_{11}^{(Q)}(n-k)]$$

$$+ \alpha_2 e_{2,I}(n) X_{21}^{(I)}(n-k) - \alpha_2 e_{2,Q}(n) X_{21}^{(Q)}(n-k)]$$

$$\begin{aligned} \hat{W}_{1,k}(n+1) = & \hat{W}_{1,k}(n) + \mu [\alpha_1 e_{1,I}(n) X_{11}^{(Q)}(n-k) - \alpha_1 e_{1,Q}(n) X_{11}^{(I)}(n-k)] \\ & + \alpha_2 e_{2,I}(n) X_{21}^{(Q)}(n-k) - \alpha_2 e_{2,Q}(n) X_{21}^{(I)}(n-k)] \end{aligned}$$

$$\begin{aligned} \hat{W}_{2,k}(n+1) = & \hat{W}_{2,k}(n) + \mu [\alpha_1 (e_{1,I}(n) X_{12}^{(I)}(n-k) + \alpha_1 e_{1,Q}(n) X_{12}^{(Q)}(n-k)) \\ & + \alpha_2 e_{2,I}(n) X_{22}^{(I)}(n-k) + \alpha_2 e_{2,Q}(n) X_{22}^{(Q)}(n-k)] \end{aligned}$$

$$\begin{aligned} \hat{W}_{2,k}(n+1) = & \hat{W}_{2,k}(n) + \mu [\alpha_1 (e_{1,I}(n) X_{11}^{(Q)}(n-k) + \alpha_1 e_{1,Q}(n) X_{12}^{(I)}(n-k)) \\ & + \alpha_2 (e_{2,I}(n) X_{22}^{(Q)}(n-k) + \alpha_2 e_{2,Q}(n) X_{22}^{(I)}(n-k))] \end{aligned}$$

7.8 Using NLMS algorithm,

$$\begin{aligned} \text{suppose } \mathbf{x}_{1,1} &= \left[ x_{11}(n), x_{11}(n-1), \dots, x_{11}\left(n - \frac{M}{2} + 1\right) \right]^T \\ \mathbf{x}_{2,1} &= \left[ x_{21}(n), x_{21}(n-1), \dots, x_{21}\left(n - \frac{M}{2} + 1\right) \right]^T \end{aligned}$$

Applying the NLMS version of Eq. (7.91) of the text, we may write

$$\hat{W}_{1,k}(n+1) = \hat{W}_{1,K}(n) + \tilde{\mu} \left[ \frac{\alpha_1 e_1(n) x_{11}(n-k)}{\|\mathbf{x}_{1,1}\|^2} + \frac{\alpha_2 e_2(n) x_{21}(n-k)}{\|\mathbf{x}_{2,1}\|^2} \right]$$

Similarly, the NLMS version of Eq. (7.92) is

$$\hat{W}_{2,k}(n+1) = \hat{W}_{2,k}(n) + \tilde{\mu} \left[ \frac{\alpha_1 e_1(n) x_{12}(n-k)}{\|\mathbf{x}_{1,2}\|^2} + \frac{\alpha_2 e_2(n) x_{22}(n-k)}{\|\mathbf{x}_{2,2}\|^2} \right]$$

where

$$\begin{aligned} \mathbf{x}_{1,2} &= \left[ x_{12}(n), x_{12}(n-1), \dots, x_{12}\left(n - \frac{M}{2} + 1\right) \right]^T \\ \mathbf{x}_{2,2} &= \left[ x_{22}(n), x_{22}(n-1), \dots, x_{22}\left(n - \frac{M}{2} + 1\right) \right]^T \end{aligned}$$



The advantages of using the NLMS algorithm in subband adaptive filters:

1. Time-varying learning rate,
2. faster convergence speed, and
3. invariance to scaling of input data.

## CHAPTER 8

8.1 The Hermitian transpose of data matrix  $\mathbf{A}$  is defined by

$$\mathbf{A}^H = \begin{bmatrix} u(1, 1) & u(1, 2) & \cdots & u(1, n) \\ u(2, 1) & u(2, 2) & \cdots & u(2, n) \\ \vdots & \vdots & & \vdots \\ u(M, 1) & u(M, 2) & \cdots & u(M, n) \end{bmatrix}$$

where  $u(k, i)$  is the output of sensor  $k$  in the linear array at time  $i$ , where  $k = 1, 2, \dots, M$ , and  $i = 1, 2, \dots, n$ .

(a) The matrix product  $\mathbf{A}^H \mathbf{A}$  equals:

$$\mathbf{A}^H \mathbf{A} = \begin{bmatrix} \sum_{i=1}^n u(1, i)u^*(1, i) & \sum_{i=1}^n u(1, i)u^*(2, i) & \cdots & \sum_{i=1}^n u(1, i)u^*(M, i) \\ \sum_{i=1}^n u(2, i)u^*(1, i) & \sum_{i=1}^n u(2, i)u^*(2, i) & \cdots & \sum_{i=1}^n u(2, i)u^*(M, i) \\ \vdots & \vdots & \cdots & \vdots \\ \sum_{i=1}^n u(M, i)u^*(1, i) & \sum_{i=1}^n u(M, i)u^*(2, i) & \cdots & \sum_{i=1}^n u(M, i)u^*(M, i) \end{bmatrix}$$

This represents the  $M$ -by- $M$  spatial (deterministic) correlation matrix of the array with temporal averaging applied to each element of the matrix. This form of averaging assumes that the environment in which the array operates is temporally stationary.

(b) The matrix product  $\mathbf{A} \mathbf{A}^H$  equals

$$\mathbf{A} \mathbf{A}^H = \begin{bmatrix} \sum_{k=1}^M u^*(k, 1)u(k, 1) & \sum_{k=1}^M u^*(k, 1)u(k, 2) & \cdots & \sum_{k=1}^M u^*(k, 1)u(k, n) \\ \sum_{k=1}^M u^*(k, 2)u(k, 1) & \sum_{k=1}^M u^*(k, 2)u(k, 2) & \cdots & \sum_{k=1}^M u^*(k, 2)u(k, n) \\ \vdots & \vdots & \cdots & \vdots \\ \sum_{k=1}^M u^*(k, n)u(k, 1) & \sum_{k=1}^M u^*(k, n)u(k, 2) & \cdots & \sum_{k=1}^M u^*(k, n)u(k, n) \end{bmatrix}$$

This second matrix represents the  $n$ -by- $n$  temporal (deterministic) correlation matrix of the array with spatial averaging applied to each element of the matrix. This form of averaging assumes that the environment is spatially stationary.

8.2 We say that the least-squares estimate  $\hat{\mathbf{w}}$  is *consistent* if

$$\lim_{N \rightarrow \infty} E[\|\hat{\mathbf{w}} - \mathbf{w}_o\|^2] = 0 \quad (1)$$

We note that

$$\begin{aligned} E[\|\hat{\mathbf{w}} - \mathbf{w}_o\|^2] &= E[(\hat{\mathbf{w}} - \mathbf{w}_o)^H (\hat{\mathbf{w}} - \mathbf{w}_o)] \\ &= \text{tr} \left\{ E[(\hat{\mathbf{w}} - \mathbf{w}_o)^H (\hat{\mathbf{w}} - \mathbf{w}_o)] \right\} \\ &= E \left\{ \text{tr}[(\hat{\mathbf{w}} - \mathbf{w}_o)^H (\hat{\mathbf{w}} - \mathbf{w}_o)] \right\} \\ &= E \left\{ \text{tr}[(\hat{\mathbf{w}} - \mathbf{w}_o)(\hat{\mathbf{w}} - \mathbf{w}_o)^H] \right\} \\ &= \text{tr} \left\{ E[(\hat{\mathbf{w}} - \mathbf{w}_o)(\hat{\mathbf{w}} - \mathbf{w}_o)^H] \right\} \\ &= \text{tr}[\mathbf{K}] \\ &= \sigma^2 \text{tr}[\Phi^{-1}] \end{aligned} \quad (2)$$

where the correlation matrix  $\Phi$  is dependent on  $n$ . Substituting Eq. (2) in (1):

$$\lim_{N \rightarrow \infty} E[\|\hat{\mathbf{w}} - \mathbf{w}_o\|^2] = \lim_{N \rightarrow \infty} \sigma^2 \text{tr}[\Phi^{-1}]$$

This result shows that  $\hat{\mathbf{w}}$  is consistent if

$$\lim_{N \rightarrow \infty} \text{tr}[\Phi^{-1}] = 0$$

8.3 The data matrix is

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 1 & 2 \\ -1 & 1 \end{bmatrix}$$

The desired response vector is

$$\mathbf{d} = \begin{bmatrix} 2 \\ 1 \\ 1/34 \end{bmatrix}$$

The tap-weights vector of the linear least-squares filter is

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

We first note that

$$\begin{aligned} \mathbf{A}^T \mathbf{A} &= \begin{bmatrix} 2 & 1 & -1 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 1 & 2 \\ -1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 6 & 7 \\ 7 & 14 \end{bmatrix} \end{aligned}$$

$$\det(\mathbf{A}^T \mathbf{A}) = \begin{vmatrix} 6 & 7 \\ 7 & 14 \end{vmatrix} = 35$$

$$(\mathbf{A}^T \mathbf{A})^{-1} = \frac{1}{35} \begin{bmatrix} 14 & 7 \\ -7 & 6 \end{bmatrix} = \begin{bmatrix} \frac{2}{5} & \frac{1}{5} \\ -\frac{1}{5} & \frac{6}{35} \end{bmatrix}$$

Hence, using Eq. (1) we get

$$\begin{aligned}
\hat{\mathbf{w}} &= \begin{bmatrix} \frac{2}{5} & -\frac{1}{5} \\ -\frac{1}{5} & \frac{6}{35} \end{bmatrix} \begin{bmatrix} 2 & 1 & -1 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 1/34 \end{bmatrix} \\
&= \begin{bmatrix} \frac{2}{5} & -\frac{1}{5} \\ -\frac{1}{5} & \frac{6}{35} \end{bmatrix} \begin{bmatrix} \frac{169}{34} \\ \frac{273}{34} \end{bmatrix} \\
&= \begin{bmatrix} 0.382 \\ 0.382 \end{bmatrix}
\end{aligned}$$

8.4 Express the transfer function of the (forward) prediction-error filter as follows (see Fig. 1):

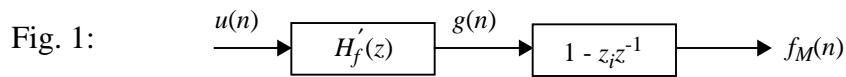
$$H_f(z) = (1 - z_i z^{-1}) H'_f(z)$$

where

$$H_f(z) = \sum_{k=0}^M a_{M,k}^* z^{-k}, \quad a_{M,0} = 1$$

and

$$z_i = \rho_i e^{j\theta_i}$$



From this figure we note that

$$Z[f_M(n)] = (1 - z_i z^{-1}) Z[g(n)]$$

Hence

$$f_M(n) = g(n) - z_i g(n-1)$$

The prediction-error energy equals (according to the autocorrelation method)

$$\begin{aligned}\epsilon_f &= \sum_{n=1}^{\infty} |f_M(n)|^2 \\ &= \sum_{n=1}^{\infty} [g(n) - z_i g(n-1)][g^*(n) - z_i^* g^*(n-1)]\end{aligned}$$

With  $z_i = \rho_i e^{j\theta_i}$ , we may expand the expression for the prediction-error energy as

$$\epsilon_f = \sum_{n=1}^{\infty} \left\{ |g(n)|^2 + \rho_i^2 |g(n-1)|^2 - 2\text{Re} \left[ \rho_i e^{j\theta_i} g(n-1) g^*(n) \right] \right\}$$

Differentiate  $\epsilon_f$  with respect to  $\rho_i$ :

$$\frac{\partial \epsilon_f}{\partial \rho_i} = 2\rho_i \sum_{n=1}^{\infty} \left\{ |g(n-1)|^2 - 2\text{Re} \left[ \sum_{n=1}^{\infty} e^{j\theta_i} g(n-1) g^*(n) \right] \right\} \quad (1)$$

From the Cauchy-Schwartz inequality, we have

$$\text{Re} \left[ \sum_{n=1}^{\infty} e^{j\theta_i} g(n-1) g^*(n) \right] \leq \left[ \sum_{n=1}^{\infty} |g^*(n)|^2 \right]^{1/2} \left[ \sum_{n=1}^{\infty} |e^{j\theta_i} g(n-1)|^2 \right]^{1/2}$$

Here we note that

$$\begin{aligned}\sum_{n=1}^{\infty} |e^{j\theta_i} g(n-1)|^2 &= \sum_{n=1}^{\infty} |g(n-1)|^2 \\ &= \sum_{n=0}^{\infty} |g(n)|^2\end{aligned} \quad (2)$$

where it is recognized that in the autocorrelation method  $g(n)$  is zero for  $n \leq 0$ . Accordingly, we may rewrite the Cauchy-Schwartz inequality as

$$\operatorname{Re} \left[ \sum_{n=1}^{\infty} e^{j\theta_i} g(n-1) g^*(n) \right] \leq \sum_{n=0}^{\infty} |g(n)|^2 \quad (3)$$

Using Eqs. (1) - (3), we thus have

$$\frac{\partial \epsilon}{\partial \rho_i} \geq 2(\rho_i - 1) \sum_{n=0}^{\infty} |g(n)|^2 \quad (4)$$

For  $\rho_i \geq 1$ , the right-hand side of Eq. (4) is always greater than or equal to zero. The derivative  $\partial \epsilon_f / (\partial \rho_i)$  is zero if, and only if,  $\rho_i = 1$  and  $g^*(n) = g(n-1)e^{j\theta_i}$  for  $n \geq 0$ . In such a case  $u(n)$  and  $g(n)$  are identically zero. Thus  $\partial \epsilon / (\partial \rho_i) \geq 1$  for  $\rho_i \geq 1$ , and so we conclude that all the zeros of the transfer function  $H_f(z)$  must lie inside the unit circle. That is,  $H_f(z)$  is minimum phase.

8.5 For forward linear prediction we have

(a) The data matrix is

$$\mathbf{A}^H = \begin{bmatrix} u(M) & u(M+1) & \dots & u(N-1) \\ u(M-1) & u(M) & \dots & u(N-2) \\ \vdots & \vdots & \dots & \vdots \\ u(1) & u(2) & \dots & u(N-M) \end{bmatrix}$$

The correlation matrix is  $\Phi = \mathbf{A}^H \mathbf{A}$

(b) The desired response vector is

$$\mathbf{d}^H = [u(M+1), u(M+2), \dots, u(N)]$$

The cross-correlation vector is  $\mathbf{A}^H \mathbf{d}$ .

(c) The minimum value of  $E_f$  is

$$E_{f, \min} = \mathbf{d}^H \mathbf{d} - \mathbf{d}^H (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{A}^H \mathbf{d}$$

8.6 For backward linear prediction we have

(a) The data matrix is

$$\begin{bmatrix} u^*(2) & u^*(3) & \dots & u^*(N-M+1) \\ u^*(3) & u^*(4) & \dots & u^*(N-M+2) \\ \vdots & \vdots & & \vdots \\ u^*(M+1) & u^*(M+2) & \dots & u^*(N) \end{bmatrix}$$

Note the difference between this data matrix and that for forward linear prediction.

The correlation matrix is

$$\Phi = \mathbf{A}^H \mathbf{A}$$

(b) The data vector is

$$\mathbf{d}^H = [u^*(1), u^*(2), \dots, u^*(N-M)]$$

The cross-correlation vector is  $\mathbf{A}^H \mathbf{d}$ .

(c) The minimum value of  $E_b$  is

$$E_{b, \min} = \mathbf{d}^H \mathbf{d} - \mathbf{d}^H (\mathbf{A}^H \mathbf{A})^{-1} \mathbf{d}$$

8.7 The data matrix is

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

The desired response vector is

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

The cost function is

$$E(\mathbf{w}) = \mathbf{d}^H \mathbf{d} - \mathbf{w}^H \mathbf{z} - \mathbf{z}^H \mathbf{w} + \mathbf{w}^H \Phi \mathbf{w}$$

where  $\mathbf{z}$  is the cross-correlation vector:

$$\mathbf{z} = \mathbf{A}^H \mathbf{d}$$



and  $\Phi$  is the correlation matrix

$$\Phi = \mathbf{A}^H \mathbf{A}$$

Differentiating the cost function with respect to the weight vector:

$$\frac{\partial E(\mathbf{w})}{\partial(\mathbf{w})} = -2\mathbf{z} + 2\Phi\mathbf{w}$$

Setting this result equal to zero and solving for the optimum  $\mathbf{w}$ , we get

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

$$8.8 \quad E_{\text{reg}} = \sum_{n=1}^K |\mathbf{w}^H \mathbf{u}(n)|^2 + \lambda(\mathbf{w}^H \mathbf{s}(\theta) - 1) + \delta \|\mathbf{w}\|^2$$

Taking the derivative of  $E_{\text{reg}}$  with respect to the weight vector  $\mathbf{w}$  and setting the result equal to zero

$$\frac{\partial E_{\text{reg}}}{\partial \mathbf{w}} = \sum_{n=1}^K 2\mathbf{w}^H \mathbf{u}(n) \mathbf{u}^H(n) + \lambda \mathbf{s}(\theta) + 2\delta \mathbf{w} = \mathbf{0}$$

Hence,

$$\hat{\mathbf{w}} = \frac{-\lambda \mathbf{s}(\theta)}{2 \left( \sum_{n=1}^K \mathbf{u}(n) \mathbf{u}^H(n) + \delta \mathbf{I} \right)}$$

$$\text{Since } \hat{\mathbf{w}}^H \mathbf{s}(\theta) = 1, \tag{1}$$

$$-\lambda = 2 \|\hat{\mathbf{w}}\|^2 \left( \sum_{n=1}^K \mathbf{u}(n) \mathbf{u}^H(n) + \delta \mathbf{I} \right) \tag{2}$$

By the virtue of Eq. (8.80) in the text, we may have

$$\Phi \hat{\mathbf{w}} = -\frac{1}{2} \lambda \mathbf{s}(\theta) \quad (3)$$

Use Eqs. (1) and (2) in (3) and rearrange terms, obtaining

$$\Phi = \sum_{n=1}^K \mathbf{u}(n) \mathbf{u}^H(n) + \delta \mathbf{I}$$

8.9 Starting with the cost function

$$E = \sum_{i=M+1}^N (|f_M(i)|^2 + |b_M(i)|^2)$$

$$f_M(i) = \mathbf{a}_M^H \mathbf{u}_{M+1}(i)$$

$$\mathbf{a}_M^H = [1, -w_1, -w_2, \dots, -w_M] = [1, -\mathbf{w}^T]$$

$$\mathbf{u}_{M+1}^H(i) = [u(i), u(i-1), \dots, u(i-M)]$$

$$b_M(i) = \mathbf{a}_M^H \mathbf{u}_{M+1}^B(i)$$

$$\mathbf{u}_{M+1}^{BH}(i) = [u(i-M), u(i-M+1), \dots, u(i)]$$

(a) Using these definitions, rewrite the cost function as

$$E = \sum_{i=M+1}^N (\mathbf{a}_M^H \mathbf{u}_{M+1}(i) \mathbf{u}_{M+1}^H(i) \mathbf{a}_M + \mathbf{a}_M^H \mathbf{u}_{M+1}^B(i) \mathbf{u}_{M+1}^{BH}(i) \mathbf{a}_M)$$

By definition,

$$\mathbf{a}_M = \begin{bmatrix} 1 \\ -\mathbf{w} \end{bmatrix} \text{ and } \mathbf{u}_{M+1}(i) = \begin{bmatrix} u(i) \\ \mathbf{u}_M(i-1) \end{bmatrix}$$

Hence,

$$E = \sum_{i=M+1}^N (|u(i)|^2 + \mathbf{w}^H \Phi_f \mathbf{w} - \theta_f^H \mathbf{w} - \mathbf{w}^H \theta_f + |u(i-M)|^2 + \mathbf{w}^H \Phi_b \mathbf{w} - \theta_b^H \mathbf{w} - \mathbf{w}^H \theta_b)$$

where

$$\Phi_f = \sum_{i=M+1}^N \mathbf{u}_M(i-1) \mathbf{u}_M^H(i-1)$$

$$\theta_f = \sum_{i=M+1}^N \mathbf{u}_M(i-1) u^*(i)$$

$$\Phi_b = \sum_{i=M+1}^N \mathbf{u}_M^{B*}(i) \mathbf{u}_M^{BH}(i)$$

$$\theta_b = \sum_{i=M+1}^N \mathbf{u}_M^B(i) u^*(i-M)$$

Setting

$$\frac{\partial E}{\partial \mathbf{w}} = 0$$

and solving for the optimum value of  $\mathbf{w}$ , we obtain

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

where

$$\Phi = \Phi_f + \Phi_b = \sum_{i=M+1}^N [\mathbf{u}_M(i-1) \mathbf{u}_M^H(i-1) + \mathbf{u}_M^{B*}(i) \mathbf{u}_M^{BH}(i)]$$

$$\theta = \theta_f + \theta_b = \sum_{i=M+1}^N [\mathbf{u}_M(i-1) u(i) + \mathbf{u}_M^{B*}(i) u(i-M)]$$

(b) Finally

$$E_{\min} = E_{f, \min} + E_{b, \min} = \sum_{i=M+1}^N [|u(i)|^2 + |u(i-M)|^2] - \theta^H \hat{\mathbf{w}}$$

$$(c) \quad \hat{\mathbf{a}} = \begin{bmatrix} 1 \\ -\hat{\mathbf{w}} \end{bmatrix}, \quad \Phi \hat{\mathbf{a}} = \begin{bmatrix} E_{\min} \\ \mathbf{0} \end{bmatrix}$$

$$\Phi = \Phi_f + \Phi_b = \sum_{i=M+1}^N [\mathbf{u}_M(i-1) \mathbf{u}_M^H(i-1) + \mathbf{u}_M^{B*}(i) \mathbf{u}_M^{BH}(i)] \quad (1)$$

Examining Eq. (1) and noting that

$$\mathbf{u}_M^H(i) = [u(i), u(i-1), \dots, u(i-M)]$$

$$\mathbf{u}_M^{BH}(i) = [u(i-M), u(i-M+1), \dots, u(i)]$$

we easily find that

$$\phi(M-k, M-t) = \phi^*(t, k), \quad 0 \leq (k, t) \leq M$$

$$\phi(k, t) = \phi^*(t, k), \quad 0 \leq (k, t) \leq M$$

8.10 The SIR maximization problem may be stated as follows:

$$\max_{\mathbf{w}} \left( \frac{\mathbf{w}^H \mathbf{S} \mathbf{S}^H \mathbf{w}}{\mathbf{w}^H \mathbf{R} \mathbf{w}} \right) \quad \text{subject to} \quad \mathbf{C}_{N-1}^H \mathbf{w} = \mathbf{f}_{N-1}$$

(a) Using Lagrang's method, the SIR maximization problem can be written as a minimization problem

$$\min J(\mathbf{w}) = \min \left\{ \frac{1}{2} \mathbf{w}^H \mathbf{R} \mathbf{w} + \lambda^H (\mathbf{C}_{N-1}^H \mathbf{w} - \mathbf{f}_{N-1}) \right\}$$

where  $\lambda^H = [\lambda_1 \dots \lambda_{N-1}]$  is the Lagrange vector ( $0 < \lambda_i < 1$ )

Set  $\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{0}$ , obtaining

That is,  $\mathbf{R}\mathbf{w} + \mathbf{C}_{N-1}^H \lambda = \mathbf{0}$

$$\mathbf{w}_{\text{opt}} = -\mathbf{R}^{-1} \mathbf{C}_{N-1} \lambda \quad (1)$$

Since  $\mathbf{C}_{N-1}^H \mathbf{w}_{\text{opt}} = \mathbf{f}_{N-1} = -\mathbf{C}_{N-1}^H \mathbf{R}^{-1} \mathbf{C}_{N-1} \lambda$

$$\lambda = -[\mathbf{C}_{N-1}^H \mathbf{R}^{-1} \mathbf{C}_{N-1}]^{-1} \mathbf{f}_{N-1} \quad (2)$$

Using Eq. (2) in (1):

$$\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1} \mathbf{C} [\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C}]^{-1} \mathbf{f}$$

(b) When  $\mathbf{w}_{\text{opt}} = \mathbf{R}^{-1} \mathbf{C} [\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C}]^{-1} \mathbf{f}$ , the SIR becomes

$$\begin{aligned} \frac{\mathbf{w}^H \mathbf{s} \mathbf{s}^H \mathbf{w}}{\mathbf{w}^H \mathbf{R} \mathbf{w}} &= \frac{\mathbf{w}^H \mathbf{s} \mathbf{s}^H \mathbf{w}}{\mathbf{f}^H [\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C}]^{-H} \mathbf{C}^H \mathbf{R}^{-H} \mathbf{C} [\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C}]^{-1} \mathbf{f}} \\ &= \frac{\mathbf{f}^H [\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C}]^{-H} \mathbf{C}^H \mathbf{R}^{-H} \mathbf{s} \mathbf{s}^H \mathbf{R}^{-1} \mathbf{C} [\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C}]^{-1} \mathbf{f}}{\mathbf{f}^H [\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C}]^{-H} \mathbf{C}^H \mathbf{R}^{-H} \mathbf{C} [\mathbf{C}^H \mathbf{R}^{-1} \mathbf{C}]^{-1} \mathbf{f}} \end{aligned}$$

(c) When there are no auxiliary constraints  $\mathbf{C}_{n-1} = 0$ , the fixed value  $\mathbf{f}_o$  will only determine the normalization of the solution yielding an unconditional maximum for SIR:

$$\mathbf{w}_{\text{opt}} = \alpha \mathbf{R}^{-1} \mathbf{s} \quad (\alpha \text{ is constant})$$

$$\text{SIR}_{\text{max}} = \frac{\mathbf{s}^H \mathbf{R}^{-H} \mathbf{s} \mathbf{s}^H \mathbf{R}^{-1} \mathbf{s}}{\mathbf{s}^H \mathbf{R}^{-H} \mathbf{s}} = \mathbf{s}^H \mathbf{R}^{-1} \mathbf{s}$$

$$(d) \hat{\mathbf{R}}(n) = \frac{1}{n} \sum_{i=1}^n \mathbf{u}(i) \mathbf{u}^H(i)$$

$$\hat{\mathbf{R}}(n) = \lambda \hat{\mathbf{R}}(n-1) + \mathbf{u}(n) \mathbf{u}^H(n)$$

where  $0 < \lambda \leq 1$ ; the parameter  $\lambda$  is a weighting factor, not to be confused with the Lagrange multiplier.

8.11 (a) We are given

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \\ 0.5 & 2 \end{bmatrix}$$

Therefore,

$$\begin{aligned} \mathbf{A}^T \mathbf{A} &= \begin{bmatrix} 1 & 0.5 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 0.5 & 2 \end{bmatrix} \\ &= \begin{bmatrix} 1.25 & 0 \\ 0 & 5 \end{bmatrix} \end{aligned}$$

This is a diagonal matrix. Hence, its eigenvalues are

$$\lambda_1 = 1.25$$

$$\lambda_2 = 5$$

The singular values of matrix  $\mathbf{A}$  are therefore

$$\sigma_1 = \sqrt{\lambda_1} = \sqrt{1.25} = \frac{\sqrt{5}}{2}$$

$$\sigma_2 = \sqrt{\lambda_2} = \sqrt{5}$$

The eigenvectors of  $\mathbf{A}^T \mathbf{A}$  are the right singular vectors of  $\mathbf{A}$ . For the problem at hand, the eigenvectors of  $\mathbf{A}^T \mathbf{A}$  constitute a unit matrix. We therefore have

$$\mathbf{V} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(b) The matrix product  $\mathbf{A} \mathbf{A}^T$  is

$$\mathbf{A} \mathbf{A}^T = \begin{bmatrix} 1 & -1 \\ 0.5 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0.5 \\ -1 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} 2 & -1.5 \\ -1.5 & 4.25 \end{bmatrix}$$

The eigenvalues of  $\mathbf{A}\mathbf{A}^T$  are the roots of the characteristic equation

$$(2 - \lambda)(4.25 - \lambda) - (1.5)^2 = 0$$

Expanding this equation:

$$\lambda^2 - 6.25\lambda + 6.25 = 0$$

Solving for the roots of this quadratic equation:

$$\lambda_1 = 1.25$$

$$\lambda_2 = 5$$

The singular values of the data matrix  $\mathbf{A}$  are therefore

$$\sigma_1 = \sqrt{\lambda_1} = \frac{\sqrt{5}}{2}$$

$$\sigma_2 = \sqrt{\lambda_2} = \sqrt{5}$$

which are identical to the values calculated in part (a).

To find the eigenvectors of  $\mathbf{A}\mathbf{A}^T$ , we note that

$$\begin{bmatrix} 2 & -1.5 \\ -1.5 & 4.25 \end{bmatrix} \begin{bmatrix} q_{11} \\ q_{12} \end{bmatrix} = 1.25 \begin{bmatrix} q_{11} \\ q_{12} \end{bmatrix}$$

Hence,

$$0.75 q_{11} - 1.5 q_{12} = 0$$

Equivalently,

$$q_{11} = 2 q_{12}$$

Setting  $q_{11}^2 + q_{12}^2 = 1$ , we find that

$$\mathbf{q}_{12} = \pm \frac{1}{\sqrt{5}} \quad \text{and} \quad q_{12} = \pm \frac{2}{\sqrt{5}}$$

We may thus write

$$\mathbf{q}_1 = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Similarly, we may show that

$$q_2 = \frac{1}{\sqrt{5}} \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

The eigenvectors of  $\mathbf{A}\mathbf{A}^T$  are the left singular vectors of  $\mathbf{A}$ . We may thus express the left singular vectors of matrix  $\mathbf{A}$  as

$$\mathbf{U} = \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix}$$

The pseudoinverse of the matrix  $\mathbf{A}$  is given by

$$\begin{aligned} \mathbf{A}^+ &= \mathbf{V} \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} \mathbf{U}^T \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{2}{\sqrt{5}} & 0 \\ 0 & \frac{1}{\sqrt{5}} \end{bmatrix} \frac{1}{\sqrt{5}} \begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix} \\ &= \frac{1}{5} \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & -1 \end{bmatrix} \\ &= \frac{1}{5} \begin{bmatrix} 2 & 4 \\ 2 & -1 \end{bmatrix} \end{aligned}$$

8.12 Given the 2 x 2 complex matrix



$$\mathbf{A} = \begin{pmatrix} 1+j & 1+0.5j \\ 0.5-j & 1-j \end{pmatrix}$$

we have

$$\mathbf{A}^H \mathbf{A} = \begin{pmatrix} 3.25 & 3 \\ 3 & 3.25 \end{pmatrix}, \quad \mathbf{A} \mathbf{A}^H = \begin{pmatrix} 3.25 & 3j \\ -3j & 3.25 \end{pmatrix}$$

The eigenvalues and eigenvectors of matrix  $\mathbf{A}$  are found in the usual way by first solving

$$|\mathbf{A} - \lambda \mathbf{I}| = 0$$

for the eigenvalues  $\lambda_i$  for  $i = 1, 2$ , and then using these values in

$$\mathbf{A} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \lambda_i \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

to determine the eigenvectors  $\mathbf{x}$ .

The two eigenvalues in descending order, for both  $\mathbf{A}^H \mathbf{A}$  and  $\mathbf{A} \mathbf{A}^H$  are  $\lambda_1 = 6.25$  and  $\lambda_2 = 0.25$ . Regarding the eigenvectors, we have for  $\mathbf{A}^H \mathbf{A}$

$$\mathbf{V} = \begin{pmatrix} x & y \\ x & -y \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

where  $x$  and  $y$  are arbitrary real numbers, and we arbitrarily make the simplest choice that gives us an orthonormal set of vectors. For  $\mathbf{A} \mathbf{A}^H$  we similarly have

$$\mathbf{U} = \begin{pmatrix} x & y \\ jx & -jy \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} j & 1 \\ 1 & j \end{pmatrix}$$

for another arbitrary choice of  $x$  and  $y$ .

The SVD of  $\mathbf{A}$  is given by

$$\mathbf{A} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^H$$

where the right singular matrix  $\mathbf{V}$  is the eigen-matrix of  $\mathbf{A}^H\mathbf{A}$  and the left singular matrix  $\mathbf{U}$  is the eigen-matrix of  $\mathbf{A}\mathbf{A}^H$ . The singular values should be the square roots of the common eigenvalues:  $\sigma_1 = \sqrt{\lambda_1} = 2.5$  and  $\sigma_2 = 0.5$  with

$$\Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$$

Interestingly enough

$$\mathbf{U}\Sigma\mathbf{V}^H = \mathbf{B} = \begin{pmatrix} 1.25 + j0.25 & -1.25 + j0.25 \\ 0.25 + j1.25 & 0.25 - j1.25 \end{pmatrix}$$

where, obviously  $\mathbf{B} \neq \mathbf{A}$ , but,  $\mathbf{B}^H\mathbf{B} = \mathbf{A}^H\mathbf{A}$  and  $\mathbf{B}\mathbf{B}^H = \mathbf{A}\mathbf{A}^H$ . In order to have  $\mathbf{B} = \mathbf{A}$ , the arbitrariness in choosing the eigenvectors of  $\mathbf{A}^H\mathbf{A}$  and  $\mathbf{A}\mathbf{A}^H$  above, should be removed. The natural constraint relation is that  $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^H$  holds true.

Starting from that premise, we take  $\mathbf{V}$  as is and define  $\mathbf{U}$  using

$$\mathbf{U} = \mathbf{A}\mathbf{V}\Sigma^{-1} = \begin{pmatrix} 0.5657 + j0.4243 & j0.7071 \\ 0.4243 - j0.5657 & -0.7071 \end{pmatrix}$$

We can easily verify that the new matrix  $\mathbf{U}$  is indeed composed of eigenvectors of  $\mathbf{A}\mathbf{A}^H$  determined above. Clearly, the new  $\mathbf{U}$  is related to the old one by some similarity transformation

$$\mathbf{U}_{\text{new}} = \mathbf{W}^H \mathbf{U}_{\text{old}} \mathbf{W}$$

where  $\mathbf{W}$  is a unitary matrix.

This problem illustrates the need for a less arbitrary procedure for determining the SVD of a matrix, even in the simplest of cases, with a 2 x 2 matrix.

8.13 We are given

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 1 & 2 \\ -1 & 1 \end{bmatrix}$$

We note that

$$\mathbf{A}^T = \begin{bmatrix} 2 & 1 & -1 \\ 3 & 2 & 1 \end{bmatrix}$$

Next, we set up

$$\begin{aligned} \mathbf{A}^T \mathbf{A} &= \begin{bmatrix} 2 & 1 & -1 \\ 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \\ 1 & 2 \\ -1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 6 & 7 \\ 7 & 14 \end{bmatrix} \end{aligned}$$

The eigenvalues of  $\mathbf{A}^T \mathbf{A}$  are roots of the characteristic equation:

$$(6-\lambda)(14-\lambda) - (7)^2 = 0$$

or,

$$\lambda^2 - 20\lambda + 35 = 0$$

Solving for the roots of this equation, we thus get

$$\lambda = 10 \pm \sqrt{65}$$

That is,

$$\lambda_1 = 10 - \sqrt{65} \approx 1.94, \quad \sigma_1 = \sqrt{\lambda_1} = 1.393$$

$$\lambda_2 = 10 + \sqrt{65} \approx 18.06, \quad \sigma_2 = \sqrt{\lambda_2} = 4.25$$

To find the eigenvectors of  $\mathbf{A}^T \mathbf{A}$ , we note that

$$\begin{bmatrix} 6 & 7 \\ 7 & 14 \end{bmatrix} \begin{bmatrix} q_{11} \\ q_{12} \end{bmatrix} = 1.94 \begin{bmatrix} q_{11} \\ q_{12} \end{bmatrix}$$

Hence,

$$4.06 q_{11} + 7 q_{12} = 0$$

or

$$q_{12} = -0.58 q_{11}$$

We also note that

$$q_{11}^2 + q_{12}^2 = 1$$

Therefore,

$$q_{11}^2 + (0.58)^2 q_{11}^2 = 1$$

or

$$q_{11} = \pm \frac{1}{\sqrt{1.336}} = \pm 0.87$$

Hence,

$$q_{12} = \mp 0.505$$

Similarly, we may show that the eigenvector associated with  $\lambda_2 = 18.06$  is defined by

$$q_{21} = -0.505$$

$$q_{22} = -0.87$$

Therefore, the right singular vectors of the data matrix  $\mathbf{A}$  constitute the matrix

$$\mathbf{V} = \begin{bmatrix} 0.87 & -0.505 \\ -0.505 & -0.87 \end{bmatrix} \quad (1)$$

Next, we set up

$$\mathbf{A}\mathbf{A}^T = \begin{bmatrix} 2 & 3 \\ 1 & 2 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & -1 \\ 3 & 2 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 13 & 8 & 1 \\ 8 & 5 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

The eigenvalues of  $\mathbf{A}\mathbf{A}^T$  are the roots of the third-order characteristic equation:

$$\begin{bmatrix} 13-\lambda & 8 & 1 \\ 8 & 5-\lambda & 1 \\ 1 & 1 & 2-\lambda \end{bmatrix} = 0$$

or

$$\lambda(\lambda^2 - 20\lambda + 25) = 0$$

The eigenvalues of  $\mathbf{A}\mathbf{A}^T$  are therefore as follows

$$\lambda_0 = 0$$

$$\lambda_1 \approx 1.94$$

$$\lambda_2 \approx 18.06$$

The two nonzero eigenvalues of  $\mathbf{A}\mathbf{A}^T$  are the same as those found for  $\mathbf{A}^T\mathbf{A}$ . To find the eigenvectors of  $\mathbf{A}\mathbf{A}^T$ , we note that for  $\lambda_1 = 1.94$ :

$$\begin{bmatrix} 13 & 8 & 1 \\ 8 & 5 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} q_{11} \\ q_{12} \\ q_{13} \end{bmatrix} = 1.94 \begin{bmatrix} q_{11} \\ q_{12} \\ q_{13} \end{bmatrix}$$

$$11.06 q_{11} + 8 q_{12} + q_{13} = 0$$

$$8 q_{11} + 3.06 q_{12} + q_{13} = 0$$

$$q_{11} + q_{12} + 0.06 q_{13} = 0$$

Using the first two equations to eliminate  $q_{13}$ :

$$3.06 q_{11} + 4.94 q_{12} = 0$$

Hence,

$$q_{12} \approx -0.619q_{11}$$

Using the first and third equations to eliminate  $q_{12}$ :

$$3.06 q_{11} + 0.52 q_{13} = 0$$

Hence,

$$q_{13} \approx -5.88q_{11}$$

Next, we note that

$$q_{11}^2 + q_{12}^2 + q_{13}^2 = 1$$

Therefore,

$$q_{11}^2 + 0.383q_{11}^2 + 34.574q_{11}^2 = 1$$

$$36q_{11}^2 \approx 1$$

$$q_{11} \approx \pm 0.167$$

Correspondingly,

$$q_{12} \approx \mp 0.109$$

$$q_{13} \approx \mp 0.98$$

We may thus set

$$\mathbf{q}_1 \approx \begin{bmatrix} 0.167 \\ -0.109 \\ -0.981 \end{bmatrix} \tag{2}$$

For the eigenvalue  $\lambda_2 = 18.06$ , we may write

$$\begin{bmatrix} 13 & 8 & 1 \\ 8 & 5 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} q_{21} \\ q_{22} \\ q_{23} \end{bmatrix} = 18.06 \begin{bmatrix} q_{21} \\ q_{22} \\ q_{23} \end{bmatrix}$$

$$\begin{aligned}
-5.06 q_{21} + 8 q_{22} + q_{23} &= 0 \\
8 q_{21} - 13.06 q_{22} + q_{23} &= 0 \\
q_{21} + q_{22} - 16.06 q_{23} &= 0
\end{aligned}$$

Using the first two equations to eliminate  $q_{23}$ :

$$\begin{aligned}
-13.06 q_{21} + 21.06 q_{22} &= 0 \\
q_{22} &= 0.62 q_{21}
\end{aligned}$$

Using the first and third equations to eliminate  $q_{22}$ :

$$\begin{aligned}
-13.06 q_{21} + 129.48 q_{23} &= 0 \\
q_{23} &\approx 0.101 q_{21}
\end{aligned}$$

We next note that

$$q_{21}^2 + q_{22}^2 + q_{23}^2 = 1$$

We therefore obtain

$$\begin{aligned}
q_{21}^2 + 0.384 q_{21}^2 + 0.01 q_{21}^2 &= 1 \\
q_{21} &\approx \pm 0.845
\end{aligned}$$

Hence,

$$\begin{aligned}
q_{22} &\approx \pm 0.524 \\
q_{23} &\approx \pm 0.085
\end{aligned}$$

We may thus set

$$\mathbf{q}_2 = \begin{bmatrix} -0.845 \\ -0.524 \\ -0.0845 \end{bmatrix} \quad (3)$$

Note that  $\mathbf{q}_2$  is orthogonal to  $\mathbf{q}_1$ , as it should be; that is,

$$\mathbf{q}_1^T \mathbf{q}_2 = 0.$$

To complete the eigencomputation, we need to determine the eigenvector associated with the zero eigenvalue of matrix  $\mathbf{A}\mathbf{A}^T$ . Here we note

$$\begin{bmatrix} 13 & 8 & 1 \\ 8 & 5 & 1 \\ 1 & 1 & 2 \end{bmatrix} \mathbf{q}_0 = \mathbf{0}, \quad \lambda_0 = 0$$

where  $\|\mathbf{q}_0\|^2 = 1$ . Solving for  $\mathbf{q}_0$ , we get

$$\mathbf{q}_0 = \begin{bmatrix} 0.5071 \\ -0.8452 \\ 0.169 \end{bmatrix} \quad (4)$$

Putting together Eqs. (2) through (4) for the eigenvectors  $\mathbf{q}_1$ ,  $\mathbf{q}_2$ , and  $\mathbf{q}_0$ , we may express the left singular vectors of the data matrix  $\mathbf{A}$  as the matrix

$$\mathbf{U} = \begin{bmatrix} 0.167 & 0.845 & 0.5071 \\ -0.109 & 0.524 & -0.8452 \\ -0.981 & 0.088 & 0.169 \end{bmatrix}$$

where the third column corresponds to the zero eigenvalue of  $\mathbf{A}\mathbf{A}^T$ . The singular value decomposition of the data matrix  $\mathbf{A}$  may therefore be expressed as

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} 0.167 & -0.845 & 0.5071 \\ -0.198 & -0.524 & -0.8452 \\ -0.98 & -0.085 & 0.169 \end{bmatrix} \begin{bmatrix} 1.393 & 0 \\ 0 & 4.25 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0.87 & -0.505 \\ -0.505 & -0.87 \end{bmatrix} \\ &= \begin{bmatrix} 0.167 & -0.845 \\ -0.109 & -0.524 \\ -0.98 & -0.085 \end{bmatrix} \begin{bmatrix} 1.393 & 0 \\ 0 & 4.25 \end{bmatrix} \begin{bmatrix} 0.87 & -0.505 \\ -0.505 & -0.87 \end{bmatrix} \end{aligned}$$

As a check, carrying out the matrix multiplication given here, we get

$$\mathbf{A} = \begin{bmatrix} 2.016 & 3.0069 \\ 0.9925 & 2.0142 \\ -1.0065 & 1.0044 \end{bmatrix}$$



which is very close to the original data matrix.

