2.3 Linear Estimation

2.3.1 Signal Estimation Problem

- The problem of estimating one signal from another is one of the most important in signal processing.
- In many applications , the desired signal is not available or observable directly. Instead, the desired signal is a degraded or distorted version of the original signal.
- The signal estimation problem is to recover , in the best way possible , the desired signal from its degraded replica.

- **Examples :**

  (1) The desired signal may be corrupted by strong additive noise , such as weak measured evoked brain potentials against the strong background of ongoing EEGs

  (2) A signal transmitted over a communications channel can suffer phase and amplitude distortions and can be subject to additive channel noise ; the problem is to recover the transmitted signal from the distorted received signal.

2.3.2 Optimum Estimation of Signals

- The signal estimation problem can be stated as follows:
  We are to estimate a random signal $x(n)$ on the basis of available observations of related signal $y(n)$.
- The available signal $y(n)$ is to be processed by an optimal processor that produces the best estimate of $x(n)$.
  The resulting estimate $x(n)$ will be a functional of the observations $y(n)$.
- If the optimal processor is linear , such as a linear filter , then the estimate $x(n)$ will be a linear function of the observations.

- **Several major criteria for designing such optimal processors will be discussed :**

  (1) Maximum a posteriori (MAP) criterion

  (2) Maximum likelihood (ML) criterion

  (3) Minimum mean-squared error (MMSE) criterion

  (4) Linear minimum mean squared error (LMMSE) criterion

  (5) Least square (LS) criterion

- Let us assume that the desired signal $x(n)$ is to be estimated over a finite time interval $n_a \leqq n \leqq n_b$

  Also we may assume that the observed signal y_n is also available over the same interval.

  Define the vector

  $$\mathbf{x} = (x(n_a)\ x(n_{a+1})\ \ldots\ x(n_b))^T \qquad (2.30)$$

  $$y = (y(n_a)\ \ y(n_{a+1})\ \ldots\ \ y(n_b))^T \qquad (2.31)$$

  For each value of n , we seek the functional dependence

  $$x(n) = x(n \mid y)$$

  of $x(n)$ on the given observation vector y to provide best estimate of the nth sample $x(n)$.

2.3.2.1 MAP Estimation

- The criterion for the MAP estimate is to maximize the *a posteriori* conditional density of $x(n)$ given that y already occurred ; namely

$$p(x(n) \mid y) = \text{maximum} \qquad (2.32)$$

In other words, the optimal estimate $x(n)$ is that $x(n)$ which maximizes this quantity for the given vector y .

2.1.2.2 ML Estimation

- The ML criterion , on the other hand, selects $x(n)$ to maximize the conditional density of y given $x(n)$ *;* that is

$$p(y \mid x(n)) = \text{maximum} \qquad (2.33)$$

This criterion selects $x(n)$ as though the already collected observations y were the most likely ones to occur .

2.3.2.3 MMSE Estimation

The MMSE criterion minimizes the mean-squared estimation error

$$E[e^2(n)] = \text{minimum}, \quad (2.34)$$

where $e(n) = x(n) - x(n \mid y)$ (2.35)

and $x(n \mid y) = E[x(n) \mid y]$

$$= \text{mean-square estimate} \quad (2.36)$$

2.3.2.4 LMMSE Estimate

- The LMMSE criterion requires the estimate to be a linear function of the observations

$$x(n) = \Sigma\, h(n; i)\; y(i) \qquad (2.37)$$

For each n , the weights $h(n ;i)$ are selected so as to minimize the mean-squared estimation error

$$E[e^2(n)] = E\,[\,(x(n) - x(n)\,)^2] = \text{minimum} \qquad (2.38)$$

Note :

With the exception of the LMMSE estimate , all the other estimates $x(n \mid y)$ are , in general , nonlinear in y

2.3.3 Linear MMSE Estimation

- Two common problems of determining the optimal weights $h(n,i)$ according to the mean-squared minimization criterion are
  (1) Optimal filtering problem
  (2) Optimal prediction problem
- In these cases , the optimal estimate of $x(n)$ at a given time instant n is given by an expression of the form

$$x(n) = \Sigma\, h(n;i)\, y(i) \qquad (2.39)$$

as a linear combination of the available observations $y(n)$ in the interval $n_a \leqq n \leqq n_b$.

2.3.3.1 Optimal Filtering

- The optimal filtering problem requires the linear operation

$$x(n) = \Sigma\, h(n; i)\, y(i) \qquad (2.40)$$

to be causal ; that is, only those observations that are in the present and past of the current samples $x(n)$ must be used in making up the estimate $x(n)$.

This requires $h(n; i) = 0$, for $n < i$ (2.41)

and then $x(n) = \Sigma\, h(n; i)\, y(i)$ (2.42)

- The estimate $x(n)$ depends on the present and all the past observations, from the fixed starting point n_a to the current time instant n . As n increases , more and more observations are taking into account in making up the estimate $x(n)$.

- **To make the optimum filter computationally efficient and manageable, only the current and the past M observations $y(i)$; $i = 1, 2, \ldots, n-M$ are taking into account. That is,**

$$x(n) = \sum h(n;i)\, y_i$$

$$= \sum h(n;i)\, y(i) \qquad (2.43)$$

This is referred to as the finite impulse response (FIR) Wiener filter.

2.3.3.2 Linear Prediction

- The linear prediction problem is a special case of the optimal filtering problem with the additional stipulation that observations only up to time instant *n-D* must be used in obtaining the current estimate $x(n)$; this equivalent to the problem of predicting *D* units of time into the future.
- If we demand that the prediction be based only on the past M samples (from the current sample) , we obtain the FIR version of the prediction problem, which can be depicted below :

$$x(n) = \sum h(n;i)\, y(i) \qquad (2.44)$$

Summary :

- In LMMSE problem , the estimate is expressed by

$$x(n) = \Sigma\, h(n;i)\, y(i) \qquad (2.45)$$

for given observations $y(n)$.

- The problem is a filtering problem when $n = n_b$,
  it is a prediction problem when $n > n_b$,
- We are to set up the general orthogonality and normal equations for the optimal weights of optimal filtering.

