Chapter 2-3

Adaptive Filters

References :

1. S. Haykin , Adaptive Filter Theory , 3rd edition 1996
2. B. Widrow, et al," Adaptive Noise Cancelling : Principles and Applications", Proc. IEEE, Dec.1975, pp.. 1692-1716.
3. B.Widrow et al, " Stationary and Nonstationary Characteristics of the LMS Adaptive Filter", Proc. IEEE , Aug.1978 , pp.1151-1162 .

Contents

2.6 Introduction to Adaptive Filtering

- Adaptive signal processing technique has been widely applied in the area of communications, control, and array processing.
- The application of adaptive technique in communications include : adaptive equalizer, adaptive echo cancellation, adaptive predictive coding of speech, etc.
- The adaptive systems for signal processing usually has all of the following characteristics :

  (1) they can automatically adapt in the face of changing environments and changing system requirements

  (2) they can be trained to perform specific filtering or decision-making tasks

  (3) there should be some "adaptive algorithm" for adjusting the system's parameters.

2.7 Adaptive Wiener Filter

- **Figure 2.x1 shows a simplified block diagram of a Wiener filter.**

$$\hat{x}(n) = \Sigma_{k=1}^{M} W(n,k)\, y(k) \qquad (2.166)$$

The optimal weights of an FIR Wiener filter is obtained by solving the normal equation

$$E[x(n)\, y(i)] = \Sigma_{k=1}^{M} W(n,k)\, E[y(k)\, y(i)]$$

$$\text{for } (n-M) \leqq i \leqq n$$

$$(2.167)$$

We can write Eq.(2.167) in vector notation as

$$\mathbf{p} = E[\,x(n)\mathbf{y}\,] \tag{2.168}$$

and $\hat{x}(n) = \mathbf{W}^T \mathbf{y}$ **(2.169)**

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

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

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

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

$$\mathbf{W}^o = E\,[x(n)\mathbf{y}\,]\,E[\mathbf{y}\mathbf{y}^T\,]^{-1}$$
$$= \mathbf{R}_{YY}^{-1}\,\mathbf{p} \qquad (2.170)$$

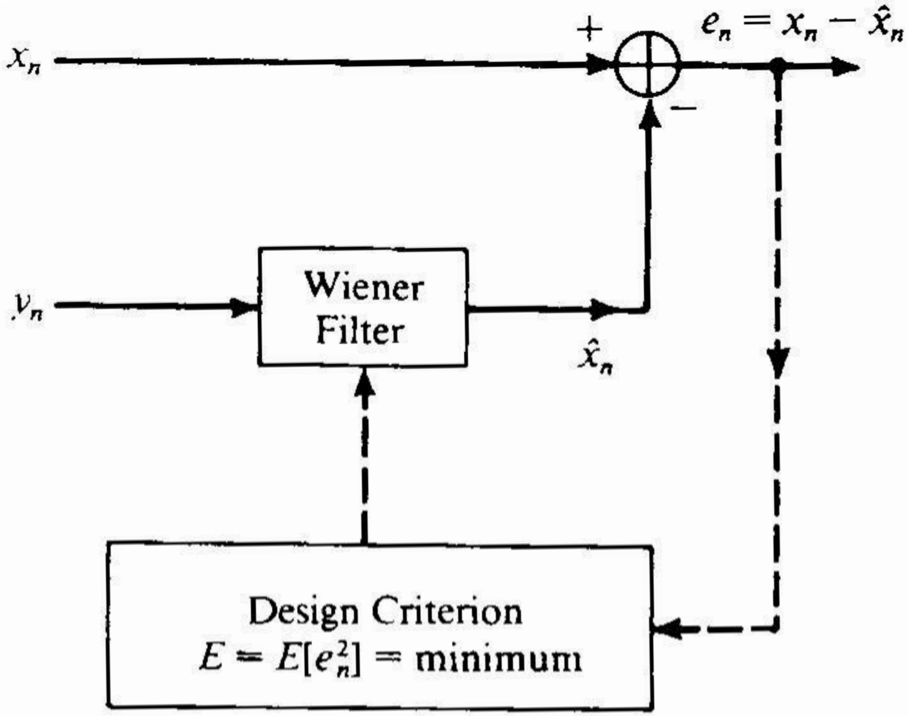
The estimated mean-squared error at time instant *n* is expressed as

$$J_n(\mathbf{w}) = E[\,|\,e_n\,|^2] = E[\,(\,|\,x(n) - x(n)\,|^2\,]$$

$$= E[e_n\,\{x(n) - \Sigma_{k=1}^{M}\,w(n,k)\,y(k)\,\}\,x(n)$$

$$= E[x^2(n)] - E[x(n)\mathbf{y}^T\,]\,E[\mathbf{y}\mathbf{y}^T\,]^{-1}\,E[\mathbf{y}x(n)] \qquad (2.171)$$

Fig.2.x1



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

- **One difficulty in practice with the Wiener solution is that the statistical quantities R and p must be known, or at least estimated, in advance.**

  This is what is done in most speech processing:

  the input speech is divided into fairly short segments known as frames , each frame is assumed to be a stationary process. The statistical correlations are estimated by sample correlations , and, finally, the optimal weights corresponding to each frame are computed.

  This procedure is block-by-block adaptive.

- **The Widrow-Hoff least –mean-square (LMS) adaptive algorithm is an alternate approach that adapts the filter's weights (or called coefficients) on a sample-by- sample basis.**

  It does not require a priori knowledge of the correlation matrices. The only requirement for this algorithm is that the statistical properties of the input signal not to be changing fast , so that the filter has time to converge to the optimal weights.

- **A typical adaptive implementation of a Wiener filter is depicted in Fig. 2.x2. The adaptive algorithm continuously monitors the output error signal e_n and attempts to minimize the output power $e_n{}^2$. At each time instant n , the current values of the weights are used to perform the filtering operation.**
- **The computed output e_n is then used by the adaptation part of the algorithm to change the weights in the direction of their optimum values.**
- **As processing of the input signals $x(n)$ and $y(n)$ take place and the filter gradually learns the statistics of these inputs,**

Its weights gradually converge to their optimum values given by the Wiener solution.

