6. LINEAR OPTIMUM FILTER
    Weiner Filter

A class of linear optimum discrete-time filters known as Weiner Filters are discussed in this section. The theory is formulated for the case of real-valued time series with the filter specified in terms of its impulse response. The analysis could be extended for complex-valued signals.

The solution to the optimum filter problem could be obtained via two different approaches that are complementary. The first approach is via the error-performance surface that describes the optimization cost as a function of the filter coefficients. The second method uses an important theorem known as the principle of orthogonality.

Following is the sub-topic outline.

6.1 Problem Statement / Preliminaries
6.2 Minimum Mean Squared Error
6.3 Weiner-Hopf Equations
6.4 Error-Performance Surface
6.5 Principle of Orthogonality
6.6 Examples on Co-ordinate Transformation

Reference: Chapter 5, “Adaptive Filter Theory”
6.1 SOME PRELIMINARIES

Consider a transversal filter with tap inputs \( x(k), x(k - 1), \ldots, x(k - M + 1) \) and a corresponding set of tap weights \( w_0(k), w_1(k), \ldots, w_{M-1}(k) \). The tap inputs represent samples drawn from a wide-sense stationary stochastic process of zero mean and correlation matrix \( R \). In addition to these inputs, the filter is supplied with a desired response \( d(k) \) that provides a frame of reference for the optimum filtering action. Figure 6.1 depicts the filtering action described herein.

The vector of tap inputs at time \( k \) is denoted by \( x(k) \), and the corresponding estimate of the desired response at the filter output is denoted by \( y(k) \). By comparing this estimate with the desired response \( d(k) \), we produce an estimation error denoted by \( \epsilon(k) \).

Figure 6.1  Structure of adaptive transversal (FIR) filter.
THE PERFORMANCE FUNCTION

\[ \varepsilon_k = d_k - w^T x_k = d_k - x_k^T w \]  \hspace{1cm} (6.1)

Here we have used the subscript \( k \) to denote the time index; the subscript has been dropped from the weight vector \( w \) for convenience, because in this discussion we do not wish to adjust the weights. We now square (6.1) to obtain the instantaneous squared error,

\[ \varepsilon_k^2 = d_k^2 - 2 d_k x_k^T w + w^T x_k x_k^T w \]  \hspace{1cm} (6.2)

We assume that \( \varepsilon_k, \ d_k \), and \( x_k \) are statistically stationary and take the expected value of (6.2) over \( k \) : i.e.,

\[ E[\varepsilon_k^2] = E[d_k^2] - 2 E[d_k x_k^T w] + w^T E[x_k x_k^T] w \]  \hspace{1cm} (6.3)

Note that the expected value of any sum is the sum of expected values, but that the expected value of a product is the product of expected values when the variables are statistically independent. The signals \( x_k \) and \( d_k \) are not generally independent.

The mean-square-error function can be more conveniently expressed as follows. Let \( R \) be defined as the square matrix

\[ R = E[x_k x_k^T] \]
This matrix is designated the "input correlation matrix." The main diagonal terms are the mean squares of the input components, and the cross terms are the cross correlations among the input components. Let \( p \) be similarly defined as the column vector

\[
P = E[d_k x_k^T] = E[d_k x_k, d_k x_{k-1}, \ldots, d_k x_{k-M+1}]^T \quad (6.5)
\]

This vector is the set of cross correlations between the desired response and the input components. The elements of both \( R \) and \( p \) are all constant second-order statistics when \( x_k \) and \( d_k \) are stationary.

We now let the mean-square error in (6.3) be designated by \( \xi \) and reexpress it in terms of (6.4) and (6.5) as

\[
MSE = \xi = E[\varepsilon_k^2] = E[d_k^2] - 2 p^T w + w^T R w \quad (6.6)
\]

It is clear from this expression that the mean-square error \( \xi \) is precisely a quadratic function of the components of the weight vector \( w \) when the input components and desired response input are stationary stochastic variables. That is, when (6.6) is expanded, the elements of \( w \) will appear in first and second degree only.
Figure 6.2  Portion of a two-dimensional quadratic performance surface. The mean-square error is plotted vertically, $w_0$ ranges from -3 to 4, $w_1$, ranges from -4 to 0, and the optimum weight vector is $w^* = (0.65, -2.10)$. The minimum $MSE$ is 0.0 in this example.

A portion of a typical two-dimensional mean-square-error function is illustrated in Figure 6.2. The vertical axis represents the mean-square error and the horizontal axes the values of the two weights. The bowl-shaped quadratic error function, or performance surface, formed in this manner is a paraboloid (a hyperparaboloid if there are more than two weights). It must be concave upward; otherwise, there would be weight settings that would result in a negative mean-square error, an impossible result.
with real, physical signals. Contours of constant mean-square error are elliptical, as can be seen by setting $\xi$ constant in (6.6). The point at the "bottom of the bowl" is projected onto the weight-vector plane as $w^*$, the optimal weight vector or point of minimum mean-square error. With a quadratic performance function there is only a single global optimum; no local minima exist.