2.4 FIR Wiener Filter

- Assuming that the signals are stationary , FIR Wiener filters are relatively simple to implement, inherently stable and more practical .
- The FIR filter is represented by the tap-weights w_k,

  $k = 0,1,2, \ldots,M$
- Denote that $W = (w_0, w_1, \ldots, w_M)^T$

The estimate can be expressed as

$$x(n) = \Sigma\, y(n-k)\, w_k \qquad (2.46)$$

Then, the estimation error in FIR Wiener filter is given by

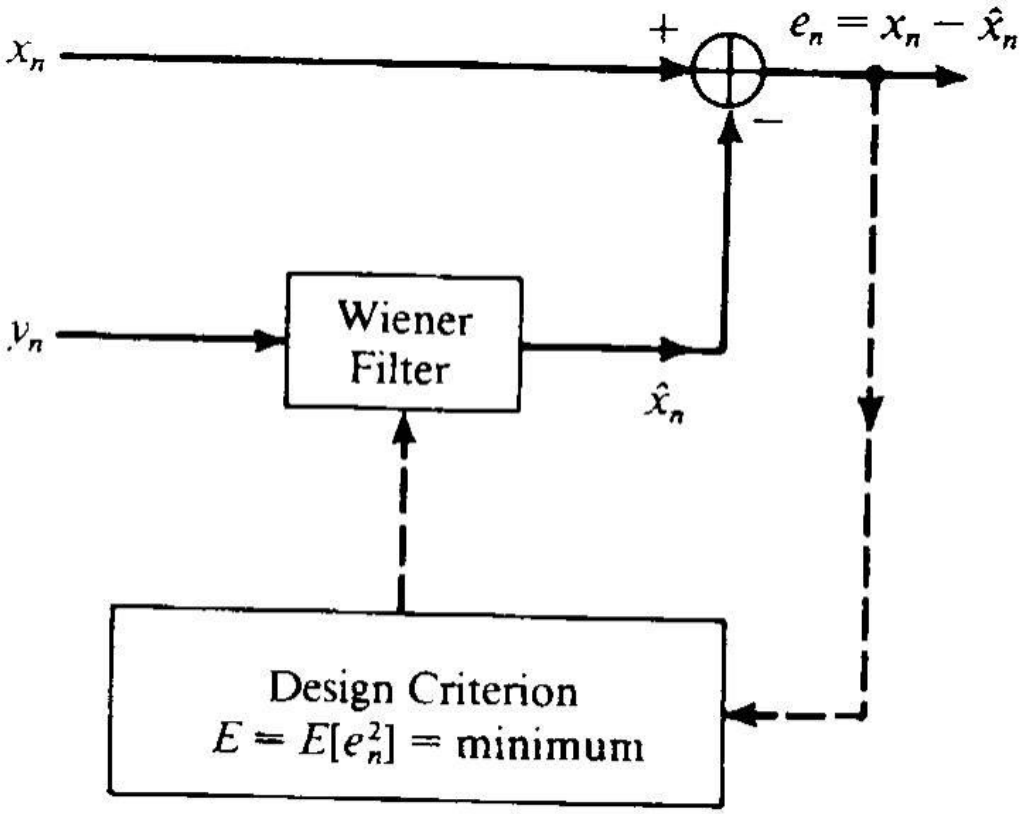
$$e(n) = x(n) - x(n) = x(n) - \Sigma\, y(n-k)\, w_k \qquad (2.47)$$

Differentiating the mean-squared estimation error with respect to each weight and setting the derivative to zero , we obtain the orthogonality equations that are enough to determine the weights :

$$\delta E[e^2(n)] / \delta w_i = 2 E[e(n)(\delta e(n) / \delta w_i)]$$
$$= -2 E[e(n) y(n-i)] = 0$$

or $R_{ey}(n) = E[e(n) y(n-i)] = 0$ for $0 \leqq i \leqq M$

(2.49)


x_n
+
$e_n = x_n - \hat{x}_n$
−
y_n
Wiener Filter
$\hat{x}_n$
Design Criterion
$E = E[e_n^2]$ = minimum

Thus , the estimation error is orthogonal (uncorrelated) to each observation y_i used in making up the estimate x_n . The orthogonality equations provided exactly as many equations as there are unknown weights.

- Inserting (2.46) for e_n , the orthogonality equations may be written in an equivalent form , known as **normal equations**

$$E\left[\left\{x(n) - \Sigma\, w_k y(n-k)\right\}\, y(n-i)\right] = 0 \qquad (2.49)$$

or

$$E[x(n)\, y(n-i)] = \Sigma\, w_k\, E[y(n-k)\, y(n-i)]$$

$$\text{for } 0 \leqq i \leqq M \qquad (2.50)$$

These **normal equations** determine the optimal weights at the current time instant n .

We can write Eq.(2.50) in vector notation as

$$\mathbf{p} = E[\,x(n)\mathbf{y}\,]$$

and $x(n) = \mathbf{W}^T \mathbf{y}$

where $\mathbf{W} = (w_0, w_1, \ldots, w_M)^T$ is the optimum weight- vector ,

$\mathbf{y} = (\, y(n) \;\; y(n-1) \ldots \; y(n-M)\,)^T$

is the vector of observations up to the current time n ,

- The optimal *weights* $\mathbf{W}^o$ and the estimate are then given by

$$\mathbf{W}^o = E\,[x(n)\mathbf{y}\,]\,E[\mathbf{y}\mathbf{y}^T]^{-1}$$

$$= \mathbf{R}_Y^{-1}\,\mathbf{p} \qquad (2.51)$$

This is identical to the correlation canceller discussed before.