(a) The pseudoinverse of matrix  $\mathbf{A}$  is

$$\mathbf{A}^+ = \sum_{i=1}^2 \frac{1}{\sigma_i} \mathbf{v}_i \mathbf{u}_i^T$$

(b) The least-squares weight vector is

$$\begin{aligned} \hat{\mathbf{w}} &= \mathbf{A}^+ \mathbf{d} \\ &= \sum_{i=1}^2 \frac{(\mathbf{u}_i^T \mathbf{d})}{\sigma_i} \mathbf{v}_i \\ &= \frac{(\mathbf{u}_1^T \mathbf{d})}{\sigma_1} \mathbf{v}_1 + \frac{(\mathbf{u}_2^T \mathbf{d})}{\sigma_2} \mathbf{v}_2 \\ &= \frac{1}{1.393} \begin{bmatrix} 0.167 & 0.109 & -0.981 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 1/24 \end{bmatrix} \begin{bmatrix} 0.87 \\ -0.505 \end{bmatrix} \\ &\quad + \frac{1}{4.250} \begin{bmatrix} 0.845 & 0.524 & 0.085 \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 1/34 \end{bmatrix} \begin{bmatrix} 0.505 \\ 0.87 \end{bmatrix} \\ &= \begin{bmatrix} 0.3859 \\ 0.3859 \end{bmatrix} \end{aligned}$$

8.14 First we set up the following identifications:

	Least-squares	Normalized LMS
Data matrix	$\mathbf{A}$	$\mathbf{x}^H(n)$
Desired data vector	$\mathbf{d}$	$e^*(n)$
Parameter vector	$\mathbf{w}$	$\mathbf{c}(n+1)$
Eigenvalue	$\sigma_i^2$	$\ \mathbf{x}(n)\ ^2$
Eigenvector	$\mathbf{u}_i$	1

Hence, the application of the linear least-squares solution yields

$$\mathbf{c}(n+1) = \frac{1}{\|\mathbf{x}(n)\|^2} \mathbf{x}(n) e^*(n)$$

That is to say,

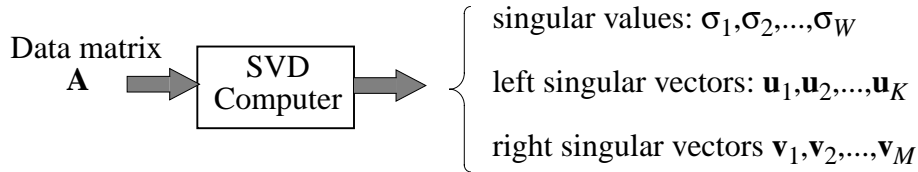
$$\delta \hat{\mathbf{w}}(n+1) = \frac{1}{\|\mathbf{u}(n)\|^2 + a} \mathbf{u}(n) e^*(n)$$

Correspondingly, we may write

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \frac{\mu}{\|\mathbf{u}(n)\|^2 + a} \mathbf{u}(n) e^*(n)$$

where  $\mu$  is the step-size parameter.

8.15 We are given an SVD computer that may be pictured as follows:



$W$ : rank of data matrix  $\mathbf{A}$

$K$ : number of rows of  $\mathbf{A}$

$M$ : number of columns of  $\mathbf{A}$

The MVDR spectrum is defined by

$$S_{\text{MVDR}}(\omega) = \frac{1}{\mathbf{s}^H(\omega) \Phi^{-1} \mathbf{s}(\omega)} \quad (1)$$

where  $\mathbf{s}(\omega)$  is the steering vector, and  $\Phi^{-1}$  is the inverse of the correlation matrix  $\Phi$ . Specifically,  $\Phi$  is defined in terms of the data matrix  $\mathbf{A}$  by

$$\Phi = \mathbf{A}^H \mathbf{A}$$

where

$$\mathbf{A} = \sum_{i=1}^W \sigma_i \mathbf{u}_i \mathbf{v}_i^H$$

$$\mathbf{A}^H = \sum_{i=1}^W \sigma_i \mathbf{v}_i \mathbf{u}_i^H$$

That is,

$$\Phi = \sum_{i=1}^W \sigma_i^2 \mathbf{v}_i \mathbf{v}_i^H, \quad \|\mathbf{u}_i\|^2 = 1 \text{ for all } i$$

Correspondingly, we may express the inverse matrix  $\Phi^{-1}$  as

$$\Phi^{-1} = \sum_{i=1}^W \frac{1}{\sigma_i^2} \mathbf{v}_i \mathbf{v}_i^H \quad (2)$$

Hence, the denominator of the MVDR spectrum may be expressed as

$$\begin{aligned} \mathbf{s}^H(\omega) \Phi^{-1} \mathbf{s}(\omega) &= \sum_{i=1}^W \frac{1}{\sigma_i^2} \mathbf{s}^H(\omega) \mathbf{v}_i \mathbf{v}_i^H \mathbf{s}(\omega) \\ &= \sum_{i=1}^W |z_i(\omega)|^2 \end{aligned} \quad (3)$$

where  $z_i(\omega)$  is a frequency-dependent scalar that is defined by the inner product:

$$z_i(\omega) = \frac{1}{\sigma_i} \mathbf{s}^H(\omega) \mathbf{v}_i, \quad i = 1, 2, \dots, W \quad (4)$$

Accordingly, Eq. (2) may be rewritten as

$$S_{\text{MVDR}}(\omega) = \frac{1}{\sum_{i=1}^W |z_i(\omega)|^2} \quad (5)$$

Thus, to evaluate the MVDR spectrum, we may proceed as follows:

- Compute the SVD of the data matrix  $\mathbf{A}$ .
- Use the right-singular vectors  $\mathbf{v}_i$  and the corresponding singular values  $\sigma_i$  in Eq. (4) to evaluate  $z_i(\omega)$  for  $i = 1, 2, \dots, W$ , where  $W$  is the rank of  $\mathbf{A}$ .
- Use Eq. (5) to evaluate the MVDR spectrum.

## CHAPTER 9

9.1 Assume that

$$\beta(n, i) = \lambda(i)\beta(n, i-1), \quad i = 1, \dots, n$$

Hence, for  $i = n$ :

$$\beta(n, n) = \lambda(n)\beta(n, n-1)$$

Since  $\beta(n, n) = 1$ , we have

$$\lambda^{-1}(n) = \beta(n, n-1)$$

Next, for  $i = n-1$ ,

$$\beta(n, n-1) = \lambda(n-1)\beta(n, n-2)$$

or equivalently,

$$\begin{aligned} \beta(n, n-2) &= \lambda^{-1}(n-1)\beta(n, n-1) \\ &= \lambda^{-1}(n-1)\lambda^{-1}(n) \end{aligned}$$

Proceeding in this way, we may thus write

$$\begin{aligned} \beta(n, i) &= \lambda^{-1}(i+1)\dots\lambda^{-1}(n-1)\lambda^{-1}(n) \\ &= \prod_{k=i+1}^n \lambda^{-1}(k) \end{aligned}$$

For  $\beta(n, i)$  to equal  $\lambda^{n-i}$ , we must have

$$\lambda^{-1}(i+1)\dots\lambda^{-1}(n-1)\lambda^{-1}(n) = \lambda^{n-i}$$

This requirement is satisfied by choosing

$$\lambda(k) = \lambda^{-1} \quad \text{for all } k$$

We thus find that

$$\begin{aligned}\beta(n, i) &= \overbrace{\lambda \dots \lambda}^{n-i \text{ terms}} \lambda \\ &= \lambda^{n-i}\end{aligned}$$

9.2 The matrix inversion lemma states that if

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

then

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B} \quad (2)$$

To prove this lemma, we multiply Eq. (1) by (2):

$$\begin{aligned}\mathbf{A}\mathbf{A}^{-1} &= (\mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^H)[\mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B}] \\ &= \mathbf{B}^{-1}\mathbf{B} - \mathbf{B}^{-1}\mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B} \\ &\quad - \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^H\mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^H\mathbf{B}\end{aligned}$$

We have to show that  $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$ . Since  $(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C}) = \mathbf{I}$ , and  $\mathbf{B}^{-1}\mathbf{B} = \mathbf{I}$ , we may rewrite this result as

$$\begin{aligned}\mathbf{A}\mathbf{A}^{-1} &= \mathbf{I} - \mathbf{C}(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B} \\ &\quad + \mathbf{C}\mathbf{D}^{-1}(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B} \\ &\quad - \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^H\mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B} \\ &= \mathbf{I} - [\mathbf{C} - \mathbf{C}\mathbf{D}^{-1}(\mathbf{D} + \cancel{\mathbf{C}^H\mathbf{B}\mathbf{C}}) - \mathbf{C}\mathbf{D}^{-1}\cancel{\mathbf{C}^H\mathbf{B}\mathbf{C}}] \cdot (\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B} \\ &= \mathbf{I} - (\mathbf{C} - \mathbf{C}\mathbf{D}^{-1}\mathbf{D})(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B}\end{aligned}$$

Since  $\mathbf{D}^{-1}\mathbf{D} = \mathbf{I}$ , the second term in this last line is zero; hence,

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

which is the desired result.

9.3 We are given

$$\Phi(n) = \delta \mathbf{I} + \mathbf{u}(n)\mathbf{u}^H(n) \quad (1)$$

Let

$$\mathbf{A} = \mathbf{B}^{-1} + \mathbf{C}\mathbf{D}^{-1}\mathbf{C}^H \quad (2)$$

Then, according to the matrix inversion lemma:

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{B}\mathbf{C}(\mathbf{D} + \mathbf{C}^H\mathbf{B}\mathbf{C})^{-1}\mathbf{C}^H\mathbf{B} \quad (3)$$

From Eqs. (1) and (2), we note:

$$\mathbf{A} = \Phi(n)$$

$$\mathbf{B}^{-1} = \delta \mathbf{I}$$

$$\mathbf{C} = \mathbf{u}(n)$$

$$\mathbf{D} = \mathbf{I}$$

Hence, using Eq. (3):

$$\begin{aligned} \Phi^{-1}(n) &= \frac{1}{\delta} \mathbf{I} - \frac{1}{\delta^2} \mathbf{u}(n) \left( 1 + \frac{1}{\delta} \mathbf{u}^H(n) \mathbf{u}(n) \right)^{-1} \mathbf{u}^H(n) \\ &= \frac{1}{\delta} \left( \frac{\mathbf{u}(n) \mathbf{u}^H(n)}{\delta + \mathbf{u}^H(n) \mathbf{u}(n)} \right) \\ &= \frac{1}{\delta} \left( \frac{\delta \mathbf{I}}{\delta + \mathbf{u}^H(n) \mathbf{u}(n)} \right) \\ &= \left( \frac{1}{\delta + \mathbf{u}^H(n) \mathbf{u}(n)} \right) \mathbf{I} \end{aligned}$$

9.4 From Section 9.6, we have

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

Define

$$\varepsilon(n) = w_o - \hat{w}(n)$$

We may thus write

$$\varepsilon(n) = \varepsilon(n-1) - k(n)\xi^*(n) \quad (2)$$

Since

$$\xi(n) = d(n) - \hat{w}^*(n-1)u(n)$$

$$e_o(n) = d(n) - w_o^*u(n)$$

We may expand Eq. (2) as follows

$$\begin{aligned} \varepsilon(n) &= \varepsilon(n-1) - k(n)(d(n) - \hat{w}^*(n-1)u(n))^* \\ &= \varepsilon(n-1) - k(n)(d(n) - w_o^*u(n) - \hat{w}^*(n-1)u(n))^* \\ &= \varepsilon(n-1) - k(n)(e_o(n) + \varepsilon^*(n-1)u(n))^* \\ &= \varepsilon(n-1) - k(n)e_o^*(n) - k(n)u^*(n)\varepsilon(n-1) \\ &= 1 - k(n)u^*(n)\varepsilon(n-1) - k(n)e_o^*(n) \end{aligned}$$

Hence,

$$a(n) = 1 - k(n)u^*(n) < 1, \text{ provided that } 0 \leq |k(n)u^*(n)| \leq 1.$$

Then under this condition,  $\varepsilon(n)$  is guaranteed to decrease with increasing  $n$ . The convergence process is perturbed by the white noise  $e_o(n)$ .

9.5 From Eq. (9.25) of the text:

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



where we have

$$\begin{aligned}\hat{\mathbf{w}}(1) &= \hat{\mathbf{w}}(0) + \mathbf{k}(1)\xi^*(1) \\ \hat{\mathbf{w}}(2) &= \hat{\mathbf{w}}(1) + \mathbf{k}(2)\xi^*(1) \\ &\vdots \\ \hat{\mathbf{w}}(n) &= \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\xi^*(n)\end{aligned}$$

Summing the above  $n$  equations, we have

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(0) + \sum_{i=1}^n \mathbf{k}(i)\xi^*(i)$$

Hence

$$\begin{aligned}\varepsilon(n) &= \mathbf{w}_o - \hat{\mathbf{w}}(n) \\ &= \mathbf{w}_o - \hat{\mathbf{w}}(0) - \sum_{i=1}^n \mathbf{k}(i)\xi^*(i) \\ &= \varepsilon(0) - \sum_{i=1}^n \mathbf{k}(i)\xi^*(i)\end{aligned}$$

9.6 The a posteriori estimation error  $e(n)$  and the a priori estimation error  $\xi(n)$  are related by

$$e(n) = \gamma(n)\xi(n)$$

where  $\gamma(n)$  is defined by [see Eq. (9.42) of the text]

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

Hence, for  $n = 1$  we have

$$e(1) = \gamma(1)\xi(1)$$

where

$$\begin{aligned}
\gamma(1) &= \frac{1}{1 + \lambda^{-1} \mathbf{u}^H(1) \Phi^{-1}(0) \mathbf{u}(1)} \\
&= \frac{1}{1 + \lambda^{-1} \delta^{-1} \mathbf{u}^H(1) \mathbf{u}(1)}, & \Phi^{-1}(0) &= \delta^{-1} \mathbf{I} \\
&= \frac{\lambda \delta}{\lambda \delta + |u(1)|^2}, & \mathbf{u}(1) &= \begin{bmatrix} u(1) \\ \mathbf{0} \end{bmatrix} \\
&\approx \frac{\lambda \delta}{|u(1)|^2}, & \delta &\ll 1
\end{aligned}$$

Hence,  $\gamma(1)$  is small.

Next, we observe that

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

For  $n = 1$ :

$$\begin{aligned}
\xi(1) &= d(1) - \hat{\mathbf{w}}^H(0) \mathbf{u}(1) \\
&= d(1), & \hat{\mathbf{w}}(0) &= \mathbf{0}
\end{aligned}$$

We thus conclude that  $e(1)$  equals a very small quantity namely  $\gamma(1)$ , multiplied by  $d(1)$ , which, in turn, makes  $e(1)$  small.

As  $n$  increases,  $\gamma(n)$  approaches towards the final value of 1. Correspondingly, the mean-square value of the estimation error  $e(n)$  increases towards its steady-state value,  $J_{\min}$ .

$$\begin{aligned}
9.7 \quad (a) \quad E[\varepsilon(n)] &= E[\mathbf{w}_o - \hat{\mathbf{w}}(n)] \\
&= \mathbf{w}_o - E[\hat{\mathbf{w}}(n)]
\end{aligned}$$

From Eq. (9.56) of the text we know that, for large  $n$ ,

$$E[\hat{\mathbf{w}}(n)] = \mathbf{w}_o - \frac{\delta}{n} \mathbf{p}$$

Therefore,

$$E[\varepsilon(n)] = \frac{\delta}{n} \mathbf{p}$$

$$E[\varepsilon(n-1)] = \frac{\delta}{n-1} \mathbf{p}$$

Hence,

$$\begin{aligned} E[\varepsilon(n)] - E[\varepsilon(n-1)] \\ = \frac{-\delta}{n(n-1)} \mathbf{p} = -\frac{1}{n} E[\varepsilon(n-1)] \end{aligned}$$

Rearranging terms:

$$E[\varepsilon(n)] \approx \left(1 - \frac{1}{n}\right) E[\varepsilon(n-1)] \quad (1)$$

- (b) Compared to the self-orthogonalizing adaptive filter discussed in Section 7.4, we can see from Eq. (7.36) that

$$E[\varepsilon(n)] = (1 - \alpha) E[\varepsilon(n-1)] \quad (2)$$

where the constant  $\alpha$  lies in the range  $0 < \alpha < 1$ . For  $\alpha = 1/n$ , Eqs. (1) and (2) are the same. This is not surprising because in RLS filter, the step-size  $\mu$  is the inverse correlation matrix  $\Phi^{-1}(n)$  whereas in self-orthogonalizing filter, step-size is  $\alpha \mathbf{R}^{-1}$  where  $\mathbf{R}$  is the correlation matrix.

- 9.8 (a) From Appendix G, using the property of complex Wishart distribution, we have: for

any given nonzero  $\alpha$  in  $R^M$ , we know that  $\frac{\alpha^H \mathbf{R}^{-1} \alpha}{\alpha^H \Phi^{-1}(n) \alpha}$  is  $\chi^2$  (chi-square) distributed with  $n-M+1$  degrees of freedom. Hence,

$$\begin{aligned} E[\alpha^H \Phi^{-1}(n) \alpha] &= \alpha \mathbf{R}^{-1} \alpha E\left[\frac{1}{\chi^2(n-M+1)}\right] \\ &= \frac{1}{n-M+1} \alpha^H \mathbf{R}^{-1} \alpha \quad (n > M+1) \end{aligned}$$

which further implies that

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

Usually  $n \gg M-1$ . Hence

$$E[\Phi^{-1}(n)] \approx \frac{1}{n} \mathbf{R}^{-1}$$

(b) By definition  $D(n) = E[\varepsilon^H(n)\varepsilon(n)]$ , and  $\mathbf{K}(n) = E[\varepsilon(n)\varepsilon^H(n)]$ ; thus

$$D(n) = \text{tr}[\mathbf{K}(n)]$$

From Eq. (9.58), we may obtain

$$\mathbf{K}(n) = \sigma_o^2 E[\Phi^{-1}(n)] = \frac{1}{n} \sigma_o^2 \mathbf{R}^{-1}$$

Hence,

$$D(n) = \frac{1}{n} \sigma_o^2 \text{tr}[\mathbf{R}^{-1}]$$

$$= \frac{1}{n} \sigma_o^2 \sum_{i=1}^M \frac{1}{\lambda_i}$$

9.9 Consider the difference:

$$X(n) = \varepsilon^H(n)\Phi(n) - \varepsilon^H(n-1)\Phi(n-1) \quad (1)$$

From RLS theory:

$$\varepsilon(n) = \varepsilon(n-1) - \Phi^{-1}(n)\mathbf{u}(n)\xi^*(n) \quad (2)$$

We may therefore rewrite Eq. (1) as

$$\begin{aligned} X(n) &= [\varepsilon(n-1) - \Phi^{-1}(n)\mathbf{u}(n)\xi^*(n)]^H \Phi(n) - \varepsilon^H(n-1)\Phi(n-1) \\ &= \varepsilon^H(n-1)[\Phi(n) - \Phi(n-1)] - \varepsilon(n-1) \end{aligned}$$

$$\begin{aligned}
& -\xi(n)\mathbf{u}^H(n)\Phi^{-1}(n)\Phi(n) \in (n-1) \\
& -\xi^*(n) \in^H (n-1)\Phi(n)\Phi^{-1}(n)\mathbf{u}(n) \\
& +|\xi(n)|^2\mathbf{u}^H(n)\Phi^{-1}(n)\Phi(n)\Phi^{-1}(n)\mathbf{u}(n) \\
& = \in^H (n-1)[\Phi(n) - \Phi^{-1}(n-1)] \in (n-1) \\
& -\xi(n)\mathbf{u}^H(n) \in (n-1) - \xi^*(n) \in^H (n-1)\mathbf{u}(n) \\
& +|\xi(n)|^2\mathbf{u}^H(n)\Phi(n)\mathbf{u}(n)
\end{aligned} \tag{3}$$

Moreover, from RLS theory:

$$\Phi(n) = \Phi(n-1) + \mathbf{u}(n)\mathbf{u}^H(n), \quad \lambda = 1 \tag{4}$$

$$r^{-1}(n) = 1 + \mathbf{u}^H(n)\Phi^{-1}(n)\mathbf{u}(n) \quad \lambda = 1 \tag{5}$$

Hence, using Eqs. (4) and (5) in (3) yields

$$\begin{aligned}
X(n) &= \in^H (n-1)\mathbf{u}(n)\mathbf{u}^H(n) \in (n-1) \\
& -\xi(n)\mathbf{u}^H(n) \in (n-1) - \xi^*(n) \in^H (n-1)\mathbf{u}(n) \\
& +|\xi(n)|^2(1 - r^{-1}(n))
\end{aligned} \tag{6}$$

Again, from RLS theory:

$$\in^H (n-1)u(n) = \xi_u(n) \tag{7}$$

$$v(n) = \xi(n) - \xi_u(n) \tag{8}$$

Hence, substituting Eqs. (7) and (8) into Eq. (6), we finally get

$$X(n) = |\xi_u(n)|^2 - \xi(n)\xi_u^*(n) - \xi^*(n)\xi_u(n) + |\xi(n)|^2(1 - r^{-1}(n))$$

$$\begin{aligned}
&= \left| \xi(n) - \xi_u(n) \right|^2 - r^{-1}(n) |\xi(n)|^2 \\
&= |v(n)|^2 - r^{-1}(n) |\xi(n)|^2
\end{aligned} \tag{9}$$

Recalling the definition of  $X(n)$  given in Eq. (1), we may thus write

$$\epsilon^H(n) \Phi(n) \in (n) - \epsilon^H(n-1) \Phi(n-1) \in (n-1) = |v(n)|^2 - r^{-1}(n) |\xi(n)|^2$$

or equivalently

$$\epsilon^H(n) \Phi(n) \in (n) + \frac{|\xi(n)|^2}{r(n)} = \epsilon^H(n-1) \Phi(n-1) \in (n-1) + |v(n)|^2$$

which is the desired result.

## CHAPTER 10

10.1 Let

$$\alpha(1) = \mathbf{y}(1) \tag{1}$$

$$\alpha(2) = \mathbf{y}(2) + \mathbf{A}_{1,1}\mathbf{y}(1) \tag{2}$$

where the matrix  $\mathbf{A}_{1,1}$  is to be determined. This matrix is chosen so as to make the innovations processes  $\alpha(1)$  and  $\alpha(2)$  uncorrelated with each other. That is,

$$E[\alpha(2)\alpha^H(1)] = \mathbf{0} \tag{3}$$

Substitute Eqs. (1) and (2) into (3):

$$E[\mathbf{y}(2)\mathbf{y}^H(1)] + \mathbf{A}_{1,1}E[\mathbf{y}(1)\mathbf{y}^H(1)] = \mathbf{0}$$

Postmultiplying both sides of this equation by the inverse of  $E[\mathbf{y}(1)\mathbf{y}^H(1)]$  and rearranging:

$$\mathbf{A}_{1,1} = -E[\mathbf{y}(2)\mathbf{y}^H(1)]\{E[\mathbf{y}(1)\mathbf{y}^H(1)]\}^{-1}$$

We may rewrite Eqs. (1) and (2) in the compact form

$$\begin{bmatrix} \alpha(1) \\ \alpha(2) \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{A}_{1,1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{y}(1) \\ \mathbf{y}(2) \end{bmatrix}$$

This relation shows that, given the observation vectors  $\mathbf{y}(1)$  and  $\mathbf{y}(2)$ , we may compute the innovations processes  $\alpha(1)$  and  $\alpha(2)$ . The block lower triangular transformation matrix

$$\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{A}_{1,1} & \mathbf{I} \end{bmatrix}$$

is invertible since its determinant equals 1. Hence, we may recover  $\mathbf{y}(1)$  and  $\mathbf{y}(2)$  from  $\alpha(1)$  and  $\alpha(2)$  by using the relation:

$$\begin{bmatrix} \mathbf{y}(1) \\ \mathbf{y}(2) \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{A}_{1,1} & \mathbf{I} \end{bmatrix}^{-1} \begin{bmatrix} \alpha(1) \\ \alpha(2) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{A}_{1,1} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \alpha(1) \\ \alpha(2) \end{bmatrix}$$

In general, we may express the innovations process  $\alpha(n)$  as a linear combination of the observation vectors  $\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(n)$  as follows:

$$\alpha(n) = \mathbf{y}(n) + \mathbf{A}_{1,1}\mathbf{y}(n-1) + \dots + \mathbf{A}_{n-1,n-1}\mathbf{y}(1)$$

$$= \sum_{k=1}^n \mathbf{A}_{n,1,k}\mathbf{y}(n-k+1), \quad n = 1, 2, \dots,$$

where  $\mathbf{A}_{n-1,0} = \mathbf{I}$ . The set of matrices  $\{\mathbf{A}_{n-1,k}\}$  is chosen to satisfy the following conditions

$$E[\alpha(n+1)\alpha^H(n)] = \mathbf{0} \quad n = 1, 2, \dots,$$

We may thus write

$$\begin{bmatrix} \alpha(1) \\ \alpha(2) \\ \vdots \\ \alpha(n) \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 & \dots & 0 \\ \mathbf{A}_{1,1} & \mathbf{I} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ \mathbf{A}_{n,1,n-1} & \mathbf{A}_{n,1,n-2} & \dots & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{y}(1) \\ \mathbf{y}(2) \\ \vdots \\ \mathbf{y}(n) \end{bmatrix}$$

The block lower triangular transformation matrix is invertible since its determinant equals one. Hence, we may go back and forth between the given set of observation vectors  $\{\mathbf{y}(1), \mathbf{y}(2), \dots, \mathbf{y}(n)\}$  and the corresponding set of innovations processes  $\{\alpha(1), \alpha(2), \dots, \alpha(n)\}$  without any loss of information.

10.2 First, we note that

$$E[\varepsilon(n, n-1)\mathbf{v}_1^H(n)] = E[\mathbf{x}(n)\mathbf{v}_1^H(n)] - E[\hat{\mathbf{x}}(n|\mathbf{y}_{n-1})\mathbf{v}_1^H(n)]$$

Since the estimate  $\hat{\mathbf{x}}(n|\mathbf{y}_{n-1})$  consists of a linear combination of the observation vectors  $\mathbf{y}(1), \dots, \mathbf{y}(n-1)$ , and since



$$E[\mathbf{y}(k)\mathbf{v}_1^H(n)] = \mathbf{0}, \quad 0 \leq k \leq n$$

it follows that

$$E[\hat{\mathbf{x}}(n|\mathbf{y}_{n-1})\mathbf{v}_1^H(n)] = \mathbf{0}$$

We also have

$$E[\mathbf{x}(n)\mathbf{v}_1^H(n)] = \Phi(n, 0)E[\mathbf{x}(0)\mathbf{v}_1^H(n)] + \sum_{i=1}^{n-1} \Phi(n, i)E[\mathbf{v}_1(i)\mathbf{v}_1^H(n)]$$

Since, by hypothesis

$$E[\mathbf{x}(0)\mathbf{v}_1^H(n)] = \mathbf{0}$$

and

$$E[\mathbf{v}_1(i)\mathbf{v}_1^H(n)] = \mathbf{0}, \quad 0 \leq i \leq n$$

it follows that

$$E[\mathbf{x}(n)\mathbf{v}_1^H(n)] = \mathbf{0}$$

Accordingly, we deduce that

$$E[\varepsilon(n, n-1)\mathbf{v}_1^H(n)] = \mathbf{0}$$

Next, we note that

$$E[\varepsilon(n, n-1)\mathbf{v}_2^H(n)] = E[\mathbf{x}(n)\mathbf{v}_2^H(n)] - E[\hat{\mathbf{x}}(n|\mathbf{y}_{n-1})\mathbf{v}_2^H(n)]$$

We have

$$E[\mathbf{x}(n)\mathbf{v}_2^H(n)] = \mathbf{0}$$

Also, since  $(\hat{\mathbf{x}}(n)|\mathbf{y}_{n-1})$  consists of a linear combination of  $\mathbf{y}(1), \dots, \mathbf{y}(n-1)$  and since

$$E[\mathbf{y}(k)\mathbf{v}_2^H(n)] = \mathbf{0}, \quad 1 \leq k \leq n-1$$

it follows that

$$E[\hat{\mathbf{x}}(n|\mathbf{y}_{n-1})\mathbf{v}_2^H(n)] = \mathbf{0}$$

We therefore conclude that

$$E[\boldsymbol{\varepsilon}(n, n-1)\mathbf{v}_2^H(n)] = \mathbf{0}$$

10.3 The estimated state-error vector equals

$$\boldsymbol{\varepsilon}(i, n) = \mathbf{x}(i) - \hat{\mathbf{x}}(i|\mathbf{y}_n)$$

$$= \mathbf{x}(i) - \sum_{k=1}^n \mathbf{b}_i(k)\alpha(k)$$

The expected value of the squared norm of  $\boldsymbol{\varepsilon}(i, n)$  equals

$$\begin{aligned} E[\|\boldsymbol{\varepsilon}(i, n)\|^2] &= E[\boldsymbol{\varepsilon}^H(i, n)\boldsymbol{\varepsilon}(i, n)] \\ &= \sum_{k=1}^n \sum_{l=1}^n \mathbf{b}_i^H(k)\mathbf{b}_i(l)E[\alpha^*(k)\alpha(l)] - \sum_{k=1}^n \mathbf{b}_i^H(k)E[\mathbf{x}(i)\alpha^*(k)] \\ &\quad - \sum_{k=1}^n E[\mathbf{x}^H(i)\alpha(k)]\mathbf{b}_i(k) + E[\mathbf{x}^H(i)\mathbf{x}(i)] \end{aligned}$$

Differentiating this index of performance with respect to the vector  $\mathbf{b}(k)$  and setting the result equal to zero, we find that the optimum value of  $\mathbf{b}_i(k)$  is determined by

$$2\mathbf{b}_i(k)E[\alpha(k)\alpha^*(k)] - 2E[\mathbf{x}(i)\alpha^*(k)] = \mathbf{0}$$

Hence, the optimum value of  $\mathbf{b}_i(k)$  equals

$$\mathbf{b}_i(k) = E[\mathbf{x}(i)\alpha^*(k)]\boldsymbol{\sigma}_\alpha^{-2}$$

where

$$\sigma_{\alpha}^2 = E[|\alpha(k)|^2]$$

Correspondingly, the estimate of the state vector equals

$$\begin{aligned}\hat{\mathbf{x}}(i|y_n) &= \sum_{k=1}^n E[\mathbf{x}(i)\alpha^*(k)]\sigma_{\alpha}^{-2}\alpha(k) \\ &= \sum_{k=1}^n E[x(i)\varphi^*(k)]\varphi(k)\end{aligned}$$

where  $\varphi(k)$  is the normalized innovation:

$$\varphi(k) = \frac{\alpha(k)}{\sigma_{\alpha}}$$

- 10.4 (a) The matrices  $\mathbf{K}(n, n-1)$  and  $\mathbf{Q}_2(n)$  are both correlation matrices and therefore nonnegative definite. In particular, we note

$$\mathbf{K}(n, n-1) = E[\boldsymbol{\varepsilon}(n, n-1)\boldsymbol{\varepsilon}^H(n, n-1)]$$

$$\mathbf{Q}_2(n) = E[\mathbf{v}_2(n)\mathbf{v}_2^H(n)]$$

where  $\boldsymbol{\varepsilon}(n, n-1)$  is the predicted state-error vector, and  $\mathbf{v}_2(n)$  is the measurement noise vector. We may therefore express  $\mathbf{R}(n)$  in the equivalent form

$$\mathbf{R}(n) = E[\mathbf{e}(n)\mathbf{e}^H(n)]$$

where

$$\mathbf{e}(n) = \mathbf{C}(n)\boldsymbol{\varepsilon}(n, n-1) + \mathbf{v}_2(n)$$

where it is noted that  $\boldsymbol{\varepsilon}(n, n-1)$  and  $\mathbf{v}_2(n)$  are uncorrelated. We now see that  $\mathbf{R}(n)$  is a correlation matrix and therefore nonnegative definite.

- (b) For  $\mathbf{R}(n)$  to be nonsingular, and therefore for the inverse matrix  $\mathbf{R}^{-1}(n)$  to exist, we demand that  $\mathbf{Q}_2(n)$  be positive definite such that the determinant of

$\mathbf{C}(n)\mathbf{K}(n,n-1)\mathbf{C}^H(n) + \mathbf{Q}_2(n)$  is nonzero. This requirement, in effect, says that no measurement is exact, hence the unavoidable presence of measurement noise and therefore  $\mathbf{Q}_2(n)$ . Such a requirement is reasonable on physical grounds.

10.5 In the limit, when  $n$  approaches infinity, we may put

$$\mathbf{K}(n+1,n) = \mathbf{K}(n,n-1) = \mathbf{K}$$

Under this condition, Eq. (10.54) of the text simplifies to

$$\mathbf{K} = (\mathbf{I} - \mathbf{GC})\mathbf{K}(\mathbf{I} - \mathbf{C}^H\mathbf{G}^H) + \mathbf{Q}_1 + \mathbf{GQ}_2\mathbf{G}^H \quad (1)$$

where it is assumed that the state-transition matrix equals the identity matrix, and  $\mathbf{G}$ ,  $\mathbf{C}$ ,  $\mathbf{Q}_1$  and  $\mathbf{Q}_2$  are the limiting values of the matrices  $\mathbf{G}(n)$ ,  $\mathbf{C}(n)$ ,  $\mathbf{Q}_1(n)$  and  $\mathbf{Q}_2(n)$ , respectively. From Eq. (10.49) we find that the limiting value of the Kalman gain equals

$$\mathbf{G} = \mathbf{KC}^H(\mathbf{CKC}^H + \mathbf{Q}_2)^{-1} \quad (2)$$

Expanding Eq. (1) and then using Eq. (2) to eliminate  $\mathbf{G}$ , we get

$$\mathbf{KC}^H(\mathbf{CKC}^H + \mathbf{Q}_2)^{-1}\mathbf{CK} - \mathbf{Q}_1 = \mathbf{0}$$

$$10.6 \quad \mathbf{x}(n+1) = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \mathbf{x}(n) + \mathbf{v}_1(n) \quad \mathbf{Q}_1 = \mathbf{I}; E[\mathbf{v}_1] = \mathbf{0}$$

$$y(n) = \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{x}(n) + v_2(n) \quad \mathbf{Q}_2 = 1; E[v_2] = 0$$

(a) Using Table 10.2, we formulate the recursions for computing the Kalman filter

Known parameters:

$$\mathbf{F}(n+1, n) = \mathbf{F} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

$$\mathbf{c}(n) = \mathbf{c} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$\mathbf{Q}_1(n) = \mathbf{I}$$

$$\mathbf{Q}_2(n) = 1$$

Unknown parameters:

$$\mathbf{K}(n, n-1) = \mathbf{K} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

We may then write

$$\mathbf{G}(n) = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \mathbf{K}(n, n-1) \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left( \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{K}(n, n-1) \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 1 \right)^{-1} = \begin{bmatrix} \frac{k_{21}(n, n-1)}{k_{11}(n, n-1) + 1} \\ \frac{k_{11}(n, n-1) + k_{21}(n, n-1)}{k_{11}(n, n-1) + 1} \end{bmatrix}$$

$$\alpha(n) = y(n) - \begin{bmatrix} 1 & 0 \end{bmatrix} \hat{\mathbf{x}}(n|Y_{n-1}) = y(n) - \hat{x}_1(n|Y_{n-1})$$

$$\hat{\mathbf{x}}(n+1|Y_n) = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{G}(n)\alpha(n) = \begin{bmatrix} \frac{\{k_{11}(n, n-1) - k_{21}(n, n-1)\}\hat{x}_1(n|Y_{n-1}) + k_{21}(n, n-1)y(n)}{k_{11}(n, n-1) + 1} \\ \frac{\{1 - k_{21}(n, n-1)\}\hat{x}_1(n|Y_{n-1}) + \{k_{11}(n, n-1) + k_{21}(n, n-1)y(n)\}}{k_{11}(n, n-1) + 1} \end{bmatrix} \quad (1)$$

$$\mathbf{K}(n) = \mathbf{K}(n, n-1) - \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \mathbf{G}(n) \begin{bmatrix} 1 & 0 \end{bmatrix} \mathbf{K}(n, n-1) =$$

$$= \mathbf{K}(n, n-1) - \begin{bmatrix} \frac{k_{11} + k_{21}}{k_{11} + 1} & 0 \\ \frac{k_{11} + 2k_{21}}{k_{11} + 1} & 0 \end{bmatrix} \mathbf{K}(n, n-1) = \mathbf{K}(n, n-1) - \begin{bmatrix} k_{11} \frac{k_{11} + k_{21}}{k_{11} + 1} & k_{12} \frac{k_{11} + k_{21}}{k_{11} + 1} \\ k_{11} \frac{k_{11} + 2k_{21}}{k_{11} + 1} & k_{12} \frac{k_{11} + k_{21}}{k_{11} + 1} \end{bmatrix}$$

$$= \begin{bmatrix} k_{11} \frac{-k_{21} + 1}{k_{11} + 1} & k_{12} \frac{-k_{21} + 1}{k_{11} + 1} \\ k_{21} - k_{11} \frac{k_{11} + 2k_{21}}{k_{11} + 1} & k_{22} - k_{12} \frac{k_{11} + 2k_{21}}{k_{11} + 1} \end{bmatrix}$$

$$\begin{aligned}
\mathbf{K}(n+1, n) &= \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \mathbf{K}(n) \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} + \mathbf{I} = \\
&= \begin{bmatrix} 1 + k_{22} - k_{12} \left( \frac{k_{11} + 2k_{21}}{k_{11} + 1} \right) & k_{21} + k_{22} - (k_{11} + k_{12}) \left( \frac{k_{11} + 2k_{21}}{k_{11} + 1} \right) \\ k_{22} - k_{12} \left( \frac{k_{11} + k_{21}}{k_{11} + 1} \right) & 1 + k_{21} + k_{22} + (k_{11} + k_{12}) \left( \frac{1 - k_{11} + k_{21}}{k_{11} + 1} \right) \end{bmatrix} \quad (2)
\end{aligned}$$

For this particular case, the Kalman filtering process is entirely described by the pair of equations (1) and (2).

(b) The generalized form of the algebraic Riccati equation is:

$$\mathbf{K} - \mathbf{F}\mathbf{K}\mathbf{F}^H + \mathbf{F}\mathbf{K}\mathbf{C}^H(\mathbf{C}\mathbf{K}\mathbf{C}^H + \mathbf{Q}_2)^{-1}\mathbf{C}\mathbf{K}\mathbf{F}^H - \mathbf{Q}_1 = \mathbf{0}$$

For our problem:

here we use the fact that the predicted state-error correlation matrix  $\mathbf{K}$  is symmetric

$$\mathbf{F} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{Q}_1 = \mathbf{I}, \quad \mathbf{K} = \begin{bmatrix} k_1 & k_2 \\ k_2 & k_3 \end{bmatrix}.$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \mathbf{Q}_2 = [1],$$

Therefore, we have:

$$\begin{aligned}
\mathbf{K} &= \overbrace{\begin{bmatrix} k_1 & k_2 \\ k_2 & k_3 \end{bmatrix} - \underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \\ k_2 & k_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}}_{\mathbf{B}} +}_{\mathbf{C}} \\
&\underbrace{\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \\ k_2 & k_3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} \left( \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \\ k_2 & k_3 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + 1 \right)^{-1} \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \\ k_2 & k_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}}_{\mathbf{A}} - \mathbf{I} = \mathbf{0}
\end{aligned}$$

$$\begin{aligned}
\mathbf{A} &= \frac{1}{k_1+1} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \\ k_2 & k_3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \\ k_2 & k_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \\
&= \frac{1}{k_1+1} \begin{bmatrix} k_2 & k_3 \\ k_1+k_2 & k_2+k_3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} k_2 & k_1+k_2 \\ k_3 & k_2+k_3 \end{bmatrix} \\
&= \frac{1}{k_1+1} \begin{bmatrix} k_2 & 0 \\ k_1+k_2 & 0 \end{bmatrix} \begin{bmatrix} k_2 & k_1+k_2 \\ k_3 & k_2+k_3 \end{bmatrix} = \frac{1}{k_1+1} \begin{bmatrix} k_2^2 & k_2(k_1+k_2) \\ k_2(k_1+k_2) & (k_1+k_2)^2 \end{bmatrix} \\
&= \begin{bmatrix} \frac{k_2^2}{k_1+1} & \frac{k_2(k_1+k_2)}{k_1+1} \\ \frac{k_2(k_1+k_2)}{k_1+1} & \frac{(k_1+k_2)^2}{k_1+1} \end{bmatrix}
\end{aligned}$$

$$\begin{aligned}
\mathbf{B} &= \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \\ k_2 & k_3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} k_2 & k_3 \\ k_1+k_2 & k_2+k_3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} k_2 & k_2+k_3 \\ k_2+k_3 & k_1+2k_2+k_3 \end{bmatrix}
\end{aligned}$$

$$\mathbf{C} = \mathbf{K} - \mathbf{B} = \begin{bmatrix} k_1-k_3 & -k_3 \\ -k_3 & -k_1-2k_2 \end{bmatrix}$$

$$\mathbf{C} + \mathbf{A} - \mathbf{I} = \mathbf{0}$$

$$\begin{bmatrix} \left( \frac{k_2^2}{k_1+1} + k_1 - k_3 - 1 \right) & \left( \frac{k_2(k_1+k_2)}{k_1+1} - k_3 \right) \\ \left( \frac{k_2(k_1+k_2)}{k_1+1} - k_3 \right) & \left( \frac{(k_1+k_2)^2}{k_1+1} - k_1 - 2k_2 - 1 \right) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$\left\{ \begin{array}{l} k_3 = \frac{k_2(k_1+k_2)}{k_1+1} \\ \frac{k_2^2 - k_2(k_1+k_2)}{k_1+1} + k_1 - 1 = 0 \\ \frac{(k_1+k_2)^2}{k_1+1} - k_1 - 2k_2 - 1 = 0 \end{array} \right. \Rightarrow \left\{ \begin{array}{l} \frac{-k_1 k_2}{k_1+1} + k_1 - 1 = 0 \\ \frac{(k_1+k_2)^2}{k_1+1} - k_1 - 2k_2 - 1 = 0 \end{array} \right. \Rightarrow$$

$$\left\{ \begin{array}{l} k_2 = \frac{k_1^2 - 1}{k_1} \\ \cancel{k_1^2 + 2k_1 k_2 + k_2^2 - k_1^2 - 2k_1 k_2 - 2k_2 - k_1 - 1 = 0} \end{array} \right. \Rightarrow \left\{ \begin{array}{l} k_2 = \frac{k_1^2 - 1}{k_1} \\ k_2^2 - 2k_2 - 2k_1 - 1 = 0 \end{array} \right.$$

$$\frac{(k_1^2 - 1)^2}{k_1^2} - 2 \frac{k_1^2 - 1}{k_1} - 2k_1 - 1 = 0$$

$$k_1^4 - 2k_1^2 + 1 - 2k_1^3 + 2k_1 - 2k_1^3 - k_1^2 = 0$$

$$k_1^4 - 4k_1^3 - 3k_1^2 + 2k_1 + 1 = 0$$

This equation has four different real-valued solutions, but it is easy to show that only one of them is meaningful in the context of our problem.