Fig.2.x2

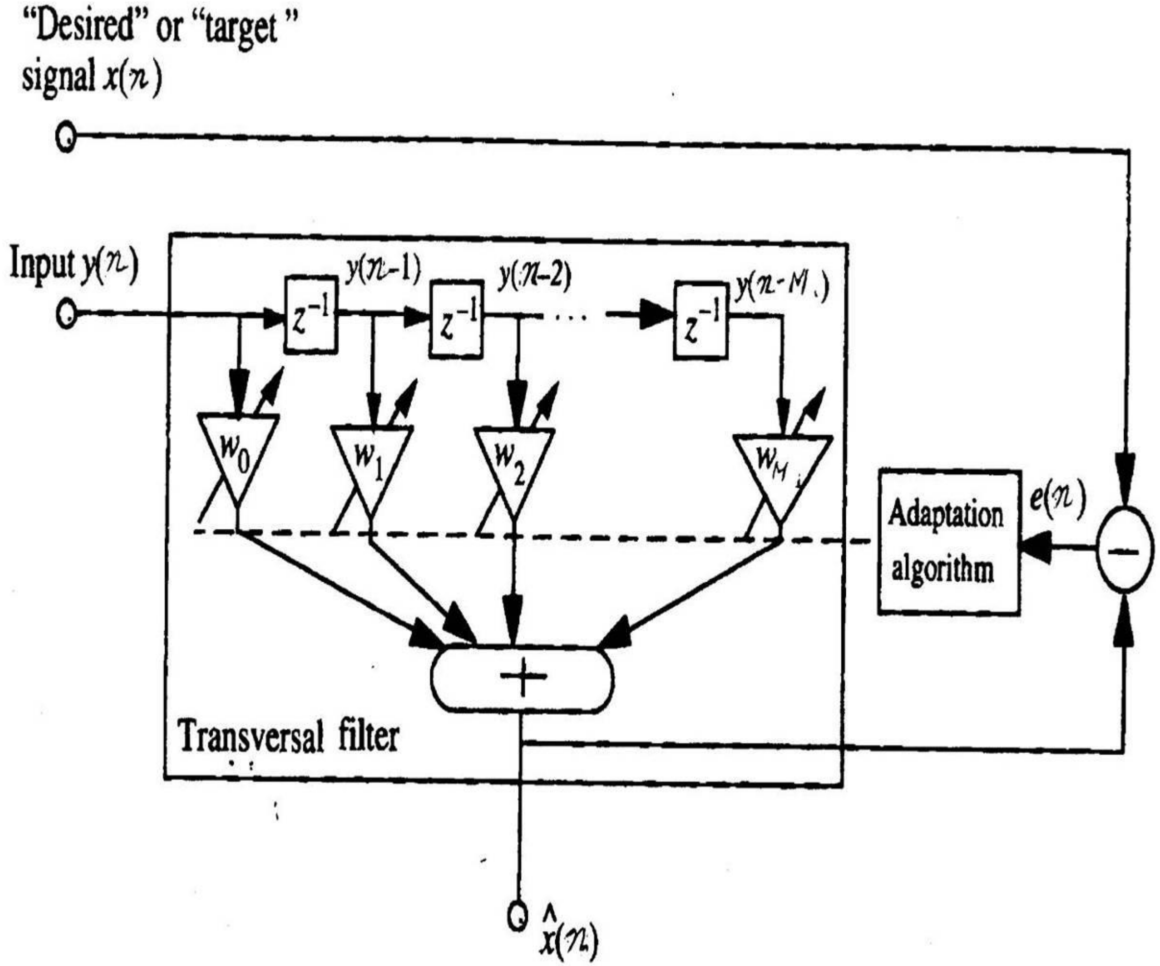


Illustration of the configuration of an adaptive filter.

- **The transversal filter (also called FIR filter) shown in Fig.2.2 is fundamental to adaptive signal processing. Because of its nonrecursive structure , it is relatively easy to implement and analyze.**
- **The output of the transversal filter is given by**

$$\hat{x}(n) = \sum_{k=0}^{M} w_k\, y(n-k) = \mathbf{W}^T . \mathbf{Y}(n) \qquad (2.172)$$

where $\mathbf{W} = (w_0, w_1, w_2, \ldots, w_M)^T$

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

and $e_n = x(n) - \hat{x}(n) \qquad (2.173)$

The mean-squared estimation error J_n is expressed as

$$J_n = E[e^2(n)] = E[\{x(n) - \mathbf{W}^T . \mathbf{Y}(n)\}^2] \qquad (2.174)$$

- **The adaptation of the tap weights toward their optimal values iteratively by using the method of steepest descent .**

Note that the direction of the negative gradient is known as the direction of steepest descent.

The weights are updated in time according to the algorithm

$$\mathbf{W}(n+1) = \mathbf{W}(n) - \mu \, \nabla_{\mathrm{w}} J_n \qquad (2.175)$$

where μ is the step- size parameter and ∇_{w} is the gradient operator defined as column vector

$$\nabla_{\mathrm{w}} = [\delta / \delta w_0 \;\; \delta / \delta w_1 \; \ldots \; \delta / \delta w_M]^T \qquad (2,176)$$

Note that the *i*th element of the gradient vector $\nabla_{\mathrm{w}} J_{\mathrm{n}}$ is

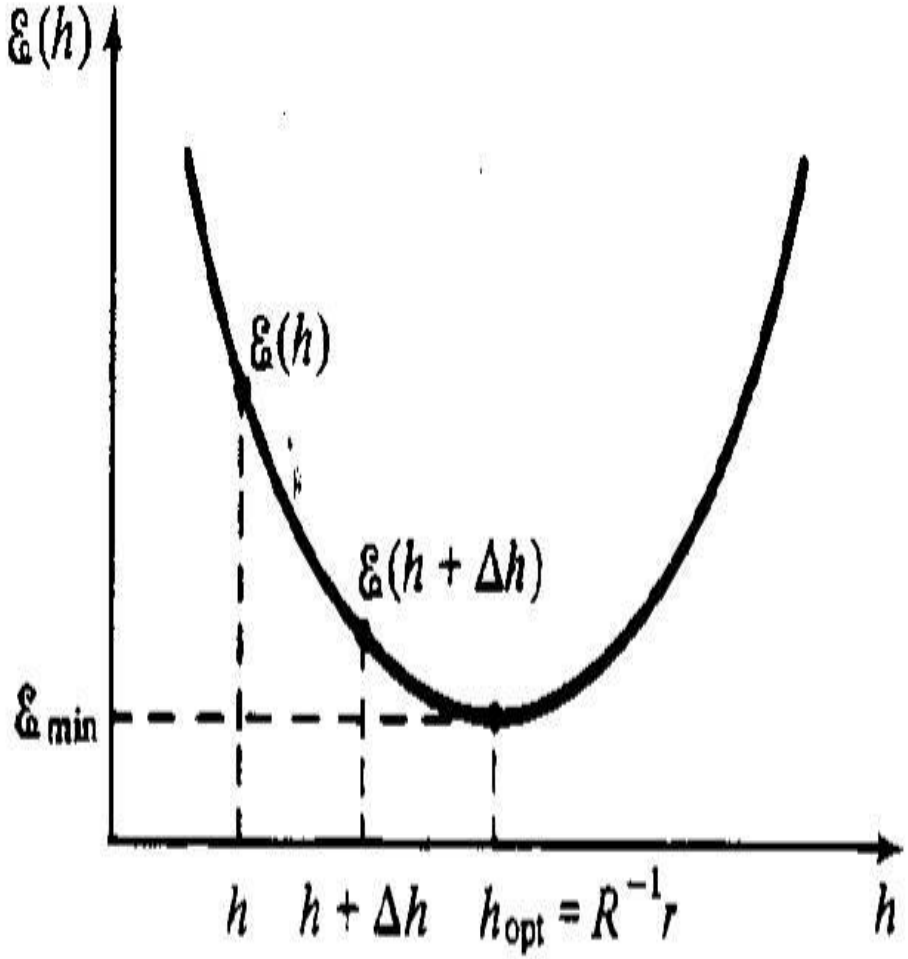
$$\delta J_n / \delta w_i = -2 \, E[e_n \, y(n-i)] \qquad (2.177)$$

and $$\nabla_{\mathrm{w}} J_{\mathrm{n}} = -2 \, E[e_n \, \mathbf{Y}(n)] \qquad (2.178)$$

Then $$\mathbf{W}(n+1) = \mathbf{W}(n) + 2 \mu \, E[e_n \, \mathbf{Y}(n)] \qquad (2.179)$$

This is the well-known steepest descent algorithm.