Note that $\mathbf{p} = (\, p(0) \;\; p(1) \ldots p(M)\,)^T$

where $p(j) = E[x(n)\, y(n-j)\,]$

The *M+1* optimal filter weights $w_0, w_1, .., w_M$ are obtained by the *(M+1)* X *(M+1)* matrix inversion of the Wiener-Hopf equations (also known as **normal equation**):

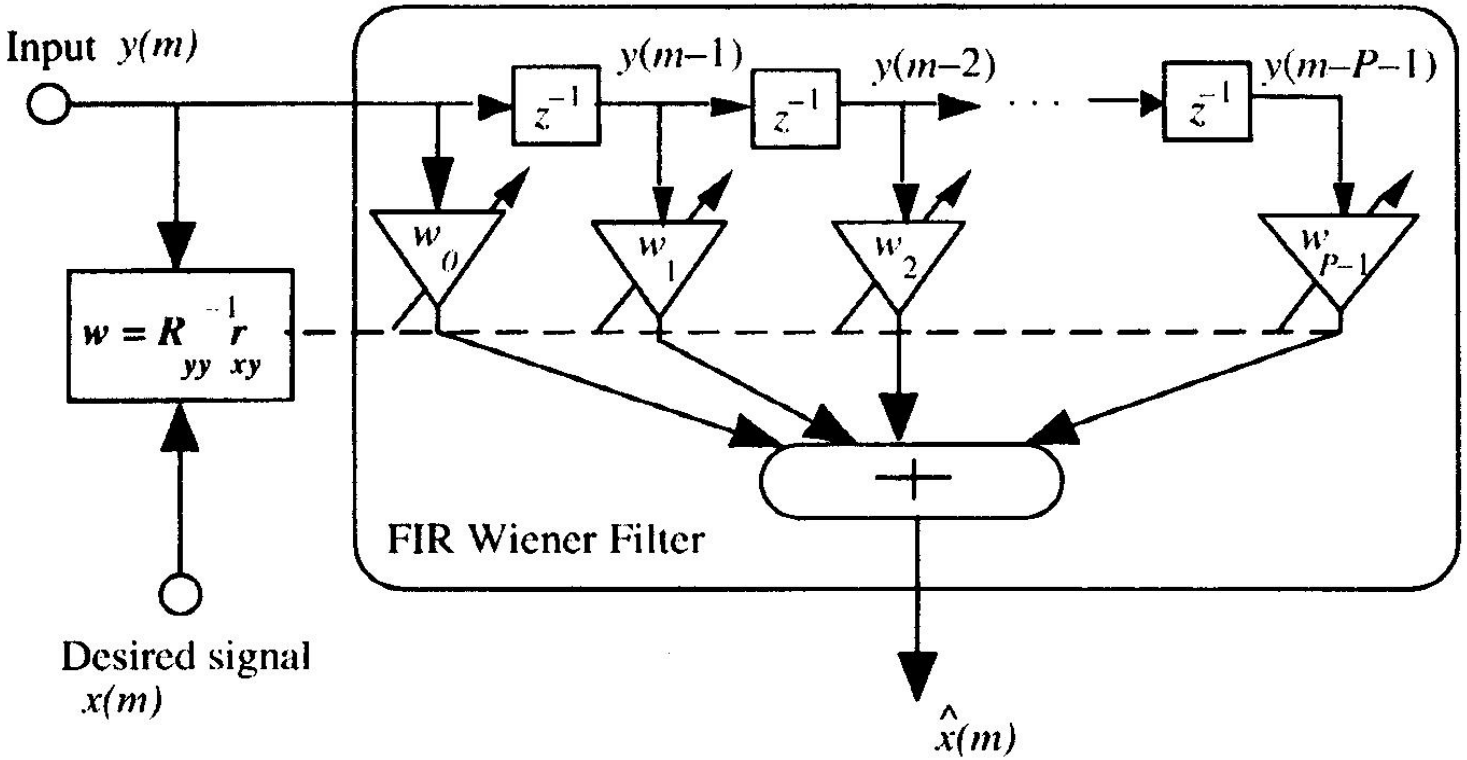


Illustration of a Wiener filter structure.

$$
\begin{matrix}
R_{yy}(0) & R_{yy}(1) & R_{yy}(2) & \dots & R_{yy}(M) \\
R_{yy}(1) & R_{yy}(0) & R_{yy}(1) & \dots & R_{yy}(M-1) \\
R_{yy}(2) & R_{yy}(1) & R_{yy}(0) & \dots & R_{yy}(M-2) \\
\\
R_{yy}(M) & R_{yy}(M-1) & R_{yy}(M-2) & \dots & R_{yy}(0)
\end{matrix}
\begin{matrix} w_0 \\ w_1 \\ w_2 \\ \\ w_M \end{matrix}
$$

$$
= \begin{matrix} p(0) \\ p(1) \\ p(2) \\ \\ p(M) \end{matrix}
$$

(2.52)

- **The minimized estimation error at time instant *n* is computed by**

$$J(n) = E[e^2(n)] = E[e(n)\,e(n)] = E[e(n)\,x(n)]$$

$$= E\left[\left\{x(n) - \Sigma\, y(n-k)\, w_k\right\} x(n)\right]$$

$$= E[x^2(n)] - E[\Sigma\, y(n-k)\, w_k\, x(n)]$$

$$= E[x^2(n)] - E[x(n)\, y^T]\, E[yy^T]^{-1}\, E[y\, x(n)] \qquad (2.53)$$

Exploiting the Toeplitz property of the matrix R_{yy} , the above matrix equation (2.53) can be solved efficiently using Levinson's algorithm.