6.2 GRADIENT AND MINIMUM MEAN-SQUARE ERROR

Many useful adaptive processes that cause the weight vector to seek the minimum of the performance surface do so by gradient methods. The gradient of the mean-square error performance surface, designated $\nabla(\xi)$ or simply $\nabla$, can be obtained by differentiating (6.6) to obtain the column vector

$$\nabla = \frac{\partial \xi}{\partial w} = \begin{bmatrix} \frac{\partial \xi}{\partial w_0} & \frac{\partial \xi}{\partial w_1} & \cdots & \frac{\partial \xi}{\partial w_{M-1}} \end{bmatrix}^T = 2Rw - 2p \quad (6.7)$$

where $R$ and $p$ are given by (6.4) and (6.5), respectively. This expression is obtained by expanding (6.6) and differentiating with respect to each component of the weight vector. Differentiation of the term $w^T R w$ can be treated as differentiation of the product $(w^T)(Rw)$.

To obtain the minimum mean-square error the weight vector $w$ is set at its optimal value $w^*$, where the gradient is zero:

$$\nabla = 0 = 2Rw^* - 2p \quad (6.8)$$
6.3 Weiner-Hopf Equation

Assuming that \( R \) is nonsingular, the optimal weight vector \( w^* \), sometimes called the Wiener weight vector, is found from (6.8) to be

\[
w^* = R^{-1}p
\]

This equation is an expression of the Wiener-Hopf equation in matrix form. The minimum mean-square error is now obtained by substituting \( w^* \) from (6.9) for \( w \) in (6.6):

\[
\xi_{\text{min}} = E[d_k^2] + w^*^T R w^* - 2p^T w^* \\
= E[d_k^2] + p^T R^{-1} R [R^{-1} p] - 2p^T [R^{-1} p]
\]

(6.10)

We now simplify this result using three rules that are of general utility in discussions of the performance surface:

1. Identity rule for any square matrix: \( A A^{-1} = I \)
2. Transpose of a matrix product: \( [A B]^T = B^T A^T \)
3. Symmetry of the input correlation matrix: \( R^T = R; [R^{-1}]^T = R^{-1} \).

Using these rules, (6.10) becomes

\[
\xi_{\text{min}} = E[d_k^2] - p^T R^{-1} p = E[d_k^2] - p^T w^*
\]

(6.11)

An example to help clarify the concepts of quadratic performance surface, gradient, and mean-square error set forth so far, will be shown later.
6.4 Canonical Form of the Error-Performance Surface

Equation (6.6) defines the expanded form of the mean-squared error $\xi$ produced by the transversal filter in Fig. 6.1. We may rewrite this equation in matrix form,

$$\xi = J(w) = E[d_k^2] - w^T p - p^T w + w^T \mathbf{R} w \quad (6.12)$$

where the mean-squared error is written as $J(w)$ to emphasize its dependence on the tapweight vector $w$. As mentioned in Chapter 3 the correlation matrix $\mathbf{R}$ is almost always positive definite, so that the inverse matrix $\mathbf{R}^{-1}$ exists. Accordingly, expressing $J(w)$ as a "perfect square" in $w$, we may rewrite Eq. (6.12) in the form

$$J(w) = E[d_k^2] - p^T \mathbf{R}^{-1} p + (w - \mathbf{R}^{-1} p)^T \mathbf{R} (w - \mathbf{R}^{-1} p) \quad (6.13)$$

From Eq. (6.13), we now immediately see that

$$\min_w J(w) = E[d_k^2] - p^T \mathbf{R}^{-1} p \quad \text{for} \quad w^* = \mathbf{R}^{-1} p$$

In effect, starting from Eq. (6.12), we have rederived the Wiener filter in a rather simple way. Moreover, we may use the defining equations for the Wiener filter to explicitly show the unique optimality of the minimizing tap-weight vector $w$, by writing

$$J(w) = J_{\text{min}} + (w - w^*)^T \mathbf{R} (w - w^*) \quad (6.14)$$

This equation shows explicitly the unique optimality of the minimizing tap-weight vector $w^*$. 