Fig. 2.x3



$\mathcal{E}(h)$
$\mathcal{E}(h)$
$\mathcal{E}(h + \Delta h)$
$\mathcal{E}_{min}$
h
$h + \Delta h$
$h_{opt} = R^{-1}r$
h

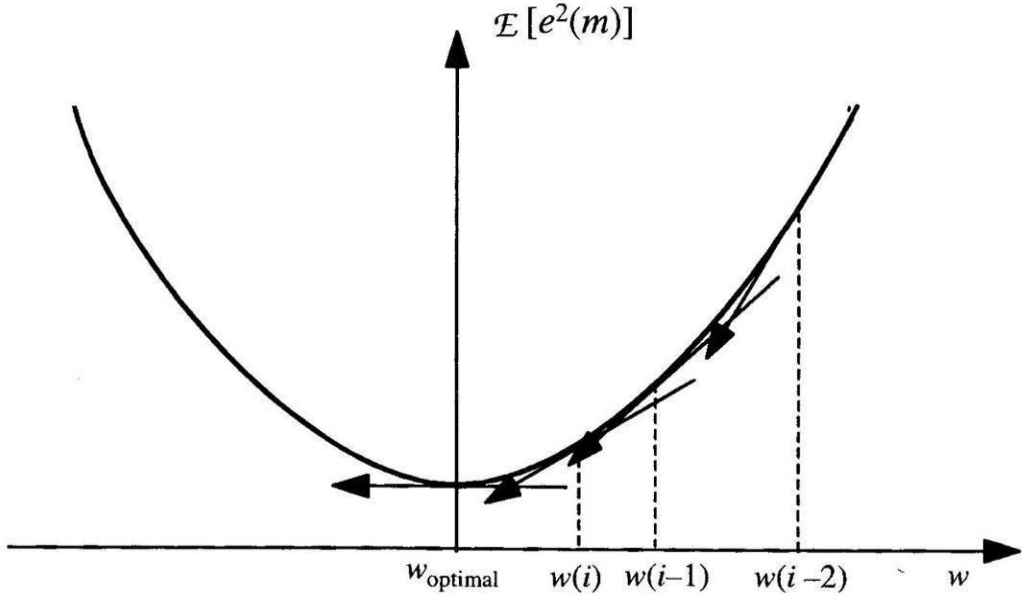


Illustration of gradient search of the mean square error surface for the minimum error point.

2.8 The LMS Algorithm

- **The main limitation of the steepest descent algorithm is that it requires exact measurement of the gradient vector in each iteration.**
- **A method of finding approximate solution of the optimal weights is the so-called least mean square (LMS) algorithm proposed by Widrow and Hoff in 1960 .**
- **The LMS algorithm is the most widely used adaptive filtering algorithm , in practice. This can be attributed to its simplicity and robustness to signal statistics.**
- **The conventional LMS algorithm is a stochastic implementation of the steepest descent algorithm .**

It simply replaces the cost function $J_n = E[\,|e_n|^2]$ by its instantaneous coarse estimate $J_n = |e_n|^2$

- **The LMS is obtained from equation (2.x13) by taking the instantaneous estimate of the gradient ,instead of statistical average. Then we have**

$$\nabla_{\mathrm{W}} J_n = -2\ e_n \mathrm{Y}(n) \qquad (2.180)$$

$$\mathrm{W}(n+1) = \mathrm{W}(n) + 2\mu\ e_n \mathrm{Y}(n) \qquad (2.181)$$

- **In summary , three steps are required to complete each iteration of the LMS algorithm in adaptive filtering :**

$$(1)\ x^{\wedge}(n) = \Sigma_{k=0}^{M}\ w(\mathrm{k})\, y(n-k) = \mathrm{W}^{\mathrm{T}} . \mathrm{Y}(n) \qquad (2.182)$$

$$(2)\ e_n = x(n) - x^{\wedge}(n) \qquad (2.183)$$

$$(3)\ \mathrm{W}(n+1) = \mathrm{W}(n) + 2\mu\ e_n \mathrm{Y}(n) \qquad (2.184)$$

2.9 Convergence Property of LMS Algorithm

- **From the tap-weight adaptation equation (2.x19),we have the relation**

$$\begin{aligned} \mathbf{W}(n+1) &= \mathbf{W}(n) + 2\mu e_n \mathbf{Y}(n) \\ &= \mathbf{W}(n) + 2\mu \mathbf{Y}(n)\{x(n) - \mathbf{W}^T \cdot \mathbf{Y}_n\} \\ &= \mathbf{W}(n) + 2\mu \mathbf{Y}(n)\{x(n) - \mathbf{Y}(n)^T \cdot \mathbf{W}\} \\ &= \{1 - 2\mu Y(n)\mathbf{Y}(n)^T\}\mathbf{W}(n) + 2\mu \mathbf{Y}(n)x(n) \end{aligned} \quad (2.185)$$

Denote that $\mathbf{c}(n) = \mathbf{W}(n) - \mathbf{W}^o(n)$

where $\mathbf{W}_o(n)$ **is the optimal weights of the Wiener solution.**

Then ,

$$\begin{aligned} \mathbf{c}(n+1) &= (1 - 2\mu \mathbf{Y}(n)\mathbf{Y}(n)^T)\,\mathbf{c}(n) \\ &\quad + 2\mu \{x(n)\mathbf{Y}(n) - \mathbf{Y}(n)\mathbf{Y}(n)^T \mathbf{W}^o(n)\} \end{aligned} \quad (2.186)$$

Taking the expectation values of both sides of Equation (2.x20), we obtain

$$E[\mathbf{c}(n+1)] = (1-2\mu \mathbf{R}_{yy}) E[\mathbf{c}(n)] + 2\mu(\mathbf{R}_{yx} - \mathbf{R}_{yy}\mathbf{W}^{o}(n)) = (1-2\mu \mathbf{R}_{yy}) E[\mathbf{c}(n)] \qquad (2.187)$$

Here we have assumed that each sample vector $\mathbf{Y}(n)$ is uncorrelated with all previous sample vector $Y(i)$, $i = 1,2,\ldots,n-1$, and each $\mathbf{Y}(n)$ is also uncorrelated with all previous samples of $x(i)$. Therefore , the weight vector $W(n)$ is independent of the input vector $\mathbf{Y}(n)$.

- **Consider the equation**

$$\mathbf{c}(n+1) = (1-2\mu \mathbf{R}_{yy})\, \mathbf{c}(n) \qquad (2.188)$$

If we diagonalize the matrix $\mathbf{R}_{yy}$ using a unitary matrix Q by $Q^T R_{yy} Q = \Lambda$, the diagonal matrix Λ consists of the eigenvalues of $\mathbf{R}_{yy}$.

Define the transformed vector

$$\mathbf{V}(n+1) = \mathbf{Q}^T \mathbf{c}(n+1)$$

Then we obtain

$$\mathbf{V}(n+1) = (1 - 2\mu\Lambda)\,\mathbf{V}(n) \qquad (2.189)$$

The solution to equation (2.102) is

$$v_j(n+1) = (1 - 2\mu\lambda_j)\, v_j(n) \qquad (2.190)$$

for $j = 1,2,\ldots,M$, where λ_j is the j-th eigenvalue of $\underline{R}_{yy}$.

Therefore we obtain

$$v_j(n) = (1 - 2\mu\lambda_j)^n\, v_j(0) \qquad (2.191)$$

The convergence of weights requires that

$$|\,1 - 2\mu\lambda_j\,| < 1$$

or , equivalently, $0 < \mu < 1/\lambda_j$ for all j .

The condition is guaranteed if

$$0 < \mu < 1/\lambda_{max} \qquad (2.192)$$

where λ_{max} is the maximum eigenvalue of R_{yy} .

- **From equations (3.13) and (3.16), we conclude that the LMS algorithm converge in the mean provided that**

$$0 < \mu < 1/\lambda_{max}$$

- **Excessive mean-squared- error (MSE) :**

When $W(n) = W^0$, the true gradient is zero. But the gradient estimated in the LMS algorithm is equal to the gradient noise, $-2\,e(n)\,Y_n$, from equation (2.180) .

The weight vector is on the average " misadjusted" from its optimal setting.