Example :

(1) For M=2

Equ.(2.55) becomes

$$\begin{bmatrix} R_{yy}(0) & R_{yy}(1) & R_{yy}(2) \\ R_{yy}(1) & R_{yy}(0) & R_{yy}(1) \\ R_{yy}(2) & R_{yy}(1) & R_{yy}(0) \end{bmatrix} \begin{bmatrix} w(0) \\ w(1) \\ w(2) \end{bmatrix} = \begin{bmatrix} r_{xy}(0) \\ r_{xy}(1) \\ r_{xy}(2) \end{bmatrix}$$

If observation $y(n) = x(n) + v(n)$

where $R_{xx}(k) = 2\,(0.8)^{|k|}$

$R_{vv}(k) = 2\,\delta(k)$

Thus $p(k) = r_{xy}(k) = E[\,x(n)\,\{x(n-k) + v(n-k)\}\,] = R_{xx}(k)$

$= 2\,(0.8)^k$

$$R_{yy}(k) = E[\,\{x(n) + v(n)\}\,\{x(n-l) + v(n-l)\}\,]$$

$$= R_{xx}(k) + R_{vv}(k) = 2\,(0.8)^k + 2\,\delta(k)$$

- **The normal equation becomes**

4.00	**1.60**	**1.28**	**w(0)**	**2.00**
1.60	**4.00**	**1.60**	**w(1)**	**1.60**
1.28	**1.60**	**4.00**	**w(2)**	**1.28**

Solving the above equations, we obtain

w(0) = 0.3824

w(1) = 0.2000

w(2) = 0.1176

Appendix :

Wiener filter for complex –valued signals

- $W = (w_0^*, w_1^*, \ldots, w_M^*)^T = (w_0, w_1, \ldots, w_M)^H$

where H denotes the complex-conjugate or Hermitian .

$p = E[x^*(n)\, y]$

$R_Y = E[yy^H]$

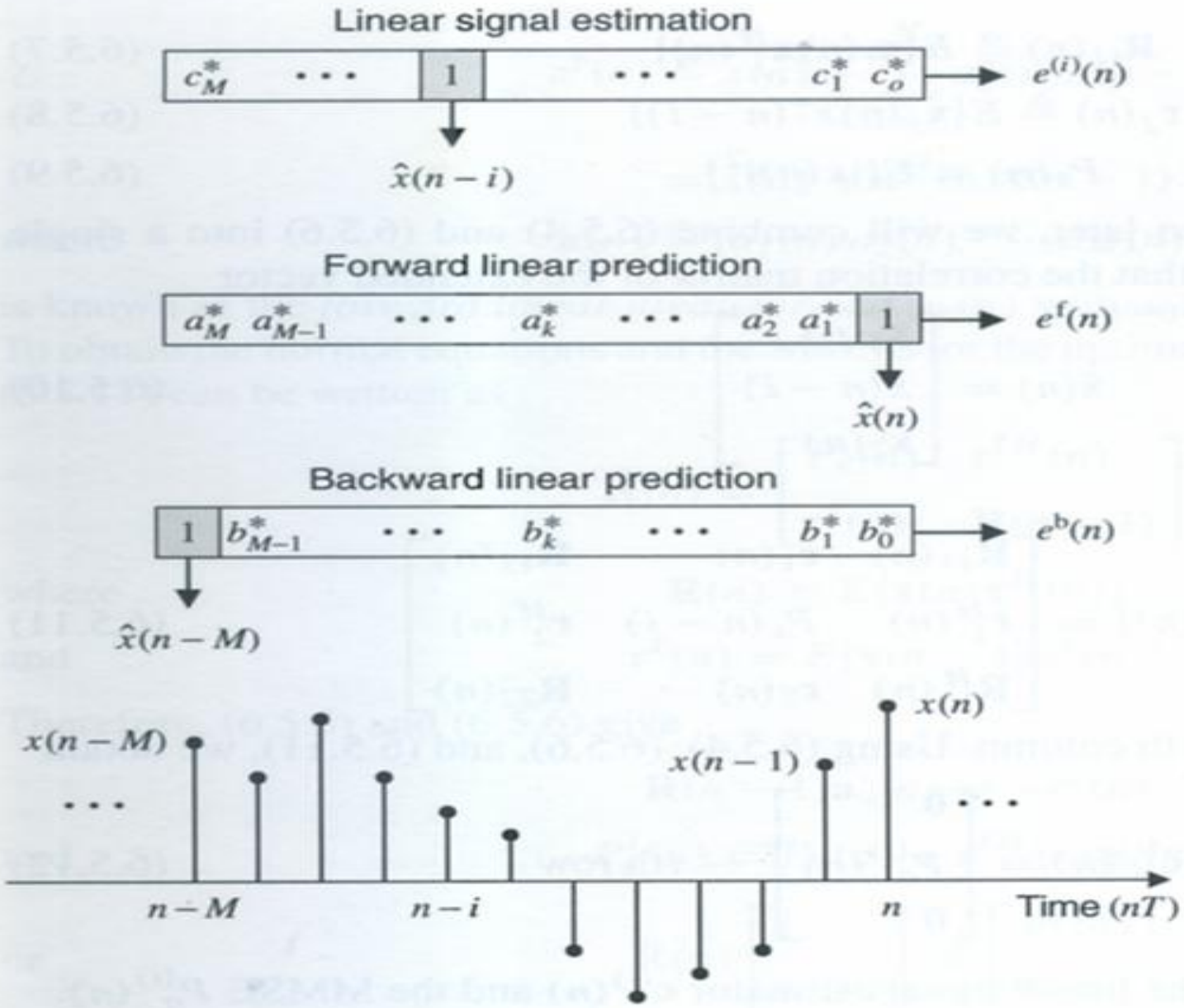
The optimal *weights* W^o and the estimate are then given by

$W^o = E[yy^H]^{-1} E[x^*(n)\, y]$

$= R_Y^{-1} p$

2.5 Linear Prediction

- The linear prediction problem is a special case of the optimal filtering problem with the additional stipulation that observations only up to time instant *n-M* must be used in obtaining the current estimate *x*(*n*) *to* predict *one* units of time into the future. This is a one-step forward prediction.
- By using similar concept of prediction, we may define a backward predictor, that predicts a sample y(n-M) from future samples , y(n-M+1), …, y(n) .