96
Although the quadratic form on the right-hand side of Eq. (6.14) is quite informative, nevertheless, it is desirable to change the basis on which it is defined so that the representation of the error-performance surface is considerably simplified. To do this, we recall from Chapter 5 that the correlation matrix \( R \) of the tap-input vector may be expressed in terms of eigenvalues and eigenvectors as follows:

\[
R = Q \Lambda Q^T \quad (6.15)
\]

where \( \Lambda \) is a diagonal matrix consisting of the eigenvalues \( \lambda_1, \lambda_2, \ldots, \lambda_M \) of the correlation matrix, and the matrix \( Q \) has for its columns the eigenvectors \( q_1, q_2, \ldots, q_M \) associated with these eigenvalues, respectively. Hence, substituting Eq. (6.15) into (6.14), we get

\[
J(w) = J_{\text{min}} + (w - w^*)^T Q \Lambda Q^T (w - w^*) \quad (6.16)
\]

Define a transformed version of the difference between the tap-weight vector \( w \) and the optimum solution \( w^* \) as

\[
v' = Q^T (w - w^*) = Q^T v \quad (6.17)
\]

Then we may put the quadratic form of Eq. (6.54) into its canonical form defined by

\[
J(w) = J_{\text{min}} + v'^T \Lambda v' \quad (6.18)
\]

This new formulation of the mean-squared error contains no cross-product terms, as shown by
\[ J(w) = J_{\text{min}} + \sum_{k=1}^{M} \lambda_k v'_k v''_k^T = J_{\text{min}} + \sum_{k=1}^{M} \lambda_k |v'_k|^2 \tag{6.19} \]

where \( v'_k \) is the \( k^{th} \) component of the vector \( v' \). The feature that makes the canonical form of Eq. (6.19) a rather useful representation of the error-performance surface is the fact that the components of the transformed coefficient vector \( v' \) constitute the \textit{principal} axes of the error-performance surface. The practical significance of this result will become apparent in later chapters.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure6.3.png}
\caption{Ellipses of constant mean-square error in the \( w_0 \ w_1 \) - plane, with translated axes \( v_0 \) and \( v_1 \), and principal axes \( v'_0 \) and \( v'_1 \). The ellipses are contours of the surface in Figure 6.2.}
\end{figure}
NUMERICAL EXAMPLE : Inverse Channel Modeling

To illustrate the filtering theory developed above, we consider the example depicted in Fig. 6.1. The desired response $d(n)$ is modeled as an AR process of order 1; that is, it may be produced by applying a white-noise process $v_1(n)$ of zero mean and variance or $\sigma_1^2 = 0.27$ to the input of an all-pole filter of order 1, whose transfer function equals [see Fig. 6.5(a)]

$$H_1(z) = \frac{1}{1 + 0.8458^{-1}}$$

The process $d(n)$ is applied to a communication channel modeled by the all-pole transfer function

$$H_2(z) = \frac{1}{1 - 0.9458^{-1}}$$

The channel output $x(n)$ is corrupted by an additive white-noise process $v_2(n)$ of zero mean and variance $\sigma_2^2 = 0.1$, so a sample of the received signal $u(n)$ equals [see Fig. 6.5(b)]

$$u(n) = x(n) + v_2(n)$$

The white-noise processes $v_1(n)$ and $v_2(n)$ are uncorrelated. It is also assumed that $d(n)$ and $u(n)$, and therefore $v_1(n)$ and $v_2(n)$, are all real valued.

The requirement is to specify a Wiener filter consisting of a transversal filter with two taps, which operates on the received signal $u(n)$ so as to produce an estimate of the desired response that is optimum in the mean-square sense.
Figure 6.4  Adaptive modeling of a multipath channel.

Figure 6.5  (a) Autoregressive model of desired response $d(n)$; (b) model of noisy communication channel.
Statistical Characterization of the Desired Response \( d(n) \) and the Received Signal \( u(n) \)

We begin the analysis by considering the difference equations that characterize the various processes described by the models of Fig. 6.5. First, the generation of the desired response \( d(n) \) is governed by the first-order difference equation

\[
d(n) + a_1 d(n - 1) = v_1(n)
\]

(6.20)

where \( a = 0.8458 \). The variance of the process \( d(n) \) equals (see Chapter 3)

\[
\sigma_d^2 = \frac{\sigma_1^2}{1 - a_1^2} = \frac{0.27}{1 - (0.8458)^2} = 0.9486
\]

(6.21)

The process \( d(n) \) acts as input to the channel. Hence, from Fig. 6.5(b), we find that the channel output \( x(n) \) is related to the channel input \( d(n) \) by the first-order difference equation

\[
x(n) + b_1 x(n - 1) = d(n)
\]

(6.22)

where \( b_1 = -0.9458 \). We also observe from the two parts of Fig. 6.5 that the channel output \( x(n) \) may be generated by applying the white-noise process \( v_1(n) \) to a second-order all pole filter whose transfer function equals

\[
H_1(z)H_2(z) = \frac{1}{(1 + 0.8458z^{-1})(1 - 0.9458z^{-1})}
\]

(6.23)
Accordingly, \( x(n) \) is a second-order AR process described by the difference equation

\[
x(n) + a_1 x(n - 1) + a_2 x(n - 2) = v(n)
\]  

(6.24)

where \( a_1 = -0.1 \) and \( a_2 = 0.8 \). Note that both AR processes \( d(n) \) and \( x(n) \) are wide sense stationary.