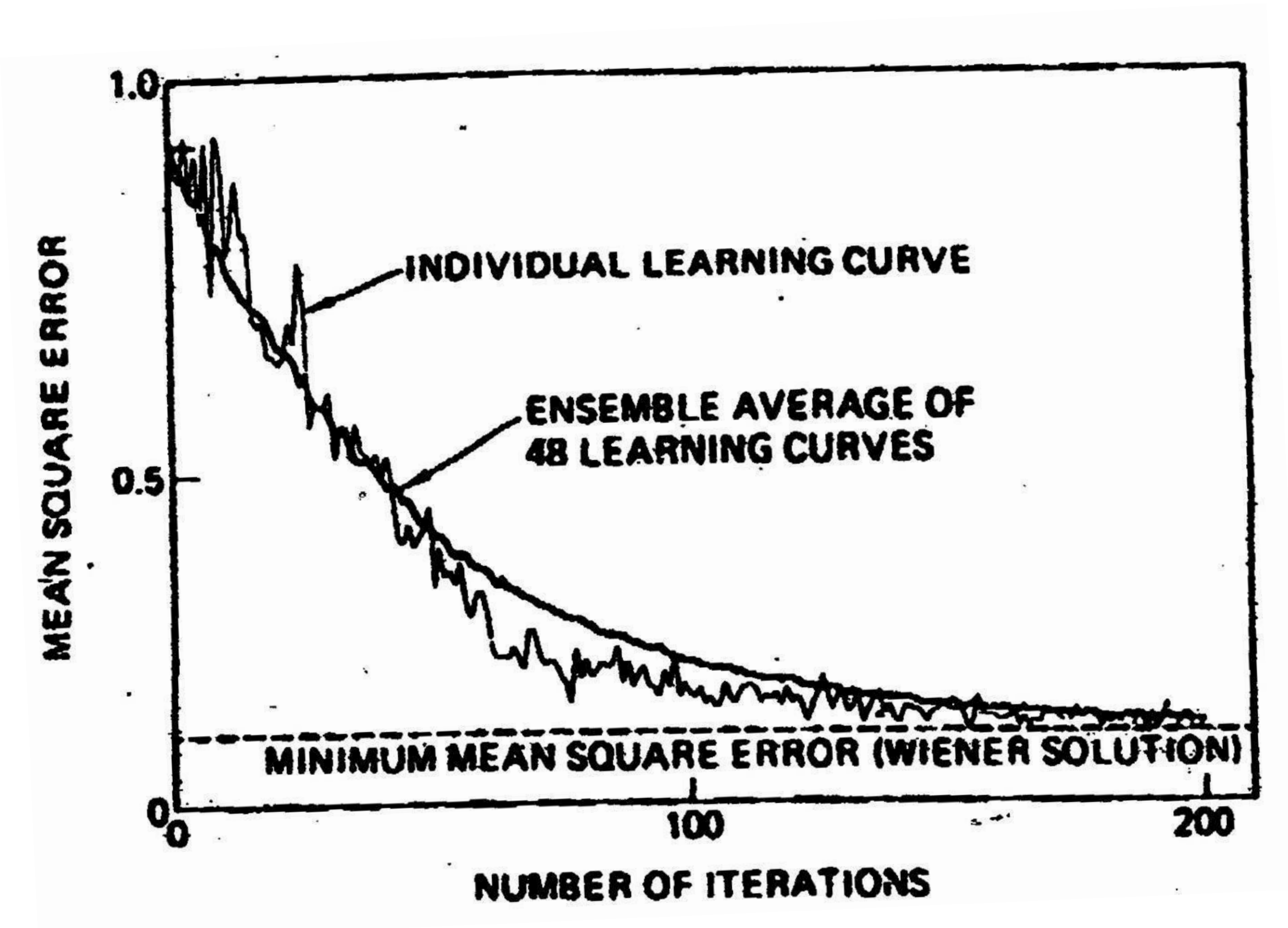
The excessive mean- squared-error (MSE) is given by

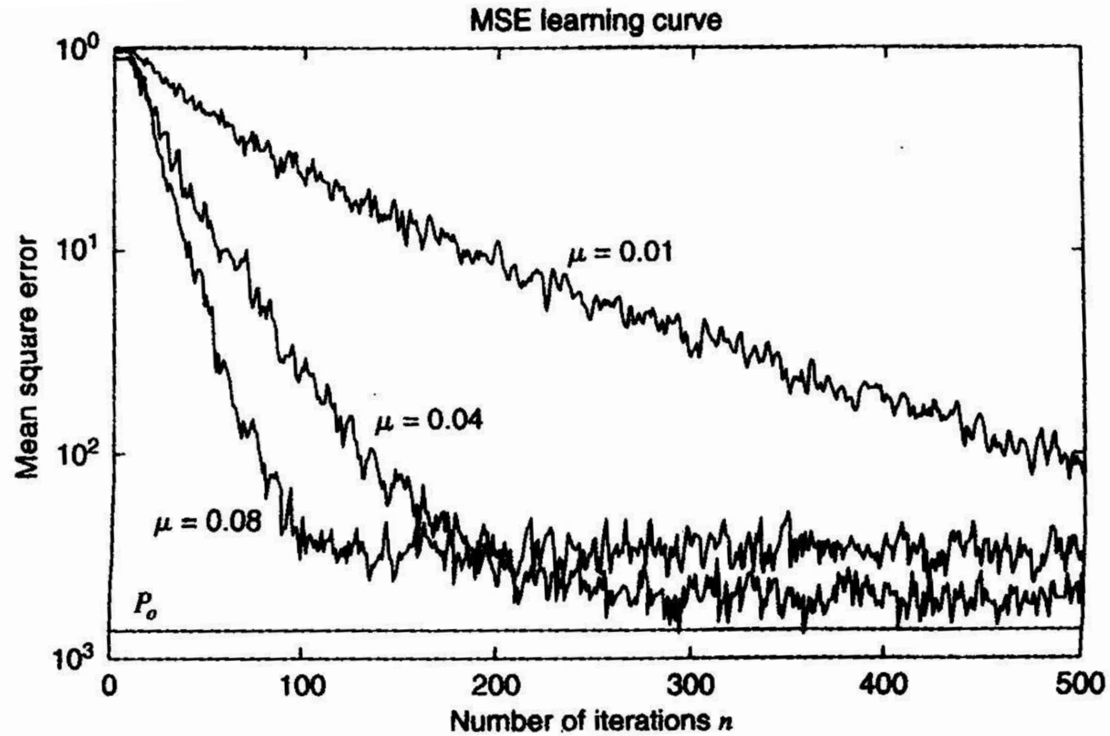
$$\text{Excessive MSE} = E[V^T \Lambda V] = \Sigma_{j=0}^{M} \lambda_j E[v_j^{\,2}] \qquad (2.193)$$

- **The convergence property of the LMS algorithm is illustrated by its learning curve , as shown in Fig. 3.4. It can be seen that the learning curve consists of noisy decaying exponentials.**

  In general , the convergence speed is inversely proportional to the step-size μ .

Fig.2.4


1.0
0.5
0
0
100
200
MEAN SQUARE ERROR
NUMBER OF ITERATIONS
INDIVIDUAL LEARNING CURVE
ENSEMBLE AVERAGE OF
48 LEARNING CURVES
MINIMUM MEAN SQUARE ERROR (WIENER SOLUTION)