Linear signal estimation
c_M^* ... 1 ... c_1^* c_o^*
$e^{(i)}(n)$
$\hat{x}(n-i)$
Forward linear prediction
a_M^* a_{M-1}^* ... a_k^* ... a_2^* a_1^* 1
$e^{f}(n)$
$\hat{x}(n)$
Backward linear prediction
1 b_{M-1}^* ... b_k^* ... b_1^* b_0^*
$e^{b}(n)$
$\hat{x}(n-M)$
$x(n-M)$
$x(n-1)$
$x(n)$
$n-M$
$n-i$
n
Time (nT)

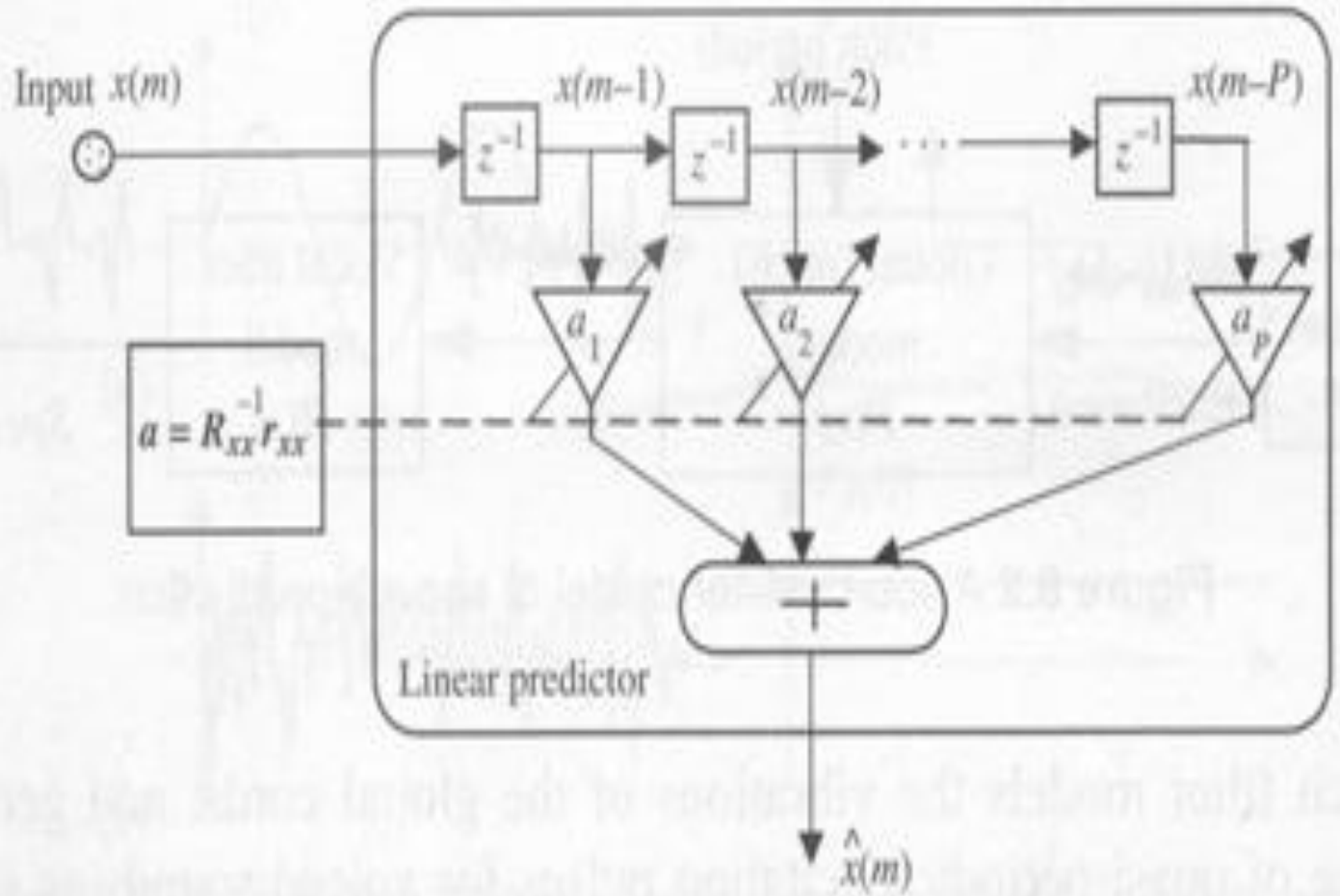


Figure 8.3 Block-diagram illustration of a linear predictor.

2.5.1 Forward Prediction

- Assuming that the signals are stationary , the prediction process can be implemented by FIR filters which are relatively simple to realized, inherently stable and more practical .
- The prediction filter is represented by the tap-weights a_k, $k = 1,2, \ldots,M$
- Denote that $a = (a_1, \ldots, a_M)$

The prediction can be expressed as

$$y(n) = \Sigma\, a_k\, y(n-k) \qquad \text{(A-1)}$$

Then, the estimation error in FIR prediction is given by

$$e(n) = y(n) - y(n) = y(n) - \Sigma\, a_k\, y(n-k) \qquad \text{(A-2)}$$

Let Y_{n-1} denote the M-dimensional linear space spanned by y(n-1) , y(n-2) ,…, y(n-M), and use $y\,(n \mid Y_{n-1})$ to denote the predicted value of y (n) given this set of samples.
Then, $e_f\,(n) = y(n) - y(n \mid Y_{n-1})$

- Differentiating the mean-squared estimation error with respect to each weight and setting the derivative to zero , we obtain the orthogonality equations that are enough to determine the weights :

$$\delta E[e^2(n)\,] / \delta a_i = 2\,E\,[e(n)\,(\delta e(n) / \delta a_i\,)\,] = -2\,E\,[e(n)\,y(n-i)\,] = 0$$

for $0 \leqq i, n \leqq M$ (A-3)

- Inserting (a-3) for e_n , the orthogonality equations may be written in an equivalent form , known as normal equations

$$E\left[\left\{ y(n) - \Sigma\, a_k y(n-k) \right\} y(n-i) \right] = 0$$

or $$E[y(n)\, y(n-i)] = \Sigma\, a_k\, E[y(n-k)\, y(n-i)]$$

for $0 \leqq n \leqq M$ (A-4)

These normal equations determine the optimal weights at the current time instant n .