10.7 (a) The state equations are

$$\mathbf{x}(n+1) = \mathbf{F}(n+1, n)\mathbf{x}(n) + \mathbf{v}_1(n) \quad (1)$$

$$\mathbf{y}(n) = \mathbf{C}(n)\mathbf{x}(n) + \mathbf{v}_2(n) \quad (2)$$

We are given

$$\begin{aligned} \mathbf{x}(n) &= [a_1(n), \dots, a_M(n), \dots, a_{M+N}(n)]^T \\ \mathbf{C}(n) &= [-y(n-1), \dots, -y(n-M), v(n-1), \dots, v(n-N)] \end{aligned}$$

We are also given

$$a_k(n+1) = a_k(n) + w_k(n), \quad k = 1, \dots, M+N$$

which means that

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

Comparing Eqs. (1) and (3) we therefore deduce that for the problem at hand

$$\mathbf{F}(n+1, n) = \mathbf{I}$$

and

$$\mathbf{v}_1(n) = \mathbf{w}(n)$$

We next note that the difference equation describing the given ARMA process may be recast as follows

$$\begin{aligned} y(n) &= -\sum_{k=1}^M a_k(n)y(n-k) + \sum_{k=1}^N a_{M+k}(n)v(n-k) + v(n) \\ &= \mathbf{C}(n)\mathbf{x}(n) + v(n) \end{aligned} \quad (4)$$

Comparing Eqs. (2) and (4) we therefore deduce

$$\begin{aligned} \mathbf{y}(n) &= y(n) \\ \mathbf{v}_2(n) &= v(n) \end{aligned}$$

This completes the evaluation of the state equations.

- (b) Since the transition matrix equals the identity matrix, we find that the predicted and filtered forms of the state vector are the same. Hence,

$$\mathbf{G}(n) = K(n, n-1) \mathbf{C}^H(n) [\mathbf{C}(n) \mathbf{K}(n, n-1) \mathbf{C}^H(n) + \mathbf{Q}_2(n)]^{-1}$$

$$\alpha(n) = y(n) - \mathbf{C}(n) \hat{\mathbf{x}}(n|Y_{n-1})$$

$$\hat{\mathbf{x}}(n+1|Y_n) = \hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{G}(n) \alpha(n)$$

$$\mathbf{x}(n|Y_n) = \hat{\mathbf{x}}(n+1|Y_n)$$

$$\mathbf{K}(n) = \mathbf{K}(n, n-1) - \mathbf{G}(n) \mathbf{C}(n) \mathbf{K}(n, n-1)$$

$$\mathbf{K}(n+1, n) = \mathbf{K}(n) + \mathbf{Q}_1(n)$$

(c) To initialize the algorithm, we set

$$\hat{\mathbf{x}}(0|Y_0) = E[\mathbf{x}(0)], \quad v(n) = v_1(n)$$

$$\mathbf{K}(0) = E[\mathbf{x}(0) \mathbf{x}^H(0)], \quad w(n) = v_2(n)$$

10.8 We are given the state equations

$$\mathbf{x}(n+1) = \mathbf{A} \mathbf{x}(n) + \mathbf{b} v(n) \quad (1)$$

$$y(n) = \mathbf{h}^T \mathbf{x}(n) + w(n) \quad (2)$$

We may therefore make the following identifications:

$$\mathbf{F}(n+1, n) = \mathbf{A}$$

$$\mathbf{v}(n) = \mathbf{b} v(n)$$

$$\mathbf{C}(n) = \mathbf{h}^T$$

$$v_2(n) = w(n)$$

Using the Kalman filtering algorithm, we have

$$\mathbf{G}(n) = \mathbf{A} \mathbf{K}(n, n-1) \mathbf{h} [\mathbf{h}^T \mathbf{K}(n, n-1) \mathbf{h} + \sigma_w^2]^{-1}$$

$$\alpha(n) = y(n) - \mathbf{h}^T \hat{\mathbf{x}}(n|Y_{n-1})$$

$$\hat{\mathbf{x}}(n+1|Y_{n-1}) = \mathbf{A}\hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{G}(n)\alpha(n)$$

$$\hat{\mathbf{x}}(n|Y_n) = \mathbf{A}\hat{\mathbf{x}}(n+1|Y_n)$$

$$\mathbf{K}(n) = \mathbf{K}(n, n-1) - \mathbf{A}\mathbf{G}(n)\mathbf{h}^T \mathbf{K}(n, n-1)$$

$$\mathbf{K}(n+1, n) = \mathbf{A}\mathbf{K}(n)\mathbf{A}^T + \sigma_v^2 \mathbf{b} \mathbf{b}^T$$

We note the following:

1. The factor  $(\mathbf{h}^T \mathbf{K}(n, n-1) \mathbf{h} + \sigma_w^2)$  is a scalar.
2. The Kalman gain  $\mathbf{G}(n)$  is a vector.
3. The matrix  $\mathbf{A}$  is  $M$ -by- $M$ . The matrix  $\mathbf{K}(n, n-1)$  is also  $M$ -by- $M$ . The vector  $\mathbf{h}$  is  $M$ -by-1. Hence, the Kalman gain is an  $M$ -by-1 vector.

The dynamics of the message source are represented by the state Eq. (1), where the  $M$ -by- $M$  matrix  $\mathbf{A}$  and the  $M$ -by-1 vector  $\mathbf{b}$  are defined by

$$\mathbf{A} = \begin{bmatrix} 0 & . & . & . & . & . & 0 \\ 1 & 0 & . & . & . & . & 0 \\ 0 & 1 & 0 & . & . & . & 0 \\ \vdots & & & & & & \vdots \\ 0 & . & . & . & . & 0 & 1 & 0 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

The elements of the state vector  $\mathbf{x}(n)$  consist of the channel input represented by  $u(n) = v(n)$  and successively delayed versions of it;  $v(n)$  is modeled as a random binary white noise sequence. Equation (1) simply states that each succeeding component at time  $n$  is equal to the previous component at time  $n-1$ . The channel output is described by Eq. (2), where  $y(n)$  is the measured output,  $\mathbf{h}^T(n)$  is the 1-by- $M$  row vector of channel coefficients, and  $w(n)$  is a Gaussian white noise sequence independent of  $v(n)$ .

For the digital communication system described by Eqs. (1) and (2), use of the Kalman filter yields

$$\hat{\mathbf{x}}(n+1|Y_n) = \mathbf{A}\hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{G}(n)\alpha(n)$$

where  $\mathbf{G}(n)$  is the Kalman gain, and

$$\alpha(n) = y(n) - \mathbf{h}^T \hat{\mathbf{x}}(n|Y_{n-1})$$

The resulting general form for the Kalman filter is depicted in Fig. 1. We see from this figure that the Kalman filter consists of an IIR filter with forward coefficients defined by the channel impulse response and feedback coefficients defined by the Kalman gain (which in this problem is a column vector).

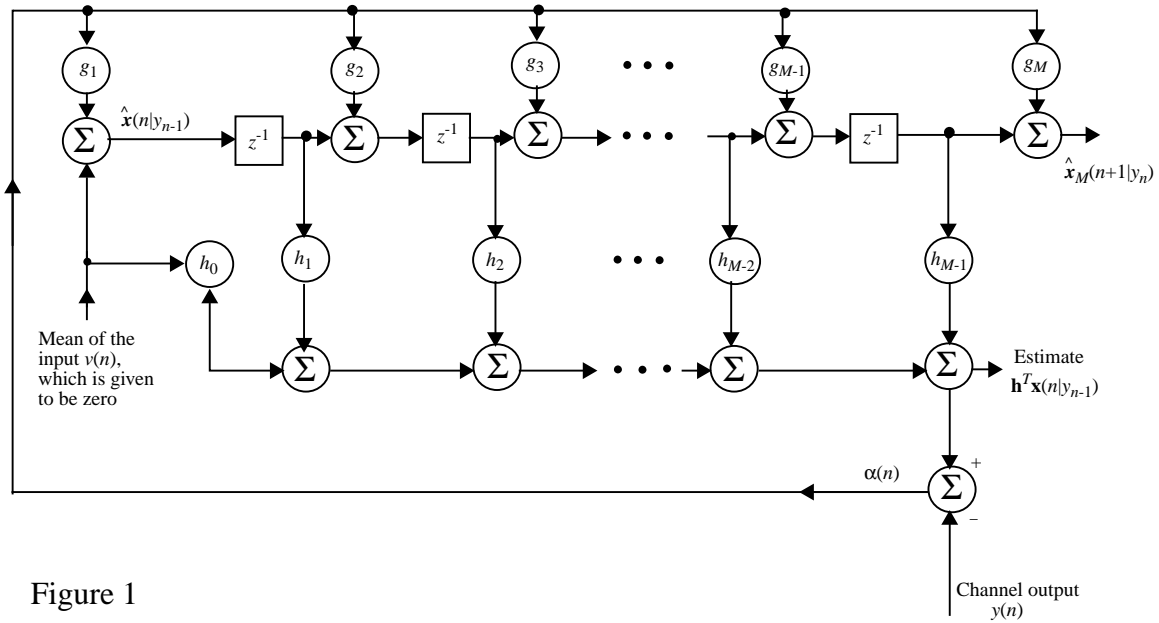


Figure 1

10.9 We start with

$$\alpha(n) = \mathbf{y}(n) - \mathbf{C}(n)\mathbf{x}(n|Y_{n-1}) \quad (1)$$

$$\hat{\mathbf{x}}(n+1|Y_n) = \mathbf{F}(n+1,n)\hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{G}(n)\alpha(n) \quad (2)$$

Substituting Eq. (1) into (2):

$$\hat{\mathbf{x}}(n+1|Y_n) = \mathbf{F}(n+1,n)\hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{G}(n)[y(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n|Y_{n-1})]$$

The filtered state estimate is

$$\begin{aligned}
\hat{\mathbf{x}}(n|Y_n) &= \mathbf{F}(n, n+1)\hat{\mathbf{x}}(n+1|Y_n) \\
&= \hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{F}(n+1, n)\mathbf{G}(n)[y(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n|Y_{n-1})] \\
&= [\mathbf{I} - \mathbf{F}(n+1, n)\mathbf{G}(n)\mathbf{C}(n)]\hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{F}(n+1, n)\mathbf{G}(n)\mathbf{y}(n)
\end{aligned}$$

But

$$\mathbf{y}(n) = \mathbf{C}(n)\mathbf{x}(n) + \mathbf{v}_2(n)$$

Hence,

$$\begin{aligned}
\hat{\mathbf{x}}(n|Y_n) &= [\mathbf{I} - \mathbf{F}(n, n+1)\mathbf{G}(n)\mathbf{C}(n)]\hat{\mathbf{x}}(n|Y_{n-1}) \\
&\quad + \mathbf{F}(n, n+1)\mathbf{G}(n)\mathbf{C}(n)\mathbf{x}(n) + \mathbf{F}(n, n+1)\mathbf{G}(n)\mathbf{v}_2(n)
\end{aligned}$$

Taking expectations, and recognizing that the measurement noise vector  $\mathbf{v}_2(n)$  has zero mean:

$$\begin{aligned}
E[\hat{\mathbf{x}}(n|Y_n)] &= [\mathbf{I} - \mathbf{F}(n, n+1)\mathbf{G}(n)\mathbf{C}(n)]E[\hat{\mathbf{x}}(n|Y_{n-1})] \\
&\quad + \mathbf{F}(n, n+1)\mathbf{G}(n)\mathbf{C}(n)E[\mathbf{x}(n)]
\end{aligned}$$

For  $n=1$ , we thus get

$$\begin{aligned}
E[\hat{\mathbf{x}}(1|Y_1)] &= [\mathbf{I} - \mathbf{F}(1, 2)\mathbf{G}(1)\mathbf{C}(1)]E[\hat{\mathbf{x}}(1|Y_0)] \\
&\quad + \mathbf{F}(1, 2)\mathbf{G}(1)\mathbf{C}(1)E[\mathbf{x}(1)]
\end{aligned} \tag{1}$$

Since the one-step prediction  $\hat{\mathbf{x}}(1|Y_0)$  must be specified, we have

$$E[\hat{\mathbf{x}}(1|Y_0)] = \hat{\mathbf{x}}(1|Y_0)$$

Accordingly, substituting the choice of the initial condition

$$\hat{\mathbf{x}}(1|Y_0) = E[\mathbf{x}(1)]$$

in Eq. (1) yields

$$\begin{aligned}
E[\hat{\mathbf{x}}(1|Y_1)] &= [\mathbf{I} - \mathbf{F}(1, 2)\mathbf{G}(1)\mathbf{C}(1)]E[\mathbf{x}(1)] \\
&\quad + \mathbf{F}(1, 2)\mathbf{G}(1)\mathbf{C}(1)E[\mathbf{x}(1)] \\
&= E[\mathbf{x}(1)]
\end{aligned}$$

By induction, we may go on to show that in general:

$$E[\hat{\mathbf{x}}|Y_n] = E[\mathbf{x}(n)]$$

In other words, the filtered estimate  $\hat{\mathbf{x}}(n|y_n)$  produced by the Kalman filter is an unbiased estimate for the specified method of initialization.

#### 10.10 MAP derivation of Kalman filter

$$(a) \ f_X(\mathbf{x}(n)|\mathbf{y}(n)) = \frac{f_{XY}(\mathbf{x}(n), Y_n)}{f_Y(Y_n)} = \frac{f_{XY}(\mathbf{x}(n), \mathbf{y}(n), Y_{n-1})}{f_Y(\mathbf{y}(n), Y_{n-1})} \quad (1)$$

where

$$\begin{aligned}
f_{XY}(\mathbf{x}(n), \mathbf{y}(n), Y_{n-1}) &= f_Y(\mathbf{y}(n)|\mathbf{x}(n), Y_{n-1})f_{XY}(\mathbf{x}(n), Y_{n-1}) \\
&= f_Y(\mathbf{y}(n)|\mathbf{x}(n), Y_{n-1})f_X(\mathbf{x}(n)|Y_{n-1})f_Y(Y_{n-1}) \\
&= f_Y(\mathbf{y}(n)|\mathbf{x}(n))f_X(\mathbf{x}(n)|Y_{n-1})f_Y(Y_{n-1}) \quad (2)
\end{aligned}$$

where we have used the fact that  $\mathbf{y}(n) = \mathbf{C}(n)\mathbf{x}(n) + \mathbf{v}_2(n)$ , and  $\mathbf{v}_2(n)$  does not depend on  $Y_{n-1}$ . Consequently,

$$f_Y(\mathbf{y}(n)|\mathbf{x}(n), Y_{n-1}) = f_Y(\mathbf{y}(n)|\mathbf{x}(n))$$

Substituting Eq. (2) into (1), we thus get

$$\begin{aligned}
f_X(\mathbf{x}(n)|Y_n) &= \frac{f_Y(\mathbf{y}(n)|\mathbf{x}(n))f_X(\mathbf{x}(n)|Y_{n-1})f_Y(Y_{n-1})}{f_Y(\mathbf{y}(n), Y_{n-1})} \\
&= \frac{f_Y(\mathbf{y}(n)|\mathbf{x}(n))f_X(\mathbf{x}(n)|Y_{n-1})\cancel{f_Y(Y_{n-1})}}{\cancel{f_Y(\mathbf{y}(n)|Y_{n-1})}\cancel{f_Y(Y_{n-1})}}
\end{aligned}$$

$$= \frac{f_Y(\mathbf{y}(n)|\mathbf{x}(n))f_X(\mathbf{x}(n)|Y_{n-1})}{f_Y(\mathbf{y}(n), Y_{n-1})} \quad (1)$$

(b) Examine first  $f_Y(\mathbf{y}(n)|\mathbf{x}(n))$ ; its mean is

$$\begin{aligned} E[\mathbf{y}(n)|\mathbf{x}(n)] &= E[\mathbf{C}(n)\mathbf{x}(n) + \mathbf{v}_2(n)|\mathbf{x}(n)] \\ &= \mathbf{C}(n)\mathbf{x}(n) \end{aligned}$$

The variance is

$$\text{var}[\mathbf{y}(n)|\mathbf{x}(n)] = \text{var}[\mathbf{v}_2(n)|\mathbf{x}(n)] = \mathbf{Q}_2(n),$$

where  $\mathbf{Q}_2(n)$  = correlation matrix of measurement noise  $\mathbf{v}_2(n)$ . Thus, assuming Gaussianity, we may write

$$f_Y(\mathbf{y}(n)|\mathbf{x}(n)) = A_1 \exp\left(-\frac{1}{2}(\mathbf{y}(n) - \mathbf{C}(n)\mathbf{x}(n))^H \mathbf{Q}_2^{-1}(n)(\mathbf{y}(n) - \mathbf{C}(n)\mathbf{x}(n))\right) \quad (2)$$

where the constant  $A_1$  is a proper scaling factor.

Consider next  $f_X(\mathbf{x}(n)|Y_{n-1})$ ; its mean is

$$\begin{aligned} E[\mathbf{x}(n)|Y_{n-1}] &= E[\mathbf{F}(n, n-1)\hat{\mathbf{x}}(n-1) + \mathbf{v}_1(n-1)|Y_{n-1}] \\ &= \mathbf{F}(n, n-1)\hat{\mathbf{x}}(n-1) \\ &= \hat{\mathbf{x}}(n|Y_{n-1}) \end{aligned}$$

The variance is

$$\begin{aligned} \text{var}[\mathbf{x}(n)|Y_{n-1}] &= \text{var}[\mathbf{x}(n) - \hat{\mathbf{x}}(n)|Y_{n-1}] \\ &= \text{var}[\boldsymbol{\varepsilon}(n, n-1)] \end{aligned}$$

where  $\boldsymbol{\varepsilon}(n, n-1)$  is the state-error vector.

Denote this variance by  $\mathbf{K}(n, n-1)$ , which is to be determined. Again, assuming Gaussianity, we may write

$$f_X(\mathbf{x}(n)|Y_{n-1}) = A_2 \exp\left(-\frac{1}{2}(\mathbf{x}(n)-\hat{\mathbf{x}}(n)|Y_{n-1})^H \mathbf{K}^{-1}(n, n-1)(\mathbf{x}(n)-\hat{\mathbf{x}}(n)|Y_{n-1})\right) \quad (3)$$

where  $A_2$  is another appropriate scaling factor. Thus, substituting Eqs. (2) and (3) into (1), we get

$$f_X(\mathbf{x}(n)|Y_n) = A \exp\left(-\frac{1}{2}(\mathbf{y}(n) - \mathbf{C}(n)\mathbf{x}(n))^H \mathbf{Q}_2^{-1}(n)(\mathbf{y}(n) - \mathbf{C}(n)\mathbf{x}(n)) - \frac{1}{2}(\mathbf{x}(n) - \hat{\mathbf{x}}(n)|Y_{n-1})^H \mathbf{K}^{-1}(n, n-1)(\mathbf{x}(n) - \hat{\mathbf{x}}(n)|Y_{n-1})\right) \quad (4)$$

where  $A = A_1 A_2$  is another constant.

(v) By definition, the MAP estimate of the state is defined by the condition

$$\left. \frac{\partial \ln f_X(\mathbf{x}(n)|Y_n)}{\partial (\mathbf{x}(n))} \right|_{\mathbf{x}(n) = \hat{\mathbf{x}}_{\text{MAP}}(n)} = \mathbf{0} \quad (5)$$

Hence, substituting Eq. (4) into (5) yields

$$\begin{aligned} \hat{\mathbf{x}}_{\text{MAP}}(n) = & [\mathbf{C}^H(n)\mathbf{Q}_2^{-1}(n)\mathbf{C}(n) + \mathbf{K}^{-1}(n, n-1)]^{-1} [\mathbf{K}^{-1}(n, n-1)\hat{\mathbf{x}}(n|Y_{n-1}) \\ & + \mathbf{C}^H(n)\mathbf{Q}_2^{-1}(n)\mathbf{y}(n)] \end{aligned} \quad (6)$$

From a computational point of view, we need to put the first inverse matrix into a more convenient form. To that end, we apply the matrix inversion lemma, which may be stated as follows (see Section 9.2 of the text).

If

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

then

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

With the expression  $[\mathbf{C}^H(n)\mathbf{Q}_2^{-1}(n)\mathbf{C}(n) + \mathbf{K}^{-1}(n, n-1)]^{-1}$  as the issue of concern, we note the following

$$\mathbf{B} = \mathbf{K}(n, n-1)$$



$$\mathbf{C} = \mathbf{C}^H(n)$$

$$\mathbf{D} = \mathbf{Q}_2(n)$$

Hence, applying the matrix inversion lemma:

$$\begin{aligned} & [\mathbf{C}^H(n)\mathbf{Q}_2^{-1}(n)\mathbf{C}(n) + \mathbf{K}^{-1}(n, n-1)]^{-1} \\ &= \mathbf{K}(n, n-1) - \mathbf{K}(n, n-1)\mathbf{C}^H(n)(\mathbf{Q}_2(n) + \mathbf{C}(n)\mathbf{K}(n, n-1)\mathbf{C}^H(n))^{-1}\mathbf{C}(n)\mathbf{K}(n, n-1) \quad (7) \end{aligned}$$

Substituting Eq. (7) into (6), and then going through some lengthy but straightforward algebraic manipulations, we get

$$\hat{\mathbf{x}}_{\text{MAP}}(n) = \hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{G}(n)[\mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n|Y_{n-1})] \quad (8)$$

where  $\mathbf{G}(n)$  is the Kalman gain defined by

$$\mathbf{G}(n) = \mathbf{F}(n+1, n)\mathbf{K}(n, n-1)\mathbf{C}^H(n)[\mathbf{C}(n)\mathbf{K}(n, n-1)\mathbf{C}^H(n) + \mathbf{Q}_2(n)]^{-1} \quad (9)$$

The one issue that is yet to be determined is  $\mathbf{K}(n, n-1)$ . Here we note

$$\begin{aligned} \epsilon(n, n-1) &= \mathbf{x}(n) - \hat{\mathbf{x}}(n|Y_{n-1}) \\ &= \mathbf{F}(n, n-1)\mathbf{x}(n-1) + \mathbf{v}_1(n-1) - \mathbf{F}(n, n-1)\hat{\mathbf{x}}_{\text{MAP}}(n-1) \\ &= \mathbf{F}(n, n-1)\epsilon_{\text{MAP}}(n-1) + \mathbf{v}_1(n-1) \end{aligned}$$

Therefore,

$$\mathbf{K}(n, n-1) = \mathbf{F}(n, n-1)\mathbf{K}(n-1)\mathbf{F}^H(n, n-1) + \mathbf{Q}_1(n-1) \quad (10)$$

where

$$\mathbf{K}(n) = \text{var}[\epsilon_{\text{MAP}}(n)]$$

There only remains for us to determine  $\mathbf{K}(n-1)$ . Here we note

$$\epsilon_{\text{MAP}}(n) = \mathbf{x}(n) - \hat{\mathbf{x}}_{\text{MAP}}(n)$$

$$= \mathbf{x}(n) + (-\hat{\mathbf{x}}(n|Y_{n-1})) + \mathbf{G}(n)[\mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n|Y_{n-1})]$$

But

$$\mathbf{y}(n) = \mathbf{C}(n)\mathbf{x}(n) + \mathbf{v}_2(n)$$

Hence, noting that  $\mathbf{e}(n, n-1) = \mathbf{x}(n) - \hat{\mathbf{x}}(n|Y_{n-1})$ ,

$$\begin{aligned} \mathbf{e}_{\text{MAP}}(n) &= \mathbf{e}(n, n-1) - \mathbf{G}(n)[\mathbf{C}(n)\mathbf{e}(n, n-1) + \mathbf{v}_2(n)] \\ &= [\mathbf{I} - \mathbf{G}(n)\mathbf{C}(n)]\mathbf{e}(n, n-1) - \mathbf{G}(n)\mathbf{v}_2(n) \end{aligned}$$

which yields

$$\begin{aligned} \mathbf{K}(n) &= \text{var}[\mathbf{e}_{\text{MAP}}(n)] \\ &= [\mathbf{I} - \mathbf{G}(n)\mathbf{C}(n)]\mathbf{K}(n, n-1)[\mathbf{I} - \mathbf{G}(n)\mathbf{C}(n)]^H + \mathbf{G}(n)\mathbf{Q}_2(n)\mathbf{G}^H(n) \end{aligned}$$

After some manipulations, this formula reduces to

$$\mathbf{K}(n) = \mathbf{K}(n, n-1) - \mathbf{F}(n, n+1)\mathbf{G}(n)\mathbf{C}(n)\mathbf{K}(n, n-1) \quad (11)$$

The algorithm for computing the MAP estimate  $\hat{\mathbf{x}}_{\text{MAP}}(n)$  is now complete. It is made up of Eqs. (8) through (11). Indeed, comparing these equations with the Kalman filtering algorithm summarized in Table 10.2 of the text, we find that the MAP estimate  $\hat{\mathbf{x}}_{\text{MAP}}(n)$  is nothing but the filtered estimate  $\hat{\mathbf{x}}(n)$  in the standard Kalman filter theory.

(d) The second dedrivative of  $\ln f_X(\mathbf{x}(n)|Y_n)$  is given by the expression

$\mathbf{K}^{-1}(n, n-1)\hat{\mathbf{x}}(n, n-1)$  which is always negative. Hence, the condition of Eq. (2) in Problem 10.10 is satisfied by the MAP estimate  $\hat{\mathbf{x}}_{\text{MAP}}(n)$ .

10.11 Given a system

$$\begin{aligned} \mathbf{x}(n+1) &= \mathbf{F}\mathbf{x}(n) \\ \mathbf{y}(n) &= \mathbf{C}\mathbf{x}(n) \end{aligned}$$

(a) Show that

$$\hat{\mathbf{x}}(n|Y_n) = \mathbf{F}(\mathbf{I} - \mathbf{G}(n)\mathbf{C})\hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{C}\mathbf{G}(n)\mathbf{y}(n)$$

$$\alpha(n) = \mathbf{y}(n) - \mathbf{C}\hat{\mathbf{x}}(n|Y_{n-1})$$

By definition, the innovation is

$$\alpha(n) = \mathbf{y}(n) - \hat{\mathbf{y}}(n|Y_{n-1})$$

Following the same reasoning used to derive Eq. (10.31) of the text, we have

$$\hat{\mathbf{y}}(n|Y_{n-1}) = \mathbf{C}\hat{\mathbf{x}}(n|Y_{n-1})$$

Therefore,

$$\alpha(n) = \mathbf{y}(n) - \mathbf{C}\hat{\mathbf{x}}(n|Y_{n-1}) \text{ which confirms the second equation of the problem.}$$

Next,

$$\mathbf{y}(n) = \mathbf{C}\hat{\mathbf{x}}(n) \Rightarrow \alpha(n) = [\mathbf{C}\mathbf{x}(n) - \hat{\mathbf{x}}(n|Y_{n-1})] = \mathbf{C}\xi(n, n-1)$$

$$\mathbf{R}(n) = E[\alpha(n)\alpha^H(n)] = \mathbf{C}E[\xi(n, n-1)\xi^H(n, n-1)]\mathbf{C}^H = \mathbf{C}\mathbf{K}(n, n-1)\mathbf{C}^H$$

Similarly to Eq. (10.45) of the text, we can write:

$$\hat{\mathbf{x}}(n|Y_{n-1}) = \mathbf{F}\hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{G}(n)\alpha(n)$$

$$\mathbf{G}(n) = E[\mathbf{x}(n+1)\alpha^H(n)]\mathbf{R}^{-1}(n) = \mathbf{F}\mathbf{K}(n, n-1)\mathbf{C}^H\mathbf{R}^{-1}(n)$$

Here we have used Eqs. (10-44)-(10.49) of the text as a model.

$$\xi(n+1, n) = [\mathbf{F} - \mathbf{G}(n)\mathbf{C}]\xi(n, n-1)$$

$$\begin{aligned} \mathbf{K}(n+1, n) &= E[\xi(n+1, n)\xi^H(n+1, n)] = [\mathbf{F} - \mathbf{G}(n)\mathbf{C}]\mathbf{K}(n, n-1)[\mathbf{F}^H - \mathbf{C}^H\mathbf{G}^H(n)] \\ &= \mathbf{F}\mathbf{K}(n)\mathbf{F}^H \end{aligned}$$

$$\mathbf{K}(n) = (\mathbf{I} - \mathbf{F}^{-1}\mathbf{G}(n)\mathbf{C})\mathbf{K}(n, n-1)$$

Similarly to Eq. (10.58) of the text,

$$\begin{aligned}
\hat{\mathbf{x}}(n|Y_n) &= \mathbf{F}^{-1}\hat{\mathbf{x}}(n+1|Y_n) = \hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{F}^{-1}\mathbf{G}(n)\alpha(n) \\
&= \hat{\mathbf{x}}(n|Y_{n-1}) + (\mathbf{I} - \mathbf{F}^{-1}\mathbf{G}(n)\mathbf{C})\mathbf{K}(n, n-1)\mathbf{F}^H\mathbf{y}(n) - \mathbf{C}\hat{\mathbf{x}}(n|Y_{n-1}) \\
&= \hat{\mathbf{x}}(n|Y_{n-1}) - \mathbf{F}^{-1}\mathbf{G}(n)\mathbf{C}(n|Y_{n-1}) + \mathbf{F}^{-1}\mathbf{G}(n)\mathbf{y}(n) \\
&= (\mathbf{I} - \mathbf{F}^{-1}\mathbf{G}(n)\mathbf{C})\hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{F}^{-1}\mathbf{G}(n)\mathbf{y}(n)
\end{aligned}$$

(Note: An error was made in the first printing of the book.)

- (b) The innovations sequence  $\alpha(1), \alpha(2), \dots, \alpha(n)$  consists of samples that are orthogonal to each other; see Property 2, p.467 of the text. Orthogonality is synonymous with whitening. Moreover, invoking the very definition of the innovations process, we may refer to the Kalman filter as a whitening filter.

$$10.12 \quad \mathbf{K}_{XY}(n) = E\left\{[\mathbf{x}(n) - \hat{\mathbf{x}}(n|Y_{n-1})][\mathbf{y}(n) - \hat{\mathbf{y}}(n|Y_{n-1})]^H\right\}$$

$$\mathbf{K}_{YY}(n) = E\left\{[\mathbf{y}(n) - \hat{\mathbf{y}}(n|Y_{n-1})][\mathbf{y}(n) - \hat{\mathbf{y}}(n|Y_{n-1})]^H\right\}$$

Show that  $\mathbf{G}_f(n) = \mathbf{K}_{XY}(n)\mathbf{K}_{YY}^{-1}(n)$

The innovation vector  $\alpha(n) = \mathbf{y}(n) - \hat{\mathbf{y}}(n|Y_{n-1})$  - by definition.

Therefore,  $\mathbf{K}_{YY}(n) = E[\alpha(n)\alpha^H(n)] = \mathbf{R}(n)$

Also, by definition,

$$\mathbf{G}_f(n) = \mathbf{F}^{-1}(n+1, n)\mathbf{G}(n) = \mathbf{F}(n, n+1)E[\mathbf{x}(n+1)\alpha^H(n)]\mathbf{R}^{-1}(n)$$

$$\begin{aligned}
E[\mathbf{x}(n+1)\alpha^H(n)] &= \mathbf{F}(n+1, n)E[\mathbf{x}(n)\alpha^H(n)] = \mathbf{F}(n+1, n)E\left\{[\mathbf{x}(n) - \hat{\mathbf{x}}(n|Y_{n-1})]\alpha^H(n)\right\} \\
&= \mathbf{F}(n+1, n)\mathbf{K}_{XY}(n)
\end{aligned}$$

Here we have used the fact that the estimate  $\hat{\mathbf{x}}(n|Y_{n-1})$  is orthogonal to  $\underline{\alpha}^H(n)$ . Therefore, adding it inside the expectation term does not change the expectation value.

Using the results obtained:

$$\mathbf{G}_f(n) = \mathbf{F}(n, n-1)\mathbf{F}(n+1, n)\mathbf{K}_{XY}(n)\mathbf{R}^{-1}(n) = \mathbf{K}_{XY}(n)\mathbf{K}_{YY}^{-1}(n)$$

10.13 Let us consider the unforced dynamical model

$$\mathbf{x}(n+1) = \mathbf{F}(n+1, n)\mathbf{x}(n) \quad (1)$$

$$y(n) = \mathbf{u}^H(n)\mathbf{x}(n) + v(n) \quad (2)$$

$$\text{where } \mathbf{F}(n+1, n) = \frac{1}{\sqrt{\lambda}}\mathbf{I}$$

From (1) we can see that

$$\mathbf{x}(n) = \lambda^{-n/2}\mathbf{x}(0) \quad (3)$$

(a) Considering a deterministic multiple regression model  $d(n) = e_o(n) + \mathbf{w}_o^H \mathbf{u}(n)$ , we can write:

$$\left\{ \begin{array}{l} d^*(0) = \mathbf{u}^H(0)\mathbf{w}_o + e_o^*(0) \\ d^*(1) = \mathbf{u}^H(1)\mathbf{w}_o + e_o^*(1) \\ \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \\ d^*(n) = \mathbf{u}^H(n)\mathbf{w}_o + e_o^*(n) \end{array} \right\}$$

which represents a deterministic system of linear equations

(b) From Eqs. (1), (2) and (3), we have:

$$\left\{ \begin{array}{l} y(0)=\mathbf{u}^H(0)\mathbf{x}(0) + v(0) \\ y(1)=\mathbf{u}^H(1)\mathbf{x}(0)\lambda^{-1/2} + v(0) \\ \vdots \\ y(n)=\mathbf{u}^H(n)\mathbf{x}(0)\lambda^{-1/2} + v(0) \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} y(0)=\mathbf{u}^H(0)\mathbf{x}(0) + v(0) \\ \lambda^{1/2}y(1)=\mathbf{u}^H(1)\mathbf{x}(0)\lambda^{-1/2} + v(0) \\ \vdots \\ \lambda^{n/2}y(n)=\mathbf{u}^H(n)\mathbf{x}(0)\lambda^{-1/2} + v(0) \end{array} \right\}$$

which represents a stochastic system of linear equations.

- (c) Both the stochastic and the deterministic systems of linear simultaneous equations describe the same problem, and therefore should have a common solution. Comparing the two systems, we can set

$$\left\{ \begin{array}{l} \mathbf{x}(0)=\mathbf{w}_o \\ y(n)=\lambda^{-n/2}d^*(n) \\ v(n)=\lambda^{-n/2}e_o^*(n) \end{array} \right\}$$

- 10.14 The reason for the RLS operating satisfactorily is the fact that the minimum mean-square error follows the recursion (see Eq. (9.30))

$$E_{\min}(n) = \lambda E_{\min}(n-1) + \xi^*(n)e(n)$$

Hence, with  $\lambda$  in the interval  $0 < \lambda \leq 1$ , the stored value of the minimum mean-square error,  $E_{\min}(n-1)$ , is reduced by the factor  $\lambda$  as the minimum mean-square error is recursively updated.

- 10.15 Consider first the correspondence between  $\hat{\mathbf{x}}(1|Y_0)$  and  $\hat{\mathbf{w}}(0)$ . For the one-step prediction, we have

$$\hat{\mathbf{x}}(n+1|Y_n): \quad \lambda^{-n/2}\hat{\mathbf{w}}(n)$$

Therefore, setting  $n=0$ , we see that  $\hat{\mathbf{x}}(1|Y_0)$  in the Kalman filter corresponds to  $\hat{\mathbf{w}}(0)$  in the RLS algorithm.

Consider next the correlation matrix of the error in state prediction, which is defined by

$$\mathbf{K}(n): \quad \lambda^{-1}\mathbf{P}(n)$$

Putting  $n=0$ , we readily see that  $\mathbf{K}(0)$  in the Kalman filter corresponds to  $\lambda^{-1}\mathbf{P}(0)$  in the RLS algorithm.

10.16 The condition number of  $\mathbf{K}(n)$  is

$$\chi(\mathbf{K}) = \frac{\lambda_{\max}}{\lambda_{\min}}$$

Given that

$$\mathbf{K}(n) = \mathbf{U}(n)\mathbf{D}(n)\mathbf{U}^H(n)$$

we may write

$$\begin{aligned}\chi(\mathbf{K}) &= \chi(\mathbf{U}\mathbf{D}\mathbf{U}^H) \\ &\leq \chi(\mathbf{U})\chi(\mathbf{D})\chi(\mathbf{U}^H)\end{aligned}\tag{1}$$

The eigenvalues of an upper triangular matrix are the same as the diagonal elements of the matrix. For the situation at hand, the upper triangular matrix has 1's for all its diagonal elements. Hence, the  $\lambda_{\max}$  and  $\lambda_{\min}$  of  $\mathbf{U}(n)$  are both equal to one, and so

$$\chi(\mathbf{U}) = \chi(\mathbf{U}^H) = 1$$

Accordingly, Eq. (1) simplifies to

$$\chi(\mathbf{D}) \geq \chi(\mathbf{K})$$

10.17 Let

$$\mathbf{K}(n) = \mathbf{K}^{1/2}(n)\mathbf{K}^{H/2}(n)$$

Hence,

$$\chi(\mathbf{K}) \leq \chi(\mathbf{K}^{1/2})\chi(\mathbf{K}^{H/2}) = (\chi(\mathbf{K}^{1/2}))^2$$

The implication of this result is that the condition number of the square root  $\mathbf{K}^{1/2}(n)$  is the square root of the condition number of the original matrix  $\mathbf{K}(n)$ .

10.18 Adding the known vector  $\mathbf{d}(n)$  to the state equation is equivalent to assuming that the state noise vector  $\mathbf{v}_1(n)$  has a mean equal to  $\mathbf{d}(n)$ . In general, we have

$$\hat{\mathbf{x}}(n+1|Y_n) = \mathbf{F}(n+1, n)\hat{\mathbf{x}}(n|Y_n) + \hat{\mathbf{v}}_1(n|Y_n)$$

If therefore  $\mathbf{v}_1(n)$  has a mean  $\mathbf{d}(n)$ , it follows that

$$\hat{\mathbf{x}}(n+1|Y_n) = \mathbf{F}(n+1, n)\hat{\mathbf{x}}(n|Y_n) + \mathbf{d}(n)$$



## CHAPTER 11

11.1 To drive the square-root information filter, we proceed as follows. Let

$$\mathbf{A}(n) = \begin{bmatrix} \lambda^{1/2} \mathbf{K}^{-H/2}(n-1) & \lambda^{1/2} \mathbf{u}(n) \\ \hat{\mathbf{x}}^H(n|Y_{n-1}) \mathbf{K}^{-H/2}(n-1) & y^*(n) \\ \mathbf{0}^T & 1 \end{bmatrix}$$

Taking the Hermitian transpose:

$$\mathbf{A}^H(n) = \begin{bmatrix} \lambda^{1/2} \mathbf{K}^{-1/2}(n-1) & \mathbf{K}^{-1/2}(n-1) \hat{\mathbf{x}}(n|Y_{n-1}) & \mathbf{0} \\ \lambda^{1/2} \mathbf{u}^H(n) & y(n) & 1 \end{bmatrix}$$

Hence, postmultiplying  $\mathbf{A}(n)$  by  $\mathbf{A}^H(n)$ :

$$\mathbf{A}(n) \mathbf{A}^H(n) = \begin{bmatrix} \lambda \mathbf{K}^{-1}(n-1) + \lambda \mathbf{u}(n) \mathbf{u}^H(n) & \lambda^{1/2} \mathbf{K}^{-1}(n-1) \hat{\mathbf{x}}(n|Y_{n-1}) + \lambda^{1/2} \mathbf{u}(n) y(n) & \lambda^{1/2} \mathbf{u}(n) \\ \left( \lambda^{1/2} \hat{\mathbf{x}}(n|Y_{n-1}) \mathbf{K}^{-1}(n-1) + \lambda^{1/2} \mathbf{u}^H(n) y^*(n) \right) & \lambda \hat{\mathbf{x}}^H(n|Y_{n-1}) \mathbf{K}^{-1}(n-1) \hat{\mathbf{x}}(n-1|Y_{n-1}) & y^*(n) \\ \lambda^{1/2} \mathbf{u}^H(n) & y(n) & 1 \end{bmatrix}$$

This result pertains to the pre-array of the square-root information filter.