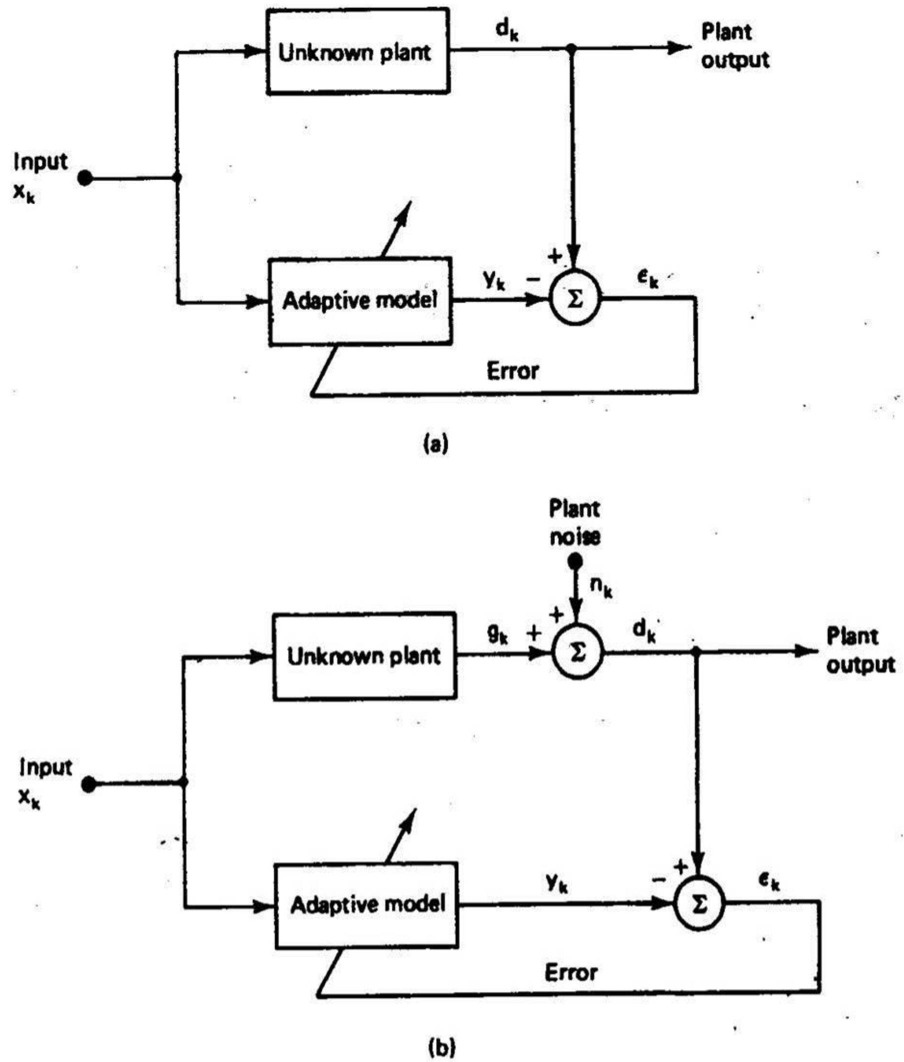


FIGU
MSE learning curves of the LMS algorithm in the adaptive equalizer: $W = 2.9$.

2.11 Applications of Adaptive Filtering

- i. System Identification
- ii . Adaptive Noise Cancellation
- iii. Adaptive Echo Cancellation
- iv. Adaptive Equalization

2.11.1 System Identification

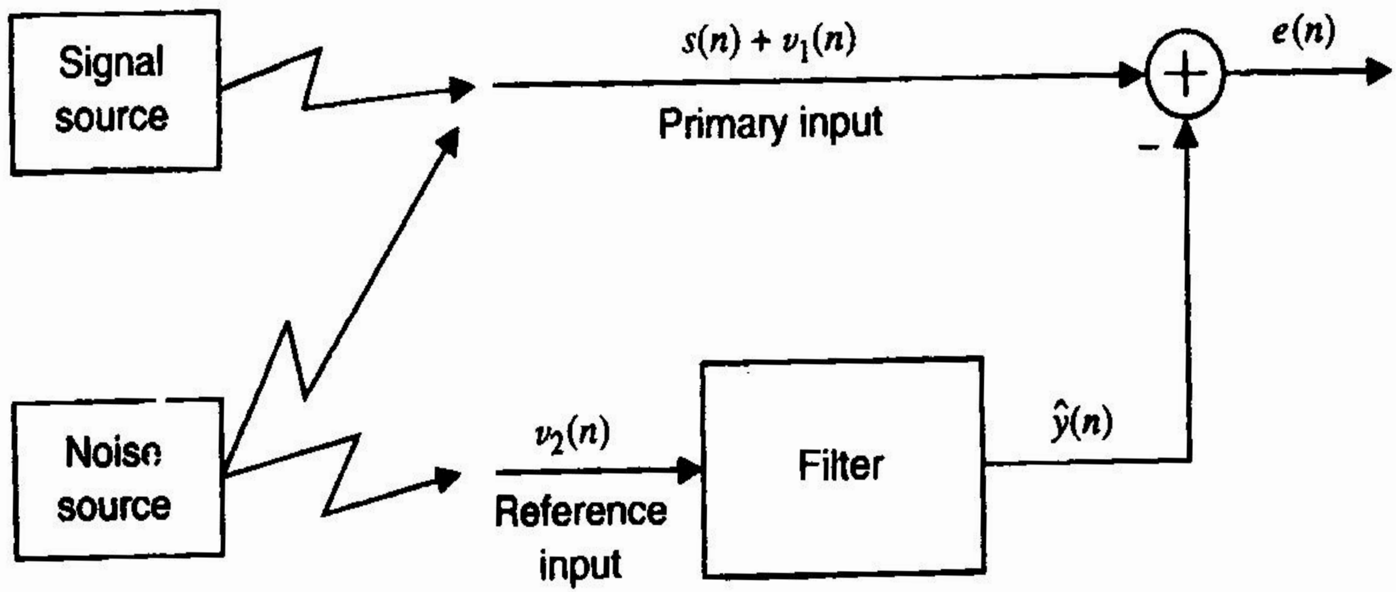


Modeling the single-input, single-output plant: (a) noise-free case; (b) noisy plant case.

From: Widrow & Stearns

- **An adaptive filter can be used in modeling a system. That is, it can imitate the behavior of physical dynamic system. In the figure, both the unknown system and the adaptive filter are driven by the same input. The adaptive adjusts itself with the goal of causing its output to match that of the unknown system.**