We can write Eq.(A-4) in vector notation as

$$p = E[\, y(n)\, Y_{n-1}\,]$$

and $y(n) = a^T Y_{n-1}$

where $a = (a_1, \ldots, a_M)^T$ is the optimum weight- vector ,

$Y_{n-1} = (y(n-1) \ldots y(n-M))^T$

is the vector of observations up to time $n-1$,

- The optimal *weights* a^o and the prediction are then given by

$$a^o = E[x(n)y]\, E[yy^T]^{-1}$$

$$= R_{Y\, n-1}^{-1}\, p \qquad \text{(A-5)} .$$

Note that $p = (p(1) \ldots p(M))^T$

where $p(j) = E[y(n)\, y(n-j)]$

$$
\begin{matrix}
R_{yy}(0) & R_{yy}(1) & R_{yy}(2) & \dots & R_{yy}(M-1) & a_1 \\
R_{yy}(1) & R_{yy}(0) & R_{yy}(1) & \dots & R_{yy}(M-2) & a_2 \\
R_{yy}(2) & R_{yy}(1) & R_{yy}(0) & \dots & R_{yy}(M-3) & a_3 \\
\\
R_{yy}(M-1) & R_{yy}(M-2) & R_{yy}(M-3) & \dots & R_{yy}(0) & a_M
\end{matrix}
=
\begin{matrix}
p(1) \\
p(2) \\
p(3) \\
\\
p(M)
\end{matrix}
$$

(A-6)

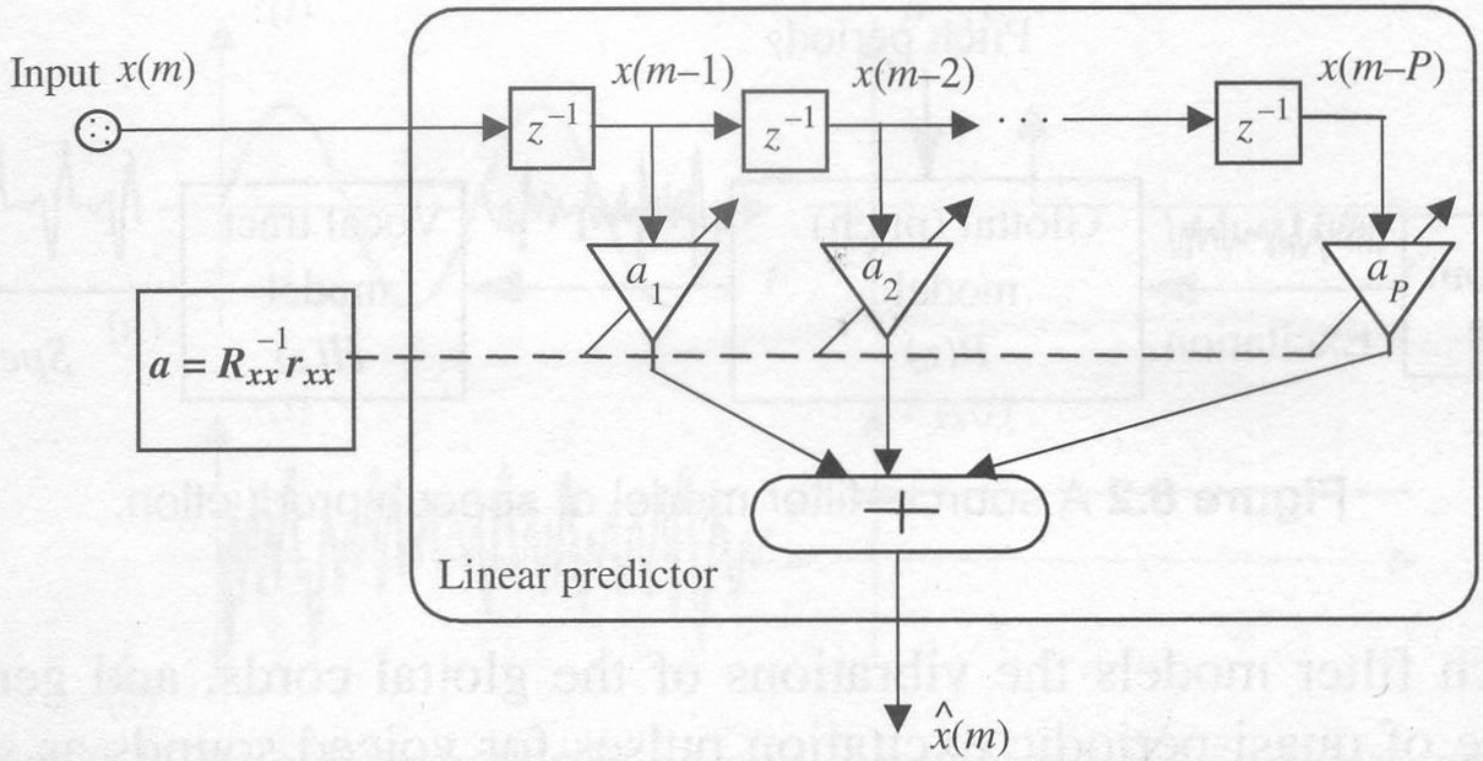


Figure 8.3 Block-diagram illustration of a linear predictor.

2.5.2 Backward Prediction

- A backward prediction filter predicts a signal sample y(n-M) from M future samples.

The predicted value can be expressed as

$$y(n-M) = \Sigma\, b_k\, y(n-k+1)$$

The backward prediction error

$$e_b(n) = y(n-M) - \Sigma\, b_k\, y(n-k+1)$$

$$= y(n-M) - y(n-P) \mid Y_{n-1})$$

- **The optimal coefficients can be obtained by he normal equation**

$$\mathbf{b} = \boldsymbol{R}_{\mathrm{Y}\,n-1}^{-1}\, \boldsymbol{p}^{B}$$

where $\mathbf{p}^{B} = (\, \mathrm{p}(M)\, \mathrm{p}(M-1) \ldots \mathrm{p}(2)\ \mathrm{p}(1)\,)^{T}$