To characterize the Wiener filter, we need to solve the Wiener-Hopf equations (6.9). This set of equations requires knowledge of two quantities: (i) the correlation matrix \( R \) pertaining to the received signal \( u(n) \), and (ii) the cross-correlation vector \( p \) between \( u(n) \) and the desired response \( d(n) \). In our example, \( R \) is a 2-by-2 matrix and \( p \) is a 2-by-1 vector, since the transversal filter used to implement the Wiener filter is assumed to have two taps.

The received signal \( u(n) \) consists of the channel output \( x(n) \) plus the additive white noise \( v_2(n) \). Since the process \( x(n) \) and \( v_2(n) \) are uncorrelated, it follows that the correlation matrix \( R \) equals the correlation matrix of \( x(n) \) plus the correlation matrix of \( v_2(n) \). That is,

\[
R = R_x + R_2
\]  

(6.25)

For the correlation matrix \( R_x \) we write [since the process \( x(n) \) is real valued]

\[
R_x = \begin{bmatrix}
r_x(0) & r_x(1) \\
r_x(1) & r_x(0)
\end{bmatrix}
\]

where \( r_x(0) \) and \( r_x(1) \) are the autocorrelation functions of the received signal \( x(n) \) for lags of 0 and 1, respectively. From Chapter 3 we have
\[ r_x(0) = \sigma_x^2 = \left( \frac{1+a_2}{1-a_2} \right) \frac{\sigma_1^2}{(1+a_2)^2-a_1^2} = \left( \frac{1-0.8}{1+0.8} \right) \frac{0.27}{(1-0.8)^2-(0.1)^2} = 1 \]

\[ r_x(1) = \left( \frac{-a_1}{1+a_2} \right) = \frac{0.1}{[1-0.8]} = 0.5 \]

Hence,

\[ R_x = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \quad (6.26) \]

Next we observe that since \( v_2(n) \) is a white-noise process of zero mean and variance \( \sigma_2^2 = 0.1 \), the 2-by-2 correlation matrix \( R_2 \) of this process equals

\[ R_2 = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \quad (6.27) \]

Thus, substituting Eqs. (6.26) and (6.27) in Eq. (6.25), we find that the 2-by-2 correlation matrix of the received signal \( x(n) \) equals

\[ R = \begin{bmatrix} 1.1 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \quad (6.28) \]

For the 2-by-1 cross-correlation vector \( p \), we write

\[ p = \begin{bmatrix} p(0) \\ p(-1) \end{bmatrix} \]

where \( p(0) \) and \( p(-1) \) are the cross-correlation functions between \( d(n) \) and \( u(n) \) for lags of 0 and -1, respectively. Since these two processes are real valued, we have
\[ p(k) = p(-k) = E[u(n-k)d(n)], \quad k = 0,1 \] (6.29)

Substituting Eq. (6.22) into Eq. (6.29), and recognizing that the channel output \( x(n) \) is uncorrelated with the white-noise process \( v_2(n) \), we get

\[ p(k) = r_x(k) + b_1 r_x(k - 1), \quad k = 0, 1 \]

Putting \( b_1 = -0.9458 \) and using the element values for the correlation matrix \( R_x \) given in Eq. (6.26), we obtain

\[
\begin{align*}
p(0) &= r_x(0) + b_1 r_x(-1) \\
&= 1 - 0.9458 \times 0.5 = 0.5272 \\
p(0) &= r_x(1) + b_1 r_x(0) \\
&= 0.5 - 0.9458 \times 1 = -0.4458
\end{align*}
\]

Hence,

\[ p = \begin{bmatrix} 0.5272 \\ -0.4458 \end{bmatrix} \] (6.30)

**Error-Performance Surface**

The dependence of the mean-squared error on the 2-by-1 tap-weight vector \( w \) is defined by Eq. (6.12). Hence, substituting Eqs. (6.21), (6.28), and (6.30) into (6.12), we get

\[
J(w_0, w_1) = 0.9486 - 2[0.5272, -0.4458] \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} + \\
\begin{bmatrix} 1.1 & 0.5 \\ 0.5 & 1.1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}
\]
\[ J(w_0, w_1) = 0.9486 - 1.0544w_0 + 0.8916w_1 + w_0w_1 + 1.1(w_0^2 + w_1^2) \]

Using a three-dimensional computer plot, the mean-squared error \( J(w_0, w_1) \) is plotted versus the tap weights \( w_0 \) and \( w_1 \). The result is shown in Fig. 6.6.