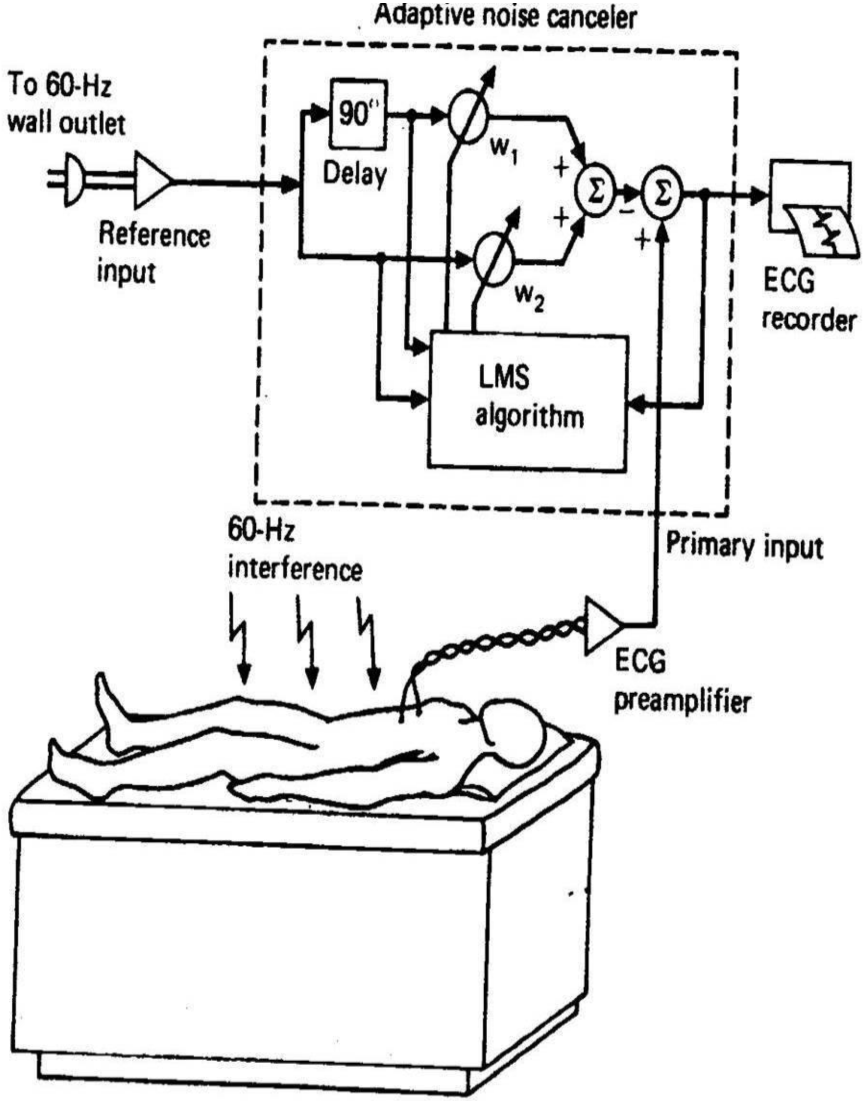
2.11.2 Adaptive Noise cancellation



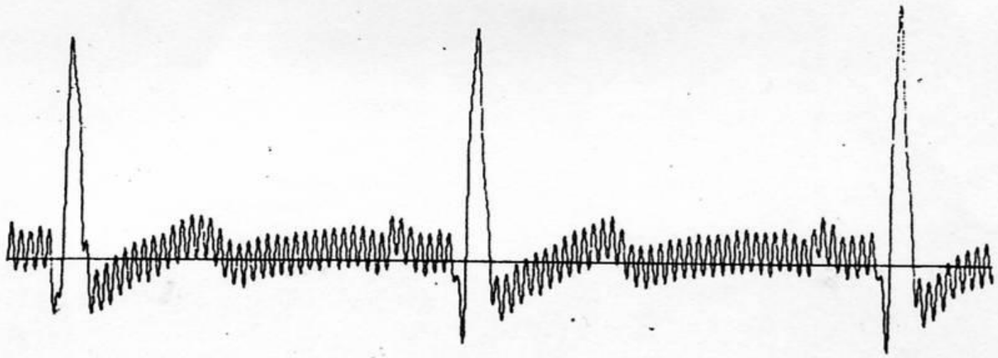
Principle of adaptive noise cancelation using a reference input.

- **In an adaptive noise canceller, two input signals d_k and x_k are available. The primary input d_k is composed of a desired signal plus undesired noise interference , and the other input x_k is composed of noise interference which is correlated with the noise part of the primary input.**

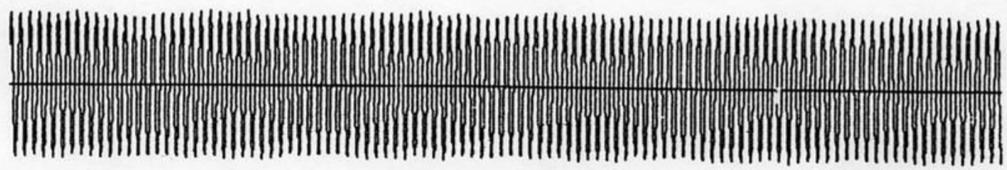
- **The adaptive noise canceller operates as a correlation canceller. It produces the best possible replica of the noise component of d_k and proceeding to cancel it.**
- **When the secondary signal x_k is purely sinusoidal at some frequency f_0 *, the adaptive filter behaves as a notch filier at the sinusoidal frequency*.**



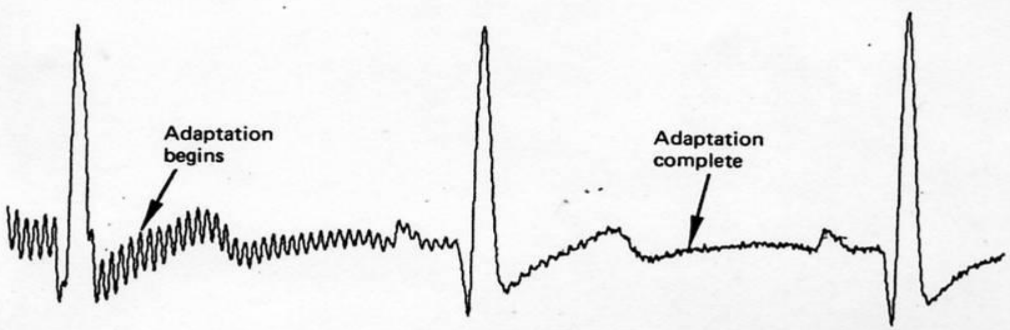
Cancelling 60-Hz interference in electrocardiography. From B. Widrow et al., *Adaptive Noise Canceling: Principles and Applications*, © December 1975, IEEE.



(a)



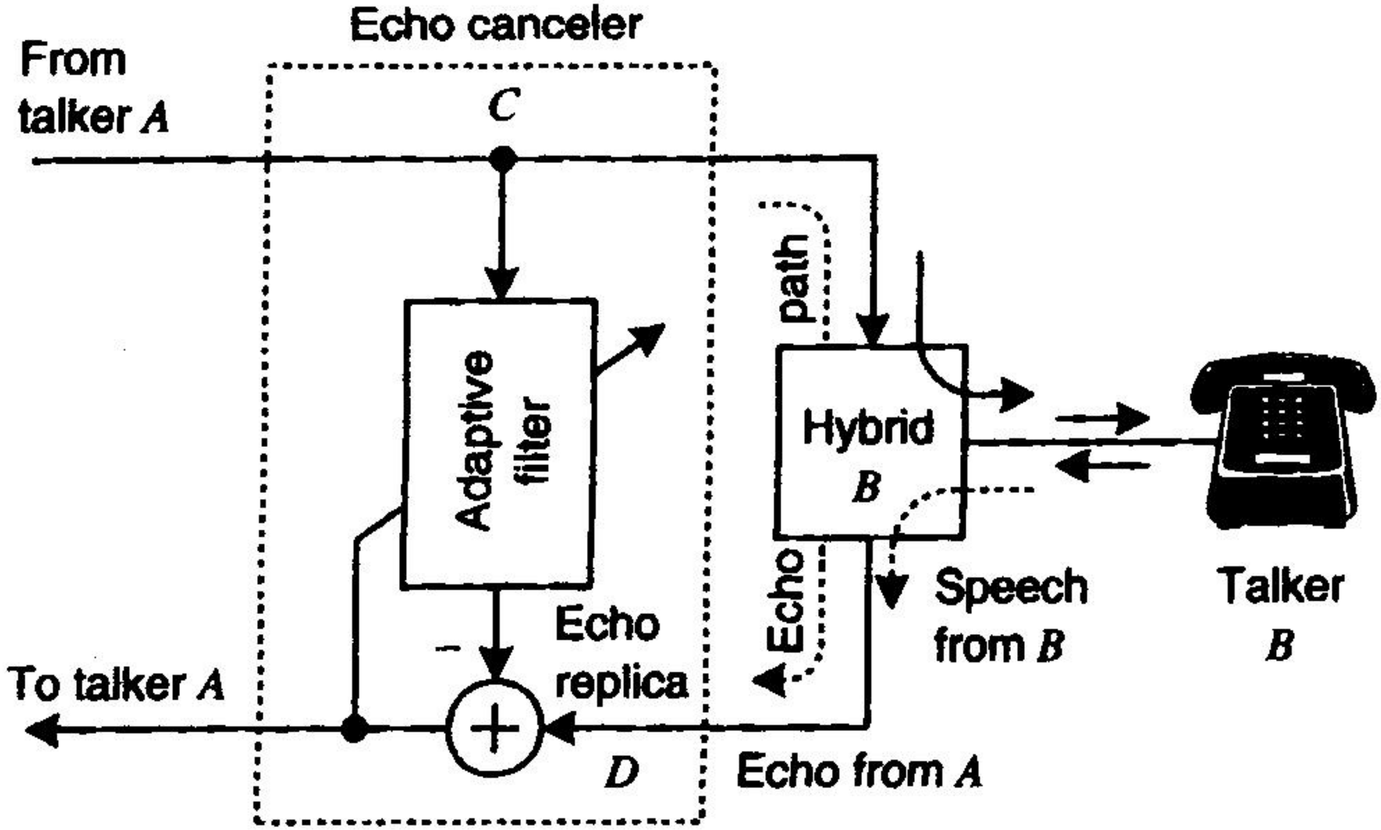
(b)



(c)

Figure 12.15 Electrocardiographic noise canceling: (a) primary unit; (b) reference input; (c) noise canceler output. From B. Widrow et al., *Adaptive Noise Canceling: Principles and Applications*, © December 1975, IEEE.