Consider next the post-array of the square-root information filter. Let

$$\mathbf{B}(n) = \begin{bmatrix} \mathbf{B}_{11}(n) & \mathbf{b}_{21}(n) & \mathbf{b}_{31}(n) \\ \mathbf{0}^T & \mathbf{b}_{22}(n) & \mathbf{b}_{32}(n) \end{bmatrix}$$

Taking the Hermitian transpose:

$$\mathbf{B}^H(n) = \begin{bmatrix} \mathbf{B}_{11}^H(n) & \mathbf{0} \\ \mathbf{b}_{21}^H(n) & \mathbf{b}_{22}^*(n) \\ \mathbf{b}_{31}^H(n) & \mathbf{b}_{32}^*(n) \end{bmatrix}$$

Hence, pre-multiplying  $\mathbf{B}(n)$  by  $\mathbf{B}^H(n)$ :

$$\mathbf{B}^H(n)\mathbf{B}(n) = \begin{bmatrix} \mathbf{B}_{11}^H(n)\mathbf{B}_{11}(n) & \mathbf{B}_{11}^H(n)\mathbf{b}_{21}(n) & \mathbf{B}_{11}^H(n)\mathbf{b}_{31}(n) \\ \mathbf{b}_{21}^H(n)\mathbf{B}_{11}(n) & \mathbf{b}_{21}^H(n)\mathbf{b}_{21}(n) + |\mathbf{b}_{22}(n)|^2 & \mathbf{b}_{21}^H(n)\mathbf{b}_{31}(n) + \mathbf{b}_{22}^*(n)\mathbf{b}_{32}(n) \\ \mathbf{b}_{31}^H(n)\mathbf{B}_{11}(n) & \mathbf{b}_{31}^H(n)\mathbf{b}_{31}(n) + \mathbf{b}_{32}^*(n)\mathbf{b}_{22}(n) & \mathbf{b}_{31}^H(n)\mathbf{b}_{31}(n) + |\mathbf{b}_{32}(n)|^2 \end{bmatrix}$$

Equating terms in  $\mathbf{A}(n)\mathbf{A}^H(n)$  to corresponding terms in  $\mathbf{B}^H(n)\mathbf{B}(n)$ , we get,

$$\begin{aligned} 1. \quad \mathbf{B}_{11}^H(n)\mathbf{B}_{11}(n) &= \lambda \mathbf{K}^{-1}(n-1) + \lambda \mathbf{u}(n)\mathbf{u}^H(n) \\ &= \mathbf{K}^{-1}(n) = \mathbf{K}^{-H/2}(n)\mathbf{K}^{-1/2}(n) \end{aligned}$$

Hence,

$$\mathbf{B}_{11}^H(n) = \mathbf{K}^{-H/2}(n)$$

$$\begin{aligned} 2. \quad \mathbf{B}_{11}^H(n)\mathbf{b}_{21}(n) &= \lambda^{1/2}\mathbf{K}^{-1}(n-1)\hat{\mathbf{x}}(n|Y_{n-1}) + \lambda^{1/2}\mathbf{u}(n)y(n) \\ &= \mathbf{K}^{-1}(n)\hat{\mathbf{x}}(n+1|Y_n) \end{aligned}$$

Hence,

$$\mathbf{b}_{21}(n) = \mathbf{K}^{-H/2}(n)\hat{\mathbf{x}}(n+1|Y_n)$$

$$3. \quad \mathbf{B}_{11}^H(n)\mathbf{b}_{31}(n) = \lambda^{1/2}\mathbf{u}(n)$$

Hence,

$$\mathbf{b}_{31}(n) = \lambda^{1/2} \mathbf{K}^{H/2}(n) \mathbf{u}(n)$$

$$4. \quad \mathbf{b}_{31}^H(n) \mathbf{b}_{31}(n) + |\mathbf{b}_{32}(n)|^2 = 1$$

Hence,

$$\begin{aligned} |\mathbf{b}_{32}(n)|^2 &= 1 - \lambda \mathbf{u}^H(n) \mathbf{K}(n) \mathbf{u}(n) \\ &= r^{-1}(n) \end{aligned}$$

That is,

$$\mathbf{b}_{32} = r^{-1/2}(n)$$

$$5. \quad \mathbf{b}_{21}^H(n) \mathbf{b}_{21}(n) + |\mathbf{b}_{22}(n)|^2 = \hat{\mathbf{x}}^H(n|Y_{n-1}) \mathbf{K}^{-1}(n-1) \hat{\mathbf{x}}(n|Y_{n-1}) + |y(n)|^2$$

Hence,

$$\begin{aligned} |\mathbf{b}_{22}(n)|^2 &= \hat{\mathbf{x}}^H(n|Y_{n-1}) \mathbf{K}^{-1}(n-1) \hat{\mathbf{x}}(n|Y_{n-1}) + |y(n)|^2 \\ &\quad - \hat{\mathbf{x}}(n+1|Y_n) \mathbf{K}^{-1}(n) \hat{\mathbf{x}}(n+1|Y_n) \end{aligned} \tag{1}$$

$$6. \quad \mathbf{b}_{31}^H(b) \mathbf{b}_{21}(n) + \mathbf{b}_{32}^*(n) b_{22}(n) = y(n)$$

Hence,

$$r^{-1/2}(n) \mathbf{b}_{22}(n) = y(n) - \lambda^{1/2} \mathbf{u}^H(n) \mathbf{K}^{1/2}(n) \mathbf{K}^{-1/2}(n) \hat{\mathbf{x}}(n+1|Y_n)$$

That is,

$$\mathbf{b}_{22}(n) = r^{1/2}(n) [y(n) - \lambda^{1/2} \mathbf{u}^H(n) \hat{\mathbf{x}}(n+1|Y_n)]$$

But, we know that

$$\hat{\mathbf{x}}(n+1|Y_n) = \lambda^{-1/2} \hat{\mathbf{x}}(n|Y_{n-1}) + \alpha(n) \mathbf{g}(n)$$

where  $\mathbf{g}(n)$  is the Kalman gain and  $\alpha(n)$  is the innovation. Therefore,

$$\begin{aligned}
\mathbf{b}_{22}(n) &= r^{1/2}(n)[y(n) - \mathbf{u}^H(n)\hat{\mathbf{x}}(n|Y_{n-1}) - \lambda^{1/2}\alpha(n)\mathbf{u}^H(n)\mathbf{g}(n)] \\
&= r^{1/2}(n)[\alpha(n) - \lambda^{1/2}\alpha(n)\mathbf{u}^H(n)\mathbf{g}(n)] \\
&= r^{1/2}(n)\alpha(n)[1 - \lambda^{1/2}\mathbf{u}^H(n)\mathbf{g}(n)]
\end{aligned}$$

But,

$$\lambda^{1/2}\mathbf{g}(n) = \mathbf{K}^{-1}(n)\mathbf{u}(n) \cdot r^{-1}(n)$$

Therefore,

$$\begin{aligned}
\mathbf{b}_{22}(n) &= r^{1/2}(n)\alpha(n)[1 - \mathbf{u}^H(n)\mathbf{K}^{-1}(n)\mathbf{u}(n)r^{-1}(n)] \\
&= r^{-1/2}(n)\alpha(n)[r(n) - \mathbf{u}^H(n)\mathbf{K}^{-1}(n)\mathbf{u}(n)] \\
&= r^{-1/2}(n)\alpha(n)
\end{aligned} \tag{2}$$

where we have used  $r(n) = 1 + \mathbf{u}^H(n)\mathbf{K}^{-1}(n)\mathbf{u}(n)$ .

#### Final Check

In a laborious way, it can be shown that Eq. (1) is satisfied exactly by the value of  $\mathbf{b}_{22}(n)$  defined in Eq. (2).

11.2 Consider a linear dynamical system described by the state-space model:

$$\mathbf{x}(n+1) = \lambda^{-1/2}\mathbf{x}(n) \tag{1}$$

$$y(n) = \mathbf{u}^H(n)\mathbf{x}(n) + v(n) \tag{2}$$

where  $v(n)$  is a Gaussian variable of zero mean and variance  $Q(n)$ . The Kalman filtering algorithm for the model is described by:

$$\mathbf{g}(n) = \lambda^{-1/2}\mathbf{K}(n-1)\mathbf{u}(n)R^{-1}(n)$$

$$R(n) = \mathbf{u}^H(n)\mathbf{K}(n-1)\mathbf{u}(n) + Q(n)$$

$$\alpha(n) = y(n) - \mathbf{u}^H(n)\hat{\mathbf{x}}(n|Y_{n-1})$$

$$\hat{\mathbf{x}}(n+1|Y_n) = \lambda^{-1/2}\hat{\mathbf{x}}(n|Y_{n-1}) + \mathbf{g}(n)\alpha(n)$$

$$\mathbf{K}(n) = \lambda^{-1}\mathbf{K}(n-1) - \lambda^{-1/2}\mathbf{g}(n)\mathbf{u}^H(n)\mathbf{K}(n-1)$$

Then, proceeding in a manner similar to that described in Chapter 11, we may formulate the extended square-root information filter for the state-space model of Eqs. (1) and (2) as follows:

$$\mathbf{K}^{-1}(n) = \lambda[\mathbf{K}^{-1}(n-1) + Q^{-1}(n)\mathbf{u}(n)\mathbf{u}^H(n)] \quad (3)$$

$$\mathbf{K}^{-1}(n)\hat{\mathbf{x}}(n+1|Y_n) = \lambda^{-1/2}[\mathbf{K}^{-1}(n-1)\hat{\mathbf{x}}(n|Y_{n-1}) + Q^{-1}(n)\mathbf{u}(n)y(n)] \quad (4)$$

where  $\mathbf{g}(n)$  is the Kalman gain (vector).

Thus, in light of Eqs. (3) and (4), we may formulate the following array structure for the square-root information filter:

$$\begin{bmatrix} \lambda^{1/2}\mathbf{K}^{-H/2}(n-1) & \lambda^{-1/2}Q^{-1/2}(n)\mathbf{u}(n) \\ \hat{\mathbf{x}}^H(n|Y_{n-1})\mathbf{K}^{-H/2}(n-1) & Q^{-1/2}(n)y(n) \\ \mathbf{0}^T & Q^{-1/2}(n) \end{bmatrix} \Theta(n) = \begin{bmatrix} \mathbf{K}^{-H/2}(n) & \mathbf{0} \\ \hat{\mathbf{x}}^H(n+1|Y_n)\mathbf{K}^{-H/2}(n) & R^{-1/2}(n)\alpha^*(n) \\ \lambda^{-1/2}Q^{-1}(n)\mathbf{u}^H(n)\mathbf{K}^{1/2}(n) & R^{-1/2}(n) \end{bmatrix}$$

This equation includes an ordinary square-root information filter as a special case. Specifically, putting  $Q(n) = 0$  we get the pre-array-to-post-array transformation for that filter.

11.3 (a) Verify the expression for the extended square-root information filter given by

$$\begin{bmatrix} \lambda^{1/2}\mathbf{K}^{-H/2}(n-1) & \lambda^{1/2}\mathbf{u}(n) \\ \hat{\mathbf{x}}(n|Y_{n-1})\mathbf{K}^{-H/2}(n-1) & y^*(n) \\ \mathbf{0}^T & 1 \\ \lambda^{1/2}\mathbf{K}^{1/2}(n-1) & \mathbf{0} \end{bmatrix} \Theta(n) = \begin{bmatrix} \mathbf{K}^{-H/2}(n-1) & \mathbf{0} \\ \hat{\mathbf{x}}(n+1|Y_n)\mathbf{K}^{-H/2}(n-1) & r^{1/2}(n)\alpha^*(n) \\ \lambda^{1/2}\mathbf{u}^H(n)\mathbf{K}^{1/2}(n) & r^{1/2}(n) \\ \mathbf{K}^{1/2}(n) & -\mathbf{g}(n)r^{1/2}(n) \end{bmatrix}$$

Let

$$\mathbf{A}(n) = \begin{bmatrix} \lambda^{1/2} \mathbf{K}^{-H/2}(n-1) & \lambda^{1/2} \mathbf{u}(n) \\ \hat{\mathbf{x}}(n|Y_{n-1}) \mathbf{K}^{-H/2}(n-1) & y^*(n) \\ \mathbf{0}^T & 1 \\ \lambda^{1/2} \mathbf{K}^{-1/2}(n-1) & \mathbf{0} \end{bmatrix}$$

$$\mathbf{B}^H(n) = \begin{bmatrix} \mathbf{B}_{11}^H(n) & \mathbf{0} \\ \mathbf{b}_{21}^H(n) & \mathbf{b}_{22}^*(n) \\ \mathbf{b}_{31}^H(n) & b_{32}^*(n) \\ \mathbf{B}_{41}^H(n) & \mathbf{B}_{42}^H(n) \end{bmatrix}$$

Then, equating the matrix product  $\mathbf{A}(n) \mathbf{A}^H(n)$  to the matrix product  $\mathbf{B}^H(n) \mathbf{B}(n)$ , and comparing their corresponding terms, we may say the following:

- Points 1 through 6 discussed in the solution to Problem 11.1 remain valid. Accordingly, the entries  $\mathbf{B}_{11}(n)$ ,  $\mathbf{b}_{21}(n)$ ,  $\mathbf{b}_{31}(n)$ ,  $\mathbf{b}_{22}(n)$ , and  $\mathbf{b}_{32}(n)$  will all have the values determined there.
- We have the following additional equations to consider:

$$1. \quad \mathbf{B}_{41}^H(n) \mathbf{B}_{11}^H(n) = \mathbf{I}$$

since  $\mathbf{B}_{11}(n) = \mathbf{K}^{-1/2}(n)$ , it follows that

$$\mathbf{B}_{41}^H(n) = \mathbf{K}^{1/2}(n) \tag{1}$$

$$2. \quad \mathbf{B}_{41}^H(n) \mathbf{b}_{31}(n) + \mathbf{B}_{42}^H(n) \mathbf{b}_{32}(n) = \mathbf{0}$$

Hence,

$$\mathbf{B}_{42}^H(n) = \mathbf{B}_{41}^H(n) \mathbf{b}_{31}(n) / \mathbf{b}_{32}(n)$$

Since  $\mathbf{b}_{32}(n) = r^{-1/2}(n)$  and  $\mathbf{b}_{31}^H(n) = \lambda^{1/2} \mathbf{u}^H(n) \mathbf{K}^{1/2}(n)$ ,

it follows that

$$\begin{aligned}\mathbf{B}_{42}^H(n) &= \lambda^{1/2} r^{1/2}(n) \mathbf{K}^{1/2}(n) \mathbf{K}^{-H/2}(n) \mathbf{u}(n) \\ &= -\lambda^{1/2} r^{1/2}(n) \mathbf{K}(n) \mathbf{u}(n)\end{aligned}\quad (2)$$

But from Kalman filter theory for the dynamical system being considered here, we know that

$$\lambda^{1/2} \mathbf{K}(n) \mathbf{u}(n) = \mathbf{g}(n)$$

Accordingly, we may simplify Eq. (2) to

$$\mathbf{B}_{42}^H(n) = -r^{1/2}(n) \mathbf{g}(n)$$

where  $\mathbf{g}(n)$  is the Kalman gain. This completes the evaluation of the post-array for the extended square-root information filter.

3. In addition to the two equations described above, we have several other relations that follow from  $\mathbf{A}(n) \mathbf{A}^H(n) = \mathbf{B}^H(n) \mathbf{B}(n)$ ; these additional relations merely provide a means to check the values already determined for the entries of the post-array.

(b) Using results from part (a), derive the extended QR-RLS algorithm.

In order to derive the extended QR-RLS algorithm we make use of the following relationships between the Kalman and RLS variables:

$$\begin{aligned}\mathbf{K}^{-1}(n) &\rightarrow \lambda \Phi(n) & \alpha(n) &\rightarrow \lambda^{-n/2} \xi^*(n) \\ r^{-1}(n) &\rightarrow \gamma(n) & y(n) &\rightarrow \lambda^{-n/2} d^*(n) \\ \mathbf{g}(n) &\rightarrow \lambda^{-1/2} \mathbf{k}(n) & \hat{\mathbf{x}}(n|Y_{n-1}) &\rightarrow \lambda^{-n/2} \hat{\mathbf{w}}(n-1)\end{aligned}$$

Therefore, the extended QR-RLS could be written as:

$$\begin{bmatrix} \lambda^{1/2} \Phi^{1/2}(n) & \mathbf{u}(n) \\ \lambda^{1/2} \mathbf{p}^H(n-1) & d(n) \\ \mathbf{0}^T & 1 \\ \lambda^{-1/2} \Phi^{-H/2}(n-1) & \mathbf{0} \end{bmatrix} \Theta(n) = \begin{bmatrix} \Phi^{1/2}(n) & \mathbf{0} \\ \mathbf{p}^H(n) & \xi(n) \gamma^{1/2}(n) \\ \mathbf{u}^H(n) \Phi^{-H/2}(n) & \gamma^{1/2}(n) \\ \Phi^{-H/2}(n) & -\mathbf{k}(n) \gamma^{-1/2}(n) \end{bmatrix}$$

where we used the same reasoning as while obtaining equations (11.40)-(11.47)

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + [\mathbf{k}(n)\gamma^{-1/2}(n)][\xi(n)\gamma^{-1/2}(n)]^*$$

11.4 We start with

$$\mathbf{Q}(n)\mathbf{A}(n) = \begin{bmatrix} \mathbf{R}(n) \\ \mathbf{O} \end{bmatrix}$$

Let

$$\mathbf{Q}(n) = \begin{bmatrix} \mathbf{Q}_1(n) \\ \mathbf{Q}_2(n) \end{bmatrix}$$

Hence,

$$\begin{bmatrix} \mathbf{Q}_1(n) \\ \mathbf{Q}_2(n) \end{bmatrix} \mathbf{A}(n) = \begin{bmatrix} \mathbf{R}(n) \\ \mathbf{O} \end{bmatrix}$$

$$\mathbf{A}(n) = [\mathbf{Q}_1^H, \mathbf{Q}_2^H(n)] \begin{bmatrix} \mathbf{R}(n) \\ \mathbf{O} \end{bmatrix}$$

$$= \mathbf{Q}_1^H(n) \mathbf{R}(n)$$

$$\mathbf{A}^H(n) = \mathbf{R}^H(n) \mathbf{Q}_1(n)$$

The projection matrix is therefore

$$\begin{aligned} \mathbf{P}(n) &= \mathbf{A}(n)(\mathbf{A}^H(n)\mathbf{A}(n))^{-1} \mathbf{A}^H(n) \\ &= \mathbf{Q}_1^H(n) \mathbf{R}(n)(\mathbf{R}^H(n) \mathbf{Q}_1(n) \mathbf{Q}_1^H(n) \mathbf{R}(n))^{-1} \mathbf{R}^H(n) \mathbf{Q}_1(n) \end{aligned}$$

Since  $\mathbf{Q}_1(n) \mathbf{Q}_1^H(n) = \mathbf{I}$ , we have

$$\mathbf{P}(n) = \mathbf{Q}_1^H(n) \mathbf{R}(n)(\mathbf{R}^H(n) \mathbf{R}(n))^{-1} \mathbf{R}^H(n) \mathbf{Q}_1(n)$$



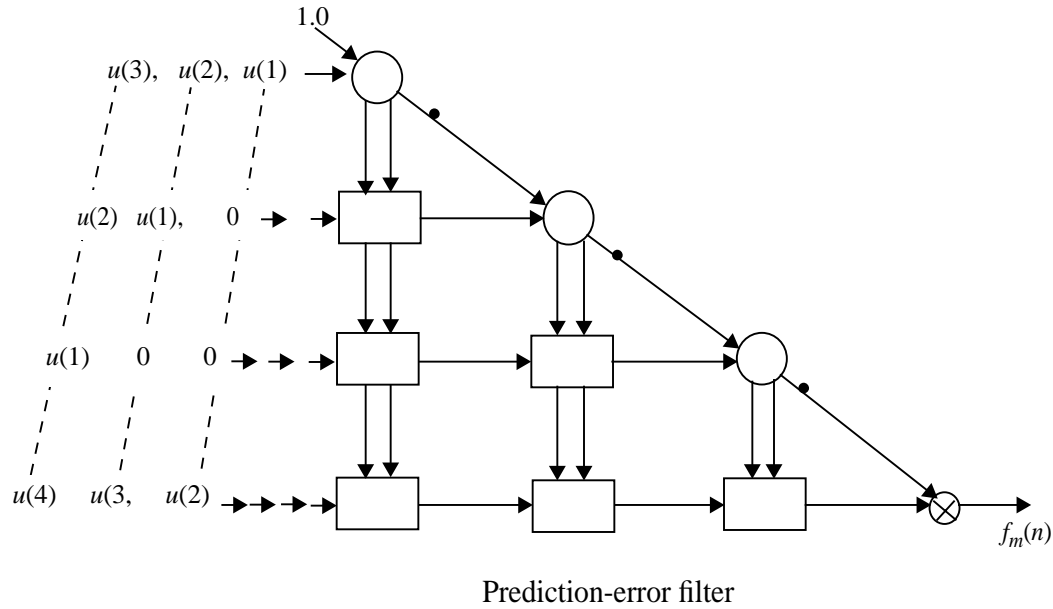
We also note that for an upper triangular matrix  $\mathbf{R}(n)$ ,

$$\mathbf{R}(n)(\mathbf{R}^H(n)\mathbf{R}(n))^{-1}\mathbf{R}^H(n) = \text{identity matrix}$$

Hence,

$$\mathbf{P}(n) = \mathbf{Q}_1^H(n)\mathbf{Q}_1(n)$$

- 11.5 In a prediction-error filter, the input  $u(n)$  represents the desired response and the tap inputs  $u(n-1), \dots, u(n-M)$  represent the variables used to estimate  $u(n)$ . Hence, we may restructure the inputs of the systolic array in Fig. 11.4 of the text in the following manner so that it operates as a prediction-error filter (illustrated here for order  $M = 3$ ):



- 11.6 We are given that

$$\mathbf{R}^H(n)\mathbf{a}(n) = \mathbf{s}$$

Hence,

$$\mathbf{a}(n) = \mathbf{R}^{-H}(n)\mathbf{s}$$

We note that  $\mathbf{R}^H(n)$  is a lower triangular matrix. Hence, given  $\mathbf{R}(n)$  and  $\mathbf{s}$ , the vector  $\mathbf{a}(n)$  may be computed by means of a linear section using forward substitution.

- 11.7 The output of the last interval cell in the bottom row of the triangular section is given by (see Fig. 11.2(a) of the text):

$$u_{\text{out}} = c u_{\text{in}} - s^* \lambda^{1/2} x$$

where  $c$  and  $s$  are the Givens parameters,  $u_{\text{in}}$  is the input to the cell and  $x$  is the stored value to the cell. At time  $n$ , we have

$$u_{\text{in}}(n) = d(n)$$

$$s^*(n) \lambda^{1/2} x(n-1) = c(n) \hat{\mathbf{w}}^H(n-1) \mathbf{u}(n)$$

where  $d(n)$  is the desired response,  $\mathbf{u}(n)$  is the input vector, and  $\hat{\mathbf{w}}(n-1)$  is the previous value of the least-squares weight vector. Hence,

$$\begin{aligned} u_{\text{out}} &= c(n) d(n) - c(n) \hat{\mathbf{w}}^H(n-1) \mathbf{u}(n) \\ &= c(n) [d(n) - \hat{\mathbf{w}}^H(n-1) \mathbf{u}(n)] \end{aligned}$$

Recognizing that

$$c(n) = \gamma^{1/2}(n)$$

and

$$d(n) - \hat{\mathbf{w}}^H(n-1) \mathbf{u}(n) = \xi(n),$$

we finally get

$$u_{\text{out}} = \gamma^{1/2}(n) \xi(n).$$

- 11.8 (a) We note that

$$\mathbf{R}^H(n) \mathbf{a}(n) = \mathbf{s}(\phi)$$

Hence,

$$\mathbf{a}(n) = \mathbf{R}^{-H}(n) \mathbf{s}(\phi)$$

Taking Hermitian transpose:

$$\mathbf{a}^H(n) = \mathbf{s}^H(\phi)\mathbf{R}^{-1}(n)$$

(b) For an MVDR beamformer:

$$\mathbf{R}(n)\hat{\mathbf{w}}(n) = \frac{\mathbf{a}(n)}{\mathbf{a}^H(n)\mathbf{a}(n)}$$

Hence,

$$\hat{\mathbf{w}}(n) = \frac{\mathbf{R}^{-1}(n)\mathbf{a}(n)}{\mathbf{a}^H(n)\mathbf{a}(n)}$$

We note that  $\mathbf{a}^H(n)\mathbf{a}(n)$  is a scalar. With  $\mathbf{R}(n)$  being an upper triangular matrix, it follows that the linear section performs backward substitution. The resulting output of this section is the weight vector  $\hat{\mathbf{w}}(n)$ .

11.9 Reformulating the prearray of the extended QR-RLQ algorithm in Problem 2 of Chapter 11 by making use of the correspondences in Table 11.3, we may write

$$\begin{bmatrix} \lambda^{1/2}\Phi^{1/2}(n-1) & \mathbf{u}(n) \\ \lambda^{1/2}\mathbf{a}^H(n-1) & 0 \\ \mathbf{0}^T & 1 \end{bmatrix} \Theta(n) = \begin{bmatrix} \Phi^{1/2}(n) & \mathbf{0} \\ \mathbf{a}^H(n) & -e'(n)\gamma^{-1/2}(n) \\ \mathbf{u}^H(n)\Phi^{-H/2}(n) & \gamma^{1/2}(n) \end{bmatrix}$$

Squaring both sides of this equation, and retaining the terms on the second rows:

$$\lambda\|\mathbf{a}(n-1)\|^2 = \|\mathbf{a}(n)\|^2 + \gamma^{-1}(n)|e'(n)|^2$$

The term  $e'(n)/\gamma^{1/2}(n)$  is recognized as an estimation error denoted by  $\epsilon(n)$ . Hence,

$$\lambda\|\mathbf{a}(n-1)\|^2 = \|\mathbf{a}(n)\|^2 + |\epsilon(n)|^2$$

11.10 The standard RLS filter is the covariance version of the Kalman filter. The inverse QR-RLS filter is the square-root version of the covariance Kalman filter. It follows therefore that the inverse QR-RLS filter is the square-root RLS filter.

## CHAPTER 12

$$\begin{aligned}
 12.1 \quad J_{fb, m} &= \frac{1}{2}(E[|f_{m-1}(n)|^2] + E[|b_{m-1}(n-1)|^2])(1 + |\kappa_m|^2) \\
 &\quad + \kappa_m E[f_{m-1}(n)b_{m-1}^*(n-1)] + \kappa_m^* E[b_{m-1}(n-1)f_{m-1}^*(n)]
 \end{aligned}$$

Differentiating with respect to a complex variable:

$$\begin{aligned}
 \frac{\partial J_{fb, m}}{\partial \kappa_m} &= \kappa_m (E[|f_{m-1}(n)|^2] + E[|b_{m-1}(n-1)|^2]) \\
 &\quad + E[b_{m-1}(n-1)f_{m-1}^*(n)] + E[b_{m-1}(n-1)f_{m-1}^*(n)] \\
 &= \kappa_m (E[|f_{m-1}(n)|^2] + E[|b_{m-1}(n-1)|^2]) \\
 &\quad + 2E[b_{m-1}(n-1)f_{m-1}^*(n)]
 \end{aligned}$$

12.2 (a) Suppose we write

$$f_m(i) = f_{m-1}(i) + \hat{\kappa}_m(n)b_{m-1}(i-1)$$

$$b_m(i) = b_{m-1}(i) + \hat{\kappa}_m(n)f_{m-1}(i)$$

Then substituting these relations into Burg's formula:

$$\begin{aligned}
 \hat{\kappa}_m(n) &= -\frac{2 \sum_{i=1}^n b_{m-1}(i-1)f_{m-1}^*(i)}{\sum_{i=1}^n |f_{m-1}(i)|^2 + |b_{m-1}(i-1)|^2} \\
 &= -\frac{2 \sum_{i=1}^n b_{m-1}(i-1)f_{m-1}^*(i) + 2b_{m-1}(n-1)f_{m-1}^*(n)}{E_{m-1}(n-1) + |f_{m-1}(n)|^2 + |b_{m-1}(n-1)|^2},
 \end{aligned}$$

Cross-multiplying and proceeding in a manner similar to that described after Eqs. (12.12) and (12.13) in the text, we finally get

$$\hat{\mathbf{k}}_m(n) = \hat{\mathbf{k}}_m(n-1) - \frac{f_{m-1}^*(n)b_m(n) + b_{m-1}(n-1)f_m^*(n)}{E_{m-1}(n-1)}$$

- (b) The algorithm so formulated is impractical because to compute the updated forward prediction error  $f_m(n)$  and backward prediction error  $b_m(n)$  we need to know the updated  $\hat{\mathbf{k}}_m(n)$ . This is not possible to do because the correction term for  $\hat{\mathbf{k}}_m(n)$  requires knowledge of  $f_m(n)$  and  $b_m(n)$ .

12.3 For the transversal filter of Fig. 12.6 in the text we have:

$$\text{tap-weight vector} = \mathbf{k}_M(n)$$

$$\text{tap-input vector} = \mathbf{u}_M(i), \quad i = 1, 2, \dots, n$$

$$\text{desired response, } d(i) = \begin{cases} 1 & i = n \\ 0 & i = n-1, \dots, 1 \end{cases}$$

The a posteriori estimation error equals

$$e(i) = d(i) - \mathbf{k}_M^H(n)\mathbf{u}_M(i), \quad i = 1, 2, \dots, n$$

The deterministic cross-correlation vector  $\phi(n)$  equals

$$\begin{aligned} \phi(n) &= \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_M d^*(i) \\ &= \mathbf{u}_M(n) \end{aligned}$$

We also note that

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

Hence, the sum of weighted error squares equals

$$\begin{aligned}
E_{\min}(n) &= E_d(n) - \phi^H(n) \hat{\mathbf{w}}(n) \\
&= 1 - \mathbf{u}_M^H(n) \mathbf{k}_m(n)
\end{aligned}$$

We note that the inner product  $\mathbf{u}_M^H(n) \mathbf{k}_m(n)$  is a real-valued scalar. Hence,

$$\begin{aligned}
E_{\min}(n) &= 1 - \mathbf{k}_m^H(n) \mathbf{u}_m(n) \\
&= \gamma_M(n)
\end{aligned}$$

12.4 We start with

$$\Phi_m(n) = \lambda \Phi_m(n-1) + \mathbf{u}_m(n) \mathbf{u}_m^H(n)$$

where  $\mathbf{u}_M(n)$  is the tap-input vector,  $\Phi_m(n-1)$  is the past value of the deterministic correlation matrix, and  $\Phi_m(n)$  is its present value. Hence,

$$\begin{aligned}
\lambda \Phi_m(n-1) &= \Phi_m(n) - \mathbf{u}_m(n) \mathbf{u}_m^H(n) \\
&= \Phi_m(n) [\mathbf{I} - \mathbf{u}_m(n) \mathbf{u}_m^H(n) \Phi_m^{-1}(n)]
\end{aligned}$$

where  $\mathbf{I}$  is the identity matrix. Hence, taking the determinants of both sides:

$$\lambda \det[\Phi_m(n-1)] = \det[\Phi_m(n)] \det[\mathbf{I} - \mathbf{u}_m(n) \mathbf{u}_m^H(n) \Phi_m^{-1}(n)] \quad (1)$$

But,

$$\begin{aligned}
\det[\mathbf{I} - \mathbf{u}_m(n) \mathbf{u}_m^H(n) \Phi_m^{-1}(n)] &= \det \left[ \mathbf{I} - \mathbf{u}_m^H(n) \Phi_m^{-1}(n) \mathbf{u}_m(n) \right] \\
&= 1 - \mathbf{u}_m^H(n) \Phi_m^{-1}(n) \mathbf{u}_m(n) \\
&= \gamma_m(n)
\end{aligned}$$

We may therefore rewrite Eq. (1) as

$$\lambda \det[\Phi_m(n-1)] = \det[\Phi_m(n)] \gamma_m(n)$$

Hence, we may express the conversion factor  $\gamma_m(n)$  as

$$\gamma_m(n) = \lambda \frac{\det[\Phi_m(n-1)]}{\det[\Phi_m(n)]}$$

12.5 (a) The  $(m+1)$ -by- $(m+1)$  correlation matrix  $\Phi_{m+1}$  may be expressed in the form

$$\Phi_{m+1}(n) = \begin{bmatrix} U(n) & \phi_1^H(n) \\ \phi_1(n) & \Phi_m(n-1) \end{bmatrix} \quad (1)$$

Define the inverse of this matrix as

$$\Phi_{m+1}^{-1}(n) = \begin{bmatrix} \alpha_1 & \beta_1^H \\ \beta_1 & \Gamma_1 \end{bmatrix} \quad (2)$$

Hence, from Eqs. (1) and (2):

$$\begin{aligned} \mathbf{I}_{m+1} &= \Phi_{m+1}(n) \Phi_{m+1}^{-1}(n) \\ &= \begin{bmatrix} U(n) & \phi_1^H(n) \\ \phi_1(n) & \Phi_m(n-1) \end{bmatrix} \begin{bmatrix} \alpha_1 & \beta_1^H \\ \beta_1 & \Gamma_1 \end{bmatrix} \\ &= \begin{bmatrix} U(n)\alpha_1 + \phi_1^H(n)\beta_1 & U(n)\beta_1^H + \phi_1^H(n)\Gamma_1 \\ \phi_1(n)\alpha_1 + \Phi_m(n-1)\beta_1 & \phi_1(n)\beta_1^H + \Phi_m(n-1)\Gamma_1 \end{bmatrix} \end{aligned}$$

From this relation we deduce the following four equations:

$$U(n)\alpha_1 + \phi_1^H(n)\beta_1 = 1 \quad (3)$$

$$U(n)\beta_1^H + \phi_1^H(n)\Gamma_1 = \mathbf{0} \quad (4)$$

$$\phi_1(n)\alpha_1 + \Phi_m(n-1)\beta_1 = \mathbf{0} \quad (5)$$

$$\phi_1(n)\beta_1^H + \Phi_m(n-1)\Gamma_1 = \mathbf{I}_m \quad (6)$$

Eliminate  $\beta_1$  between Eqs. (3) and (5):

$$U(n)\alpha_1 - \phi_1^H(n)\Phi_m^{-1}(n-1)\phi_1(n)\alpha_1 = 1$$

Hence,

$$\alpha_1 = \frac{1}{U(n) - \phi_1^H(n)\Phi_m^{-1}(n-1)\phi_1(n)} \quad (7)$$

which is real-valued. Correspondingly,

$$\begin{aligned} \beta_1 &= -\Phi_m^{-1}(n-1)\phi_1(n)\alpha_1 \\ &= \frac{-\Phi_m^{-1}(n-1)\phi_1(n)}{U(n) - \phi_1^H(n)\Phi_m^{-1}(n-1)\phi_1(n)} \end{aligned} \quad (8)$$

From Eq. (6):

$$\begin{aligned} \Gamma_1 &= \Phi_m^{-1}(n-1) - \Phi_m^{-1}(n-1)\phi_1(n)\beta_1^H \\ &= \Phi_m^{-1}(n-1) + \frac{\Phi_m^{-1}(n-1)\phi_1(n)\phi_1^H(n)\Phi_m^{-1}(n-1)}{U(n) - \phi_1^H(n)\Phi_m^{-1}(n-1)\phi_1(n)} \end{aligned} \quad (9)$$

Check:

Substitute Eqs. (8) and (9) into the left hand side of Eq. (4):

$$\begin{aligned} U(n)\beta_1^H + \phi_1^H(n)\Gamma_1 &= U(n) \left( \frac{-\phi_1^H(n)\Phi_m^{-1}(n-1)}{U(n) - \phi_1^H(n)\Phi_m^{-1}(n-1)\phi_1(n)} \right) \\ &\quad + \phi_1^H(n)\Phi_m^{-1}(n-1) \end{aligned}$$



$$\begin{aligned}
& + \frac{\phi_1^H(n) \Phi_m^{-1}(n-1) \phi_1(n) \phi_1^H(n) \Phi_m^{-1}(n-1)}{U(n) - \phi_1^H(n) \Phi_m^{-1}(n-1) \phi_1(n)} \\
& = \mathbf{0}
\end{aligned}$$

This agrees with the right-hand side of Eq. (4).

From Eq. (7) we note that

$$\alpha_1 = \frac{1}{F_m(n)}$$

From Eq. (8) we note that

$$\beta_1 = -\frac{\hat{\mathbf{w}}_f(n)}{F_m(n)}$$

From Eq. (9) we note that

$$\Gamma_1 = \begin{bmatrix} 0 & \mathbf{0}_m^T \\ \mathbf{0}_m & \Phi_m^{-1}(n-1) \end{bmatrix} + \frac{\hat{\mathbf{w}}_f(n) \hat{\mathbf{w}}_f^H(n)}{F_m(n)}$$

Hence, we may express the inverse matrix of Eq. (1) as follows:

$$\begin{aligned}
\Phi_{m+1}^{-1}(n) &= \begin{bmatrix} 0 & \mathbf{0}_m^T \\ \mathbf{0}_m & \Phi_m^{-1}(n-1) \end{bmatrix} + \frac{1}{F_m(n)} \begin{bmatrix} 1 & -\hat{\mathbf{w}}_f^H(n) \\ -\hat{\mathbf{w}}_f(n) & \hat{\mathbf{w}}_f(n) \hat{\mathbf{w}}_f^H(n) \end{bmatrix} \\
&= \begin{bmatrix} 0 & \mathbf{0}_m^T \\ \mathbf{0}_m & \Phi_m^{-1}(n-1) \end{bmatrix} + \frac{1}{F_m(n)} \begin{bmatrix} 1 \\ -\hat{\mathbf{w}}_f(n) \end{bmatrix} [1, -\hat{\mathbf{w}}_f^H(n)] \\
&= \begin{bmatrix} 0 & \mathbf{0}_m^T \\ \mathbf{0}_m & \Phi_m^{-1}(n-1) \end{bmatrix} + \frac{1}{F_m(n)} \mathbf{a}_m(n) \mathbf{a}_m^H(n)
\end{aligned}$$

(b) Consider next the second form of the  $(m+1)$ -by- $(m+1)$  correlation matrix  $\Phi_{m+1}(n)$  given by

$$\Phi_{m+1}(n) = \begin{bmatrix} \Phi_m(n) & \phi_2(n) \\ \phi_2^H(n) & U(n-m) \end{bmatrix} \quad (10)$$

Define the inverse of this matrix as

$$\Phi_{m+1}^{-1}(n) = \begin{bmatrix} \Gamma_2 & \beta_2 \\ \beta_2^H & \alpha_2 \end{bmatrix} \quad (11)$$

Using Eqs. (10) and (11):

$$\begin{aligned} \mathbf{I}_{m+1} &= \Phi_{m+1}(n) \Phi_{m+1}^{-1}(n) \\ &= \begin{bmatrix} \Phi_m(n) & \phi_2(n) \\ \phi_2^H(n) & U(n-m) \end{bmatrix} \begin{bmatrix} \Gamma_2 & \beta_2 \\ \beta_2^H & \alpha_2 \end{bmatrix} \\ &= \begin{bmatrix} \Phi_m(n)\Gamma_2 + \phi_2(n)\beta_2^H & \Phi_m(n)\beta_2 + \phi_2(n)\alpha_2 \\ \phi_2^H(n)\Gamma_2 + U(n-m)\beta_2^H & \phi_2^H(n)\beta_2 + U(n-m)\alpha_2 \end{bmatrix} \end{aligned}$$

We thus deduce the following four relations:

$$\Phi_m(n)\Gamma_2 + \phi_2(n)\beta_2^H = \mathbf{I}_m \quad (12)$$

$$\Phi_m(n)\beta_2 + \phi_2(n)\alpha_2 = \mathbf{0} \quad (13)$$

$$\phi_2^H(n)\Gamma_2 + U(n-m)\beta_2^H = \mathbf{0} \quad (14)$$

$$\phi_2^H(n)\beta_2 + U(n-m)\alpha_2 = 1 \quad (15)$$

Eliminate  $\beta_2$  between Eqs. (13) and (15):

$$-\phi_2^H(n)\Phi_m^{-1}(n)\phi_2(n)\alpha_2 + U(n-m)\alpha_2 = 1$$

Hence,

$$\alpha_2 = \frac{1}{U(n-m) - \phi_2^H(n)\Phi_m^{-1}(n)\Phi_m^{-1}(n)\phi_2(n)} \quad (16)$$

Correspondingly,  $\beta_2$  equals

$$\begin{aligned} \beta_2 &= -\Phi_m^{-1}(n)\phi_2(n)\alpha_2 \\ &= -\frac{\Phi_m^{-1}(n)\phi_2(n)}{U(n-m) - \phi_2^H(n)\Phi_m^{-1}(n)\phi_2(n)} \end{aligned} \quad (17)$$

Substitute Eq. (17) in (12) and solve for  $\Gamma_2$ :

$$\begin{aligned} \Gamma_2 &= \Phi_m^{-1}(n) - \Phi_m^{-1}(n)\phi_2(n)\beta_2^H \\ &= \Phi_m^{-1}(n) + \frac{\Phi_m^{-1}(n)\phi_2(n)\phi_2^H(n)\Phi_m^{-1}(n)}{U(n-m) - \phi_2^H(n)\Phi_m^{-1}(n)\phi_2(n)} \end{aligned} \quad (18)$$

Check:

Substitute Eqs. (17) and (18) into the left-hand side of Eq. (14):

$$\begin{aligned} &\phi_2^H(n)\Phi_m^{-1}(n) + \frac{\phi_2^H(n)\Phi_m^{-1}(n)\phi_2(n)\phi_2^H(n)\Phi_m^{-1}(n)}{U(n-m) - \phi_2^H(n)\Phi_m^{-1}(n)\phi_2(n)} \\ &\quad - U(n-m)\frac{\phi_2^H(n)\Phi_m^{-1}(n)}{U(n-m) - \phi_2^H(n)\Phi_m^{-1}(n)\phi_2(n)} = \mathbf{0} \end{aligned}$$

which agrees with the right-hand side of Eq. (14).