Figure 6.7 shows contour plots of the tap weight \( w_1 \) versus \( w_0 \) for varying values of the mean-squared error \( J \). We see that the locus of \( w_1 \) versus \( w_0 \) for a fixed \( J \) is in the form of an ellipse. The elliptical locus shrinks in size as the mean-squared error \( J \) approaches the minimum value \( J_{min} \). For \( J = J_{min} \), the locus reduces to a point with coordinates \( w_0^* \) and \( w_1^* \).

![Error-performance surface of the two-tap transversal filter described in the numerical example.](image)

**Wiener Filter**

The 2-by-1 optimum tap-weight vector \( w^* \), of the Wiener filter is defined in Equation (6.9) as \( R^{-1}p \). In particular, it consists of the
inverse matrix $R^{-1}$ multiplied by the cross-correlation vector $p$. Inverting the correlation matrix $R$ of Eq. (6.28), we get

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

(6.31)

$$= \begin{bmatrix} 1.1456 & -0.5208 \\ -0.5208 & 1.1456 \end{bmatrix}$$

Figure 6.7 Contour plots of the error-performance surface depicted in Fig. 6.6.

Hence, substituting Eqs. (6.30) and (6.31) into Eq. (6.9), we get the desired result:

$$w^* = \begin{bmatrix} 1.1456 & -0.5208 \\ -0.5208 & 1.1456 \end{bmatrix} \begin{bmatrix} 0.5272 \\ -0.4458 \end{bmatrix} = \begin{bmatrix} 0.8360 \\ -0.7853 \end{bmatrix}$$

(6.32)
Minimum Mean-Squared Error

To evaluate the minimum value of the mean-squared error, $J_{\text{min}}$, which results from the use of the optimum tap-weight vector $\mathbf{w}_0$, we use Eq. (5.49). Hence, substituting Eqs. (6.21), (6.30), and (6.32) into Eq. (6.11), we get

$$J_{\text{min}} = 0.9486 - \begin{bmatrix} 0.5272 & -0.4458 \\ -0.7853 & 0.8360 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix}$$

(6.33)

The point represented jointly by the optimum tap-weight vector $\mathbf{w}^*$ of Eq. (6.32) and the minimum mean-squared error of Eq. (6.33) defines the bottom of the error-performance surface in Fig. 6.6, or the center of the contour plots in Fig. 6.7.

Canonical Error-Performance Surface

The characteristic equation of the 2-by-2 correlation matrix $\mathbf{R}$ of Eq. (6.28) is

$$(1.1 - \lambda)^2 - (0.5)^2 = 0$$

The two eigenvalues of the correlation matrix $\mathbf{R}$ are therefore

$$\lambda_1 = 1.6 \quad ; \quad \lambda_2 = 0.6$$

The canonical error-performance surface is therefore defined by [see Eq. (6.19)]

$$J(v_1', v_2') = J_{\text{min}} + 1.6v_1'^2 + 0.6v_2'^2$$

(6.34)

The locus of $v_2'$ versus $v_1'$, as defined in Eq. (6.34), traces an ellipse for a fixed value of $J - J_{\text{min}}$. In particular, the ellipse has a minor axis of $[(J - J_{\text{min}})/\lambda_1]^{1/2}$ along the $v_1'$-coordinate and a major axis of $[(J - J_{\text{min}})/\lambda_2]^{1/2}$ along the $v_2'$-coordinate; this assumes that $\lambda_1 > \lambda_2$, which is how they are related.
6.5 PRINCIPLE OF ORTHOGONALITY

Consider again the statistical filtering problem described in Fig. 6.8. The filter input is denoted by the time series \( u(0), u(1), u(2), \ldots \), and the impulse response of the filter is denoted by \( w_0, w_1, w_2, \ldots \). The filter output \( y(n) \) at discrete time \( n \) is defined by the linear convolution sum:

\[
y(n) = \sum_{k=0}^{\infty} w_k u(n-k), \quad n = 0, 1, 2, \ldots \ldots \quad (6.35)
\]

where the term \( w_k u(n-k) \) represents the inner product of the filter coefficient \( w_k \) and the filter input \( u(n-k) \).
The purpose of the filter in Fig. 6.8 is to produce an estimate of the desired response \( d(n) \). We assume that the filter input and the desired response are single realizations of *jointly wide-sense stationary stochastic processes*, both with zero mean. Accordingly, the estimation of \( d(n) \) is accompanied by an error, defined by the difference

\[
e(n) = d(n) - y(n)
\]  

(6.36)

The estimation error \( e(n) \) is the sample value of a random variable. To *optimize the filter design*, we choose to minimize the mean-square value of the estimation error \( e(n) \). We may thus define the cost function as the *mean-squared error*

\[
J = E[|e(n)|^2]
\]  

(6.37)

where \( E \) denotes the *statistical expectation operator*. The problem is therefore to determine the operating conditions for which \( J \) attains its minimum value.