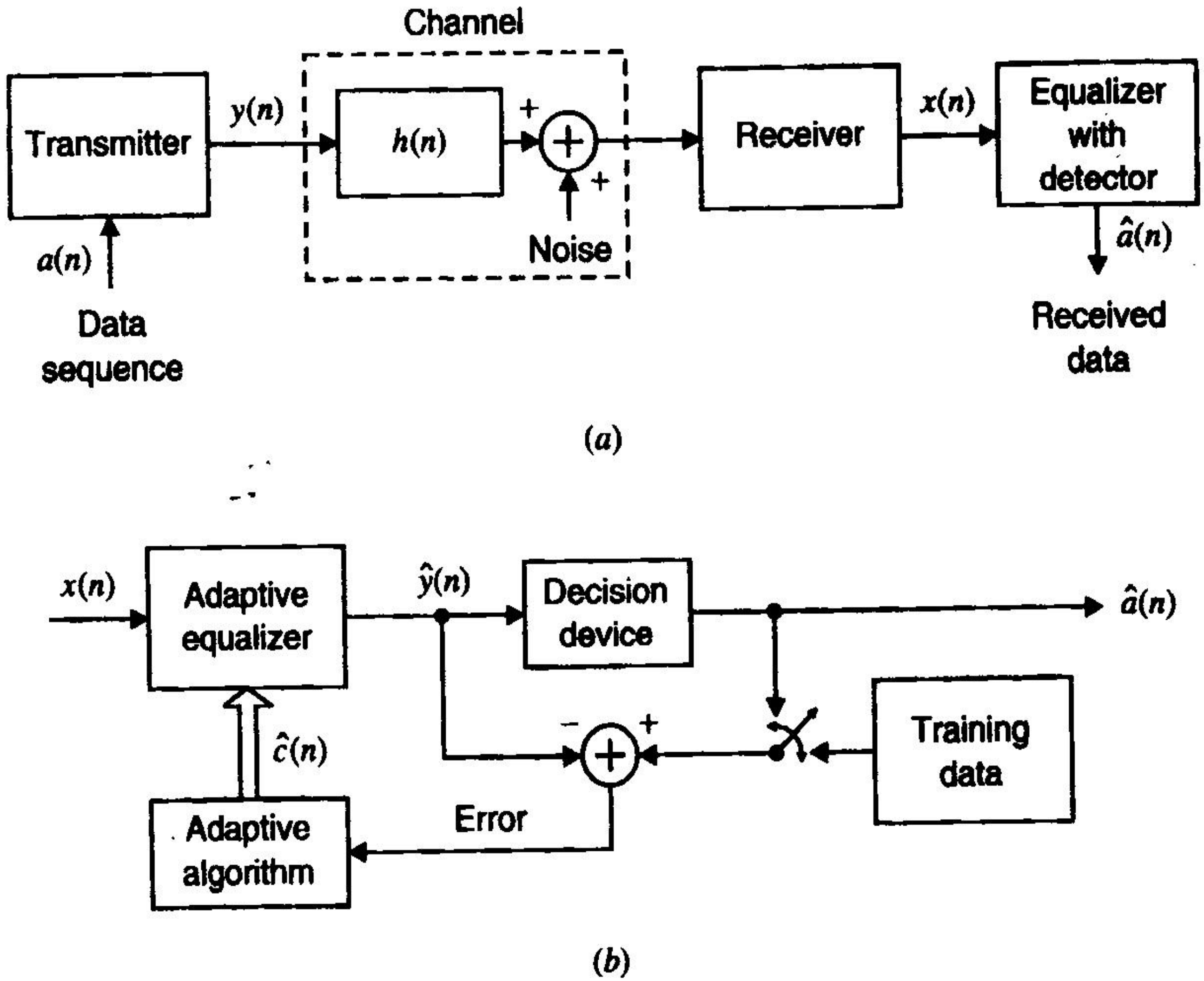
2.11.3 Adaptive Echo Canceller



Principle of echo cancelation.

- **Consider two speakers connected to each other by the telephone network . Due to various impedance mismatch (such as the hybrid connecting a four-to-two wire transmission)an echo is generated.**
- **The task of the echo canceller is to replicate the echo signal and subtract this from the incoming signal (echo plus far-end signal) . Since the echo path is unknown and can change , the echo canceller must adaptively try to produce an echo estimate.**

2.11.5 Adaptive Equalizer



Model of an adaptive equalizer in a data transmission system.

- **In the baseband part of the receiver in a transmission system, the equalizer has only the corrupted signal available , as shown in the above figure. The control signal (for adaptation) must be derived from the received signal itself , for example by taking the output signal of a detector in a decision-oriented data equalizer. The operation of the equalizer involves a training mode followed by a tracking mode. During the training , a known test signal is transmitted to probe the channel. By generating a synchronized version of the test signal in the receiver , the adaptive equalizer is supplied with a desired response d_k .**

The equalizer output is subtracted from this desired response to produce an error signal , which is in turn used to adaptively adjust the filter's coefficients to their optimum values

- **When the initial training period is completed , the coefficients of the adaptive equalizer may be continually adjusted in a decision-oriented mode. In this mode , the error signal is obtained from the final receiver estimate of the transmitted (signal) sequence . The receiver estimate is ontained by applying the adaptive equalizer output to a decision device. In normal operation , the receiver decisions are correct with a high probability , so that the estimate of the error signal is correct often enough to allow the adaptive equalizer to maintain proper adjustment of its coefficients.**

2.12 RLS Adaptive Filter

- **The LMS adaptation algorithm , based on the steepest descent method provides a gradual, iterative minimization of the performance index. The adaptive tap-weights are not optimal at each time instant , but only after convergence.**
- **Adaptive recursive least- squares (RLS) algorithms , based on the exact minimization of least squares criteria, are the time-recursive analogs of the adaptive FIR Wiener filtering .**
- **Because of their fast convergence RLS algorithms have been proposed for use in fast start-up channel equalization. They are also routinely used in real-time system identification applications.**
- **The main disadvantage is that they require a fair amount of computation (O(M^2), for M-tap filters) per time update.**

2.13 Adaptive Transversal Filters using Least Squares Method

- **Consider the stationary adaptive FIR filter shown in Fig.2.x2. If the optimal estimation criterion is to minimize the weighted sum of squares of the difference between x_i and $\Sigma_{k=0}^{M} w_k\, y(i-k)$ for $i = 1,2,\ldots,n$.**

Then the estimation is called the least square (LS) method.