Next, we note that

$$\alpha_2 = \frac{1}{B_m(n)}$$

$$\beta_2 = -\frac{\mathbf{g}(n)}{B_m(n)}$$

$$\Gamma_2 = \begin{bmatrix} \Phi_m^{-1}(n) & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{1}{B_m(n)} \mathbf{w}_b(n) \mathbf{w}_b^H(n)$$

Hence, we may express the inverse matrix  $\Phi_{m+1}^{-1}(n)$  in the alternative form:

$$\begin{aligned} \Phi_{m+1}^{-1}(n) &= \begin{bmatrix} \Phi_m^{-1}(n) & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{1}{B_m(n)} \begin{bmatrix} \mathbf{w}_b(n) \mathbf{w}_b^H(n) & -\mathbf{w}_b(n) \\ -\mathbf{w}_b^H(n) & 1 \end{bmatrix} \\ &= \begin{bmatrix} \Phi_m^{-1}(n) & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{1}{B_m(n)} \begin{bmatrix} -\mathbf{w}_b(n) \\ 1 \end{bmatrix} [-\mathbf{w}_b^H(n), 1] \\ &= \begin{bmatrix} \Phi_m^{-1}(n) & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} + \frac{1}{B_m(n)} \mathbf{c}_m(n) \mathbf{c}_m^H(n) \end{aligned}$$

12.6 (a) From the solution to part (a) of Problem 12.5, we have

$$\Phi_{m+1}^{-1}(n) = \begin{bmatrix} 0 & \mathbf{0}_m^T \\ \mathbf{0}_m & \Phi_m^{-1}(n-1) \end{bmatrix} + \frac{1}{F_m(n)} \mathbf{a}_m(m) \mathbf{a}_m^H(n) \quad (1)$$

Correspondingly, the input vector  $\mathbf{u}_{m+1}(n)$  is partitioned as follows:

$$\mathbf{u}_{m+1}(n) = \begin{bmatrix} u(n) \\ \mathbf{u}_{m+1}(n-1) \end{bmatrix} \quad (2)$$

From the definition of the conversion factor, we have

$$\gamma_{m+1}(n) = 1 - \mathbf{u}_{m+1}^H(n-1)\Phi_{m+1}^{-1}(n)\mathbf{u}_{m+1}(n) \quad (3)$$

Therefore, substituting Eqs. (1) and (2) into (3):

$$\begin{aligned} \gamma_{m+1}(n) &= 1 - \mathbf{u}_m^H(n-1)\Phi_m^{-1}(n-1)\mathbf{u}_m(n-1) \\ &\quad - \frac{1}{F_m(n)}\mathbf{u}_{m+1}^H(n)\mathbf{a}_m(m)\mathbf{a}_m^H(n)\mathbf{u}_{m+1}(n) \\ &= \gamma_{m+1}(n-1) - \frac{|f_m(n)|^2}{F_m(n)} \end{aligned} \quad (4)$$

where we have used the fact that

$$f_m(n) = \mathbf{a}_m^H(n)\mathbf{u}_{m+1}(n)$$

(b) From the solution to part (b) of Problem 12.5, we have

$$\Phi_{m+1}^{-1}(n) = \begin{bmatrix} \Phi_m^{-1}(n) & \mathbf{0}_m \\ \mathbf{0}_m^T & 0 \end{bmatrix} + \frac{1}{B_m(n)}\mathbf{c}_m(n)\mathbf{c}_m^H(n) \quad (5)$$

This time, we partition the input vector  $\mathbf{u}_{m+1}(n)$  as follows:

$$\mathbf{u}_{m+1}(n) = \begin{bmatrix} \mathbf{u}_m(n) \\ u(n-m) \end{bmatrix} \quad (6)$$

Therefore, substituting Eqs. (5) and (6) into (3):

$$\begin{aligned} \gamma_{m+1}(n) &= 1 - \mathbf{u}_m^H(n)\Phi_m^{-1}(n)\mathbf{u}_m(n) \\ &\quad - \frac{1}{B_m(n)}\mathbf{u}_{m+1}^H(n)\mathbf{c}_m(n)\mathbf{c}_m^H(n)\mathbf{u}_{m+1}(n) \\ &= \gamma_m(n) - |b_m(n)|^2/B_m(n) \end{aligned} \quad (7)$$

where we have made use of the fact that

$$b_m(n) = \mathbf{c}_m^H(n) \mathbf{u}_{m+1}(n)$$

(c) We invoke the following property of the conversion factor:

$$\gamma_m(n-1) = \frac{f_m(n)}{\eta_m(n)} \quad (8)$$

Also, we note that

$$F_m(n) = \lambda F_m(n-1) + \eta_m(n) f_m^*(n) \quad (9)$$

Therefore eliminating  $\eta_m(n)$  between Eqs. (8) and (9):

$$F_m(n) = \lambda F_m(n-1) + \frac{|f_m(n)|^2}{\gamma_m(n-1)} \quad (10)$$

Finally, eliminating  $|f_m(n)|^2$  between Eqs. (4) and (10):

$$\begin{aligned} \gamma_{m+1}(n) &= \gamma_m(n-1) - \frac{\gamma_m(n-1)}{F_m(n)} [F_m(n) - \lambda F_m(n-1)] \\ &= \lambda F_m(n-1) \gamma_m(n-1) / F_m(n) \end{aligned}$$

(d) Next, we invoke another property of the conversion factor:

$$\gamma_m(n) = \frac{b_m(n)}{\beta_m(n)} \quad (11)$$

We also note that

$$B_m(n) = \lambda B_m(n-1) + b_m^*(n) \beta_m(n) \quad (12)$$

Therefore, eliminating  $\beta_m(n)$  between Eqs. (11) and (12):

$$B_m(n) = \lambda B_m(n-1) + |b_m(n)|^2 / \gamma_m(n) \quad (13)$$

Eliminating  $|b_m(n)|^2$  between Eqs. (7) and (13):

$$\begin{aligned} \gamma_{m+1}(n) &= \gamma_m(n) - \frac{\gamma_m(n)}{B_m(n)} [B_m(n) - \lambda B_m(n-1)] \\ &= \lambda B_m(n-1) \gamma_m(n) / B_m(n) \end{aligned}$$

$$\begin{aligned} 12.7 \quad (a) \quad F_m(n) &= \sum_{i=1}^n \lambda^{n-i} |f_m(i)|^2 \\ &= \sum_{i=1}^n \lambda^{n-i} f_m(i) f_m^*(i) \\ &= \sum_{i=1}^n \lambda^{n-i} [u(i) - \hat{\mathbf{w}}_{f,m}^H(n) \mathbf{u}_m(i-1)] f_m^*(i) \\ &= \sum_{i=1}^n \lambda^{n-i} u(i) f_m^*(i) - \left( \hat{\mathbf{w}}_{f,m}^H(n) \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_m(i-1) f_m^*(i) = \mathbf{0} \right) \\ &= \sum_{i=1}^n \lambda^{n-i} u(i) f_m^*(i) \\ &= \sum_{i=1}^n \lambda^{n-i} [\eta_m(i) + \hat{\mathbf{w}}_{f,m}^H(n-1) \mathbf{u}_m(i-1)] f_m^*(i) \\ &= \sum_{i=1}^n \lambda^{n-i} \eta_m(i) f_m^*(i) + \left( \hat{\mathbf{w}}_{f,m}^H(n-1) \sum_{i=1}^n \mathbf{u}_m(i-1) f_m^*(i) = \mathbf{0} \right) \\ &= \sum_{i=1}^n \lambda^{n-i} \eta_m(i) f_m^*(i) \end{aligned}$$

$$\begin{aligned}
&= \sum_{i=1}^n \lambda^{n-i} \eta_m(i) f_m^*(i) + \eta_m(n) f_m^*(n) \\
&= \lambda F_m(n-1) + \eta_m(n) f_m^*(n)
\end{aligned}$$

(b) Following a similar procedure, we may use the relations

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

and

$$\sum_{i=1}^n \lambda^{n-i} \mathbf{u}_m(i) b_m^*(i) = \mathbf{0}$$

to derive the recursion

$$B_m(n) \lambda B_m(n-1) + \beta_m(n) b_m^*(n)$$

12.8 (a) We start with

$$\begin{aligned}
\Delta_{m-1}(n) &= \sum_{i=1}^n \lambda^{n-i} b_{m-1}(i-1) f_{m-1}^*(i) \\
&= \sum_{i=1}^n \lambda^{n-i} [u(i-m) - \hat{\mathbf{w}}_{b, m-1}^H(n-2) \mathbf{u}_{m-1}(i-1)] f_{m-1}^*(i) \\
&= \sum_{i=1}^n \lambda^{n-i} u(i-m) f_{m-1}^*(i) - \left( \hat{\mathbf{w}}_{b, m-1}^H(n-2) \sum_{i=1}^n \cancel{\mathbf{u}_{m-1}(i-1) f_{m-1}^*(i)} = \mathbf{0} \right) \\
&= \sum_{i=1}^n \lambda^{n-i} u(i-m) f_{m-1}^*(i)
\end{aligned}$$



$$\begin{aligned}
&= \sum_{i=1}^n \lambda^{n-i} [\beta_{m-1}(i-1) + \hat{\mathbf{w}}_{b, m-1}^H(n-1) \mathbf{u}_{m-1}(i)] f_{m-1}^*(i) \\
&= \sum_{i=1}^n \lambda^{n-i} \beta_{m-1}(i-1) f_{m-1}^*(i) + \left( \hat{\mathbf{w}}_{b, m-1}^H(n-1) \sum_{i=1}^n \cancel{\mathbf{u}_{m-1}(i-1)} f_{m-1}^*(i) = \mathbf{0} \right) \\
&= \sum_{i=1}^n \lambda^{n-i} \beta_{m-1}(i-1) f_{m-1}^*(i) \\
&= \sum_{i=1}^n \lambda^{n-i} \beta_{m-1}(i-1) f_{m-1}^*(i) + \beta_{m-1}(n-1) f_{m-1}^*(n) \\
&= \lambda \Delta_{m-1}(n-1) + \beta_{m-1}(n-1) f_{m-1}^*(n) \tag{1}
\end{aligned}$$

Comparing this equation with Eq. (12.69) of the text, we deduce the equivalence

$$\eta_{m-1}^*(n) b_{m-1}(n-1) = \beta_{m-1}(n-1) f_{m-1}^*(n) \tag{2}$$

(b) Applying Eqs. (12.48) and (12.49) of the text, we may write

$$\eta_{m-1}^*(n) = \frac{f_{m-1}^*(n)}{\gamma_{m-1}(n-1)} \tag{3}$$

$$b_{m-1}(n-1) = \gamma_{m-1}(n-1) \beta_{m-1}(n-1) \tag{4}$$

Multiplying Eqs. (3) and (4):

$$\begin{aligned}
\eta_{m-1}^*(n) b_{m-1}(n-1) &= \frac{f_{m-1}^*(n)}{\gamma_{m-1}(n-1)} \cdot \gamma_{m-1}(n-1) \beta_{m-1}(n-1) \\
&= f_{m-1}^*(n) \beta_{m-1}(n-1)
\end{aligned}$$

which proves Eq. (2)

- 12.9 (a) Let  $\Phi_{m+1}(n)$  denote the  $(m+1)$ -by- $(m+1)$  correlation matrix of the tap-input vector  $\mathbf{u}_{m+1}(i)$  applied to the forward prediction error filter of order  $m$ , where  $1 \leq i \leq n$ . Let  $\mathbf{a}_m(n)$  denote the tap-weight vector of this filter, and  $F_m(n)$  denote the corresponding sum of weighted prediction-error squares. We may characterize this filter by the augmented normal equations

$$\Phi_{m+1}(n)\mathbf{a}_m(n) = \begin{bmatrix} F_m(n) \\ \mathbf{0}_m \end{bmatrix} \quad (1)$$

where  $\mathbf{0}_m$  is the  $m$ -by-1 null vector. The correlation matrix  $\Phi_{m+1}(n)$  may be partitioned in two different ways, depending on how we interpret the first or last element of the tap-input vector  $\mathbf{u}_{m+1}(i)$ . The form of partitioning that we like to use first is the one that enables us to relate the tap-weight vector  $\mathbf{a}_m(n)$ , pertaining to prediction order  $m$ , to the tap-weight vector  $\mathbf{a}_{m-1}(n)$ , pertaining to prediction order  $m-1$ . This aim is realized by using

$$\Phi_{m+1}(n) = \begin{bmatrix} \Phi_m(n) & \vdots & \phi_2(n) \\ \hline \phi_2^H(n) & \vdots & U_2(n) \end{bmatrix} \quad (2)$$

where  $\Phi_m(n)$  is the  $m$ -by- $m$  correlation matrix of the tap-input vector  $\mathbf{u}_m(i)$ ,  $\phi_2(n)$  is the  $m$ -by-1 cross-correlation vector between  $\mathbf{u}_m(i)$  and  $u(i-m)$ , and  $U_2(n)$  is the sum of weighted squared values of the input  $\mathbf{u}(i-m)$  for  $1 \leq i \leq n$ . Note that  $U_2(n)$  is zero for  $n - m \leq 0$ . We postmultiply both sides of Eq. (2) by an  $(m+1)$ -by-1 vector whose first  $m$  elements are defined by the vector  $\mathbf{a}_{m-1}(n)$  and whose last element equals zero. We may thus write

$$\begin{aligned} \Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} &= \begin{bmatrix} \Phi_m(n) & \vdots & \phi_2(n) \\ \hline \phi_2^H(n) & \vdots & U_2(n) \end{bmatrix} \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \Phi_m(n)\mathbf{a}_{m-1}(n) \\ \phi_2^H(n)\mathbf{a}_{m-1}(n) \end{bmatrix} \end{aligned} \quad (3)$$

Both  $\Phi_m(n)$  and  $\mathbf{a}_{m-1}(n)$  have the same time argument  $n$ . Furthermore, in the first line of Eq. (3), they are both positioned in such a way that when the matrix multiplication

is performed  $\Phi_m(n)$  becomes postmultiplied by  $\mathbf{a}_{m-1}(n)$ . For a forward prediction-error filter of order  $m-1$ , evaluated at time  $n$ , the set of augmented normal equations defined in Eq. (1) takes the form

$$\Phi_m(n)\mathbf{a}_{m-1}(n) = \begin{bmatrix} F_{m-1}(n) \\ \mathbf{0}_{m-1} \end{bmatrix}$$

Define the scalar

$$\Delta_{m-1}(n) = \phi_2^H(n)\mathbf{a}_{m-1}(n) \quad (4)$$

Accordingly, we may rewrite Eq. (3) as

$$\Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} = \begin{bmatrix} F_{m-1}(n) \\ \mathbf{0}_{m-1} \\ \Delta_{m-1}(n) \end{bmatrix} \quad (5)$$

(b) For a definition of  $\Delta_{m-1}(n)$ , we have

$$\Delta_{m-1}(n) = \sum_{i=1}^n \lambda^{n-i} f_{m-1}^*(i) b_{m-1}(i-1) \quad (6)$$

For another definition of this same quantity, we have

$$\Delta_{m-1}(n) = \phi_2^H(n)\mathbf{a}_{m-1}(n) \quad (7)$$

where

$$\phi_2(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_m(i) u^*(i-1) \quad (8)$$

To show that these two definitions are equivalent, we first substitute Eq. (8) into (7):

$$\Delta_{m-1}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{u}_m^H(i) \mathbf{a}_{m-1}(n) u(i-m) \quad (9)$$

From the definition of the forward prediction error, we have

$$f_{m-1}(i) = \mathbf{a}_{m-1}^H(n) \mathbf{u}_m(i), \quad 1 \leq i \leq n$$

We may therefore rewrite Eq. (9) as

$$\Delta_{m-1}(n) = \sum_{i=1}^n \lambda^{n-i} f_{m-1}^*(i) u(i-m) \quad (10)$$

Next, from the definition of the backward prediction error, we have

$$b_{m-1}(i) = u(i-m) - \sum_{k=1}^{m-1} w_{b, m-1, k}^*(n) u(i-k) \quad (11)$$

where  $w_{b, m-1, k}(n)$  is the  $k$ th element of the backward predictor's coefficient vector. Therefore, eliminating  $u(i-m)$  between Eqs. (10) and (11):

$$\begin{aligned} \Delta_{m-1}(n) &= \sum_{i=1}^n \lambda^{n-i} f_{m-1}^*(i) b_{m-1}(i) \\ &\quad + \sum_{i=1}^n \sum_{k=1}^{m-1} \lambda^{n-i} w_{b, m-1, k}^*(n) f_{m-1}^*(i) u(i-k) \end{aligned} \quad (12)$$

But, the tap inputs  $u(i-1), u(i-2), \dots, u(i-m+1)$  are the very inputs involved in computing the forward prediction error. From the principle of orthogonality, we therefore have

$$\sum_{k=1}^{m-1} \lambda^{n-i} f_{m-1}^*(i) u(i-k) = 0 \quad \text{for all } i$$

That is,

$$\Delta_{m-1}(n) = \sum_{i=1}^n \lambda^{n-i} f_{m-1}^*(i) b_{m-1}(i) = \phi_2^H(n) \mathbf{a}_{m-1}(n)$$

which is the desired result.

- (c) Consider next the backward prediction-error filter of order  $m$ . Let  $\mathbf{c}_m(n)$  denote its tap-weight vector, and  $B_m(n)$  denote the corresponding sum of weighted prediction-error squares. This filter is characterized by the augmented normal equations written in the matrix form:

$$\Phi_{m+1}(n)\mathbf{c}_m(n) = \begin{bmatrix} \mathbf{0}_m \\ B_m(n) \end{bmatrix} \quad (13)$$

where  $\Phi_{m+1}(n)$  is as defined previously, and  $\mathbf{0}_m$  is the  $m$ -by-1 null vector. This time we use the other partitioned form of the correlation matrix  $\Phi_m(n)$ , as shown by

$$\Phi_{m+1}(n) = \begin{bmatrix} U_1(n) & \vdots & \phi_1^H(n) \\ \text{---} & \text{---} & \text{---} \\ \phi_1(n) & \vdots & \Phi_1(n-1) \end{bmatrix} \quad (14)$$

where  $U_1(n)$  is the sum of weighted squared values of the input  $u(i)$  for the time interval  $1 \leq i \leq n$ ,  $\phi_1(n)$  is the  $m$ -by-1 cross-correlation vector between  $u(i)$  and the tap-input vector  $\mathbf{u}_m(i-1)$ , and  $\Phi_1(n-1)$  is the  $m$ -by- $m$  correlation matrix of  $\mathbf{u}_m(i-1)$ . Correspondingly, we postmultiply  $\Phi_{m+1}(n)$  by an  $(m+1)$ -by-1 vector whose first element is zero and whose  $m$  remaining elements are defined by the tap-weight vector  $\mathbf{c}_{m-1}(n-1)$  that pertains to a backward prediction-error filter of order  $m-1$ . We may thus write

$$\begin{aligned} \Phi_{m+1}(n) \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} &= \begin{bmatrix} U_1(n) & \phi_1^H(n) \\ \phi_1(n) & \Phi_m(n-1) \end{bmatrix} \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} \\ &= \begin{bmatrix} \phi_1^H(n)\mathbf{c}_{m-1}(n-1) \\ \Phi_m(n-1)\mathbf{c}_{m-1}(n-1) \end{bmatrix} \end{aligned} \quad (15)$$

Both  $\Phi_m(n-1)$  and  $\mathbf{c}_{m-1}(n-1)$  have the same time argument,  $n-1$ . Also, they are both positioned in the first line of Eq. (15) in such a way that, when the matrix multiplication is performed,  $\Phi_m(n-1)$  becomes postmultiplied by  $\mathbf{c}_{m-1}(n-1)$ . For a

backward prediction-error filter of order  $m-1$ , evaluated at time  $n-1$ , the set of augmented normal equations in Eq. (13) takes the form

$$\Phi_m(n-1)\mathbf{c}_{m-1}(n-1) = \begin{bmatrix} \mathbf{0}_{m-1} \\ B_{m-1}(n-1) \end{bmatrix}$$

Define the second scalar

$$\Delta'_{m-1}(n-1) = \phi_1^H(n)\mathbf{c}_{m-1}(n-1) \quad (16)$$

where the prime is intended to distinguish this new parameter from  $\Delta_{m-1}(n-1)$ . Accordingly, we may rewrite Eq. (15) as

$$\Phi_{m+1}(n) \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} = \begin{bmatrix} \Delta'_{m-1}(n) \\ \mathbf{0}_{m-1} \\ B_{m-1}(n-1) \end{bmatrix} \quad (17)$$

- (d) The parameters  $\Delta_{m-1}(n-1)$  and  $\Delta'_{m-1}(n-1)$ , defined by Eqs. (4) and (16), respectively, are in actual fact the complex conjugate of one another; that is,

$$\Delta'_{m-1}(n) = \Delta_{m-1}^*(n) \quad (18)$$

where  $\Delta_{m-1}^*(n)$  is the complex conjugate of  $\Delta_{m-1}(n)$ . We prove this relation in three stages:

1. We premultiply both sides of Eq. (5) by the row vector

$$[0, \mathbf{c}_{m-1}^H(n-1)]$$

where the superscript  $H$  denotes Hermitian transposition. The result of this matrix multiplication is the scalar

$$[0, \mathbf{c}_{m-1}^H(n-1)]\Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} = [0, \mathbf{c}_{m-1}^H(n-1)] \begin{bmatrix} F_{m-1}(n-1) \\ \mathbf{0}_{m-1} \\ \Delta_{m-1}(n) \end{bmatrix}$$

$$= \Delta_{m-1}(n) \quad (19)$$

where we have used the fact that the last element of  $\mathbf{c}_{m-1}(n-1)$  equals unity.

2. We apply Hermitian transposition to both sides of Eq. (17), and use the Hermitian property of the correlation matrix  $\Phi_{m+1}(n)$ , thereby obtaining

$$[0, \mathbf{c}_{m-1}^H(n-1)]\Phi_{m+1}(n) = [\Delta_{m-1}'^*(n), \mathbf{0}_{m-1}^T, B_{m-1}(n-1)]$$

where  $\Delta_{m-1}'^*(n)$  is the complex conjugate of  $\Delta_{m-1}'(n)$  and  $B_{m-1}(n-1)$  is real valued. Next we use this relation to evaluate the scalar

$$\begin{aligned} [0, \mathbf{c}_{m-1}(n-1)]\Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} \\ = \Delta_{m-1}'^*(n) \mathbf{0}_{m-1}^T, B_{m-1}(n-1) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} \\ = \Delta_{m-1}'^*(n) \end{aligned} \quad (20)$$

where we have used the fact that the first element of  $\mathbf{a}_{m-1}(n)$  equals unity.

3. Comparison of Eqs. (19) and (20) immediately yields the relation of Eq. (18) between the parameters  $\Delta_{m-1}(n)$  and  $\Delta_{m-1}'(n)$ .
- (e) We are now equipped with the relations needed to derive the desired time-update for recursive computation of the parameter  $\Delta_{m-1}(n)$ .

Consider the  $m$ -by-1 tap-weight vector  $\mathbf{a}_{m-1}(n-1)$  that pertains to a forward prediction-error filter of order  $m-1$ , evaluated at time  $n-1$ . The reason for considering time  $n-1$  will become apparent presently. Since the leading element of the vector  $\mathbf{a}_{m-1}(n-1)$  equals unity, we may express  $\Delta_{m-1}(n)$  as follows [see Eq. (18) and (20)]:

$$\Delta_{m-1}(n) = [\Delta_{m-1}(n), \mathbf{0}_{m-1}^T, B_{m-1}(n-1)] \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \quad (21)$$

Taking the Hermitian transpose of both sides of Eq. (17), recognizing the Hermitian property of  $\Phi_{m+1}(n)$  and using the relation of Eq. (19), we get

$$[0, \mathbf{c}_{m-1}(n-1)]\Phi_{m+1}(n) = [\Delta_{m-1}(n), \mathbf{0}_{m-1}^T, B_{m-1}(n-1)] \quad (22)$$

Hence, substitution of Eq. (22) into (21) yields

$$\Delta_{m-1}(n) = [0, \mathbf{c}_{m-1}(n-1)]\Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \quad (23)$$

But the correlation matrix  $\Phi_{m+1}(n)$  may be time-updated as follows:

$$\Phi_{m+1}(n) = \lambda\Phi_{m+1}(n-1) + \mathbf{u}_{m+1}(n)\mathbf{u}_{m+1}^H(n) \quad (24)$$

Accordingly, we may use this relation for  $\Phi_{m+1}(n)$  to rewrite Eq. (23) as

$$\begin{aligned} \Delta_{m-1}(n) &= \lambda[0, \mathbf{c}_{m-1}^H(n-1)]\Phi_{m+1}(n-1) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \\ &\quad + [0, \mathbf{c}_{m-1}^H(n-1)]\mathbf{u}_{m+1}(n)\mathbf{u}_{m+1}^H(n) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \end{aligned} \quad (25)$$

Next, we recognize from the definition of forward a priori prediction error that

$$\begin{aligned} \mathbf{u}_{m+1}^H(n) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} &= [\mathbf{u}_m^H(n), u^*(n-m)] \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \\ &= \mathbf{u}_m^H(n)\mathbf{a}_{m-1}(n-1) \\ &= \eta_{m-1}^*(n) \end{aligned} \quad (26)$$

and from the definition of the backward a posteriori prediction error that



$$\begin{aligned}
[0, \mathbf{c}_{m-1}^H(n-1)] \mathbf{u}_{m+1}(n) &= [0, \mathbf{c}_{m-1}^H(n-1)] \begin{bmatrix} u(n) \\ \mathbf{u}_m(n-1) \end{bmatrix} \\
&= \mathbf{c}_{m-1}^H(n-1) \mathbf{u}_m(n-1) \\
&= b_{m-1}(n-1)
\end{aligned} \tag{27}$$

Also, by substituting  $n-1$  for  $n$  into Eq. (5), we have

$$\Phi_{m+1}(n-1) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} = \begin{bmatrix} F_{m-1}(n-1) \\ \mathbf{0}_{m-1} \\ \Delta_{m-1}(n-1) \end{bmatrix}$$

Hence, using this relation and the fact that the last element of the tap-weight vector  $\mathbf{c}_{m-1}(n-1)$ , pertaining to the backward prediction-error filter, equals unity, we may write the first term on the right-hand side of Eq. (25), except for  $\lambda$ , as

$$\begin{aligned}
[0, \mathbf{c}_{m-1}^H(n-1)] \Phi_{m+1}(n-1) \begin{bmatrix} \mathbf{a}_{m-1}(n-1) \\ 0 \end{bmatrix} \\
&= [0, \mathbf{c}_{m-1}^H(n-1)] \begin{bmatrix} F_{m-1}(n-1) \\ \mathbf{0}_{m-1} \\ \Delta_{m-1}(n-1) \end{bmatrix} \\
&= \Delta_{m-1}(n-1)
\end{aligned} \tag{28}$$

Finally, substituting Eqs. (26), (27), and (28) into (25), we may express the time-update recursion for  $\Delta_{m-1}(n)$  simply as

$$\Delta_{m-1}(n) = \lambda \Delta_{m-1}(n-1) + b_{m-1}(n-1) \eta_{m-1}^*(n) \tag{29}$$

which is the desired result.

12.10 (a) We start with the relations

$$\Phi_{m+1}(n) \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} = \begin{bmatrix} F_{m-1}(n) \\ \mathbf{0}_{m-1} \\ \Delta_{m-1}(n) \end{bmatrix} \quad (1)$$

and

$$\Phi_{m+1}(n) \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} = \begin{bmatrix} \Delta_{m-1}^*(n) \\ \mathbf{0}_{m-1} \\ B_{m-1}(n-1) \end{bmatrix} \quad (2)$$

Multiplying Eq. (2) by the ratio  $\Delta_{m-1}(n)/B_{m-1}(n-1)$  and subtracting the result from Eq. (1):

$$\begin{aligned} \Phi_{m+1}(n) \left[ \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} - \frac{\Delta_{m-1}(n)}{B_{m-1}(n-1)} \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} \right] \\ = \begin{bmatrix} F_{m-1}(n) - |\Delta_{m-1}(n)|^2 / B_{m-1}(n-1) \\ \mathbf{0}_m \end{bmatrix} \end{aligned} \quad (3)$$

Equation (3) represents the augmented normal equations for forward prediction with order  $m$ , as shown by

$$\Phi_{m+1}(n) \mathbf{a}_{m-1}(n) = \begin{bmatrix} F_m(n) \\ \mathbf{0}_m \end{bmatrix} \quad (4)$$

on the basis of which we may immediately write

$$F_m(n) = F_{m-1}(n) - \frac{|\Delta_{m-1}(n)|^2}{B_{m-1}(n-1)}$$

(b) Multiplying Eq. (1) by  $\Delta_{m-1}^*(n)/F_{m-1}(n)$  and subtracting the result from Eq. (2):

$$\begin{aligned}
\Phi_{m+1}(n) & \left[ \begin{bmatrix} 0 \\ \mathbf{c}_{m-1}(n-1) \end{bmatrix} - \frac{\Delta_{m-1}^*(n)}{F_{m-1}(n)} \begin{bmatrix} \mathbf{a}_{m-1}(n) \\ 0 \end{bmatrix} \right] \\
& = \begin{bmatrix} \mathbf{0}_m \\ B_{m-1}(n-1) - |\Delta_{m-1}(n)|^2 / F_{m-1}(n) \end{bmatrix}
\end{aligned} \tag{5}$$

Equation (5) represents the augmented normal equations for backward prediction with order  $m$ , as shown by

$$\Phi_{m+1}(n) \mathbf{c}_{m-1}(n) = \begin{bmatrix} \mathbf{0}_m \\ B_{m-1}(n) \end{bmatrix} \tag{6}$$

on the basis of which we may immediately write

$$B_m(n) = B_{m-1}(n-1) - |\Delta_{m-1}(n)|^2 / F_{m-1}(n)$$

12.11 (a) From part (a) of problem 12.5, we have

$$\Phi_{M+1}^{-1}(n) = \begin{bmatrix} 0 & \mathbf{0}_M^T \\ \mathbf{0}_M & \Phi_M^{-1}(n-1) \end{bmatrix} + \frac{1}{F_M(n)} \mathbf{a}_M(n) \mathbf{a}_M^H(n).$$

Similarly, from part (b) of the same problem, we have

$$\Phi_{M+1}^{-1}(n) = \begin{bmatrix} \Phi_M^{-1}(n) & \mathbf{0}_M \\ \mathbf{0}_M^T & 0 \end{bmatrix} + \frac{1}{B_M(n)} \mathbf{c}_M(n) \mathbf{c}_M^H(n).$$

Subtracting these two equations gives  $\Phi_{M+1}^{-1}(n) - \Phi_{M+1}^{-1}(n) = \mathbf{0}$ , or

$$\mathbf{0} = \begin{bmatrix} 0 & \mathbf{0}_M^T \\ \mathbf{0}_M & \Phi_M^{-1}(n-1) \end{bmatrix} + \frac{1}{F_M(n)} \mathbf{a}_M(n) \mathbf{a}_M^H(n) - \begin{bmatrix} \Phi_M^{-1}(n) & \mathbf{0}_M \\ \mathbf{0}_M^T & 0 \end{bmatrix} - \frac{1}{B_M(n)} \mathbf{c}_M(n) \mathbf{c}_M^H(n)$$

This is easily rearranged as

$$\begin{bmatrix} \Phi_M^{-1}(n) & \mathbf{0}_M \\ \mathbf{0}_M^T & 0 \end{bmatrix} - \begin{bmatrix} 0 & \mathbf{0}_M^T \\ \mathbf{0}_M & \Phi_M^{-1}(n-1) \end{bmatrix} = \frac{1}{F_M(n)} \mathbf{a}_M(n) \mathbf{a}_M^H(n) - \frac{1}{B_M(n)} \mathbf{c}_M(n) \mathbf{c}_M^H(n).$$

(b) From page 441, equation (9.16) we have the basic matrix recursion

$$\Phi_M^{-1}(n) = \lambda^{-1} \Phi_M^{-1}(n-1) - \lambda^{-1} \frac{\lambda^{-1} \Phi_M^{-1}(n-1) \mathbf{u}(n) \mathbf{u}^H(n) \Phi_M^{-1}(n-1)}{1 + \lambda^{-1} \mathbf{u}^H(n) \Phi_M^{-1}(n-1) \mathbf{u}(n)}$$

From Eq. (9.18) of the text, we may introduce the gain vector

$$\mathbf{k}_M(n) = \frac{\lambda^{-1} \Phi_M^{-1}(n-1) \mathbf{u}(n)}{1 + \lambda^{-1} \mathbf{u}^H(n) \Phi_M^{-1}(n-1) \mathbf{u}(n)}$$

which yields the expression

$$\Phi_M^{-1}(n) = \lambda^{-1} \Phi_M^{-1}(n-1) - \lambda^{-1} \mathbf{k}_M(n) \mathbf{u}^H(n) \Phi_M^{-1}(n-1)$$

Now, again from Eq. (9.18) of the text, we have

$$\lambda^{-1} \mathbf{u}^H(n) \Phi_M^{-1}(n-1) = \mathbf{k}_M^H(n) (1 + \lambda^{-1} \mathbf{u}^H(n) \Phi_M^{-1}(n-1) \mathbf{u}(n))$$

But from Eq. (10.100) of the text, we may introduce the conversion factor as

$$\gamma_M(n) = \frac{1}{1 + \lambda^{-1} \mathbf{u}^H(n) \Phi_M^{-1}(n-1) \mathbf{u}(n)}$$

so that the basic matrix inversion lemma becomes

$$\Phi_M^{-1}(n) = \lambda^{-1} \Phi_M^{-1}(n-1) - \frac{\mathbf{k}_M(n) \mathbf{k}_M^H(n)}{\gamma_M(n)}$$

(c) The result of part (b) can be rearranged as

$$\Phi_M^{-1}(n-1) = \lambda \Phi_M^{-1}(n) + \lambda \frac{\mathbf{k}_M(n) \mathbf{k}_M^H(n)}{\gamma_M(n)}$$

Inserting this recursion into part (a), we have

$$\begin{bmatrix} \Phi_M^{-1}(n) & \mathbf{0}_M \\ \mathbf{0}_M^T & 0 \end{bmatrix} - \begin{bmatrix} 0 & \mathbf{0}_M^T \\ \mathbf{0}_M & \lambda \Phi_M^{-1}(n) + \lambda \mathbf{k}_M(n) \mathbf{k}_M^H(n) / \gamma_M(n) \end{bmatrix} \\ = \frac{1}{F_M(n)} \mathbf{a}_M(n) \mathbf{a}_M^H(n) - \frac{1}{B_M(n)} \mathbf{c}_M(n) \mathbf{c}_M^H(n)$$

This, in turn, can be rearranged as

$$\begin{bmatrix} \Phi_M^{-1}(n) & \mathbf{0}_M \\ \mathbf{0}_M^T & 0 \end{bmatrix} - \lambda \begin{bmatrix} 0 & \mathbf{0}_M^T \\ \mathbf{0}_M & \Phi_M^{-1}(n) \end{bmatrix} \\ = \frac{\mathbf{a}_M(n) \mathbf{a}_M^H(n)}{F_M(n)} + \lambda \begin{bmatrix} 0 \\ \mathbf{k}_M(n) \end{bmatrix} \frac{\begin{bmatrix} 0 & \mathbf{k}_M^H(n) \end{bmatrix}}{\gamma_M(n)} - \frac{\mathbf{c}_M(n) \mathbf{c}_M^H(n)}{B_M(n)}.$$

(d) Now we multiply the solution from (c) from the left by

$$[1, (z/\sqrt{\lambda}), (z/\sqrt{\lambda})^2, \dots, (z/\sqrt{\lambda})^M], \text{ and from the right by}$$

$$[1, (w/\sqrt{\lambda}), ((w)/\sqrt{\lambda})^2, \dots, w/\sqrt{\lambda})^M]^H. \text{ First note that, if we set}$$

$$P(z, w^*) = [1(z/\sqrt{\lambda}) \dots (z/\sqrt{\lambda})^M] \begin{bmatrix} \Phi_M^{-1}(n) & \mathbf{0}_M \\ \mathbf{0}_M^T & 0 \end{bmatrix} \begin{bmatrix} 1 \\ w^*/\sqrt{\lambda} \\ \vdots \\ (w^*/\sqrt{\lambda})^M \end{bmatrix}$$

then, upon displacing this matrix one position along the main diagonal, we obtain

$$[1(z/\sqrt{\lambda})\dots(z/\sqrt{\lambda})^M]\lambda \begin{bmatrix} 0 & \mathbf{0}_M^T \\ \mathbf{0}_M & \Phi_M^{-1}(n) \end{bmatrix} \begin{bmatrix} 1 \\ w^*/\sqrt{\lambda} \\ \vdots \\ (w^*/\sqrt{\lambda})^M \end{bmatrix} = zw^*P(z, w^*)$$

Thus the left-hand side of part (c) yields the two-variable polynomial

$$(1 - zw^*)P(z, w^*)$$

Similarly, rewrite the right-hand side as the sum and difference of dyads:

$$\frac{\mathbf{a}_M(n)}{\sqrt{F_M(n)}} \frac{\mathbf{a}_M^H(n)}{\sqrt{F_M(n)}} + \sqrt{\frac{\lambda}{\gamma_M(n)}} \begin{bmatrix} 0 \\ \mathbf{k}_M(n) \end{bmatrix} \left( \sqrt{\frac{\lambda}{\gamma_M(n)}} \begin{bmatrix} 0 & \mathbf{k}_M^H(n) \end{bmatrix} \right) - \frac{\mathbf{c}_M(n)}{\sqrt{B_M(n)}} \frac{\mathbf{c}_M^H(n)}{\sqrt{B_M(n)}}$$

Accordingly, if we set

$$A(z) = [1, z/\sqrt{\lambda}\dots(z/\sqrt{\lambda})^M] \frac{\mathbf{a}_M(n)}{\sqrt{F_M(n)}}$$

$$K(z) = [1, z/\sqrt{\lambda}\dots(z/\sqrt{\lambda})^M] \sqrt{\frac{\lambda}{\gamma_M(n)}} \begin{bmatrix} 0 \\ \mathbf{k}_M(n) \end{bmatrix}$$

$$C(z) = [1, z/\sqrt{\lambda}\dots(z/\sqrt{\lambda})^M] \frac{\mathbf{c}_M(n)}{\sqrt{B_M(n)}}$$

then we will have

$$[1, z/\sqrt{\lambda}\dots(z/\sqrt{\lambda})^M] \frac{\mathbf{a}_M(n)}{\sqrt{F_M(n)}} \frac{\mathbf{a}_M^H(n)}{\sqrt{F_M(n)}} \begin{bmatrix} 1 \\ w^*/\sqrt{\lambda} \\ \vdots \\ (w^*/\sqrt{\lambda})^M \end{bmatrix} = A(z)A^*(w),$$

and likewise for the remaining terms. Putting all this together, we finally obtain

$$(1 - zw^*)P(z, w^*) = A(z)A^*(w) + K(z)K^*(w) - B(z)B^*(w)$$

(e) If we set  $z = w = e^{j\omega}$ , then the result of part (d) reads as

$$\underbrace{(1 - e^{j\omega} e^{(-j)\omega})}_{= 0} P(e^{j\omega}, e^{-j\omega}) = A(e^{j\omega})A^*(e^{j\omega}) + K(e^{j\omega})K^*(e^{j\omega}) - C(e^{j\omega})C^*(e^{j\omega})$$

This gives

$$|A(e^{j\omega})|^2 + |K(e^{j\omega})|^2 = |C(e^{j\omega})|^2, \quad \text{for all } \omega.$$

(f) If we set  $w^* = z^*$  in the result of part (d), we have

$$\begin{aligned} (1 - |z|^2)P(z, z^*) &= A(z)A^*(z) + K(z)K^*(z) - C(z)C^*(z) \\ &= |A(z)|^2 + |K(z)|^2 - |C(z)|^2 \end{aligned} \quad (1)$$

Now, since  $\Phi_M^{-1}(n)$  is a positive definite matrix, it is true that

$$P(z, z^*) = [1, z/\sqrt{\lambda}, \dots, (z/\sqrt{\lambda})^{M-1}] \Phi_m^{-1}(n) \begin{bmatrix} 1 \\ z^*/\sqrt{\lambda} \\ \vdots \\ (z^*/\sqrt{\lambda})^M \end{bmatrix} > 0, \quad \text{for all } z.$$

Accordingly, the right-hand side of Eq. (1) above must take the same sign as  $(1 - |z|^2)$ , which gives

$$|A(z)|^2 + |K(z)|^2 - |C(z)|^2 \begin{cases} < 0, & |z| > 1; \\ = 0, & |z| = 1; \\ > 0, & |z| < 1. \end{cases}$$

(g) From the first inequality of part (f), we have

$$|A(z)|^2 + |K(z)|^2 - |C(z)|^2 < 0, \quad \text{in } |z| > 1. \quad (2)$$

Now, if  $C(z)$  were to have a zero with modulus greater than one, that is,

$$C(z_0) = 0, \quad \text{with } |z_0| > 1$$

then we would have

$$|A(z_0)|^2 + |K(z_0)|^2 - \underbrace{|C(z_0)|^2}_0 = |A(z_0)|^2 + |K(z_0)|^2 \geq 0.$$

With  $|z_0| > 1$ , we obtain a contradiction to Eq. (2) above. Accordingly,  $C(z)$  can have no zeros in  $|z| > 1$ , so that it may be determined without phase ambiguity from  $A(z)$  and  $K(z)$  by way of spectral factorization using the result of part (e).

#### 12.12 (a) Backward prediction

The first three lines of Table 12.4 follow from the state-space models of Eq. (12.87) through (12.95), together with Table 10.3 of Chapter 10 in the text. For the remaining three entries, we may proceed as follows:

$$\begin{aligned} \text{(a) } \mathbf{K}(n-1) &\leftrightarrow \lambda^{-1} \Phi^{-1}(n-1) = \lambda^{-1} \left( \sum_{i=1}^{n-1} \lambda^{n-i+1} |\varepsilon_{f, m-1}(i-1)|^2 \right)^{-1} \\ &= \lambda^{-1} F_{m-1}^{-1}(n-1) \end{aligned}$$

$$\begin{aligned} \text{(b) } \mathbf{g}(n) &\leftrightarrow \lambda^{-1/2} \mathbf{k}(n) = \lambda^{-1/2} F_{m-1}^{-1}(n) \varepsilon_{f, m-1}(n) \\ &= \lambda^{-n/2} \gamma_{m-1}^{1/2}(n-1) \beta_{m-1}^*(n-1) + \eta_{m-1}^*(n) \kappa_{b, m}(n-1) \\ &= \lambda^{-n/2} \gamma_{m-1}^{1/2}(n-1) \beta_{m-1}^*(n) \\ e(n) &\leftrightarrow \lambda^{-n/2} (\varepsilon_{b, m-1}^*(n-1) + \varepsilon_{f, m-1}^*(n) \kappa_{b, m}(n)) \\ &= \lambda^{-n/2} \gamma_{m-1}^{1/2}(n-1) (b_{m-1}^*(n-1) + f_{m-1}^*(n) \kappa_{b, m}(n)) \\ &= \lambda^{-n/2} \gamma_{m-1}^{1/2}(n-1) b_m^*(n) \end{aligned}$$

Hence,

$$r(n) = \frac{\alpha(n)}{e(n)} = \gamma_{m-1}(n-1) \left( \frac{B_m(n)}{b_m(n)} \right)$$



$$= \frac{\gamma_{m-1}(n-1)}{\gamma_m(n)}$$

(b) The relations for joint-process estimation follow a similar procedure.