For the cost function \( J \) to attain its minimum value, all the elements of the gradient vector \( \nabla J \) must be simultaneously equal to zero, as shown by

\[
\nabla_k J = 0, \quad k = 0, 1, 2, \ldots \quad (6.38)
\]

Under this set of conditions, the filter is said to be *optimum in the mean–squared error sense*. According to Eq. (6.37), the cost function \( J \) is a scalar independent of time \( n \). Hence, substituting the first line of Eq. (6.37), we get
\[ \nabla_k J = 2E \left[ \frac{\partial e(n)}{\partial w_k} e(n) \right] \]  

(6.39)

Using Eq. (6.36) we get the following partial derivatives:

\[ \frac{\partial e(n)}{\partial w_k} = -u(n - k) \]

Thus, substituting these partial derivatives in Eq. (6.39), we get the result

\[ \nabla_k J = -2E[u(n - k)e(n)] \]  

(6.40)

We are now ready to specify the operating conditions required for minimizing the cost function \( J \). Let \( e_o \) denote the special value of the estimation error that results when the filter operates in its optimum condition. We then find that the conditions specified in Eq. (6.38) are indeed equivalent to

\[ E[u(n - k) e_o(n)] = 0, \quad k = 0, 1, 2.... \]  

(6.41)

In words, Eq. (6.41) states the following:

The necessary and sufficient condition for the cost function \( J \) to attain its minimum value is that the corresponding value of the estimation error \( e_o(n) \) is orthogonal to each input sample that enters into the estimation of the desired response at time \( n \).

Indeed, this statement constitutes the **principle of orthogonality**; it represents one of the most elegant theorems in the subject of linear optimum filtering. It also provides the mathematical basis of a procedure for testing that the linear filter is operating in its optimum condition.
Corollary to the Principle of Orthogonality

There is a corollary to the principle of orthogonality that we may derive by examining the correlation between the filter output $y(n)$ and the estimation error $e(n)$. Using Eq. (6.35), we may express this correlation as follows:

$$E[y(n)e(n)] = E\left[\sum_{k=0}^{\infty} w_k u(n-k)e(n)\right] = \sum_{k=0}^{\infty} w_k E[u(n-k)e(n)](6.42)$$

Let $y_0$ denote the output produced by the filter optimized in the mean-squared-error sense, with $e_0(n)$ denoting the corresponding estimation error. Hence, using the principle of orthogonality described by Eq. (6.41), we get the desired result:

$$E[y_0(n)e_0(n)] = 0 \quad (6.43)$$

We may thus state the corollary to the principle of orthogonality as follows:

When the filter operates in its optimum condition, the estimate of the desired response defined by the filter output, $y_0(n)$, and the corresponding estimation error, $e_0(n)$, are orthogonal to each other.

WIENER-HOPF EQUATIONS

The principle of orthogonality, described in Eq. (6.41), specifies the necessary and sufficient condition for the optimum operation of the filter. We may reformulate the necessary and sufficient condition for optimality by substituting Eqs. (6.35) and (6.36) in (6.41). In particular, we may write
\[
E\left[ u(n-k)\left( d(n) - \sum_{i=0}^{\infty} w_{0i}u(n-i) \right) \right] = 0 \quad k = 0, 1, 2, \ldots
\]

where \( w_{0i} \) is the \( i^{th} \) coefficient in the impulse response of the optimum filter. Expanding this equation and rearranging terms, we get

\[
\sum_{i=0}^{\infty} w_{0i} E[u(n-k)u(n-i)] = E[u(n-k)d(n)], \quad k = 0,1,2,\ldots \quad (6.44)
\]

The two expectations in Eq. (6.44) may be interpreted as follows:

1. The expectation \( E[u(n-k)u(n-i)] \) is equal to the autocorrelation function of the filter input for a lag of \( i-k \). We may thus express this expectation as

\[
r(i-k) = E[u(n-k)u(n-i)] \quad (6.45)
\]

2. The expectation \( E[u(n-k)d(n)] \) is equal to the cross-correlation between the filter input \( u(n-k) \) and the desired response \( d(n) \) for a lag of \( -k \). We may thus express this second expectation as

\[
p(-k) = E[u(n-k)d(n)] \quad (6.46)
\]

Accordingly, using the definitions of Eqs. (6.45) and (6.46) in (6.44), we get an infinitely large system of equations as the necessary and sufficient condition for the optimality of the filter:

\[
\sum_{i=0}^{\infty} w_{0i} r(i-k) = p(-k), \quad k = 0, 1, 2,\ldots \quad (6.47)
\]
The system of equations (6.47) defines the optimum filter coefficients, in the most general setting, in terms of two correlation functions: the autocorrelation function of the filter input, and the cross-correlation between the filter input and the desired response. These equations are called the *Wiener-Hopf equations*.