$\mathbf{b} = (\, b_1\ b_2 \qquad b_{M-1}\ b_M\ )$

$$
\begin{matrix}
R_{yy}(0) & R_{yy}(1) & R_{yy}(2) & \dots & R_{yy}(M-1) & b_1 \\
R_{yy}(1) & R_{yy}(0) & R_{yy}(1) & \dots & R_{yy}(M-2) & b_2 \\
R_{yy}(2) & R_{yy}(1) & R_{yy}(0) & \dots & R_{yy}(M-3) & b_3 \\
\\
R_{yy}(M-1) & R_{yy}(M-2) & R_{yy}(M-3) & \dots & R_{yy}(0) & b_M
\end{matrix}
=
\begin{matrix}
p(M) \\
p(M-1) \\
p(M-2) \\
\\
p(1)
\end{matrix}
$$

(A-6)

2.6 Least Squares Optimal Filtering

(C.W. Therrien, pp.519-524)

- In the minimum mean-squared estimation , the sequences (or vectors) y and x were regarded as random processes with known (or previously estimated) second-order (moment) statistics.
- Here , we are to consider the optimal filtering problem from a slightly different approach , the least square (LS) method.

2.6.1 LS Method

Historical Notes :

The principle of least squares was introduced by the German mathematician Carl F. Gauss , who used it to determine the orbit of the asteroid Ceres in 1821 by formulating the estimation problem as an optimization problem.

Carl Gauss (1777 – 1855) was born in Braunschweig , Germany.

- There is no presumed knowledge of the statistical properties of random vectors x and y beforehand . It is assumed that a typical data sequence of both x and y has been measured and recorded and that these sequences can be used to design the filter .
- We are to estimate $x = [\, x(0), x(1), \ldots, x(M-1) \,]^T$

  given the observation $y = [\, y(0), y(1), \ldots, y(M-1) \,]^T$

- If a causal FIR filter of length M is used , then the estimate for the given data sequence is

$$x(n) = \Sigma\, h(i)\, y(n-i) \qquad (2.67)$$

and the estimation error can be defined as

$$\varepsilon(n) = x(n) - x(n) \qquad (2.68)$$

The approach here is to design the filter to minimize the sum of squared errors

$$S = \Sigma \mid \varepsilon(n) \mid^2 \qquad (2.69)$$

where n_1 and n_f are some initial and final values of n that define the interval over which to perform the minimization.

Note that no probabilistic statements have been made in defining this problem.

- The criterion (2.69) is called a **least squares criterion**.
- In matrix form, we can express (2.67) as

$$\mathbf{x} = \mathbf{Y}\,\mathbf{h} \qquad (2.70)$$

where $\mathbf{x} = [\, x(n_l)\ \ x(n_l+1) \ldots x(n_f)\,]^T$

$$\mathbf{Y} = \begin{matrix} y(n_l) & y(n_l-1) \ldots & y(n_l-M+1) \\ y(n_l+1) & y(n_l) & \ldots\ y(n_l-M+2) \\ & & \\ y(n_f) & y(n_f-1) \ldots & y(n_f-M+1) \end{matrix}$$

and $\mathbf{h} = [\, h(0)\ \ h(1)\ \ \ldots\ \ h(M-1)\,]^T$

- The matrix Y is called the data matrix and has dimension $K \times M$ where $K = n_f - n_l + 1$.

  It will be assumed that $K >> M$.

- Define the error vector as

$$\varepsilon = x - \hat{x} \qquad (2.71)$$

  x and ε are K-dimensional vectors.

  Then the problem is to minimize

$$S = \| \varepsilon \|^2 = \varepsilon^{*T} \varepsilon \qquad (2.72)$$

2.6.2 LS Optimal Filtering

- A direct approach to this problem would be as follows .

  Substitute (2.70) and (2.71) into (2.72) and expand the result to obtain

$$S = (x - Yh)^{*T} (x - Yh)$$

$$= x^{*T} x - h^{*T} Y^{*T} x - x^{*T} Yh + h^{*T} Y^{*T} Y h \qquad (2.73)$$

- Then by formal methods of differentiation , a necessary condition for the minimum can be found to be

$$(Y^{*T} Y)\, h = Y^{*T} x \qquad (2.74)$$

  This is the least squares Wiener –Hopf equation.

- If Y has independent columns (i.e. , if it is of full rank) , then $Y^{*T} Y$ is also of full rank and (2.74) has the solution

$$h = (Y^{*T} Y)^{-1} Y^{*T} x \qquad (2.75)$$

- The sum of the squared errors for the optimal filter can be found by returning to (2.32) and writing

$$S = (x - Yh)^{*T} (x - Yh)$$
$$= x^{*T} (x - Yh) - (Yh)^{*T} (x - Yh)$$
$$= x^{*T}x - x^{*T} Yh - h^{*T} (Y^{*T}x - Y^{*T}Yh)$$
$$= x^{*T}x - x^{*T} Yh \qquad (2.76)$$

- Denote that $Y^{+} = (Y^{*T}Y)^{-1} Y^{*T}$ (2.77)

Then we can write the optimal solution (2.75) of h as

$$h = Y^{+} x \qquad (2.78)$$

The matrix Y^{+} is known as Moore-Penrose pseudoinverse .

2.6.3 Least Squares Orthogonality

- **Theorem (Least Squares Orthogonality):**

  Let $x = Yh$ and $\varepsilon = x - x$

  Then h minimizes the sum of squared errors

  $$S = \| \varepsilon \|^2 \qquad (2.79)$$

  if h is chosen such that $Y^{*T} \varepsilon = 0$

  Further , the sum of squared errors is given by

  $$S = x^{*T} \varepsilon \qquad (2.80)$$

Proof :

Let h be any vector of filter coefficients and $h_{\perp}$ be the vector that results in orthogonality.

Further , let ε be the error vector corresponding to h and $\varepsilon_{\perp}$ be the error vector corresponding to $h_{\perp}$.