12.13 (a) The correction in the update equation for  $E_{\min}(n)$  is defined by the product term  $\alpha_m(n)e_m^*(n)$ . This correction term is also equal to  $|\varepsilon_{m+1}(n)|^2$ . Hence

$$\varepsilon_{m+1}(n)\varepsilon_{m+1}^*(n) = \alpha_m(n)e_m^*(n)$$

By definition

$$|\varepsilon_m(n)| = \sqrt{|e_m(n)| \cdot |\alpha_m(n)|}$$

We know  $\alpha_m(n)e_m^*(n)$  is real. Hence, we must have

$$\arg[\varepsilon_m(n)] = \arg[e_m(n)]$$

Moreover, it is natural to have

$$\arg[\varepsilon_m(n)] = \arg[e_m(n)]$$

which goes to show that

$$\arg[\varepsilon_m(n)] = \arg[e_m(n)] = \arg[\alpha_m(n)]$$

(b) The correction term in the update equation for  $B_{\min}(n)$  is defined by the product term  $\psi_m(n)b_m^*(n)$ . This correction term is also equal to  $|\varepsilon_{b,m}(n)|^2$ . Hence,

$$\varepsilon_{b,m}(n)\varepsilon_{b,m}^*(n) = \psi_m(n)b_m^*(n)$$

By definition,

$$|\varepsilon_{b,m}(n)| = \sqrt{|b_m(n)| \cdot |\psi_m(n)|}$$

We know  $\psi_m(n)b_m^*(n)$  is real. Hence, we must have

$$\arg[b_m(n)] = \arg[\psi_m(n)]$$

Moreover, it is natural to have

$$\arg[\varepsilon_{b,m}(n)] = \arg[b_m(n)]$$

Therefore,

$$\arg[\varepsilon_{b,m}(n)] = \arg[b_m(n)] = \arg[\psi_m(n)]$$

- (c) The correction term in the update equation for  $F_m(n)$  is  $\eta_m(n)f_m^*(n)$  which is also equal to  $|\varepsilon_{f,m}(n)|^2$ . Hence,

$$\varepsilon_{f,m}(n)\varepsilon_{f,m}^*(n) = \eta_m(n)f_m^*(n)$$

By definition,

$$|\varepsilon_{f,m}(n)| = \sqrt{|f_m(n)| \cdot |\eta_m(n)|}$$

We know  $\eta_m(n)f_m^*(n)$  is real. Hence, we must have

$$\arg[f_m(n)] = \arg[\eta_m(n)]$$

Moreover, it is natural to have

$$\arg[\varepsilon_{f,m}(n)] = \arg[f_m(n)]$$

Therefore,

$$\arg[\varepsilon_{f,m}(n)] = \arg[f_m(n)]\arg[\eta_m(n)]$$

12.14 We start with

$$c_{b,m-1}(n) = \frac{\lambda^{1/2} B_{m-1}^{1/2}(n-1)}{B_{m-1}^{1/2}(n)} \quad (1)$$

Also, we note that

$$c_{f, m-1}(n) = \frac{\lambda^{1/2} F_{m-1}^{1/2}(n-1)}{F_{m-1}^{1/2}(n)} \quad (2)$$

But from the solution to part (d) of Problem 12.6:

$$\frac{\lambda^{1/2} B_{m-1}^{1/2}(n-1)}{B_{m-1}^{1/2}(n)} \gamma_{m-1}^{1/2}(n) = \gamma_m^{1/2}(n) \quad (3)$$

We next adapt the relation of part (c) of Problem 12.6 to our present situation:

$$\gamma_m^{1/2}(n) = \frac{\lambda^{1/2} F_{m-1}^{1/2}(n-1)}{F_m^{1/2}(n)} \gamma_{m-1}^{1/2}(n) \quad (4)$$

Hence, using Eqs. (1) and (3):

$$\gamma_m^{1/2}(n) = c_{b, m-1}(n) \gamma_{m-1}^{1/2}(n) \quad (5)$$

and using Eqs. (2) and (4):

$$\gamma_m^{1/2}(n) = c_{f, m-1}(n) \gamma_{m-1}^{1/2}(n) \quad (6)$$

Multiplying Eqs. (1) and (2), we get

$$\gamma_m(n) = c_{b, m-1}(n) c_{f, m-1}(n) \gamma_{m-1}(n) \quad (7)$$

which is the desired result.

12.15 The matrix product  $\Phi_{m+1}(n) L_m^H(n)$  equals

$$\Phi_{m+1}(n) \mathbf{L}_m^H(n) = \Phi_{m+1}(n) \begin{bmatrix} 1 & c_{1,1}(n) & \dots & c_{m-1,m-1}(n) & c_{m,m}(n) \\ 0 & 1 & \dots & c_{m-1,m-2}(n) & c_{m,m-1}(n) \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \dots & 1 & c_{m,1}(n) \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix} \quad (1)$$

(1) From the augmented system of normal equations, we have

$$\Phi_{m+1}(n) \begin{bmatrix} c_{m,m}(n) \\ c_{m,m-1}(n) \\ \vdots \\ c_{m,1}(n) \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ B_m(n) \end{bmatrix}$$

(2) From the solution to Problem 12.9 we note that

$$\Phi_{m+1}(n) = \begin{bmatrix} \Phi_m(n) & \phi_2(n) \\ \phi_2^H(n) & U_2(n) \end{bmatrix}$$

Hence

$$\begin{aligned} \Phi_{m+1}(n) \begin{bmatrix} c_{m-1,m-1}(n) \\ c_{m-1,m-2}(n) \\ \vdots \\ 1 \\ 0 \end{bmatrix} &= \begin{bmatrix} \Phi_m(n) \begin{bmatrix} c_{m-1,m-1}(n) \\ c_{m-1,m-2}(n) \\ \vdots \\ 1 \end{bmatrix} \\ \phi_2^H(n) \mathbf{c}_{m-1}(n) \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ 0 \\ \vdots \\ B_{m-1}(n) \\ \phi_2^H(n) \mathbf{c}_{m-1}(n) \end{bmatrix} \end{aligned}$$

(3) Thus far we have dealt with the last two columns of the matrix on the right-hand side of Eq. (1). Similarly, we may go onto show that

$$\Phi_{m+1}(n) \begin{bmatrix} c_{1,1}(n) \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ B_1(n) \\ \vdots \\ \times \\ \times \end{bmatrix}$$

where the crosses refer to some nonzero elements. Finally, we arrive at

$$\Phi_{m+1}(n) = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} B_0(n) \\ \times \\ \vdots \\ \times \\ \times \end{bmatrix}$$

where again the crosses refer to some other nonzero elements.

Putting all of these pieces together, we conclude that

$$\Phi_{m+1}(n) \mathbf{L}_m^H(n) = \begin{bmatrix} B_0(n) & 0 & \dots & 0 & 0 \\ \times & B_1(n) & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ \times & \times & \dots & B_{m-1}(n) & 0 \\ \times & \times & \dots & \times & B_m(n) \end{bmatrix}$$

This shows that  $\Phi_{m+1}(n) \mathbf{L}_m^H(n)$  is a lower triangular matrix. Since  $\mathbf{L}_m(n)$  is itself a lower triangular matrix, all the elements of  $\mathbf{L}_m(n)$  above the main diagonal are zero.

Moreover, since  $\Phi_{m+1}(n)$  is Hermitian symmetric,  $\mathbf{L}_m(n) \Phi_{m+1}(n) \mathbf{L}_m^H(n)$  is likewise Hermitian symmetric. Accordingly, all the elements of this product below the main diagonal are also zero. Finally, since all the diagonal elements of  $\mathbf{L}_m(n)$  equal unity, we

conclude that  $\mathbf{L}_m(n) \Phi_{m+1}(n) \mathbf{L}_m^H(n)$  is a diagonal matrix, as shown by

$$\mathbf{L}_m(n) \Phi_{m+1}(n) \mathbf{L}_m^H(n) = \text{diag}[B_0(n), B_1(n), \dots, B_{m-1}(n), B_m(n)]$$

12.16 The joint probability density function of the time series  $u(n), u(n-1), \dots, u(n-M)$  equals

$$f_{\mathbf{U}}(\mathbf{u}) = \frac{1}{[2\pi \det(\mathbf{R})]^{1/2}} \exp -\frac{1}{2} \mathbf{u}^H(n) \mathbf{R}^{-1} \mathbf{u}(n)$$

where  $\mathbf{u}^T(n) = [u(n), u(n-1), \dots, u(n-M)]$  and  $\mathbf{R}$  is the  $(M+1)$ -by- $(M+1)$  ensemble-averaged correlation matrix of the input vector  $\mathbf{u}(n)$ . The log-likelihood function equals

$$\begin{aligned} L &= \ln f_{\mathbf{U}}(\mathbf{u}) \\ &= -\frac{1}{2} \ln[2\pi \det(\mathbf{R})] - \frac{1}{2} \mathbf{u}^H(n) \mathbf{R}^{-1} \mathbf{u}(n) \end{aligned} \quad (1)$$

For  $n \geq M+1$ , we may approximate the correlation matrix  $\mathbf{R}$  in terms of the deterministic correlation matrix  $\Phi(n)$  as

$$\mathbf{R} \approx \frac{1}{n} \Phi(n), \quad n \geq M+1 \quad (2)$$

Hence, we may rewrite Eq. (1) as

$$L \approx -\frac{1}{2} \ln[2\pi \det(\mathbf{R})] - \frac{1}{2n} \mathbf{u}^H(n) \Phi^{-1}(n) \mathbf{u}(n) \quad (3)$$

The second term on the right side of Eq. (3) equals

$$\frac{1}{2n} \mathbf{u}^H(n) \Phi^{-1}(n) \mathbf{u}(n) = \frac{1}{2n} [1 - \gamma(n)]$$

where  $\gamma(n)$  is the conversion factor.

12.17 The a posteriori estimation  $e_m(n)$  is order-updated by using the recursion

$$e_m(n) = e_{m-1}(n) - \kappa_m^*(n) b_m(n), \quad m = 1, 2, \dots, M \quad (1)$$

By definition, we have

$$e_m(n) = d(n) - \hat{d}(n|U_{n-m}) \quad (2)$$

where  $d(n)$  is the desired response and  $\hat{d}(n|U_{n-m})$  is the least-squares estimate of  $d(n)$  given the input samples  $u(n), \dots, u(n-m+1), u(n-m)$  that span the space  $U_{n-m}$ . Similarly,

$$e_{m-1}(n) = d(n) - \hat{d}(n|U_{n-m+1}) \quad (3)$$

where  $\hat{d}(n|U_{n-m+1})$  is the least-squares estimate of  $d(n)$  given the input samples  $u(n), \dots, u(n-m+1)$ . Substituting Eqs. (1) and (2) into (3), we get

$$\hat{d}(n|U_{n-m}) = \hat{d}(n|U_{n-m+1}) + h_m^*(n)b_m(n) \quad (4)$$

Equation (4) shows that given the regression coefficient  $h_m$ , we only need  $b_m(n)$  as the new piece of information for updating the least-squares estimate of the desired response. Hence,  $b_m(n)$  may be viewed as a form of innovation, which is in perfect agreement with the discussion presented in Section 10.1.

12.18 From Eqs. (12.159) of the text:

$$\mathbf{D}_{m+1}(n) = \mathbf{L}_m(n)\Phi_{m+1}(n)\mathbf{L}_m^H(n)$$

Equivalently, we may write

$$\Phi_{m+1}^{-1}(n) = \mathbf{L}_m^H(n)\mathbf{D}_{m+1}^{-1}(n)\mathbf{L}_m(n)$$

Hence,

$$\begin{aligned} \mathbf{P}(n) &= \Phi_{m+1}^{-1}(n) \\ &= [\mathbf{D}^{-1/2}(n)\mathbf{L}(n)]^H [\mathbf{D}^{-1/2}(n)\mathbf{L}(n)] \end{aligned}$$

12.19 (a) From Eq. (12.152),

$$\hat{\mathbf{x}}(n+1|Y_n) = \lambda^{-1/2} \hat{\mathbf{x}}(n+1|Y_{n-1}) + \mathbf{g}(n)\alpha(n) \quad (1)$$

From Eqs. (12.97) and the solution to Problem 12.12:

$$\mathbf{g}(n) \leftrightarrow \lambda^{-1/2} F_{m-1}^{-1}(n) \epsilon_{f, m-1}(n) \quad (2)$$

$$\alpha(n) \leftrightarrow \lambda^{-n/2} \gamma_{m-1}^{1/2}(n-1) \beta_m^*(n) \quad (3)$$

(It is important to note that the relation corresponding to  $\alpha(n)$  must involve the error signal appropriate to the estimation problem of interest.)

Also from Table 12.4 (under backward prediction)

$$\hat{\mathbf{x}}(n+1|Y_{n-1}) \leftrightarrow (-\lambda^{-n/2} \kappa_{b,m}(n-1)) \quad (4)$$

Hence, substituting Eqs. (2), (3) and (4) into (1), and cancelling the common term  $\lambda^{-(1/2)/2}$ :

$$\kappa_{b,m}(n) = \kappa_{b,m}(n-1) - \left( \frac{\gamma_{m-1}^{1/2}(n-1) \varepsilon_{f,m-1}(n)}{F_{m-1}(n)} \right) \beta_m^*(n) \quad (5)$$

But, from Eq. (12.80):

$$\varepsilon_{f,m-1}(n-1) = \gamma_{m-1}^{1/2}(n-1) \eta_{m-1}(n) \quad (6)$$

Thus Eq. (5) simplifies to

$$\kappa_{b,m}(n) = \kappa_{b,m}(n-1) - \left( \frac{\gamma_{m-1}^{1/2}(n-1) \eta_{m-1}(n)}{F_{m-1}(n)} \right) \beta_m^*(n) \quad (7)$$

which is the desired result; see Eq. (12.154).

(b) From Table 12.4 (under joint-process estimation) and the solution to Problem 12.12:

$$\hat{\mathbf{x}}(n|Y_{n-1}) \leftrightarrow \lambda^{-n/2} h_{m-1}(n-1) \quad (10)$$

$$\mathbf{g}(n) \leftrightarrow \lambda^{-1/2} \mathbf{B}_{m-1}^{-1}(n) \varepsilon_{b,m-1}(n) \quad (11)$$

$$\alpha(n) \leftrightarrow \lambda^{-n/2} \gamma_{m-1}^{1/2}(n-1) \xi_m^*(n) \quad (12)$$

Hence, using Eqs. (10) through (12) in (1), we get



$$h_{m-1}(n) = h_{m-1}(n-1) + \left( \frac{\gamma_{m-1}^{1/2}(n-1)\varepsilon_{b,m-1}(n)}{B_{m-1}(n)} \right) \xi_m^*(n) \quad (13)$$

From Eq. (12.80):

$$\varepsilon_{b,m-1}(n) = \gamma_{m-1}(n-1)\beta_{m-1}(n-1)$$

Hence,

$$h_{m-1}(n) = h_{m-1}(n-1) + \left( \frac{\gamma_{m-1}(n)\beta_{m-1}(n)}{B_{m-1}(n)} \right) \xi_m^*(n) \quad (14)$$

which is exactly the same as Eq. (12.155).

12.20 For all  $m$ , when we put

$$\kappa_{b,m}(n) = \kappa_{f,m}^*(n) = \kappa_m^*(n)$$

$$\gamma_m = 1$$

in Table 12.7, we get

$$F_{m-1}(n) = \lambda F_{m-1}(n-1) + |f_{m-1}(n)|^2 \quad (1)$$

$$B_{m-1}(n-1) = \lambda B_{m-1}(n-2) + |b_{m-1}(n-1)|^2 \quad (2)$$

$$f_m(n) = f_{m-1}(n) + \kappa_m^*(n-1)b_{m-1}(n-1) \quad (3)$$

$$b_m(n) = b_{m-1}(n-1) + \kappa_m(n-1)f_{m-1}(n) \quad (4)$$

$$\kappa_m(n) = \kappa_m(n-1) - \frac{b_{m-1}(n-1)f_m^*(n)}{B_{m-1}(n-1)} \quad (5)$$

$$= \kappa_m(n-1) - \frac{f_{m-1}^*(n)b_m(n)}{F_{m-1}(n)} \quad (6)$$

Adding Eqs. (1) and (2), and setting

$$E_{m-1}(n) = F_{m-1}(n) + B_{m-1}(n-1)$$

we get

$$E_{m-1}(n) = \lambda E_{m-1}(n-1) + (|f_{m-1}(n)|^2 + |b_{m-1}(n-1)|^2) \quad (7)$$

Equation (7) is recognized as the update equation (12.9) for the GAL algorithm in the text.

The only outstanding issue is the recursive relation for the reflection coefficient. Adding Eqs. (5) and (6) and setting

$$\kappa_m(n) = \hat{\kappa}_m(n)$$

and dividing by two, we get

$$\hat{\kappa}_m(n) = \hat{\kappa}_m(n-1) - \frac{1}{2} \left[ \frac{b_{m-1}(n-1)f_m^*(n)}{B_{m-1}(n-1)} + \frac{f_{m-1}^*(n)b_m(n)}{F_{m-1}(n)} \right] \quad (8)$$

Now assuming that

$$B_{m-1}(n-1) = F_{m-1}(n) = \frac{1}{2}E_{m-1}(n) \quad (9)$$

then Eq. (8) takes the form

$$\hat{\kappa}_m(n) = \hat{\kappa}_m(n-1) - \frac{1}{E_{m-1}(n)} [b_{m-1}(n-1)f_m^*(n) + f_{m-1}^*(n)b_m(n)] \quad (10)$$

Equation (10) is recognized as the update equation (12.15) of the algorithm in the text.

Conclusions: The GAL algorithm is a special case of the recursive LSL algorithm, using a posteriori estimation errors under the following settings:

$$\kappa_{b,m}(n) = \hat{\kappa}_{f,m}(n)$$

$$\gamma_m(n) = 1$$

$$F_{m-1}(n) = B_{m-1}(n-1) = \frac{1}{2}E_{m-1}(n)$$

12.21 The forward reflection coefficient equals

$$\begin{aligned}
\kappa_{f,m}(n) &= \frac{\Delta_{m-1}(n)}{B_{m-1}(n-1)} \\
&= -\lambda \frac{\Delta_{m-1}(n-1)}{B_{m-1}(n-1)} - \frac{b_{m-1}(n-1)f_{m-1}^*(n)}{\gamma_{m-1}(n-1)B_{m-1}(n-1)} \\
&= -\lambda \frac{B_{m-1}(n-2)}{B_{m-1}(n-1)} \frac{\Delta_{m-1}(n-1)}{B_{m-1}(n-2)} - \frac{b_{m-1}(n-1)f_{m-1}^*(n)}{\gamma_{m-1}(n-1)B_{m-1}(n-1)} \xi_m^*(n)
\end{aligned}$$

where, in the first term, we have multiplied and divided by  $B_{m-1}(n-2)$ . But

$$\kappa_{f,m}(n-1) = -\frac{\Delta_{m-1}(n-1)}{B_{m-1}(n-2)}$$

Hence,

$$\kappa_{f,m}(n) = \frac{B_{m-1}(n-2)}{B_{m-1}(n-1)} \left[ \kappa_{f,m}(n-1) - \frac{b_{m-1}(n-1)f_{m-1}^*(n)}{\lambda B_{m-1}(n-2)\gamma_{m-1}(n-1)} \right] \quad (1)$$

We now use the relations (see part d of Problem 12.6)

$$\frac{\lambda B_{m-1}(n-2)}{B_{m-1}(n-1)} = \frac{\gamma_m(n-1)}{\gamma_{m-1}(n-1)}$$

Hence, we may rewrite Eq. (1) as

$$\kappa_{f,m}(n) = \frac{\gamma_m(n-1)}{\gamma_{m-1}(n-1)} \left[ \kappa_{f,m}(n-1) - \frac{b_{m-1}(n-1)f_{m-1}^*(n)}{\lambda B_{m-1}(n-2)\gamma_{m-1}(n-1)} \right]$$

Next, the backward reflection coefficient equals

$$\kappa_{b,m}(n) = -\frac{\Delta_{m-1}^*(n)}{F_{m-1}(n)}$$

$$\begin{aligned}
&= -\lambda \frac{\Delta_{m-1}^*(n-1)}{F_{m-1}(n)} - \frac{b_{m-1}^*(n-1)f_{m-1}(n)}{F_{m-1}(n)\gamma_{m-1}(n-1)} \\
&= \lambda \kappa_{b,m}(n-1) \frac{F_{m-1}(n-1)}{F_{m-1}(n)} - \frac{b_{m-1}^*(n-1)f_{m-1}(n)}{F_{m-1}(n)\gamma_{m-1}(n-1)} \\
&= \lambda \frac{F_{m-1}(n-1)}{F_{m-1}(n)} \left[ \kappa_{b,m}(n-1) - \frac{b_{m-1}^*(n-1)f_{m-1}(n)}{\lambda F_{m-1}(n-1)\gamma_{m-1}(n-1)} \right]
\end{aligned} \tag{2}$$

We now recognize that (see part c of Problem 12.6)

$$\lambda \frac{F_{m-1}(n-1)}{F_{m-1}(n)} = \frac{\gamma_m(n)}{\gamma_{m-1}(n-1)}$$

Hence, we may rewrite Eq. (2) as

$$\kappa_{b,m}(n) = \frac{\gamma_m(n)}{\gamma_{m-1}(n-1)} \left[ \kappa_{b,m}(n-1) - \frac{b_{m-1}^*(n-1)f_{m-1}(n)}{\lambda F_{m-1}(n-1)\gamma_{m-1}(n-1)} \right]$$

12.22 The forward a posteriori prediction error equals

$$\begin{aligned}
f_m(n) &= f_{m-1}(n) + \kappa_{f,m}^*(n)b_{m-1}(n-1) \\
&= f_{m-1}(n) - \frac{\Delta_{m-1}^*(n)}{B_{m-1}(n-1)}b_{m-1}(n-1)
\end{aligned}$$

Hence, the normalized value of  $f_m(n)$  equals

$$\begin{aligned}
\bar{f}_m(n) &= \frac{f_m(n)}{F_m^{1/2}(n)\gamma_m^{1/2}(n-1)} \\
&= \frac{f_{m-1}(n)}{F_m^{1/2}(n)\gamma_m^{1/2}(n-1)} - \frac{\Delta_{m-1}^*(n)}{B_{m-1}(n-1)F_m^{1/2}(n)\gamma_m^{1/2}(n-1)}b_{m-1}(n-1)
\end{aligned} \tag{1}$$

But

$$\begin{aligned}
F_m(n) &= F_{m-1}(n) \left[ 1 - \frac{|\Delta_{m-1}(n)|^2}{F_{m-1}(n)B_{m-1}(n-1)} \right] \\
&= F_{m-1}(n)[1 - \bar{\Delta}_{m-1}(n)]
\end{aligned} \tag{2}$$

where

$$\bar{\Delta}_{m-1}(n) = \frac{\Delta_{m-1}(n)}{F_{m-1}^{1/2}(n)B_{m-1}^{1/2}(n-1)}$$

Similarly,

$$\begin{aligned}
B_m(n) &= B_{m-1}(n-1) \left[ 1 - \frac{|\Delta_{m-1}(n)|^2}{B_{m-1}(n-1)F_{m-1}(n)} \right] \\
&= B_{m-1}(n-1)[1 - |\bar{\Delta}_{m-1}(n)|^2]
\end{aligned} \tag{3}$$

Also, we may write

$$\begin{aligned}
\gamma_m(n-1) &= \gamma_{m-1}(n-1) \left[ 1 - \frac{|b_{m-1}(n-1)|^2}{\gamma_{m-1}(n-1)B_{m-1}(n-1)} \right] \\
&= \gamma_{m-1}(n-1)[1 - |\bar{b}_{m-1}(n-1)|^2]
\end{aligned} \tag{4}$$

where

$$\bar{b}_{m-1}(n-1) = \frac{b_{m-1}(n-1)}{\gamma_{m-1}^{1/2}(n-1)B_{m-1}^{1/2}(n-1)} \tag{5}$$

Hence, we may use Eqs. (2) and (4) to express the first term on the right side of Eq. (1) as

$$\frac{f_{m-1}(n)}{F_m^{1/2}(n)\gamma_m^{1/2}(n-1)} = \frac{f_{m-1}(n)/[F_{m-1}^{1/2}(n)\gamma_{m-1}^{1/2}(n-1)]}{[1 - |\bar{\Delta}_{m-1}(n)|^2]^{1/2} [1 - |\bar{b}_{m-1}(n-1)|^2]^{1/2}}$$

$$= \frac{\bar{f}_{m-1}(n)}{[1 - |\bar{\Delta}_{m-1}(n)|^2]^{1/2} [1 - |\bar{b}_{m-1}(n-1)|^2]^{1/2}} \quad (6)$$

Next, we use Eqs. (2) to (4) to express the second term on the right side of Eq. (1) as

$$\begin{aligned} \frac{\Delta_{m-1}^*(n)b_{m-1}(n-1)}{B_{m-1}(n-1)F_m^{1/2}(n)\gamma_m^{1/2}(n-1)} &= \frac{\Delta_{m-1}^*(n)b_{m-1}(n-1)}{B_{m-1}(n-1)F_m^{1/2}(n)\gamma_m^{1/2}(n-1)[1 - |\bar{\Delta}_{m-1}(n)|^2]^{1/2}[1 - |\bar{b}_{m-1}(n-1)|^2]^{1/2}} \\ &= \frac{\bar{\Delta}_{m-1}^*(n)\bar{b}_{m-1}(n-1)}{[1 - |\bar{\Delta}_{m-1}(n)|^2]^{1/2}[1 - |\bar{b}_{m-1}(n-1)|^2]^{1/2}} \end{aligned} \quad (7)$$

Substituting Eqs. (6) and (7) into (1), we may thus write

$$\bar{f}_m(n) = \frac{\bar{f}_{m-1}(n) - \bar{\Delta}_{m-1}^*(n)\bar{b}_{m-1}(n-1)}{[1 - |\bar{\Delta}_{m-1}(n)|^2]^{1/2} [1 - |\bar{b}_{m-1}(n-1)|^2]^{1/2}} \quad (8)$$

Next, the backward a posteriori prediction error equals

$$b_m(b) = b_{m-1}(n-1) - \frac{\Delta_{m-1}(n)}{F_{m-1}(n)}f_{m-1}(n)$$

Hence, the normalized value of  $b_m(n)$  equals

$$\bar{b}_m(b) = \frac{b_{m-1}(n-1)}{B_m^{1/2}(n)\gamma_m^{1/2}(n)} - \frac{\Delta_{m-1}(n)f_{m-1}(n)}{F_{m-1}(n)B_m^{1/2}(n)\gamma_m^{1/2}(n)} \quad (9)$$

The conversion factor may be updated by using the recursion:

$$\begin{aligned} \gamma_m(n) &= \gamma_{m-1}(n-1) - \frac{|f_{m-1}(n)|^2}{F_{m-1}(n)} \\ &= \gamma_{m-1}(n-1) \left[ 1 - \frac{|f_{m-1}(n)|^2}{F_{m-1}(n)\gamma_{m-1}(n-1)} \right] \end{aligned}$$

$$= \gamma_{m-1}(n-1)[1 - |\bar{f}_{m-1}(n)|^2] \quad (10)$$

where

$$\bar{f}_{m-1}(n) = \frac{f_{m-1}(n)}{F_{m-1}^{1/2}(n)\gamma_{m-1}^{1/2}(n-1)}$$

Using Eqs. (3) and (10), we may express the first term on the right side of (Eq. 9) as

$$\begin{aligned} \frac{b_{m-1}(n-1)}{B_m^{1/2}(n)\gamma_m^{1/2}(n)} &= \frac{b_{m-1}(n-1)/[B_{m-1}^{1/2}(n-1)\gamma_{m-1}^{1/2}(n-1)]}{[1 - |\bar{\Delta}_{m-1}(n)|^2]^{1/2} [1 - |\bar{f}_{m-1}(n)|^2]^{1/2}} \\ &= \frac{\bar{b}_{m-1}(n-1)}{[1 - |\bar{\Delta}_{m-1}(n)|^2]^{1/2} [1 - |\bar{f}_{m-1}(n)|^2]^{1/2}} \end{aligned} \quad (11)$$

Using Eqs. (3) and (9), we may also rewrite the second term on the right side of Eq. (9) as

$$\begin{aligned} \frac{\Delta_{m-1}(n)f_{m-1}(n)}{F_{m-1}(n)B_m^{1/2}(n)\gamma_m^{1/2}(n)} &= \frac{\Delta_{m-1}(n)f_{m-1}(n)}{F_{m-1}(n)B_{m-1}^{1/2}(n-1)\gamma_{m-1}^{1/2}(n-1)[1 - |\bar{\Delta}_{m-1}(n)|^2]^{1/2} [1 - |\bar{f}_{m-1}(n)|^2]^{1/2}} \\ &= \frac{\bar{\Delta}_{m-1}(n)\bar{f}_{m-1}(n)}{[1 - |\bar{\Delta}_{m-1}(n)|^2]^{1/2} [1 - |\bar{f}_{m-1}(n)|^2]^{1/2}} \end{aligned} \quad (12)$$

Hence, the use of Eqs. (11) and (12) in (9) yields

$$\bar{b}_m(n) = \frac{\bar{b}_{m-1}(n-1) - \bar{\Delta}_{m-1}(n)\bar{f}_{m-1}(n)}{[1 - |\bar{\Delta}_{m-1}(n)|^2]^{1/2} [1 - |\bar{f}_{m-1}(n)|^2]^{1/2}} \quad (13)$$

Finally, we note that

$$\Delta_{m-1}(n) = \lambda\Delta_{m-1}(n-1) + \frac{b_{m-1}(n-1)f_{m-1}^*(n)}{\gamma_{m-1}(n-1)}$$

The normalized value of  $\Delta_{m-1}(n)$  equals

$$\begin{aligned}\bar{\Delta}_{m-1}(n) &= \frac{\Delta_{m-1}(n)}{F_{m-1}^{1/2}(n)B_{m-1}^{1/2}(n-1)} \\ &= \lambda \frac{\Delta_{m-1}(n)}{F_{m-1}^{1/2}(n)B_{m-1}^{1/2}(n-1)} + \frac{b_{m-1}(n-1)f_{m-1}^*(n)}{F_{m-1}^{1/2}(n)B_{m-1}^{1/2}(n-1)\gamma_{m-1}(n-1)}\end{aligned}\quad (14)$$

Using Eqs. (2) and (3), we may rewrite the first term on the right side of Eq. (14) as

$$\lambda \frac{\Delta_{m-1}(n-1)}{F_{m-1}^{1/2}(n)B_{m-1}^{1/2}(n-1)} = \frac{\Delta_{m-1}(n-1)}{F_{m-1}^{1/2}(n-1)B_{m-1}^{1/2}(n-2)} \frac{\gamma F_{m-1}^{1/2}(n-1)B_{m-1}^{1/2}(n-2)}{F_{m-1}^{1/2}(n)B_{m-1}^{1/2}(n-1)} \quad (15)$$

We next use the relations:

$$\frac{\lambda F_{m-1}(n-1)}{F_{m-1}(n)} = \frac{\gamma_m(n)}{\gamma_{m-1}(n-1)}$$

$$\frac{\lambda B_{m-1}(n-2)}{B_{m-1}(n-1)} = \frac{\gamma_m(n-1)}{\gamma_{m-1}(n-1)}$$

$$\bar{\Delta}_{m-1}(n-1) = \frac{\Delta_{m-1}(n-1)}{F_{m-1}^{1/2}(n-1)B_{m-1}^{1/2}(n-2)}$$

Hence, we may rewrite Eq. (15) as

$$\lambda \frac{\Delta_{m-1}(n-1)}{F_{m-1}^{1/2}(n)B_{m-1}^{1/2}(n-1)} = \bar{\Delta}_{m-1}(n-1) \frac{\gamma_m^{1/2}(n)\gamma_m^{1/2}(n-1)}{\gamma_{m-1}(n-1)} \quad (16)$$

Next, multiplying Eq. (4) by (10), we have

$$\frac{\gamma_m(n)\gamma_m(n-1)}{\gamma_{m-1}^2(n-1)} = [1 - |\bar{f}_{m-1}(n)|^2][1 - |\bar{b}_{m-1}(n-1)|^2]$$

Accordingly, we may rewrite (16) as



$$\lambda \frac{\Delta_{m-1}(n-1)}{F_{m-1}^{1/2}(n)B_{m-1}^{1/2}(n-1)} = \bar{\Delta}_{m-1}(n-1)[1 - |\bar{f}_{m-1}(n)|^2]^{1/2} [1 - |\bar{b}_{m-1}(n-1)|^2]^{1/2} \quad (17)$$

Next, the second term on the right side of Eq. (14) equals

$$\frac{b_{m-1}(n-1)f_{m-1}^*(n)}{F_{m-1}^{1/2}(n)B_{m-1}^{1/2}(n-1)\gamma_{m-1}(n-1)} = \bar{b}_{m-1}(n-1)\bar{f}_{m-1}^*(n) \quad (18)$$

Thus, substituting Eqs. (17) and (18) into (14), we get the desired time-update

$$\begin{aligned} \bar{\Delta}_{m-1}(n) = & \bar{\Delta}_{m-1}(n-1)[1 - |\bar{f}_{m-1}(n)|^2]^{1/2} [1 - |\bar{b}_{m-1}(n-1)|^2]^{1/2} \\ & + \bar{b}_{m-1}(n-1)\bar{f}_{m-1}^*(n) \end{aligned} \quad (19)$$

Equations (19), (8) and (13), in that order, constitute the normalized LSL algorithm.

## CHAPTER 13

13.1 The analog (infinite-precision) form of the LMS algorithm is described by

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu \mathbf{u}(n)[d(n) - \mathbf{w}^T(n)\mathbf{u}(n)], \quad n = 0, 1, 2, \dots$$

where  $\hat{\mathbf{w}}(n)$  is the tap-weight vector estimate at time  $n$ ,  $\mathbf{u}(n)$  is the tap-input vector,  $d(n)$  is the desired response, and  $\mu$  is the step-size parameter. The digital (finite-precision) counterpart of this update may be expressed as

$$\hat{\mathbf{w}}_q(n+1) = \hat{\mathbf{w}}_q(n) + Q[\mu \mathbf{u}_q(n)e_q(n)]$$

where

$$e(n) = d(n) - \mathbf{w}^T(n)\mathbf{u}(n)$$

and the use of subscripts  $q$  signifies the use of finite-precision arithmetic. Let

$$Q[\mu \mathbf{u}_q(n)e_q(n)] = \mu \mathbf{u}_q(n)e_q(n) + \mathbf{v}(n)$$

where the quantizing noise vector  $\mathbf{v}(n)$  is determined by the manner in which the term  $\mathbf{u}_q(n)e_q(n)$  is computed. Hence,

$$\hat{\mathbf{w}}_q(n+1) = \hat{\mathbf{w}}_q(n) + \mu \mathbf{u}_q(n)e_q(n) + \mathbf{v}(n) \quad (1)$$

The quantized value of the estimation error  $e(n)$  may be expressed as

$$e_q(n) = e(n) - \Delta \mathbf{u}^T(n)\mathbf{w}(n) - \mathbf{u}^T(n)\Delta \mathbf{w}(n) + \zeta(n)$$

where  $\zeta(n)$  denotes residual error. The quantized value of  $\mathbf{u}(n)$  equals

$$\mathbf{u}_q(n) = \mathbf{u}(n) + \Delta \mathbf{u}(n)$$

Hence, we may express  $\mathbf{u}_q(n)e_q(n)$ , ignoring second-order effects, as follows

$$\begin{aligned} \mathbf{u}_q(n)e_q(n) &= \mathbf{u}(n)e(n) + \Delta \mathbf{u}(n)e(n) - \mathbf{u}(n)\Delta \mathbf{u}^T(n)\hat{\mathbf{w}}(n) \\ &\quad - \mathbf{u}(n)\mathbf{u}^T(n)\Delta \hat{\mathbf{w}}(n) + \mathbf{u}(n)\zeta(n) \end{aligned}$$

We may therefore rewrite Eq. (1) as

$$\begin{aligned}
\hat{\mathbf{w}}_q(n+1) &= \hat{\mathbf{w}}_q(n) + \mu \mathbf{u}_q(n) e_q(n) + \mu \Delta \mathbf{u}(n) e(n) \\
&\quad - \mu \mathbf{u}(n) \Delta \mathbf{u}^T(n) \hat{\mathbf{w}}(n) - \mu \mathbf{u}(n) \mathbf{u}^T(n) \Delta \hat{\mathbf{w}}(n) + \mu \mathbf{u}(n) \zeta(n) \\
&\quad + \mathbf{v}(n)
\end{aligned} \tag{2}$$

But

$$\hat{\mathbf{w}}_q(n+1) = \hat{\mathbf{w}}(n+1) + \Delta \hat{\mathbf{w}}(n+1)$$

$$\hat{\mathbf{w}}_q(n) = \hat{\mathbf{w}}(n) + \Delta \hat{\mathbf{w}}(n)$$

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \mu \mathbf{u}(n) e(n)$$

Hence, from Eq. (2) we deduce that

$$\Delta \mathbf{w}(n+1) = \mathbf{F}(n) \Delta \mathbf{w}(n) + \mathbf{t}(n)$$

where

$$\mathbf{F}(n) = \mathbf{I} - \mu \mathbf{u}(n) \mathbf{u}^T(n)$$

$$\mathbf{t}(n) = \mu \Delta \mathbf{u}(n) e(n) - \mu \mathbf{u}(n) \hat{\mathbf{w}}^T(n) \Delta \mathbf{u}(n) + \mu \mathbf{u}(n) \zeta(n) + \mathbf{v}(n)$$

Note that

$$\Delta \mathbf{u}^T(n) \hat{\mathbf{w}}(n) = \hat{\mathbf{w}}^T(n) \Delta \mathbf{u}(n)$$

13.2 Building on the solution to Problem 13.1, assume that  $\Delta \mathbf{w}(n)$  is stationary. Then the expectation of  $\Delta \mathbf{w}(n)$  is zero because the expectation of  $\mathbf{t}(n)$  is zero.

13.3 We note that

$$y_I(n) = \sum_i w_i u(n-i)$$

$$y_{II}(n) = \sum_i w_{iq} u(n-i)$$

Hence,

$$\epsilon(n) = y_I(n) - y_{II}(n) = \sum_i (w_i - w_{iq})u(n-i)$$

The mean-square value of  $\epsilon(n)$  is

$$\begin{aligned} E[\epsilon^2(n)] &= E\left[\sum_i \sum_j (w_i - w_{iq})(w_j - w_{jq})u(n-i)u(n-j)\right] \\ &= \sum_i \sum_j (w_i - w_{iq})(w_j - w_{jq})E[u(n-i)u(n-j)] \end{aligned} \quad (1)$$

Assuming that

$$E[u(n-i)u(n-j)] = \begin{cases} A_{\text{rms}}^2, & j = i \\ 0, & j \neq i \end{cases}$$

we may simplify Eq. (1) as

$$E[\epsilon^2(n)] = A_{\text{rms}}^2 \sum_i (w_i - w_{iq})^2$$

13.4 (a) The digital residual error is

$$e_D(n) = \frac{\text{LSB}}{\mu A_{\text{rms}}}$$

With 12-bit quantization, the least significant bit is

$$\text{LSB} = 2^{-12} \approx 0.25 \times 10^{-3}$$

We are given

$$\mu = 0.07$$

$$A_{\text{rms}} = 1$$

Hence, the digital residual error is

$$e_D(n) = \frac{2^{-12}}{0.07 \times 1} \approx \frac{0.25 \times 10^{-3}}{0.07} = 0.35 \times 10^{-2}$$

(b) The rms quantization error is

$$(QE)_{\text{rms}} = \sqrt{E[\varepsilon^2(n)]} = A_{\text{rms}} \left( \sum_{i=0}^{M-1} (w_i - w_{iq})^2 \right)^{1/2} \leq (A_{\text{rms}} M^{1/2}) (\text{LSB})$$

For the problem at hand we have

$$(QE)_{\text{rms}} \approx \sqrt{17} \times 0.25 \times 10^{-3} \approx 10^{-3}$$

We thus see that  $(QE)_{\text{rms}}$  is about 3.5 times worse than the digital residual error.