**Solution of the Wiener-Hopf Equations for Linear Transversal Filters**

The solution of the set of Wiener-Hopf equations is greatly simplified for the special case when a *linear transversal filter*, or FIR filter, is used to perform the estimation of desired response $d(n)$. Consider then the structure shown in Fig. 6.8. The transversal filter involves a combination of three basic operations: *storage, multiplication, and addition*, as described here:

1. The storage is represented by a cascade of $M - 1$ one-sample delays, with the block for each such unit labeled as $z^{-1}$. We refer to the various points at which the one-sample delays are accessed as *tap points*. The tap inputs are denoted by $u(n)$, $u(n - 1)$, \ldots, $u(n - M + 1)$. Thus, with $u(n)$ viewed as the current value of the filter input, the remaining $M - 1$ tap inputs, $u(n - 1)$, \ldots, $u(n - M + 1)$, represent *past values of the input*.

2. The scalar *inner products* of tap inputs $u(n)$, $u(n - 1)$, \ldots, $u(n - M + 1)$ and *tap weights* $w_0$, $w_1$, \ldots, $w_{M-1}$, respectively, are formed by using a corresponding set of multipliers. In particular, the multiplication involved in forming the scalar inner product of $u(n)$ and $w_0$ is represented by a block labeled $w_0$, and so on for the other inner products.
3. The function of the adders is to sum the multiplier outputs to produce an overall output for the filter.

The impulse response of the transversal filter in Fig. 6.8 is defined by the finite set of tap weights $w_0, w_1, \ldots, w_{M-1}$. Accordingly, the Wiener-Hopf equations reduce to a system of $M$ simultaneous equations, as shown by

$$\sum_{i=0}^{\infty} w_{o,i} r(i - k) = p(-k), \quad k = 0, 1, 2, \ldots, M-1 \quad (6.48)$$

where $w_{o,0}, w_{o,1}, \ldots, w_{o,M-1}$ are the optimum values of the tap weights of the filter.

**Matrix Formulation of the Wiener-Hopf Equations**

Let $R$ denote the $M$-by-$M$ correlation matrix of the tap inputs $u(n), u(n-1), \ldots, u(n - M + 1)$ in the transversal filter of Fig. 6.8:

$$R = E[u(n)u^T(n)] \quad (6.49)$$

where $u(n)$ is the $M$-by-1 tap-input vector:

$$u(n) = [u(n), u(n - 1), \ldots, u(n - M + 1)]^T \quad (6.50)$$

In expanded form, we have
Correspondingly, let $p$ denote the $M$-by-$1$ cross-correlation vector between the tap inputs of the filter and the desired response $d(n)$:

In expanded form, we have

$$p = E[u(n)d(n)]$$

(6.52)

$$p = [p(0), p(-1), \ldots, p(1-M)]^T$$

(6.53)

Note that the lags used in the definition of $p$ are either zero or else negative. We may thus rewrite the Wiener-Hopf equations (6.48) in the compact matrix form:

$$Rw^* = p$$

(6.54)

**EXAMPLE 2**

A simple example of a single-input adaptive linear combiner with two weights is shown in Figure 6.9. The input and desired signals are sampled sinusoids at the same frequency, with $N$ samples per cycle. We assume that $N > 2$ so that the input samples are not all zero. We are not concerned here with the origin of these signals, only with the resulting performance surface and its properties.
To obtain the performance function [i.e., $\xi$ in (6.6)], we need the expected signal products in (6.4) and (6.5). Note that we must change the subscripts of $x$ for the single-input case. The expected products may be found for any product of sinusoidal functions by averaging over one or more periods of the product. Thus,

$$E[x_k x_{k-n}] = \frac{1}{N} \sum_{n=1}^{N} \sin \left( \frac{2\pi k}{N} \right) \sin \left( \frac{2\pi (k-n)}{N} \right) = 0.5 \cos \left( \frac{2\pi n}{N} \right) \quad (6.55)$$

$$E[d_k x_{k-n}] = \frac{2}{N} \sum_{n=1}^{N} \cos \left( \frac{2\pi k}{N} \right) \sin \left( \frac{2\pi (k-n)}{N} \right) = -\sin \left( \frac{2\pi n}{N} \right) \quad (6.56)$$

We note further that obviously $E[x_{k-1}^2] = E[x_k^2]$, because the average is over $k$.