Then it follows that

$$\varepsilon = x - Yh = (x - Y h_{\perp}) + Y(h_{\perp} - h)$$

$$= \varepsilon_{\perp} + Y(h_{\perp} - h)$$

so that

$$\varepsilon^{*T} \varepsilon = [\varepsilon_{\perp} + Y(h_{\perp} - h)]^{*T} [\varepsilon_{\perp} + Y(h_{\perp} - h)]$$

$$= \varepsilon_{\perp}^{*T} \varepsilon_{\perp} + (h_{\perp} - h)^{*T} Y^{*T} \varepsilon_{\perp}$$

$$+ \varepsilon_{\perp}^{*T} Y (h_{\perp} - h) + (h_{\perp} - h)^{*T} Y^{*T} Y (h_{\perp} - h)$$

Since $\varepsilon_{\perp}^{*T} Y$ and $Y^{*T} \varepsilon_{\perp}$ are both zero by assumption , this leads to

$$S = \varepsilon^{*T} \varepsilon = \varepsilon_{\perp}^{*T} \varepsilon_{\perp} + (h_{\perp} - h)^{*T} Y^{*T} Y (h_{\perp} - h)$$

which is clearly minimized when $h = h_{\perp}$.

The minimum sum of squared errors is then

$$S = \varepsilon^{*T} \varepsilon = (x - Y h)^{*T} \varepsilon = x^{*T} \varepsilon$$

The last step follows because $Y^{*T} \varepsilon = 0$.

This proves the theorem.

The results of the above Theorem can now be applied to solve the optimal LS filtering problem. The Theorem requires that

$$Y^{*T}\varepsilon = Y^{*T}(x - Yh) = 0$$

which leads to the Wiener-Hopf equation (2.74)

$$Y^{*T}x = Y^{*T}Yh$$

- Further , from the theorem , the minimum sum of squared errors is given by

$$S = x^{*T}\varepsilon = x^{*T}(x - Yh) = x^{*T}x - x^{*T}Yh$$

as before (2.76) .

Example

The sequence {x(n); $0 \leqq n \leqq 4$ } = { 1,-1,1, -1,1 } is to be estimated from the observation sequence { y(n)} = { 1,-2,3, -4 ,5} using an FIR filter of length $P=2$ **. Choosing** $n_I = 1$, $n_F = 4$ **.**

We obtain the following least-squares problem :

The pseudo inverse of the data matrix is

The filter is the given by

Example #2 (Estimating Expected Values from Data)

- **Assuming that $x^{(1)}$ $x^{(2)}$ … $x^{(K)}$ are samples of a random vector x.**

  The expectation of a function φ(x) can be approximated as

  $$[\varphi(x)] = (1/K) \Sigma \varphi(x^K)$$

- **When estimating for expectation of a quantity involving two random vectors from samples $x^{(1)}$ $x^{(2)}$ … $x^{(K)}$ and $y^{(1)}$ $y^{(2)}$ … $y^{(K)}$.**

  The estimate for the expectation takes the form

  $$[\varphi(x,y)] = (1/K) \Sigma \varphi(x^K, y^K)$$

- **For complex-valued random vectors , define the data matrix X by**

$$\mathbf{X} = \left(\mathbf{x}^{(1)*} \; \mathbf{x}^{(2)*} \; \ldots \; \mathbf{x}^{(K)*} \right)^{T}$$

Then since $\mathbf{X}^{*T}\mathbf{X} = \Sigma\, \mathbf{x}^{(k)*}\, \mathbf{x}^{(k)*T}$

The correlation matrix can be written as

$$\mathbf{R}_x = (1/K)\, \mathbf{X}^{*T}\mathbf{X}$$

Then a typical element $r_{kl} = (1/K)\, x_k^{*T} x_l$

Toeplitz Matrix

- **A Toeplitz matrix** or **diagonal-constant matrix**, **named after Otto Toeplitz, is a matrix in which each descending diagonal from left to right is constant. For instance, the following matrix is a Toeplitz matrix:**

a b c d e
f a b c d
e f a b c
d e f a b
c d e f a

Toeplitz systems of form $Ax = b$ can be solved by the Levinson- Durbin in $\Theta(n^2)$ time.

Note that Levinson recursion or Levinson-Durbin recursion is a procedure in linear algebra to recursively calculate the solution to an equation involving a Toeplitz matrix.

The algorithm runs in $\theta(n^2)$ time,

Appendix

Norbert Wiener (1894-1964)

- Norbert Wiener was born in Columbia, Missouri.

  He entered Tufts College at age 11.

  Harvard awarded Wiener a Ph.D. in 1912, when he was a mere 18, for a dissertation on mathematical logic.

  In 1914, Wiener traveled to Europe, to study under Bertrand Russell and G.H.Hardy at Cambridge University, and under David Hilbert and Edmund Landau at the Univesity of Gottingen .

- Wiener 's position in the mathematics Department at MIT began in 1919. He was promoted to Professorship in 1932.

  He was a pioneer in the study of stochastic and noise processes, contributing work relevant to electronic engineering, communications and control systems.

- **Wiener is perhaps best known as the founder of cybernetics, a field that formalizes the notion of feedback and has implications for engineering , systems control , computer science , biology , philosophy, and the organization of society.**

- **In 1942 ,Wiener developed theory of signal transmission in the presence of a perturbative noise, in a classified monograph ,nicknamed "the yellow peril“ and then in Extrapolation, Interpolation and Smoothing of Stationary Time Series with Engineering Applications (1949).**

  In this book Wiener applies generalized harmonic analysis to stationary random signals and solves the problem of optimal elimination of the perturbative noise and of optimal prediction of the signal itself, with the help of a filtering operator.

  Quite independently, A.N. Kolmogoroff had announced results in the same domain at a time (1941) when scientific communications were interrupted.

Norman Levinson (, 1912 –1975)

was an American mathematician. He worked closely with Norbert Wiener in his early career. He joined the faculty of the MIT in 1937.

He received both his BS degree and his master degree in electrical engineering from MITin 1934, where he had studied under Wiener and took almost all of the graduate-level courses in mathematics. He received the MIT Redfield Protor Traveling Fellowship to study at the University of Cambridge , with the assurance that MIT would reward him with a Ph. D. upon his return regardless of whatever he produced at Cambridge. Within the first four months in Cambridge, he had already produced two papers. In 1935, MIT awarded him with the Ph. D. in mathematics.