13.5 We start with

$$\begin{aligned} \frac{1}{\kappa_q(n)} &= \frac{1}{\lambda + \pi_q(n)\mathbf{u}(n)} \\ &\approx \frac{1}{\lambda + \pi(n)\mathbf{u}(n)} \left( 1 - \frac{\eta_{\pi}(n)\mathbf{u}(n)}{\lambda + \pi(n)\mathbf{u}(n)} \right) \end{aligned} \quad (1)$$

where it is noted that

$$\pi_q(n) = \pi(n) + \eta_{\pi}(n)$$

Next, we note that

$$\mathbf{k}_q(n) = \frac{\mathbf{P}_q(n-1)\mathbf{u}(n)}{\kappa_q(n)} \quad (2)$$

Let

$$\mathbf{P}_q(n-1) = \mathbf{P}(n-1) + \eta_P(n-1) \quad (3)$$

Therefore, using Eqs. (1) and (3) in (2):

$$\eta_k(n) = \mathbf{k}_q(n) - \mathbf{k}(n)$$

$$\begin{aligned}
&= [(\mathbf{P}(n-1) + \eta_{P(n-1)})\mathbf{u}(n)] \left[ \frac{1}{\lambda + \pi(n)\mathbf{u}(n)} \left( 1 - \frac{\eta_{\pi(n)}\mathbf{u}(n)}{\lambda + \pi(n)\mathbf{u}(n)} \right) \right] - \mathbf{k}(n) \\
&\approx \frac{\mathbf{P}(n-1)\mathbf{u}(n)}{\lambda + \pi(n)\mathbf{u}(n)} + \frac{\eta_{P(n-1)}\mathbf{u}(n)}{\lambda + \pi(n)\mathbf{u}(n)} - \frac{\mathbf{P}(n-1)\mathbf{u}(n)\eta_{\pi(n)}\mathbf{u}(n)}{(\lambda + \pi(n)\mathbf{u}(n))^2} - \mathbf{k}(n) \\
&= \frac{\eta_{P(n-1)}\mathbf{u}(n)}{\lambda + \pi(n)\mathbf{u}(n)} - \frac{\mathbf{P}(n-1)\mathbf{u}(n)\eta_{\pi(n)}\mathbf{u}(n)}{(\lambda + \pi(n)\mathbf{u}(n))^2}
\end{aligned}$$

We next calculate (using the relation  $\eta_{\pi(n)} = \mathbf{u}^H(n)\eta_{P(n-1)}$ )

$$\begin{aligned}
\eta_{P'(n)} &= \eta_k(n)\pi(n) + \mathbf{k}(n)\eta_{\pi(n)} \\
&\approx \frac{\eta_{P(n-1)}\mathbf{u}(n)\pi(n)}{\lambda + \pi(n)\mathbf{u}(n)} - \frac{\mathbf{P}(n-1)\mathbf{u}(n)\eta_{\pi(n)}\mathbf{u}(n)\pi(n)}{(\lambda + \pi(n)\mathbf{u}(n))^2} + \frac{\mathbf{P}(n-1)\mathbf{u}(n)\eta_{\pi(n)}}{\lambda + \pi(n)\mathbf{u}(n)} \\
&= \frac{\eta_{P(n-1)}\mathbf{u}(n)\pi(n)}{\lambda + \pi(n)\mathbf{u}(n)} - \frac{\mathbf{P}(n-1)\mathbf{u}(n)\mathbf{u}^H(n)\eta_{P(n-1)}\mathbf{u}(n)\pi(n)}{(\lambda + \pi(n)\mathbf{u}(n))^2} \\
&\quad + \frac{\mathbf{P}(n-1)\mathbf{u}(n)\mathbf{u}^H(n)\eta_{P(n-1)}}{\lambda + \pi(n)\mathbf{u}(n)}
\end{aligned}$$

Finally, we calculate

$$\begin{aligned}
\eta_P(n) &= \frac{1}{\lambda}(\eta_{P(n-1)} - \eta_{P'(n-1)}) \\
&= \frac{1}{\lambda} \left( \eta_{P(n-1)} - \frac{\eta_{P(n-1)}\mathbf{u}(n)\pi(n)}{\lambda + \pi(n)\mathbf{u}(n)} + \frac{\mathbf{P}(n-1)\mathbf{u}(n)\mathbf{u}^H(n)\eta_{P(n-1)}\mathbf{u}(n)\pi(n)}{(\lambda + \pi(n)\mathbf{u}(n))^2} \right. \\
&\quad \left. - \frac{\mathbf{P}(n-1)\mathbf{u}(n)\mathbf{u}^H(n)\eta_{P(n-1)}}{\lambda + \pi(n)\mathbf{u}(n)} \right)
\end{aligned}$$

We now note that

$$\frac{\mathbf{P}(n-1)\mathbf{u}(n)}{\lambda + \pi(n)\mathbf{u}(n)} = \mathbf{k}(n)$$

$$\pi(n) = \mathbf{u}^H(n)\mathbf{P}(n-1)$$

Hence, assuming that  $\mathbf{P}(n-1)$  is Hermitian:

$$\begin{aligned}\eta_P(n) &= \frac{1}{\lambda} \left( \eta_P(n-1) - \eta_P(n-1)\mathbf{u}(n)\mathbf{k}^H(n) + \mathbf{k}(n)\mathbf{u}^H(n)\eta_P(n-1)\mathbf{u}(n)\mathbf{k}^H(n) \right. \\ &\quad \left. - \mathbf{k}(n)\mathbf{u}^H(n)\eta_P(n-1) \right) \\ &= \frac{1}{\lambda} (\mathbf{I} - \mathbf{k}(n)\mathbf{u}^H(n))\eta_{cP}(n) - \frac{1}{\lambda} (\mathbf{I} - \mathbf{k}(n)\mathbf{u}^H(n))\eta_P(n-1)\mathbf{u}(n)\mathbf{k}^H(n) \\ &= \frac{1}{\lambda} (\mathbf{I} - \mathbf{k}(n)\mathbf{u}^H(n))\eta_P(n-1)(\mathbf{I} - \mathbf{k}(n)\mathbf{u}^H(n))^H\end{aligned}$$

From this result we readily see that  $\eta_P^T(n) = \eta_P(n)$  which demonstrates the symmetry-preserving-property of the RLS algorithm summarized in Table 13.1.

13.6 The condition for persistent excitation (assuming real data) may be expressed as

$$a\mathbf{I} \leq \sum_{i=n_0}^n \lambda^{n-i} \mathbf{u}(i)\mathbf{u}^T(i) \leq b\mathbf{I}$$

where  $a$  and  $b$  are both positive numbers. Premultiplying by  $\mathbf{z}^T$  and postmultiplying by  $\mathbf{z}$ :

$$a\mathbf{z}^T\mathbf{z} \leq \sum_{i=n_0}^n \lambda^{n-i} \mathbf{z}^T\mathbf{u}(i)\mathbf{u}^T(i)\mathbf{z} \leq b\mathbf{z}^T\mathbf{z}$$

We now recognize that

$$\mathbf{z}^T\mathbf{u}(i) = \mathbf{u}^T(i)\mathbf{z}$$

Hence,

$$a\mathbf{z}^T\mathbf{z} \leq \sum_{i=n_0}^n \lambda^{n-i} |\mathbf{z}^T \mathbf{u}(i)|^2 \leq b\mathbf{z}^T\mathbf{z}$$

For a nonzero vector  $\mathbf{z}$ , this condition requires that we have

$$\mathbf{z}^T \mathbf{u}(i) > \alpha \quad \text{for } n_0 \leq i \leq n$$

This is another way of defining the condition for persistent excitation.

13.7 We start with the matrix relation:

$$\begin{bmatrix} \lambda^{1/2} \Phi^{1/2}(n-1) & \mathbf{u}(n) \\ \lambda^{1/2} \mathbf{p}^H(n-1) & d(n) \\ \mathbf{0}^T & 1 \\ \lambda^{1/2} \Phi^{H/2}(n-1) & \mathbf{0} \end{bmatrix} \Theta(n) = \begin{bmatrix} \Phi^{1/2}(n) & \mathbf{0} \\ \mathbf{p}^H(n) & \xi(n) \gamma^{1/2}(n) \\ \mathbf{u}^H(n) \Phi^{-H/2}(n) & \gamma^{1/2}(n) \\ \Phi^{-H/2}(n) & -\mathbf{k}(n) \gamma^{-1/2}(n) \end{bmatrix} \quad (1)$$

where  $\Theta(n)$  is unitary rotation.

Let

$$\mathbf{X}(n) = \Phi^{-H/2}(n) + \eta_x(n) \quad (2)$$

where  $\eta_x(n)$  represents the effect of round-off error.

Assuming there are no additional local errors introduced at time  $n$ , the recursion pertaining to the bottom parts of the prearray and postarray of Eq. (1) takes on the following form:

$$\begin{bmatrix} \lambda^{-1/2} \mathbf{X}(n-1), & \mathbf{0} \end{bmatrix} \Theta(n) = \begin{bmatrix} \mathbf{X}(n), & -\mathbf{y}(n) \end{bmatrix} \quad (3)$$

The vector  $\mathbf{y}(n)$  is the quantized version of  $\mathbf{k}(n) \gamma^{1/2}(n)$ :

$$\mathbf{y}(n) = \mathbf{k}(n) \gamma^{-1/2}(n) + \eta_y(n) \quad (4)$$

where  $\mathbf{k}(n)$  is the gain vector,  $\gamma(n)$  is the conversion factor, and  $\eta_y(n)$  is the round-off error.



Substituting Eq. (2) and (4) into (3), we may express the quantized version of the last rows in Eq. (1) as follows:

$$\begin{aligned} & \begin{bmatrix} \lambda^{-1/2} \Phi^{-H/2}_{(n-1)} + \lambda^{-1/2} \eta_x(n-1), & \mathbf{0} \end{bmatrix} \Theta(n) \\ &= \begin{bmatrix} \Phi^{-H/2}_{(n)} + \eta_x(n), & -\mathbf{k}(n) \gamma^{-1/2}_{(n)} - \eta_y(n) \end{bmatrix} \end{aligned} \quad (5)$$

Under infinite-precision arithmetic, we have from the last rows of Eq. (1):

$$\begin{bmatrix} \lambda^{-1/2} \Phi^{-H/2}_{(n-1)} & \mathbf{0} \end{bmatrix} \Theta(n) = \begin{bmatrix} \Phi^{-H/2}_{(n)}, & -\mathbf{k}(n) \gamma^{-1/2}_{(n)} \end{bmatrix} \quad (6)$$

Hence, comparing Eqs. (5) and (6), we infer

$$\begin{bmatrix} \lambda^{-1/2} \eta_x(n-1) & \mathbf{0} \end{bmatrix} \Theta(n) = \begin{bmatrix} \eta_x(n) & -\eta_y(n) \end{bmatrix} \quad (7)$$

Equation (7) reveals that the error propagation due to  $\eta_x(n-1)$  is NOT necessarily stable, in that the local errors tend to grow unboundedly. The unlimited error growth is due to (1) the amplification produced by the factor  $\lambda^{-1/2}$  for  $\lambda < 1$ , and (2) the fact that the unitary rotation  $\Theta$  is independent of the error  $\eta_x$ . Consequently, as the recursion progresses the stored values of  $\Phi^{1/2}$  and  $\Phi^{-H/2}$  deviate more and more from each other's Hermitian transpose, thereby contradicting the very premise on which the extended QR-RLS algorithm is based.

13.8 We start with the relations (see Problem 12.5):

$$\Phi_{M+1}^{-1}(n) = \begin{bmatrix} \Phi_M^{-1}(n) & \mathbf{0}_M \\ \mathbf{0}_M^T & 0 \end{bmatrix} + \frac{1}{B_M(n)} \mathbf{c}_M(n) \mathbf{c}_M^H(n)$$

$$\mathbf{u}_{M+1}(n) = \begin{bmatrix} \mathbf{u}_M(n) \\ u(n-M) \end{bmatrix}$$

Hence,

$$\mathbf{k}_{M+1}(n) = \Phi_{M+1}^{-1}(n) \mathbf{u}_{M+1}(n)$$

$$= \begin{bmatrix} \mathbf{k}_M(n) \\ 0 \end{bmatrix} + \frac{b_M(n)}{B_M(n)} \mathbf{c}_M(n) \quad (1)$$

Let  $k_{M+1,M+1}(n)$  denote the last element of the gain vector  $\mathbf{k}_{M+1}(n)$ . Then, recognizing that the last element of  $\mathbf{c}_M(n)$  is unity by definition, we immediately deduce from Eq. (1) that

$$k_{M+1,M+1}(n) = \frac{b_M(n)}{B_M(n)}$$

Normalizing with respect to  $\gamma_{M+1}(n)$ , we write

$$\tilde{k}_{M+1,M+1}(n) = \frac{k_{M+1,M+1}(n)}{\gamma_{M+1}(n)} = \frac{b_M(n)}{\gamma_{M+1}(n)B_M(n)} \quad (2)$$

Next, we use the relation (see part (d) of the solution to Problem 12.6)

$$\gamma_{M+1}(n) = \lambda \frac{B_M(n-1)}{B_M(n)} \gamma_M(n) \quad (3)$$

But

$$B_M(n) = \lambda B_M(n-1) + b_M^*(n) \beta_M(n)$$

or

$$\frac{\lambda B_M(n-1)}{B_M(n)} = 1 - \frac{b_M^*(n) \beta_M(n)}{B_M(n)}$$

We may therefore rewrite Eq. (3) in the equivalent form:

$$\gamma_M(n) = \left( 1 - \frac{b_M^*(n) \beta_M(n)}{B_M(n)} \right)^{-1} \gamma_{M+1}(n)$$

The rescue variable is thus defined by

$$R = \frac{\gamma_{M+1}(n)}{\gamma_M(n)} = 1 - \frac{b_M^*(n) \beta_M(n)}{B_M(n)} \quad (4)$$

Ideally, we have  $0 < R \leq 1$ . Eliminating  $b_M(n)/B_M(n)$  between Eqs. (2) and (4), we may express the rescue variable in the equivalent form:

$$R = 1 - \gamma_{M+1}(n) \tilde{k}_{M+1,M+1}(n) \beta_M(n)$$

## CHAPTER 14

- 14.1 In an adaptive equalizer, the input signal equals the channel output and the desired response equals the channel input (i.e., transmitted signal). In a stationary environment, both of these signals are stationary with the result that the error-performance surface is fixed in all respects. On the other hand, in a nonstationary environment, the channel output (i.e., equalizer input) is nonstationary with the result that both the correlation matrix  $\mathbf{R}$  of the input vector and the cross-correlation vector  $\mathbf{p}$  between the input vector and desired response take on time-varying forms. Consequently, the error-performance surface is continually changing its shape and is also in a constant state of motion.
- 14.2 In adaptive prediction applied to a nonstationary process, both the input vector (defined by a set of past values of the process) and the desired response (defined by the present value of the process) are nonstationary. Accordingly, in such a case the error-performance surface behaves in a manner similar to that described for adaptive equalization in Problem 14.1. Specifically, the error-performance surface constantly changes its shape and constantly moves. In contrast, the error-performance surface for the adaptive prediction of a stationary process is completely fixed.
- 14.3 We have, by definition,

$$\epsilon_1(n) = \hat{\mathbf{w}}(n) - E[\hat{\mathbf{w}}(n)]$$

$$\epsilon_2(n) = E[\hat{\mathbf{w}}(n)] - \mathbf{w}_o$$

We may therefore expand

$$\begin{aligned} E[\epsilon_1^H(n) \epsilon_2(n)] &= E[(\hat{\mathbf{w}}(n) - E[\hat{\mathbf{w}}(n)])^H (E[\hat{\mathbf{w}}(n)] - \mathbf{w}_o)] \\ &= E[\hat{\mathbf{w}}^H(n) E[\hat{\mathbf{w}}(n)] - E[\hat{\mathbf{w}}^H(n)] E[\hat{\mathbf{w}}(n)] \\ &\quad - \hat{\mathbf{w}}^H(n) \mathbf{w}_o + E[\hat{\mathbf{w}}^H(n)] \mathbf{w}_o] \\ &= -E[\hat{\mathbf{w}}^H(n) \mathbf{w}_o] + E[\hat{\mathbf{w}}^H(n)] E[\mathbf{w}_o] \end{aligned}$$

Invoking the assumption that  $\mathbf{w}(n)$  and  $\mathbf{w}_o$  are statistically independent, we may go on to write

$$\begin{aligned} E[\epsilon_1^H(n) \epsilon_2(n)] &= (-E[\hat{\mathbf{w}}^H(n)] E[\mathbf{w}_o]) + E[\hat{\mathbf{w}}^H(n)] E[\mathbf{w}_o] \\ &= \mathbf{0} \end{aligned} \tag{1}$$

From this result we immediately deduce that we also have

$$E[\epsilon_2^H(n) \epsilon_1(n)] = \mathbf{0} \quad (2)$$

Finally, we note that

$$\begin{aligned} E[\|\epsilon(n)\|^2] &= E[\epsilon^H(n) \epsilon(n)] \\ &= E[(\epsilon_1(n) + \epsilon_2(n))^H (\epsilon_1(n) + \epsilon_2(n))] \\ &= E[\|\epsilon_1(n)\|^2] + E[\epsilon_1^H(n) \epsilon_2(n)] \\ &\quad + E[\epsilon_2^H(n) \epsilon_1(n)] + E[\|\epsilon_2(n)\|^2] \\ &= E[\|\epsilon_1(n)\|^2] + E[\|\epsilon_2(n)\|^2] \end{aligned}$$

where in the last line we have made use of Eqs. (1) and (2).

14.4 Invoking the low-pass filtering action of the LMS filter for small  $\mu$ , we note that  $\epsilon_1(n)$  and  $\epsilon_2(n)$  are both independent of the input vector  $\mathbf{u}(n)$ . We may therefore write:

$$\begin{aligned} 1. \quad E[\epsilon_1^H(n) \mathbf{u}(n) \mathbf{u}^H(n) \epsilon_1(n)] &= \text{tr} \left\{ E[\epsilon_1^H(n) \mathbf{u}(n) \mathbf{u}^H(n) \epsilon_1(n)] \right\} \\ &= E \left\{ \text{tr}[\epsilon_1^H(n) \mathbf{u}(n) \mathbf{u}^H(n) \epsilon_1(n)] \right\} \\ &= E \left\{ \text{tr}[\mathbf{u}(n) \mathbf{u}^H(n) \epsilon_1(n) \epsilon_1^H(n)] \right\} \\ &= \text{tr} \left\{ E[\mathbf{u}(n) \mathbf{u}^H(n) \epsilon_1(n) \epsilon_1^H(n)] \right\} \\ &= \text{tr} \left\{ E[\mathbf{u}(n) \mathbf{u}^H(n)] E[\epsilon_1(n) \epsilon_1^H(n)] \right\} \end{aligned}$$

$$= \text{tr}[\mathbf{R}\mathbf{K}_1(n)]$$

2. Similarly, we may show that

$$E[\in_2^H(n)\mathbf{u}(n)\mathbf{u}^H(n)\in_2^H(n)] = \text{tr}[\mathbf{R}\mathbf{K}_2(n)]$$

$$\begin{aligned} 3. \quad E[\in_1^H(n)\mathbf{u}(n)\mathbf{u}^H(n)\in_2^H(n)] &= \text{tr}\left\{E[\in_1^H(n)\mathbf{u}(n)\mathbf{u}^H(n)\in_2^H(n)]\right\} \\ &= \text{tr}\left\{E[\mathbf{u}(n)\mathbf{u}^H(n)]E[\in_2(n)\in_1^H(n)]\right\} \\ &= \text{tr}\left\{\mathbf{R}E[\in_2(n)\in_1^H(n)]\right\} \end{aligned} \quad (1)$$

Next we note that

$$E[\in_2^H(n)\in_1^H(n)] = \mathbf{0}$$

It follows therefore that

$$E[\in_1^H(n)\mathbf{u}(n)\mathbf{u}^H(n)\in_2^H(n)] = \mathbf{0}$$

Similarly, we may show that

$$E[\in_2^H(n)\mathbf{u}(n)\mathbf{u}^H(n)\in_1^H(n)] = \mathbf{0}$$

14.5 Evaluating the mean-square values of both sides of Eq. (14.27) yields

$$\begin{aligned} E[|v_k(n+1)|^2] &= (1 - \mu\lambda_k)^2 E[|v_k(n)|^2] + E[|\phi_k(n)|^2] \\ &= (1 - 2\mu\lambda_k + \mu^2\lambda_k^2)E[|v_k(n)|^2] + \sigma_{\phi_k}^2 \end{aligned} \quad (1)$$

For small  $\mu$ , we may ignore the term  $\mu^2 \lambda_k^2$  in comparison to the unity term, and so approximate Eq. (1) as

$$E[|v_k(n+1)|^2] \approx (1 - 2\mu\lambda_k)E[|v_k(n)|^2] + \sigma_{\phi_k}^2 \quad (2)$$

Under steady-state conditions,  $v_k(n+1) \rightarrow v_k(n)$  as  $n \rightarrow \infty$ , in which case Eq. (2) reduces further to

$$\begin{aligned} 2\mu\lambda_k E[|v_k(n)|^2] &\approx \sigma_{\phi_k}^2 \\ &\approx \mu^2 \sigma_v^2 \lambda_{u,k} + \lambda_{\omega,k} \end{aligned}$$

Solving for  $E[|v_k(n)|^2]$ , we get

$$E[|v_k(n)|^2] \approx \frac{\sigma_v^2}{2}\mu + \frac{\lambda_{\omega,k}}{2\lambda_{u,k}} \quad (3)$$

Hence, the mean-square deviation is (see Eq. (14.31))

$$\begin{aligned} D(n) &= E[\|\varepsilon_0(n)\|^2] \\ &= \sum_{k=1}^M E[|v_k(n)|^2] \\ &= \frac{M}{2}\sigma_v^2\mu + \frac{1}{2}\sum_{k=1}^M \frac{\lambda_{\omega,k}}{\lambda_{u,k}} \\ &= \frac{M}{2}\sigma_v^2\mu + \frac{1}{2}\text{tr}[\mathbf{R}_u^{-1}\mathbf{R}_\omega] \end{aligned} \quad (4)$$

where  $\mathbf{R}_u = E[\mathbf{u}(n)\mathbf{u}^H(n)]$ ,  $\mathbf{u}(n)$  = tap-input vector

$\mathbf{R}_\omega = E[\omega(n)\omega^H(n)]$ ,  $\omega(n)$  = process noise

14.6 The misadjustment of the LMS algorithm is given by (see Eqs. (5.91) and (14.36))

$$M \approx \frac{1}{2\sigma_v} \sum_{k=1}^M \lambda_{u,k} E[|v_k(n)|^2] \quad (1)$$

where  $M$  is the filter length. From the solution to Problem 14.5, we have

$$E[|v_k(n)|^2] \approx \frac{\mu}{2}\sigma_v^2 + \frac{1}{2\mu} \frac{\lambda_{\omega,k}}{\lambda_{u,k}} \quad (2)$$

Substituting Eq. (2) into (1), we get

$$\begin{aligned} M &\approx \frac{1}{2\sigma_v} \sum_{k=1}^M \lambda_{u,k} \left( \frac{\mu}{2}\sigma_v^2 + \frac{1}{2\mu} \frac{\lambda_{\omega,k}}{\lambda_{u,k}} \right) \\ &= \frac{\mu}{2} \sum_{k=1}^M \lambda_{u,k} + \frac{1}{2\mu\sigma_v^2} \sum_{k=1}^M \lambda_{\omega,k} \\ &= \frac{\mu}{2} + r[\mathbf{R}_u] + \frac{1}{2\mu\sigma_v^2} \text{tr}[\mathbf{R}_\omega] \end{aligned} \quad (3)$$

which is the desired result.

14.7 To simplify the presentation, we use the following notations in the solution to this problem:

$$\mathbf{R}_u = \mathbf{R} \quad \text{and} \quad \mathbf{R}_\omega = \mathbf{Q}$$

(a) The minimum misadjustment for the LMS algorithm is

$$M_{\min}^{\text{LMS}} = \frac{1}{\sigma_v} \sqrt{\text{tr}[\mathbf{R}] \text{tr}[\mathbf{Q}]} \quad (1)$$

The corresponding value for the RLS algorithm is

$$M_{\min}^{\text{RLS}} = \frac{1}{\sigma_v} \sqrt{M \text{tr}[\mathbf{RQ}]} \quad (2)$$



For  $\mathbf{Q} = c_1 \mathbf{R}$ , Eqs. (1) and (2) yield the ratio:

$$\frac{M_{\min}^{\text{LMS}}}{M_{\min}^{\text{RLS}}} = \frac{\text{tr}[\mathbf{R}]}{\sqrt{M \text{tr}[\mathbf{R}^2]}} \quad (3)$$

Now

$$\mathbf{R} = \sum_{i=1}^M \lambda_i \mathbf{q}_i \mathbf{q}_i^H$$

$$\mathbf{R}^2 = \sum_{i=1}^M \lambda_i^2 \mathbf{q}_i \mathbf{q}_i^H$$

We may therefore write

$$\text{tr}[\mathbf{R}] = \sum_{i=1}^M \lambda_i \quad (4)$$

$$\text{tr}[\mathbf{R}^2] = \sum_{i=1}^M \lambda_i^2 \quad (5)$$

Let

$$\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \dots, \lambda_M]^T$$

$$\mathbf{1} = [1, 1, \dots, 1]^T$$

We may then reformulate Eqs. (4) and (5) as follows, respectively:

$$\text{tr}[\mathbf{R}] = \boldsymbol{\lambda}^T \mathbf{1}$$

$$\text{tr}[\mathbf{R}^2] = \boldsymbol{\lambda}^T \boldsymbol{\lambda} = \|\boldsymbol{\lambda}\|^2$$

Applying the Cauchy-Schwarz inequality to the matrix product  $\boldsymbol{\lambda}^T \mathbf{1}$ :

$$|\lambda^T \mathbf{1}|^2 \leq \|\lambda\|^2 \cdot \|\mathbf{1}\|^2$$

Since  $\|\mathbf{1}\|^2 = M$ , it follows that

$$(\text{tr}[\mathbf{R}])^2 \leq \text{tr}[\mathbf{R}^2] \cdot M$$

or, equivalently,

$$\frac{\text{tr}[\mathbf{R}]}{\sqrt{M \text{tr}[\mathbf{R}^2]}} \leq 1 \quad (6)$$

Accordingly, we may rewrite Eq. (3) as

$$\frac{M_{\min}^{\text{LMS}}}{M_{\min}^{\text{RLS}}} \leq 1$$

or

$$M_{\min}^{\text{LMS}} \leq M_{\min}^{\text{RLS}}, \quad \mathbf{Q} = c_1 \mathbf{R}$$

- (b) Consider next the minimum mean-square deviation as the criterion of interest. For the LMS algorithm, we have

$$D_{\min}^{\text{LMS}} = \sigma_v \sqrt{M \text{tr}[\mathbf{R}^{-1} \mathbf{Q}]}$$

and for the RLS algorithm:

$$D_{\min}^{\text{RLS}} = \sigma_v \sqrt{\text{tr}[\mathbf{R}^{-1}] \text{tr}[\mathbf{Q}]}$$

Therefore,

$$\frac{D_{\min}^{\text{LMS}}}{D_{\min}^{\text{RLS}}} = \sqrt{\frac{M \text{tr}[\mathbf{R}^{-1} \mathbf{Q}]}{\text{tr}[\mathbf{R}^{-1}] \text{tr}[\mathbf{Q}]}}$$

For  $\mathbf{Q} = c_2 \mathbf{R}^{-1}$ ,

$$\frac{D_{\min}^{\text{LMS}}}{D_{\min}^{\text{RLS}}} = \frac{\sqrt{M \text{tr}[\mathbf{R}^{-2}]}}{\text{tr}[\mathbf{R}^{-1}]} \quad (7)$$

Since,

$$\mathbf{R}^{-1} = \sum_{i=1}^M \lambda_i^{-1} \mathbf{q}_i \mathbf{q}_i^H$$

and

$$\mathbf{R}^{-2} = \sum_{i=1}^M \lambda_i^{-2} \mathbf{q}_i \mathbf{q}_i^H,$$

it follows that

$$\text{tr}[\mathbf{R}^{-1}] = \sum_{i=1}^M \lambda_i^{-1}$$

and

$$\text{tr}[\mathbf{R}^{-2}] = \sum_{i=1}^M \lambda_i^{-2}$$

Let

$$\lambda_{\text{inv}} = [\lambda_1^{-1}, \lambda_2^{-1}, \dots, \lambda_M^{-1}]^T$$

$$\mathbf{1} = [1, 1, \dots, 1]^T$$

Hence,

$$\text{tr}[\mathbf{R}^{-1}] = \lambda_{\text{inv}}^T \mathbf{1}$$

$$\text{tr}[\mathbf{R}^{-2}] = \lambda_{\text{inv}}^T \lambda_{\text{inv}} = \|\lambda_{\text{inv}}\|^2$$

Applying the Cauchy-Schwarz inequality to the matrix product  $\lambda_{\text{inv}}^T \mathbf{1}$ , we may write

$$|\lambda_{\text{inv}}^T \mathbf{1}|^2 \leq \|\lambda_{\text{inv}}\|^2 \|\mathbf{1}\|^2$$

That is,

$$(\text{tr}[\mathbf{R}^{-1}])^2 \leq \text{tr}[\mathbf{R}^{-2}] \cdot M$$

or equivalently,

$$\frac{\sqrt{M \text{tr}[\mathbf{R}^{-2}]}}{\text{tr}[\mathbf{R}^{-1}]} \geq 1 \quad (8)$$

Accordingly, we may rewrite Eq. (7) as

$$\frac{D_{\min}^{\text{LMS}}}{D_{\min}^{\text{RLS}}} \geq 1$$

That is,

$$D_{\min}^{\text{LMS}} \geq D_{\min}^{\text{RLS}}, \quad \mathbf{Q} = c_2 \mathbf{R}^{-1}$$

14.8 As with Problem 14.7, here again we simplify the presentation by using the notations

$$\mathbf{R}_u = \mathbf{R} \text{ and } \mathbf{R}_\omega = \mathbf{Q}$$

(a) LMS Algorithm

For  $\mathbf{Q} = c_1 \mathbf{R}$ , or equivalently,  $\mathbf{R}^{-1} \mathbf{Q} = c_1 \mathbf{I}$ :

$$\begin{aligned} \text{(i)} \quad D_{\min}^{\text{LMS}} &= \sigma_v \sqrt{M} (\text{tr}[\mathbf{R}^{-1} \mathbf{Q}])^{1/2} \\ &= \sigma_v \sqrt{M} (\text{tr}[c_1 \mathbf{I}])^{1/2} \\ &= \sigma_v M \sqrt{c_1} \end{aligned} \quad (1)$$

$$\begin{aligned}\mu_{\text{opt}} &= \frac{1}{\sigma_v \sqrt{M}} (\text{tr}[\mathbf{R}^{-1} \mathbf{Q}])^{1/2} \\ &= \sqrt{c_1} / \sigma_v\end{aligned}\tag{2}$$

$$\begin{aligned}\text{(ii)} \quad M_{\text{min}}^{\text{LMS}} &= \frac{1}{\sigma_v} (\text{tr}[\mathbf{R}] \text{tr}[\mathbf{Q}])^{1/2} \\ &= \frac{\sqrt{c_1}}{\sigma_v} \text{tr}[\mathbf{R}]\end{aligned}\tag{3}$$

Givben the two-by-two correlation matrix

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{21} \\ r_{21} & r_{22} \end{bmatrix}$$

we may write

$$\text{tr}[\mathbf{R}] = r_{11} + r_{22}\tag{4}$$

Therefore, substituting Eq. (4) into (3):

$$M_{\text{min}}^{\text{LMS}} = \frac{\sqrt{c_1} (r_{11} + r_{22})}{\sigma_v}\tag{5}$$

The optimum step-size parameter is

$$\mu_{\text{opt}} = \frac{1}{\sigma_v} \left( \frac{\text{tr}[\mathbf{Q}]}{\text{tr}[\mathbf{R}]} \right)^{1/2} = \frac{\sqrt{c_1}}{\sigma_v}\tag{6}$$

which is the same as the  $\mu_{\text{opt}}$  for minimum  $D^{\text{LMS}}$ .

Consider next the case of

$$\mathbf{Q} = c_2 \mathbf{R}^{-1}:$$

$$\text{(iii)} \quad D_{\text{min}}^{\text{LMS}} = \sigma_v \sqrt{M} (\text{tr}[\mathbf{R}^{-1} \mathbf{Q}])^{1/2} = \sigma_v \sqrt{2c_2} (\text{tr}[\mathbf{R}^{-2}])^{1/2}\tag{7}$$

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{21} \\ r_{21} & r_{22} \end{bmatrix}$$

$$\mathbf{R}^{-1} = \frac{1}{\Delta_r} \begin{bmatrix} r_{22} & -r_{21} \\ -r_{21} & r_{11} \end{bmatrix}, \quad \Delta_r = r_{11}r_{22} - r_{21}^2$$

$$\mathbf{R}^{-2} = \mathbf{R}^{-1}\mathbf{R}^{-1} = \frac{1}{\Delta_r^2} \begin{bmatrix} r_{22}^2 + r_{21}^2 & -r_{21}(r_{11} + r_{22}) \\ -r_{21}(r_{11} + r_{22}) & r_{11}^2 + r_{21}^2 \end{bmatrix}$$

$$\text{tr}[\mathbf{R}^{-2}] = \frac{1}{\Delta_r^2} [r_{11}^2 + 2r_{21}^2 + r_{22}^2] \quad (8)$$

Substituting Eq. (8) into (7), and noting that  $M = 2$ :

$$D_{\min}^{\text{LMS}} = \sigma_v \sqrt{2c_2} \frac{\sqrt{r_{11}^2 + 2r_{21}^2 + r_{22}^2}}{r_{11}r_{22} - r_{21}^2} \quad (9)$$

The optimum step-size parameter is

$$\begin{aligned} \mu_{\text{opt}} &= \frac{1}{\sigma_v \sqrt{M}} (\text{tr}[\mathbf{R}^{-1}\mathbf{Q}])^{1/2} \\ &= \frac{1}{\sigma_v} \sqrt{\frac{c_2}{2}} (\text{tr}[\mathbf{R}^{-2}])^{1/2} \\ &= \frac{1}{\sigma_v} \sqrt{\frac{c_2}{2}} \frac{\sqrt{r_{11}^2 + 2r_{21}^2 + r_{22}^2}}{r_{11}r_{22} - r_{21}^2} \end{aligned} \quad (10)$$

$$\text{(iv)} \quad M_{\min}^{\text{LMS}} = \frac{1}{\sigma_v} (\text{tr}[\mathbf{R}]\text{tr}[\mathbf{Q}])^{1/2}$$

$$\begin{aligned}
&= \frac{\sqrt{c_2}}{\sigma_v} (\text{tr}[\mathbf{R}] \text{tr}[\mathbf{R}^{-1}])^{1/2} \\
&= \frac{\sqrt{c_2}}{\sigma_v} \frac{r_{11} + r_{22}}{(r_{11}r_{22} - r_{21}^2)^{1/2}}
\end{aligned} \tag{11}$$

$$\begin{aligned}
\mu_{\text{opt}}^{\text{LMS}} &= \frac{1}{\sigma_v} \left( \frac{\text{tr}[\mathbf{Q}]}{\text{tr}[\mathbf{R}]} \right)^{1/2} \\
&= \frac{\sqrt{c_2}}{\sigma_v} \left( \frac{\text{tr}[\mathbf{R}^{-1}]}{\text{tr}[\mathbf{R}]} \right)^{1/2} \\
&= \frac{\sqrt{c_2}}{\sigma_v} (r_{11}r_{22} - r_{21}^2)^{-1/2}
\end{aligned} \tag{12}$$

which is different from the  $\mu_{\text{opt}}$  for minimum  $D^{\text{LMS}}$ .

(b) RLS Algorithm

For  $\mathbf{Q} = c_1 \mathbf{R}$ :

$$\begin{aligned}
\text{(i)} \quad D_{\text{min}}^{\text{RLS}} &= \sigma_v (\text{tr}[\mathbf{R}^{-1}] \text{tr}[\mathbf{Q}])^{1/2} \\
&= \sigma_v \sqrt{c_1} (\text{tr}[\mathbf{R}^{-1}] \text{tr}[\mathbf{R}])^{1/2} \\
&= \sigma_v \sqrt{c_1} \frac{r_{11} + r_{22}}{(r_{11}r_{22} - r_{21}^2)^{1/2}}
\end{aligned} \tag{13}$$

$$\begin{aligned}
\lambda_{\text{opt}} &= 1 - \frac{1}{\sigma_v} \left( \frac{\text{tr}[\mathbf{Q}]}{\text{tr}[\mathbf{R}^{-1}]} \right)^{1/2} \\
&= 1 - \frac{\sqrt{c_1}}{\sigma_v} \sqrt{r_{11}r_{22} - r_{21}^2}
\end{aligned} \tag{14}$$

$$\begin{aligned}
\text{(ii)} \quad M_{\min}^{\text{RLS}} &= \frac{1}{\sigma_v} (M \text{tr}[\mathbf{R}\mathbf{Q}])^{1/2} \\
&= \frac{\sqrt{c_1}}{\sigma_v} (2 \text{tr}[\mathbf{R}^2])^{1/2}, \quad M=2 \\
&= \frac{\sqrt{2c_1}}{\sigma_v} \sqrt{r_{11}^2 + 2r_{21}^2 + r_{22}^2}
\end{aligned} \tag{15}$$

$$\begin{aligned}
\lambda_{\text{opt}} &= 1 - \frac{1}{\sigma_v} \left( \frac{1}{M} \text{tr}[\mathbf{R}\mathbf{Q}] \right)^{1/2}, \quad M=2 \\
&= 1 - \frac{1}{\sigma_v} \sqrt{\frac{c_1}{2}} \sqrt{r_{11}^2 + 2r_{21}^2 + r_{22}^2}
\end{aligned} \tag{16}$$

which is different from the  $\lambda_{\text{opt}}$  for minimum  $D^{\text{RLS}}$ .

Consider next the case of  $\mathbf{Q} = c_2 \mathbf{R}^{-1}$  or equivalently  $\mathbf{R}\mathbf{Q} = c_2 \mathbf{I}$ :

$$\begin{aligned}
\text{(iii)} \quad D_{\min}^{\text{RLS}} &= \sigma_v (\text{tr}[\mathbf{R}^{-1}] \text{tr}[\mathbf{Q}])^{1/2} \\
&= \sigma_v \sqrt{c_2} (\text{tr}[\mathbf{R}^{-1}]) \\
&= \sigma_v \sqrt{c_2} \frac{r_{11} + r_{22}}{r_{11}r_{22} - r_{21}^2}
\end{aligned} \tag{17}$$

$$\begin{aligned}
\lambda_{\text{opt}} &= 1 - \frac{1}{\sigma_v} \left( \frac{\text{tr}[\mathbf{Q}]}{\text{tr}[\mathbf{R}^{-1}]} \right)^{1/2} \\
&= 1 - \frac{\sqrt{c_2}}{\sigma_v} \sqrt{r_{11}r_{22} - r_{21}^2}
\end{aligned} \tag{18}$$

$$\text{(iv)} \quad M_{\min}^{\text{RLS}} = \frac{1}{\sigma_v} (M \text{tr}[\mathbf{R}\mathbf{Q}])^{1/2}, \quad M=2$$



$$= \frac{2}{\sigma_v} \sqrt{c_2} \quad (19)$$

$$\begin{aligned} \lambda_{\text{opt}} &= 1 - \frac{1}{\sigma_v} \left( \frac{1}{M} \text{tr}[c_2 \mathbf{I}] \right), \quad M = 2 \\ &= 1 - \frac{\sqrt{c_2}}{\sigma_v} \end{aligned} \quad (20)$$

which is different from the  $\lambda_{\text{opt}}$  for minimum  $D_{\text{min}}^{\text{RLS}}$ .

Comparisons of LMS and RLS algorithms:

1.  $\mathbf{Q} = c_1 \mathbf{R}$

$$\frac{D_{\text{min}}^{\text{LMS}}}{D_{\text{min}}^{\text{RLS}}} = \frac{2\sqrt{r_{11}r_{22} - r_{21}^2}}{r_{11} + r_{22}}$$

$$\frac{M_{\text{min}}^{\text{LMS}}}{M_{\text{min}}^{\text{RLS}}} = \frac{r_{11} + r_{22}}{\sqrt{2}\sqrt{r_{11}^2 + 2r_{21}^2 + r_{22}^2}}$$

2.  $\mathbf{Q} = c_2 \mathbf{R}^{-1}$

$$\frac{D_{\text{min}}^{\text{LMS}}}{D_{\text{min}}^{\text{RLS}}} = \sqrt{2} \frac{\sqrt{r_{11}^2 + 2r_{21}^2 + r_{22}^2}}{r_{11} + r_{22}}$$

$$\frac{M_{\text{min}}^{\text{LMS}}}{M_{\text{min}}^{\text{RLS}}} = \frac{r_{11} + r_{22}}{2\sqrt{r_{11}r_{22} - r_{21}^2}}$$

## CHAPTER 15

15.1 For the output error method

$$y(n) = \sum_{i=0}^M a_i(n)u(n-i) + \sum_{i=1}^N b_i(n)y(n-i) \quad (1)$$

Taking the  $z$ -transform of both sides of Eq. (1)

$$Y(z) = A(z)U(z) + B(z)Y(z) \quad (2)$$

For the equation error method

$$\begin{aligned} y'(n) &= \sum_{i=0}^M a_i(n)u(n-i) + \sum_{i=1}^N b_i(n)d(n-i) \\ &= \sum_{i=0}^M a_i(n)u(n-i) + \sum_{i=1}^N b_i(n)[e'(n-i) + y'(n-i)] \\ &= \sum_{i=0}^M a_i(n)u(n-i) + \sum_{i=1}^N b_i(n)e'(n-i) + \sum_{i=1}^N b_i(n)y'(n-i) \end{aligned} \quad (3)$$

Taking the  $z$ -transform of both sides of Eq. (3):

$$\begin{aligned} Y'(z) &= A(z)U(z) + B(z)E'(z) + B(z)Y'(z) \\ &= A(z)U(z) + B(z)(1 - B(z))E(z) + B(z)Y'(z) \\ (1 - B(z))Y'(z) &= A(z)U(z) + B(z)(1 - B(z))E(z) \\ Y'(z) &= \frac{1}{1 - B(z)}A(z)U(z) + B(z)E(z), \end{aligned} \quad (4)$$

which explains why the transfer function  $\frac{1}{1 - B(z)}$  comes in.

15.2 The approximation used in Eqs. (15.18) and (15.19), reproduced here for convenience of presentation,

$$\alpha_j(n) \approx \alpha_0(n-j) = \frac{\partial y(n-j)}{\partial a_0(n-j)} \quad j = 1, 2, \dots, M$$

$$\beta_j(n) \approx \beta_1(n-j+1) \quad j = 2, \dots, N$$

is based on the following observation: When the adaptive filtering algorithm reaches a convergent point, then the parameters of the filter could be held constant, at which point the two equations become exact. Hence,

$$\alpha_j(n) = \frac{\partial y(n)}{\partial a_j(n)} \approx \frac{\partial y(n-j)}{\partial a_0(n)} = \alpha_0(n-j) \quad j = 1, 2, \dots, M$$

$$\beta_j(n) = \frac{\partial y(n)}{\partial b_j(n)} \approx \frac{\partial y(n-j+1)}{\partial b_1(n)} = \beta_1(n-j+1) \quad j = 2, \dots, N$$

15.3 To mitigate the stability problem in adaptive IIR filters, the following measures can be born in mind in formulating the algorithm:

1. Use of the “equation error method” to replace “output error method” for rough approximation.
2. Use of a lattice structure for the IIR filter<sup>1</sup>
3. Combine IIR and FIR structures to devise hybrid filter, e.g., Laguerre transversal filter.

15.4 LMS Algorithm for Laguerre transversal filter:

Initialization:

Initialize the weights  $w_0, w_1, \dots, w_M$  by setting them equal to zero, or else by assigning them small random values.