With these results, the input correlation matrix $R$ and the correlation vector $p$ can be obtained from (6.4) and (6.5) for this two-dimensional, single-input example:
\[ R = E\left[ \begin{bmatrix} x_k^2 & x_k x_{k-1} \\ x_{k-1} x_k & x_{k-1}^2 \end{bmatrix} \right] = \begin{bmatrix} 0.5 & 0.5\cos\left(\frac{2\pi}{N}\right) \\ 0.5\cos\left(\frac{2\pi}{N}\right) & 0.5 \end{bmatrix} \] (6.57)

\[ p = E[d_k x_k \quad d_k x_{k-1}]^T = \begin{bmatrix} 0 & -\sin\left(\frac{2\pi}{N}\right) \end{bmatrix} \] (6.58)

As in (6.55) and (6.56) we also obtain \( E[d_k^2] = 2 \). Using these results in (6.6), we obtain the performance function for this example:

\[ \xi = E[d_k^2] + w^T R w - 2p^T w = \\
2 + 0.5\begin{bmatrix} w_0 & w_1 \end{bmatrix} \begin{bmatrix} 1 & \cos\left(\frac{2\pi}{N}\right) \\ \cos\left(\frac{2\pi}{N}\right) & 1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} - 2 \begin{bmatrix} 0 & -\sin\left(\frac{2\pi}{N}\right) \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \\
= 0.5(w_0^2 + w_1^2) + w_0 w_1 \cos\left(\frac{2\pi}{N}\right) + 2w_1 \sin\left(\frac{2\pi}{N}\right) + 2 \] (6.59)

This performance surface was plotted in Figure 6.2 for \( N=5 \) samples per cycle. Note that it is quadratic in \( w_0 \) and \( w_1 \), and has a single global minimum. The gradient vector at any point \( w_0, w_1 \), can be found by substituting (6.57) and (6.58) into (6.7), and is

\[ \nabla = 2R w - 2p = \begin{bmatrix} 1 & \cos\left(\frac{2\pi}{N}\right) \\ \cos\left(\frac{2\pi}{N}\right) & 1 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} - 2 \begin{bmatrix} 0 & -\sin\left(\frac{2\pi}{N}\right) \end{bmatrix} \]
The Wiener weight vector for this example, \( w^* \), may be found formally from (6.9) by inverting \( R \), or it may be found by setting \( \nabla \) equal to zero in (6.60). These are, of course, equivalent operations, and in either case the result is

\[
\begin{bmatrix}
w_0 + w_1 \cos\left(\frac{2\pi}{N}\right) \\
w_0 \cos\left(\frac{2\pi}{N}\right) + w_1 + 2 \sin\left(\frac{2\pi}{N}\right)
\end{bmatrix}
\]

(6.60)

Finally, the minimum mean-square error for this example is obtained by substituting (6.58) and (6.61) into (6.11):

\[
\xi_{\text{min}} = E[d_k^2] - p^T w^* = 2 - \begin{bmatrix} 0 & -\sin\left(\frac{2\pi}{N}\right) \\
2\cot\left(\frac{2\pi}{N}\right) & -2 \csc\left(\frac{2\pi}{N}\right) \end{bmatrix} = 0
\]

(6.62)

This result, which says in effect that the weights in Figure 6.9 can be adjusted to reduce \( \varepsilon_k \) to zero for any value of \( N \), may seem surprising at first. The unit delay by itself can change \( x_k \) from a sine into a cosine function only when \( N = 4 \), that is, only when the unit delay is one-fourth of a cycle. Note that in this case (6.61) gives \( w_0^* = 0 \) and \( w_1^* = -2 \). However, with two weights in addition to the delay, the adaptive linear combiner can always shift \( x_k \) so that it becomes the proper cosine function, for any \( N \) greater than 2.
6.7 Coordinate Transformation

How to transform an Ellipse to a Circle:

(i) Coordinate System is \( W \):

\[
\mathbf{f} = E(d^2) - 2P^TW + W^TRW
\]

\( w_0 \) and \( w_1 \) are coupled in this system.

(ii) Coordinate System is \( V = W - W^* \):

\[
\mathbf{f} = \mathbf{f}_{\text{min}} + V^TRV
\]

\( v_0 \) and \( v_1 \) are still coupled.
(iii) **Coordinate Rotation to $V'$**:

$$V_0' = (\cos \theta) V_0 + (\sin \theta) V_1,$$

$$V_1' = (-\sin \theta) V_0 + (\cos \theta) V_1,$$

$$\begin{pmatrix} V_0' \\ V_1' \end{pmatrix} = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} V_0 \\ V_1 \end{pmatrix} \Rightarrow V' = QV$$

- Choose direction $\theta$ along the eigenvalues

(iv) **Scale $V'$ axis by $\lambda$**:

$$\lambda_0 = \sqrt{\lambda_0} V_0' \quad ; \quad \lambda_1 = \sqrt{\lambda_1} V_1' \Rightarrow \chi = \sqrt{\lambda} V'$$

$$\ell = \ell_{\min} + \chi_0^2 + \chi_1^2 \Rightarrow \text{a circle!}$$