Computation:

$$L(z, \alpha) = \frac{z^{-1} - \alpha}{1 - \alpha z^{-1}}$$

$$L_i(z, \alpha) = L_o(z, \alpha)(L(z, \alpha))^i \quad i = 0, 1, 2, \dots$$

$$U_i(z, \alpha) = L_i(z, \alpha)U(z) \quad i = 0, 1, 2, \dots$$

---

<sup>1</sup>. Miao, K.X., et al., IEEE Trans. Signal Processing, **42**, pp. 721-742, 1994.

$$y(n) = \sum_{i=0}^M w_i Z^{-1} [U_i(z, \alpha)] = \sum_{i=0}^M w_i u_i(n, \alpha)$$

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

Update:

$$w_i(n+1) = w_i(n) + \tilde{\mu} z^{-1} [U_i(z, \alpha)] e(n)$$

$$= w_i(n) + \tilde{\mu} u_M(n, \alpha) e(n)$$

15.5 In order to prove the shift invariance property, we need to prove a Lemma first.

**Lemma:** Let  $y_1(n)$  and  $y_2(n)$  be two random signals obtained by linear time-invariant filtering of the (wide-sense) stationary signal  $x(n)$ , then

$$E \left\{ L[y_1(n)] \{L[y_2(n)]\}^* \right\} = E \left\{ y_1(n) y_2^*(n) \right\}$$

where  $L$  denotes the filtering operation.

**Proof:**

Let  $H_1(z)$  and  $H_2(z)$  be the filters used to get  $y_1(n)$  and  $y_2(n)$  from  $x(n)$ . If  $y_1(n)$  and  $y_2(n)$  are jointly wide-sense stationary, and

$$E \left\{ y_1(n) y_2^*(n) \right\} = \frac{1}{2\pi j} \oint_{z=e^{j\omega}} H_1(z) H_2^*(z) S_x(z) \frac{dz}{z}$$

where  $S_x(e^{j\omega})$  is the power spectrum of input signal  $x(n)$ , then we may write

$$E \left\{ L[y_1(n)] \{L[y_2(n)]\}^* \right\}$$

$$= \frac{1}{2\pi j} \oint_{z=e^{j\omega}} |L_1(z)|^2 H_1(z) H_2^*(z) S_x(z) \frac{dz}{z}$$

$$= \frac{1}{2\pi j} \oint_{z=e^{j\omega}} H_1(z) H_2^*(z) S_x(z) \frac{dz}{z} \quad \text{for } |L_1(z)|^2 = 1$$

$$= E \left\{ y_1(n) y_2^*(n) \right\}$$

With this Lemma at hand, now examine  $\xi_j(n)$  and  $\xi_k^*(n)$  and assume  $j \geq k$  without loss of generality.

Applying the Lemma given above, we have

$$E \left\{ \xi_j(n) \xi_k^*(n) \right\} = \frac{1}{2\pi} \int_{-\pi}^{\pi} L_j(e^{j\omega}) L_k^*(e^{j\omega}) S_x(e^{j\omega}) d\omega$$

Since  $L_j(z) = L_0(z) \left( \frac{z^{-1} - a}{1 - az^{-1}} \right)^j$ , and  $\left| \frac{z^{-1} - a}{1 - az^{-1}} \right| = 1$ , for  $|z| = 1$  it follows that

$$\begin{aligned} & E \left\{ \xi_j(n) \xi_k^*(n) \right\} \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} L_0(e^{j\omega})^2 \left( \frac{e^{-j\omega} - a}{1 - ae^{-j\omega}} \right)^{j-k} S_x(e^{j\omega}) d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} L_{j-k}(e^{j\omega}) L_0^*(e^{j\omega}) S_x(e^{j\omega}) d\omega \\ &= E \left\{ \xi_{j-k}(n) \xi_0^*(n) \right\} \end{aligned}$$

The above proof naturally holds for the real-valued situation as a special case:

$$E \{ \xi_j(n) \xi_k(n) \} = E \{ \xi_{j-k}(n) \xi_0(n) \}$$

$$15.6 \quad \xi(n) = [\xi_0(n), \xi_1(n), \dots, \xi_M(n)]^T$$

From the solution to Problem 15.5, we know  $\xi_j(n)$  satisfies the shift-invariant property

$$E[\xi_j(n) \xi_k(n)] = E[\xi_{j-k}(n) \xi_0(n)] = C_\xi(j-k) \quad (1)$$

Now examine the correlation matrix  $\mathbf{R} = E[\xi(n)\xi^T(n)]$  which is a  $M$ -by- $M$  matrix. For any element  $(j,k)$  of this matrix ( $1 \leq j, k \leq M$ ) we know that Eq. (1) holds.

For any  $(j+1, k+1)$  element,

$$\begin{aligned} E[\xi_{j+1}(n)\xi_{k+1}(n)] &= E[\xi_{j-k}(n)\xi_0(n)] = E[\xi_j(n)\xi_k(n)] \\ &= C_\xi(j-k) \end{aligned}$$

For the  $(j+M, k+M)$  element we have

$$E[\xi_{j+M}(n)\xi_{k+M}(n)] = E[\xi_j(n)\xi_k(n)]$$

Thus the diagonal and sub-diagonal elements of  $\mathbf{R}$  will be the same, hence  $\mathbf{R}$  is a Toeplitz matrix.

- 15.7 (a) The derivation of the algorithm summarized in Table 15.3 is based on the Burg formula for reflection coefficients of a lattice predictor for real-valued data [see Eq. (2.7)]

$$\begin{aligned} \hat{\kappa}_m(n) &= -\frac{2 \sum_{i=1}^n b_{m-1}(i-1)f_{m-1}(i)}{\sum_{i=1}^n (|f_{m-1}(i)|^2 + |b_{m-1}(i-1)|^2)} \\ &= -\frac{\Delta_m(n)}{D_m(n)} \end{aligned} \tag{1}$$

where

$$\begin{aligned} \Delta_m(n) &= 2 \sum_{i=1}^n b_{m-1}(i-1)f_{m-1}(i) \\ D_m(n) &= \sum_{i=1}^n f_{m-1}^2(i) + b_{m-1}^2(i-1) \end{aligned}$$

For recursive computations of the numerator  $\Delta_m(n)$  and denominator  $D_m(n)$ , we proceed by writing

$$\begin{aligned}
\Delta_m(n) &= 2 \sum_{i=1}^n b_{m-1}(i-1) f_{m-1}(i) + 2b_{m-1}(n-1) f_{m-1}(n) \\
&= \Delta_m(n-1) + 2b_{m-1}(n-1) f_{m-1}(n) \\
D_m(n) &= \sum_{i=1}^n (f_{m-1}^2(i) + b_{m-1}^2(i-1)) + f_{m-1}^2(n) + b_{m-1}^2(n-1) \\
&= D_m(n-1) + f_{m-1}^2(n) + b_{m-1}^2(n-1)
\end{aligned}$$

To exercise designer control on these two recursive computations, we modify them respectively as follows:

$$\Delta_m(n) = \lambda \Delta_m(n-1) + 2b_{m-1}(n-1) f_{m-1}(n) \quad (2)$$

$$D_m(n) = \lambda D_m(n-1) + f_{m-1}^2(n) + b_{m-1}^2(n-1) \quad (3)$$

where  $\lambda$  is a design parameter,  $0 < \lambda \leq 1$ .

Finally, we apply Eq. (15.35) of the text in accordance with the Laguerre formulation of the lattice filter:

$$\tilde{b}_{m-1}(n) = b_{m-1}(n-1) + \alpha(\tilde{b}_{m-1}(n-1) - b_{m-1}(n)) \quad (4)$$

where  $\alpha$  is another design parameter. Thus, using  $\tilde{b}_{m-1}(n)$  in place of  $b_{m-1}(n-1)$  in Eqs. (2) and (3), we get the entries in the first two lines of the time updates. All that remains for us to do is to use Eq. (1) to compute the reflection coefficient  $\hat{\kappa}_m(n)$ ; hence, the third line of the time updates.

- (b) Using GAL algorithm, compared to conventional GAL algorithm, the lattice structure has a faster initial convergence and less computational complexity.

## CHAPTER 16

16.1 (a) The received signal of a digital communication system in baseband form is given by

$$u(t) = \sum_{m=-\infty}^{\infty} x_m h(t - mT) + v(t)$$

where  $x_k$  is the transmitted symbol,  $h(t)$  is the overall impulse response,  $T$  is the symbol period, and  $v(t)$  is the channel noise. Evaluating  $u(t)$  at times  $t_1$  and  $t_2$ :

$$u(t_1) = \sum_{m=-\infty}^{\infty} x_m h(t_1 - mT) + v(t_1)$$

$$u(t_2) = \sum_{m=-\infty}^{\infty} x_l h(t_2 - lT) + v(t_2)$$

Hence, the autocorrelation function of  $u(t)$  is

$$\begin{aligned} r_u(t_1, t_2) &= E[u(t_1)u^*(t_2)] \\ &= E \left[ \sum_{m=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} x_m x_l^* h(t_1 - mT) h(t_2 - lT) \right] \\ &\quad + E[v(t_1)v^*(t_2)] \\ &= \sum_{m=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} r_x(mT - lT) h(t_1 - mT) h^*(t_2 - lT) \\ &\quad + \sigma_v^2 \delta(t_1 - t_2) \end{aligned} \tag{1}$$

where  $r_x(mT - lT)$  is the autocorrelation function of the transmitted signal. From Eq. (1) we immediately see that

$$r_u(t_1 + T, t_2 + T) = r_u(t_1, t_2)$$

which shows that  $u(t)$  is indeed cyclostationary in the wide sense.



(b) Applying the definitions

$$r_u^\alpha(\tau) = \frac{1}{T} \int_{-T/2}^{T/2} r_u\left(t + \frac{\tau}{2}, t - \frac{\tau}{2}\right) e^{j2\pi\alpha t} dt$$

$$S_u^\alpha(\omega) = \int_{-\infty}^{\infty} r_u^\alpha(\tau) e^{-j2\pi f\tau} d\tau, \quad \omega = 2\pi f$$

$$\alpha = k/T, \quad k = 0, \pm 1, \pm 2, \dots$$

to the result obtained in part (a), we may show that

$$\begin{aligned} S_u^{k/T} &= \frac{1}{T} H(e^{j\omega + jk\pi/T}) H^*(e^{j\omega} - e^{jk\pi/T}) S_x\left(\omega + \frac{k\pi}{T}\right) \\ &\quad + \sigma_v^2 \delta(k), \quad k = 0, \pm 1, \pm 2, \dots \end{aligned} \quad (2)$$

As a check, we see that for  $k = 0$ , Eq. (2) reduces to the standard result:

$$S_u(\omega) = \frac{1}{T} |H(e^{j\omega})|^2 S_x + \sigma_v^2$$

Let  $\psi_k(\omega)$  denote the phase response of  $S_u^{k/T}$  and  $\Phi(\omega)$  denote the phase response of the channel. Then recognizing that the power spectral density  $S_x(\omega)$  of the transmitted signal is real valued, we readily find from Eq. (2) that

$$\Psi_k(\omega) = \Phi\left(\omega + \frac{k\pi}{T}\right) - \Phi\left(\omega - \frac{k\pi}{T}\right), \quad k = 0, \pm 1, \pm 2, \dots \quad (3)$$

(c) From the formula for the inverse Fourier transform, we have

$$\psi_k(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Psi_k(\omega) e^{j\omega\tau} d\omega$$

$$\phi(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Phi_k(\omega) e^{j\omega\tau} d\omega$$

Applying these definitions to Eq. (3):

$$\psi_k(\tau) = \phi(\tau) e^{-j\pi k\tau/T} - \phi(\tau) e^{j\pi k\tau/T}$$

$$= -2j\phi(\tau) \sin\left(\frac{k\pi\tau}{T}\right), \quad k = 0, \pm 1, \pm 2, \dots \quad (4)$$

On the basis of Eq. (4), we may make two important observations:

- (1) For  $k = 0$  and  $k\tau/T$  equal to an integer,  $\psi_k(\tau)$  is identically zero. For these values of  $k$ ,  $\phi(\tau)$  cannot be determined. This means that for an arbitrary channel, the unknown phase response  $\Phi(f)$  cannot be identified for  $k = 0$  or  $k\tau/T = \text{integer}$  by using cyclostationary second-order statistics of the channel output.
- (2) For  $k = \pm 2$  and higher in absolute value, the use of  $\psi_k(\tau)$  does not reveal any more information about the channel phase response than what can be obtained with  $k = \pm 1$ . We may therefore just as well work with  $k = 1$ , for which  $\psi_k(\tau)$  has the largest support, as shown by

$$\psi_1(\tau) = -2j\phi(\tau) \sin\left(\frac{\pi\tau}{T}\right)$$

That is,

$$\phi(\tau) = \frac{j\psi_1(\tau)}{2\sin(\pi\tau/T)}$$

which shows that  $\phi(\tau)$  is identifiable from  $\psi_1(\tau)$  except for  $\tau = mT$ , where  $m$  is an integer.

16.2 In the noise-free case, we have

$$\mathbf{U}_n = \mathbf{H}\mathbf{x}_n \quad (1)$$

where  $\mathbf{H}$  is the  $LN$ -by- $(M+N)$  multichannel filtering matrix,  $\mathbf{x}_n$  is the  $(M+N)$ -by-1 transmitted signal vector, and  $\mathbf{u}_n$  is the  $LN$ -by-1 multichannel received signal vector. Let  $\mathbf{u}_n$  be applied to a multichannel structure characterized by the  $(M+N)$ -by- $LN$  filtering matrix  $\mathbf{T}$  such that we have

$$\mathbf{T}\mathbf{U}_n = \mathbf{x}_n \quad (2)$$

This zero-forcing condition ensures the perfect recovery of  $\mathbf{x}_n$  from  $\mathbf{u}_n$ . Substituting Eq. (1) into (2):

$$\mathbf{T}\mathbf{H} = \mathbf{I} \quad (3)$$

where  $\mathbf{I}$  is the identity matrix. From Eq. (3) we immediately deduce that

$$\mathbf{T} = \mathbf{H}^+$$

where  $\mathbf{H}^+$  is the pseudoinverse of  $\mathbf{H}$ .

- 16.3 The relationship between  $\mathbf{g}$  and  $\mathbf{H}$  on the one hand and  $\mathbf{h}$  and  $\mathbf{g}_k$  on the other hand, as described in Eq. (16.27), namely,

$$\mathbf{g}_k \mathbf{H} \mathbf{H}^H \mathbf{g}_k = \mathbf{h}^H \mathbf{g}_k \mathbf{g}_k \mathbf{h}$$

follows directly from the two sets of definitions

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}^{(0)} \\ \mathbf{H}^{(1)} \\ \vdots \\ \mathbf{H}^{(L-1)} \end{bmatrix}, \quad \mathbf{H}^{(l)} = \begin{bmatrix} h_0^{(l)} & \dots & h_M^{(l)} & \dots & 0 \\ 0 & \dots & h_{M-1}^{(l)} & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & h_0^{(l)} & \dots & h_M^{(l)} \end{bmatrix}$$

$$\mathbf{g}_k = \begin{bmatrix} \mathbf{g}_k^{(0)} \\ \mathbf{g}_k^{(1)} \\ \vdots \\ \mathbf{g}_k^{(L-1)} \end{bmatrix}$$

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{G}_k^{(0)} \\ \mathbf{G}_k^{(1)} \\ \vdots \\ \mathbf{G}_k^{(L-1)} \end{bmatrix}, \quad \mathbf{G}_k^{(l)} = \begin{bmatrix} g_{k,0}^{(l)} & \dots & g_{k,N-1}^{(l)} & \dots & 0 \\ 0 & \dots & g_{k,N-2}^{(l)} & \dots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \dots & g_{k,0}^{(l)} & \dots & g_{k,N-1}^{(l)} \end{bmatrix}$$

$$\mathbf{h} = \begin{bmatrix} \mathbf{h}^{(0)} \\ \mathbf{h}^{(1)} \\ \vdots \\ \mathbf{h}^{(L-1)} \end{bmatrix}$$

16.4 For a noiseless channel, the received signal is

$$u_n^{(l)} = \sum_{m=0}^M h_n^{(l)} x_{n-m}, \quad l = 0, 1, \dots, L-1 \quad (1)$$

By definition, we have

$$H^l(z) = \sum_{m=0}^M h_m^{(l)} z^{-m}$$

where  $z$  is the unit-delay operator. We therefore rewrite Eq. (1) in the equivalent form:

$$u_n^{(l)} = H^l(z)[x_n] \quad (2)$$

where  $H^l(z)$  acts as an operator. Multiplying Eq. (2) by  $G^l(z)$  and then summing over  $l$ :

$$\sum_{l=0}^{L-1} G^l(z)[u_n^{(l)}] = \sum_{l=0}^{L-1} G^l(z)H^l(z)[x_n] \quad (3)$$

According to the generalized Bezout identity:

$$\sum_{l=0}^{L-1} G^l(z)H^l(z) = 1$$

We may therefore simplify Eq. (3) to

$$\sum_{l=0}^{L-1} G^l(z)[u_n^{(l)}] = x_n$$

Let

$$y^{(l)}(n) = G^l(z)[u_n^{(l)}], \quad l = 0, 1, \dots, L-1$$

Let  $G^{(l)}(z)$  be written in the expanded form:

$$G^l(z) = \sum_{i=0}^M g_i^{(l)} z^{-i}$$

Hence,

$$\begin{aligned} y^{(l)}(n) &= \sum_{i=0}^M g_i^{(l)} z^{-i} [u_n^{(l)}] \\ &= \sum_{i=0}^M g_i^{(l)} u_{n-i}^{(l)} \end{aligned}$$

From linear prediction theory, we recognize that

$$\hat{u}_{n+1}^{(l)} = \sum_{i=0}^M g_i^{(l)} u_{n-i}^{(l)}$$

It follows therefore that

$$y^{(l)}(n) = z^{-i} [\hat{u}_{n+1}^{(l)}]$$

and so we may rewrite

$$x_n = \sum_{l=0}^{L-1} z^{-i} [\hat{u}_{n+1}^{(l)}]$$

16.5 We are given

$$\hat{x} = \frac{1}{f_Y(y)} \int_{-\infty}^{\infty} x f_V(y - c_0 x) f_X(x) dx$$

where

$$f_X(x) = \begin{cases} 1/2\sqrt{3} & -\sqrt{3} \leq x < \sqrt{3} \\ 0, & \text{otherwise} \end{cases}$$

$$f_V(v) = \frac{1}{\sqrt{2\pi}\sigma} e^{-v^2/2\sigma^2},$$

and

$$f_Y(y) = \int_{-\infty}^{\infty} f_X(x) f_V(y - c_0 x) dx$$

Hence,

$$\hat{x} = \frac{\int_{-\sqrt{3}}^{\sqrt{3}} x e^{-(y-c_0 x)^2/2\sigma^2} dx}{\int_{-\sqrt{3}}^{\sqrt{3}} e^{-(y-c_0 x)^2/2\sigma^2} dx} \quad (1)$$

Let

$$t = (y - c_0 x)/\sigma$$

$$dt = -c_0 dx/\sigma$$

Then, we may rewrite Eq. (1) as

$$\begin{aligned} \hat{x} &= \frac{\int_{(y-\sqrt{3}c_0)/\sigma}^{(y+\sqrt{3}c_0)/\sigma} \frac{\sigma}{c_0} (y - t\sigma) e^{-t^2/2} dt}{\int_{(y-\sqrt{3}c_0)/\sigma}^{(y+\sqrt{3}c_0)/\sigma} \frac{\sigma}{c_0} e^{-t^2/2} dt} \\ &= \frac{1}{c_0} y - \frac{\sigma}{c_0} \frac{\int_{(y-\sqrt{3}c_0)/\sigma}^{(y+\sqrt{3}c_0)/\sigma} t e^{-t^2/2} dt}{\int_{(y-\sqrt{3}c_0)/\sigma}^{(y+\sqrt{3}c_0)/\sigma} e^{-t^2/2} dt} \end{aligned} \quad (2)$$

Next we recognize the following two results:

$$\begin{aligned}
\int_{(y-\sqrt{3}c_0)/\sigma}^{(y+\sqrt{3}c_0)/\sigma} t e^{-t^2/2} dt &= \left[ e^{-t^2/2} \right]_{(y-\sqrt{3}c_0)/\sigma}^{(y+\sqrt{3}c_0)/\sigma} \\
&= \sqrt{2\pi} \left\{ Z\left(\frac{y+\sqrt{3}c_0}{\sigma}\right) - Z\left(\frac{y-\sqrt{3}c_0}{\sigma}\right) \right\} \\
\int_{(y-\sqrt{3}c_0)/\sigma}^{(y+\sqrt{3}c_0)/\sigma} e^{-t^2/2} dt &= \int_{(y-\sqrt{3}c_0)/\sigma}^{\infty} e^{-t^2/2} dt - \int_{(y+\sqrt{3}c_0)/\sigma}^{\infty} e^{-t^2/2} dt \\
&= \sqrt{2\pi} \{ Q((y-\sqrt{3}c_0)/\sigma) - Q((y+\sqrt{3}c_0)/\sigma) \}
\end{aligned}$$

We may therefore rewrite Eq. (2) in the compact form:

$$\hat{x} = \frac{1}{c_0} y + \frac{\sigma}{c_0} \frac{Z((y+\sqrt{3}c_0)/\sigma) - Z((y-\sqrt{3}c_0)/\sigma)}{Q((y-\sqrt{3}c_0)/\sigma) - Q((y+\sqrt{3}c_0)/\sigma)}$$

16.6 For convergence of a Bussgang algorithm:

$$E[y(n)y(n+k)] = E[y(n)g(y(n+k))]$$

For large  $n$ :

$$E[y^2(n)] = E[y(n)g(y(n))]$$

For  $y(n) = x(n)$  to achieve perfect equalization:

$$E[\hat{x}^2] = E[xg(x)]$$

With  $\hat{x}$  being of zero mean and unit variance, we thus have

$$E[\hat{x}g(x)] = 1$$

16.7 We start with (for real-valued data)

$$\begin{aligned}
y(n) &= \sum_i \hat{w}_i(n) u(n-i) \\
&= \hat{\mathbf{w}}^T(n) \mathbf{u}(n) = \mathbf{u}^T(n) \hat{\mathbf{w}}(n)
\end{aligned}$$

Let

$$\mathbf{y} = [y(n_1), y(n_2), \dots, y(n_K)]^T$$

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}^T(n_1) \\ \mathbf{u}^T(n_2) \\ \vdots \\ \mathbf{u}^T(n_K) \end{bmatrix}$$

Assuming that  $\hat{\mathbf{w}}(n)$  has a constant value  $\hat{\mathbf{w}}$ , averaged over the whole block of data, then

$$\mathbf{y} = \mathbf{U}\hat{\mathbf{w}}$$

The solution for  $\hat{\mathbf{w}}$  is

$$\hat{\mathbf{w}} = \mathbf{U}^+ \mathbf{y}$$

where  $\mathbf{U}^+$  is the pseudoinverse of the matrix  $\mathbf{U}$ .

- 16.8 (a) For the binary PSK system, a plot of the error signal versus the equalizer output has the form shown by the continuous curve in Fig. 1:

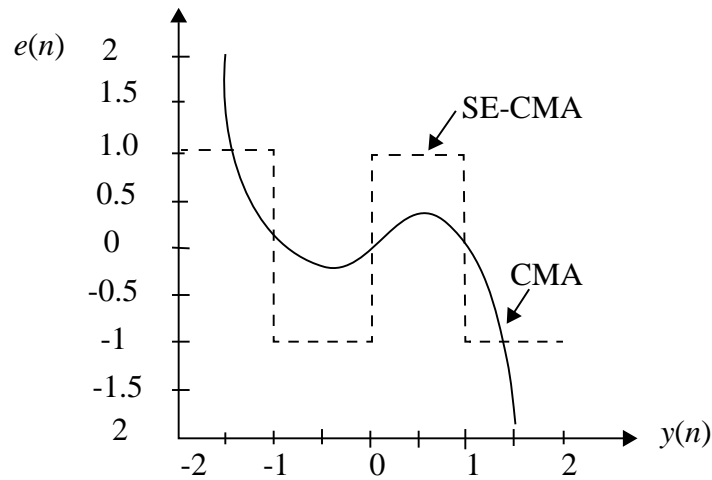


Fig. 1



(b) The corresponding plot for the signed-error (SE) version of the CMA is shown by the dashed line in Fig. 1

(c) CMA is a stochastic algorithm minimizing the Godard criterion

$$J_{cm} = \frac{1}{4}E\left\{[|y_n|^2 - R_2]^2\right\}$$

where the positive constant  $R_2$  is the dispersion constant, which is chosen in accordance with the source statistics. For a fractionally spaced equalizer (FSE) update algorithm, the algorithm is described by

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{u}(n) \underbrace{y^*(n) \gamma^2 - |y(n)|^2}_{\Psi(y_n)}, \quad \gamma^2 = 1 + R_2$$

where  $\mu$  is small positive step size and  $\Psi(\cdot)$  is called the CMA error function.

The signed-error (SE)-CMA algorithm is described as follows:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{u}(n) \text{sgn} \Psi(y(n))$$

where  $\text{sgn}(\cdot)$  denotes the signum function. Specifically,

$$\text{sgn}(x) = \begin{cases} 1 & \text{for } x > 0 \\ -1 & \text{for } x < 0 \end{cases}$$

The SE-CMA is computationally more efficient than CMA.

16.9 The update function for the constant modulus algorithm (CMA) is described by

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

where

$$e(n) = y(n)(\gamma^2 - |y(n)|^2), \quad \gamma^2 = 1 + R_2$$

In quadriphase-shift keying (QPSK), the output signal  $y(n)$  is complex-valued, as shown by

$$y(n) = y_I(n) + jy_Q(n), \quad \begin{array}{l} y_I(n) = \text{in-phase component} \\ y_Q(n) = \text{quadrature component} \end{array}$$

Hence,

$$e_I(n) = y_I(n)(R_2 - y_I^2(n) - y_Q^2(n))$$

$$e_Q(n) = y_Q(n)(R_2 - y_I^2(n) - y_Q^2(n))$$

For the signed CMA, we thus have

$$\begin{aligned} \hat{\mathbf{w}}(n+1) &= \hat{\mathbf{w}}(n) + \mu \mathbf{u}(n) \text{sgn} e(n) \\ &= \hat{\mathbf{w}}(n+1) + \mu \mathbf{u}(n) \text{sgn}[e_I(n) + je_Q(n)] \end{aligned}$$

The weights are complex valued. Hence, following the analysis presented in Section 5.3 of the text, we may write

$$\hat{\mathbf{w}}_I(n+1) = \hat{\mathbf{w}}_I(n) + \mu(\mathbf{u}_I(n) \text{sgn}[e_I(n)] - \mathbf{u}_Q(n) \text{sgn}[e_Q(n)]) \quad (1)$$

$$\hat{\mathbf{w}}_Q(n+1) = \hat{\mathbf{w}}_Q(n) + \mu(\mathbf{u}_Q(n) \text{sgn}[e_I(n)] - \mathbf{u}_I(n) \text{sgn}[e_Q(n)]) \quad (2)$$

where

$$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}_I(n) + j\hat{\mathbf{w}}_Q(n)$$

$$\mathbf{u}(n) = \mathbf{u}_I(n) + j\mathbf{u}_Q(n)$$

The standard version of the complex CMA is as follows:

$$\hat{\mathbf{w}}_I(n+1) = \hat{\mathbf{w}}_I(n) + \mu(\mathbf{u}_I(n)e_I(n) - \mathbf{u}_Q(n)e_Q(n)) \quad (3)$$

$$\hat{\mathbf{w}}_Q(n+1) = \hat{\mathbf{w}}_Q(n) + \mu(\mathbf{u}_Q(n)e_I(n) - \mathbf{u}_I(n)e_Q(n)) \quad (4)$$

Both versions of the CMA, the signed version of Eqs. (1) and (2) and the standard version of Eqs. (3) and (4), can now be treated in the same way as the real-valued CMA in Problem 16.8.

16.10 Dithered signed-error version of CMA, hereafter referred to as DSE-CMA:

- (a) According to quantization theory, the operator  $\alpha \text{sgn}(v(n))$  has an effect equivalent to that of the two-level quantizer:

$$Q(v(n)) = \begin{cases} \Delta/2 & v(n) \geq 0 \\ -\Delta/2 & v(n) < 0 \end{cases}$$

where  $v(n) = e(n) + \alpha \varepsilon(n)$

Furthermore, since the samples of the dither  $\varepsilon(n)$  are i.i.d over the interval  $[-1, 1]$ ,  $\{\alpha \varepsilon(n)\}$  satisfies the requirement for a valid dither process if the constant  $\alpha$  satisfies the requirement:

$$\alpha \geq |e(n)|$$

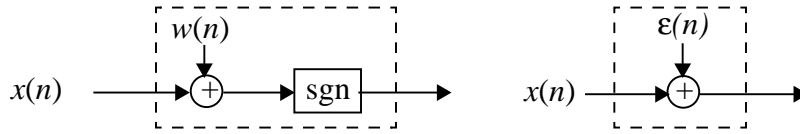


Fig. 1

The equivalent model is illustrated in Fig. 1. Hence, we may rewrite the DSE-CMA update formula:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{u}(n)(e(n) + \varepsilon(n)) \quad (1)$$

Also since  $\varepsilon(n)$  is an uncorrelated random process, its first moment is defined by

$$E[\varepsilon(n)|e(n)] = E[\varepsilon(n)] = 0 \quad (2)$$

Taking the expectation of the DSE-CMA error function, we find that it is a hard-limited version of the  $e(n)$ , as shown by

$$E[v(n)|y(n)]$$

$$= \begin{cases} \alpha, & v(n) > 0 \\ e(n), & |v(n)| \leq \alpha \\ -\alpha, & v(n) < -\alpha \end{cases}$$

which follows from Eqs. (1) and (2).

16.11 (a) The Shalvi-Weinstein equalizer is based on the cost function

$$J(n) = E[|y(n)|^4] \text{ subject to the constraint } E[|y(n)|^2] = \sigma_x^2$$

where  $y(n)$  is the equalizer output and  $\sigma_x^2$  is the variance of the original data sequence applied to the channel input. Applying the method of Lagrange multipliers, we may define a cost function for the Shalvi-Weinstein equalizer that incorporates the constraint as follows:

$$J_{SW}(n) = E[|y(n)|^4] + \lambda(E[|y(n)|^2] - \sigma_x^2) \quad (1)$$

where  $\lambda$  is the Lagrange multiplier.

From Eq. (16.105) of the text, we find that the cost function for the Godard algorithm takes the following form for  $p = 2$ :

$$\begin{aligned} J_G &= E[(|y(n)|^2 - R_2)^2] \\ &= E[|y(n)|^4] - 2R_2 E[|y(n)|^2] + R_2^2 \end{aligned} \quad (2)$$

where  $R_2$  is a positive real constant. Comparing Eqs. (1) and (2), we see that these two cost functions have the same mathematical form. Hence, we may infer that these two equalization algorithms share the same optimization criterion.

- (b) For a more detailed discussion of the equivalence between the Godard and Shalvi-Weinstein algorithms, we may proceed as follows:

First, rewrite the tap-weight vector of the equalizer in polar form (i.e., a unit-norm vector times a radial scale factor), and then optimize the Godard cost function with respect to the radial factor. The “reduced” cost function that results from this transformation is then recognized as a monotonic transformation of the corresponding Shalvi-Weinstein cost function. Since the transformation relating these two criteria is monotonic, their stationary points and local/global minima coincide to within a radial factor. By taking this approach, the equivalence between the Godard and Shalvi-Weinstein equalizers seem to hold under general conditions (linear or nonlinear channels, *iid* or correlated data sequences applied to the channel input, Gaussian or non-Gaussian channel noise, etc).<sup>1</sup>

---

<sup>1</sup>. For further details on the issues raised herein, see P.A. Regalia, “On the equivalence between the Godard and Shalvi-Weinstein schemes of blind equalization”, *Signal Processing*, vol. 73, pp.185-190, 1999.

- 16.12 For the derivation of Eq. (16.116), see the Appendix of the paper by Johnson et al.<sup>2</sup> Note, however, the CMA cost function for binary PSK given in Eq. (63) of that paper is four times that of Eq. (16.116).

---

<sup>2</sup>. C.R. Johnson, et al., "Blind equalization using the constant modulus criterion: A review", Proc. IEEE, vol. 86, pp. 1927-1950, October 1998.

## CHAPTER 17

### 17.1 (a) The complementary error function

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-t^2/2} dt$$

qualifies as a sigmoid function for two reasons:

1. The function is a monotonically increasing function of  $x$ , with

$$\varphi(-\infty) = 0$$

$$\varphi(0) = 0.5$$

$$\varphi(\infty) = 1$$

For  $x = \infty$ ,  $\varphi$  equals the total area under the probability density function of a Gaussian variable with zero mean and unit variable; this area is unity by definition.

2. The function  $\varphi(x)$  is continuously differentiable:

$$\frac{d\varphi}{dx} = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

### (b) The inverse tangent function

$$\varphi(x) = \frac{2}{\pi} \tan^{-1}(x)$$

also qualifies as a sigmoid function for two reasons:

1.  $\varphi(-\infty) = -1$

$$\varphi(0) = 0$$

$$\varphi(\infty) = +1$$

2.  $\varphi(x)$  is continuously differentiable:

$$\frac{d\varphi}{dx} = \frac{2}{\pi} \frac{1}{\sqrt{1+x^2}}$$

The complementary error function and the inverse tangent function differ from each other in the following respects:

- The complementary error function is unipolar (nonsymmetric).
- The inverse tangent function is bipolar (antisymmetric).

17.2 The incorporation of a momentum modifies the update rule for synaptic weight  $w_{kj}$  as follows:

$$\Delta w_{kj}(n) = \alpha \Delta w_{kj}(n-1) - \eta \frac{\partial E(n)}{\partial w_{kj}} \quad (1)$$

where  $\alpha$  = momentum constant  
 $\eta$  = learning-rate parameter  
 $E(n)$  = cost function to be minimized  
 $n$  = iteration number

Equation (1) represents a first-order difference equation in  $\Delta w_{kj}(n)$ . Solving it for  $\Delta w_{kj}(n)$ , we get

$$\Delta w_{kj}(n) = -\eta \sum_{t=0}^n \alpha^{n-t} \frac{\partial E(t)}{\partial w_{kj}} \quad (2)$$

For  $-1 < \alpha \leq 0$ , we may rewrite Eq. (2) as

$$\Delta w_{kj}(n) = -\eta \sum_{t=0}^n (-1)^{n-t} |\alpha|^{n-t} \frac{\partial E(t)}{\partial w_{kj}}$$

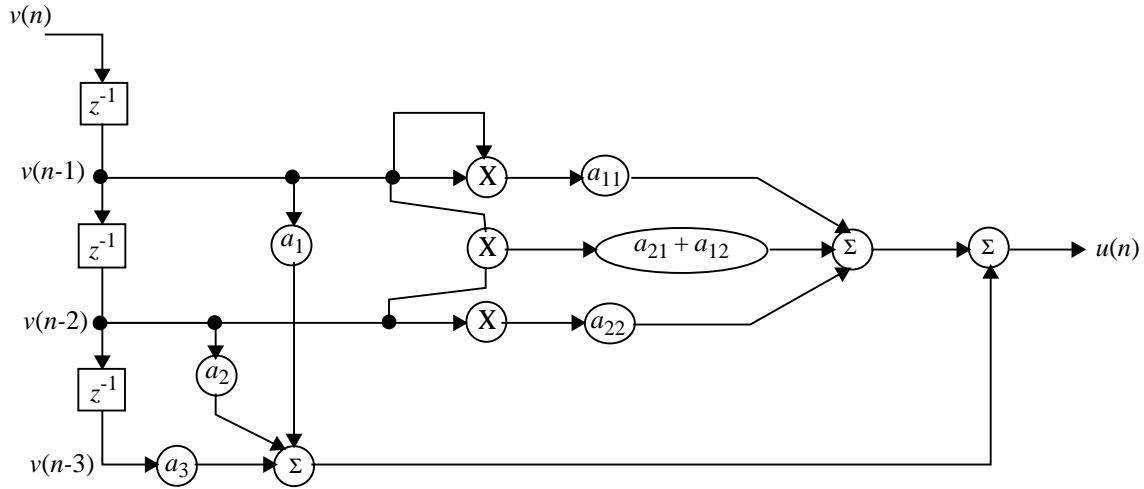
Thus, the use of  $-1 < \alpha \leq 0$  in place of the commonly used value  $0 \leq \alpha < 1$  is merely to introduce the multiplying factor  $(-1)^{n-t}$ , which (for a specified  $n$ ) alternates in algebraic sign as  $t$  increases.

17.3 Consider the real-valued version of the back-propagation algorithm, which is summarized in Table 17.2 of the text. In the backward pass, starting from the output layer, we see that as we progress from the output layer, the error signal decreases the further away we are from the output layer. This would then suggest that the learning rates used to adjust the weights in the multilayer perceptron should be *increased* to make up for the decrease in the error signal as we move away from the output layer. In so doing, the rate at which the learning process takes place in the different layers of the network is equalized, which is highly desirable.

17.4 We are given the time series

$$u(n) = \sum_{i=1}^3 a_i v(n-i) + \sum_{i=1}^2 \sum_{j=1}^2 a_{ij} v(n-1)v(n-j)$$

We may implement it as follows:

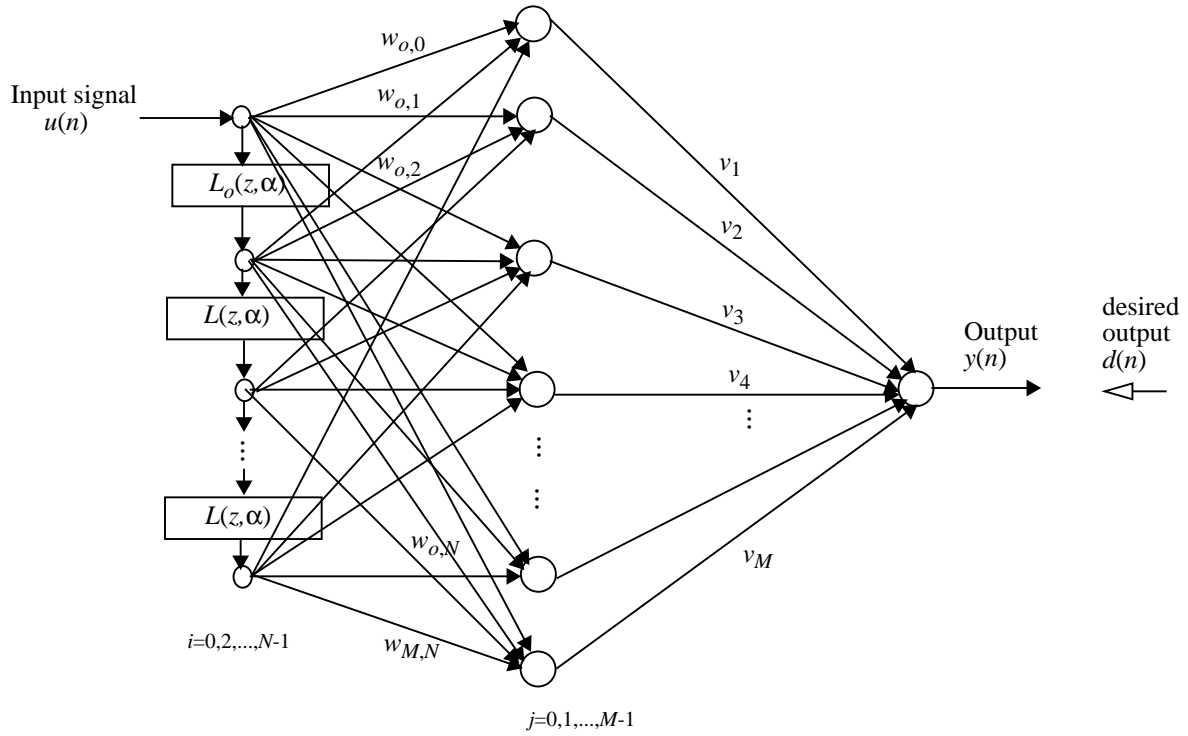


17.5 The minimum description length (MDL) criterion strives to optimize the model order. In particular, it provides the best compromise between its two components: a likelihood function, and a penalty function. Realizing that model order has a direct bearing on model complexity, it may therefore be argued that the MDL criterion tries to match the complexity of a model to the underlying complexity of the input data.

The risk  $R$  of Eq. (17.63) also has two components: one determined by the training data, and the other determined by the network complexity. Loosely speaking, the roles of these two components are comparable to those of the likelihood function and the penalty function in the MDL criterion, respectively. Increasing the first component of the risk  $R$  at the expense of the second component implies that the training data are highly reliable, whereas increasing the second component of  $R$  at the expense of the first component implies that the training data are of poor quality.



17.6 (a) Laguerre-based version of MLP has the following structure:



(b) The new BP algorithm can be devised for Laguerre-based version MLP in a manner similar to the LMS algorithm formulated for Langerre filter (Problem 4 in Chapter 15).

The only difference between the new BP algorithm and the conventional BP algorithm lies in the adjustment of the input-to-hidden layer weights, and the calculation of each hidden unit output.

Recall from the solution to Problem 4 in Chapter 15, we have

For hidden unit output:

$$h_j = \sum_{i=0}^{N-1} w_{i,j} u_i(n, \alpha) = \sum_{i=0}^{N-1} z^{-1} [U_i(z, \alpha)] \quad j = 0, 1, \dots, M-1$$

$$\phi(h_j) = \tanh(h_j), \quad y = \tanh\left(\sum_{j=0}^{M-1} v_j \phi(h_j)\right)$$

For adaptation of weights

1. Output layer (for simplicity, consider only one output unit)

$$\Delta v_j(n) = \tilde{\mu} h_j(n) \varphi' \left( \sum_{j=0}^{M-1} v_j \varphi(h_j) \right) [d - y(n)]$$

$$\mathbf{Bias}(n) = \tilde{\mu} \varphi' \left( \sum_{j=0}^{M-1} v_j \varphi(h_j) \right) [d - y(n)]$$

2. Hidden layer (for simplicity, consider only one hidden layer)

$$\Delta w_{ij}(n) = \tilde{\mu} u_i(n, \alpha) \varphi'(h_j) \sum_j v_j \varphi' \left( \sum_j v_j h_j \right) [d - y(n)]$$

$$\mathbf{Bias}(n) = \tilde{\mu} \varphi'(h_j) \sum_j v_j \varphi \left( \sum_j v_j h_j \right) [d - y(n)]$$