- **Denote that**

$$S = \Sigma_{i=1}^{n} \lambda^{n-i} \,|\, e_i \,|^{2} \qquad (2.1194)$$

and

$$e_i = (x(i) - \Sigma_{k=0}^{M} w_k\, y(i-k))$$

$$= x(i) - \mathbf{W}^T \mathbf{Y}(i) \qquad (2.195)$$

where

$$\mathbf{W}(n) = (w_0(n), w_1(n), \ldots, w_M(n))^T \qquad (2.196)$$

$$\mathbf{Y}(i) = (y(i), y(i-1), \ldots, y(i-M))^T \qquad (2.197)$$

Fig.3.2

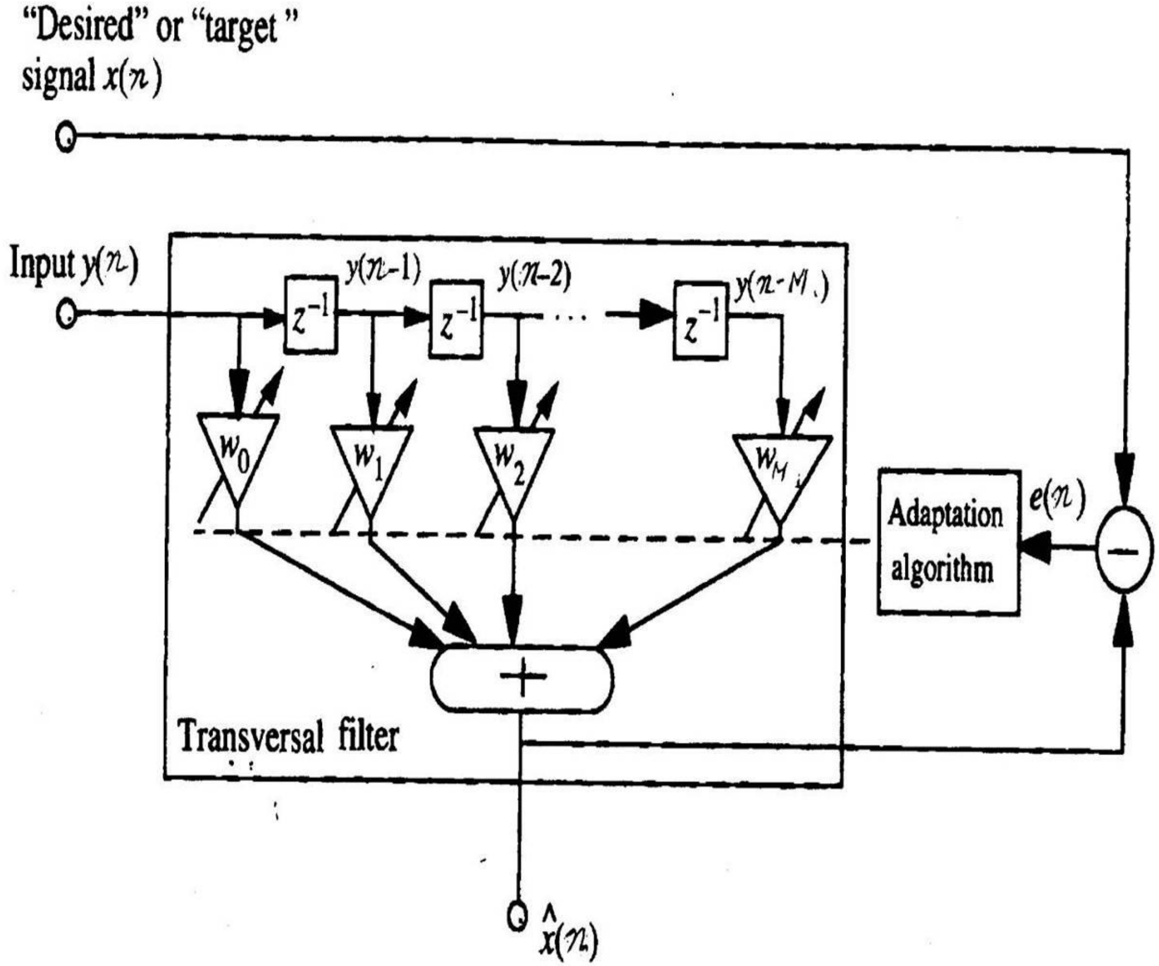


Illustration of the configuration of an adaptive filter.

- **Define the deterministic correlation function as follows.**

$$R_{yy}(n) = \sum_{i=1}^{n} \lambda^{n-i}\ Y(i)\, Y(i)^T \quad ,$$

$$R_{xy}(n) = \sum_{i=1}^{n} \lambda^{n-i}\ X(i)\, Y(i)^T$$

By minimizing *S* , we obtain the normal equations

$$R_{yy}(n)\, W_{op} = R_{xy}(n)$$

or $$W_{op} = R_{yy}(n)^{-1}\ R_{xy}(n)$$

$$= P(k)\, R_{xy}(n) \qquad (2.198)$$

where $$P(k) = R_{yy}(n)^{-1} \qquad (2.199)$$

These equations are the least square analog of the Wiener solution.

- **The recursive least square (RLS) algorithm is obtained by writing the time-dependent correlations as**

$$R_{yy}(n) = \lambda\, R_{yy}(n-1) + Y(n)\, Y(n)^T \qquad (2.200)$$

$$R_{xy}(n) = \lambda\, R_{xy}(n-1) + X(n)\, Y(n)^T \qquad (2.201)$$

The $(M+1) \times (M+1)$ correlation matrix is given by

$$\Phi(n) = \sum_{i=1}^{n} \lambda^{n-i}\, Y(i)\, Y^T(i) \qquad (2.202)$$

where the forgetting factor λ is a positive constant close to, but smaller than, one . When $\lambda < 1$, the weighting factors in (3.26), (3.27) and (3.28) give more weigh to the recent samples of the error estimate .

- **With some elaborate mathematical treatment, the recursive relation for filter coefficients (tap-weights) is given by**

$$W(n) = W(n-1) + k(n)\, P(n)\, Y(n)\, \xi(n) \qquad (2.203)$$

where $\xi(n) = x_n - W^T(n-1)\, Y(n)$ **(2.204)**

and $P(n) = \lambda^{-1} P(n-1) - \lambda^{-1} k(n)\, Y^T(n)\, P(n-1)$ **(2.205)**

$$k(n) = \lambda^{-1} P(n-1)\, Y(n) \Big/ \left[1 + \lambda^{-1} Y^T(n)\, P(n-1)\, Y(n)\right] \qquad (2.206)$$

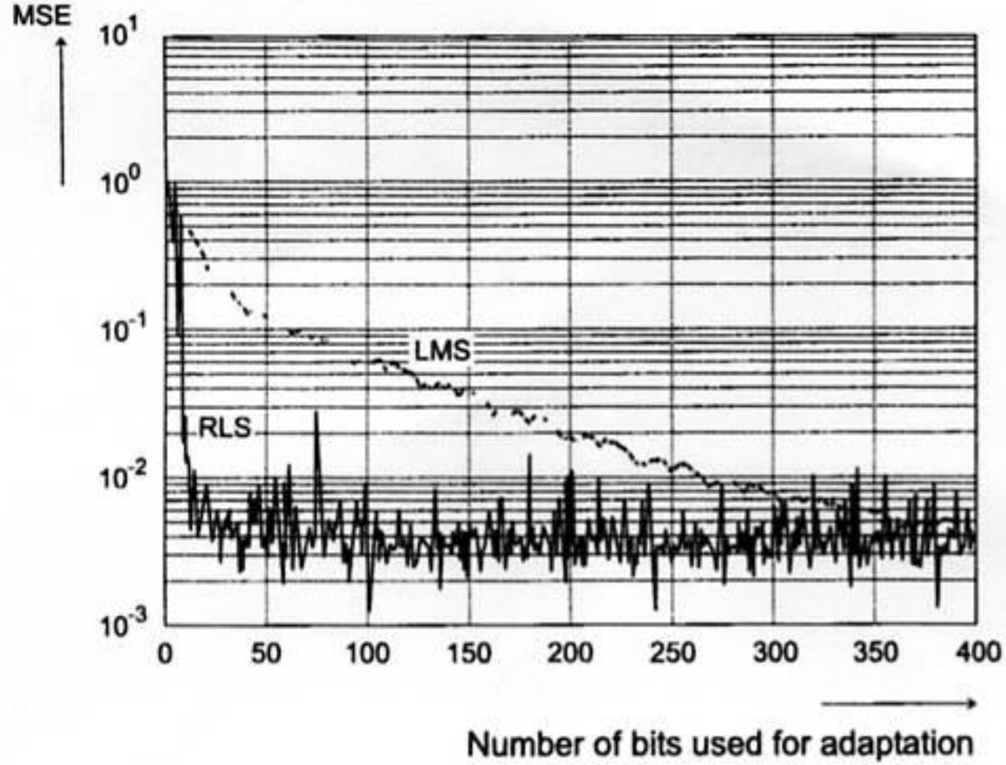
- **Initialization of the RLS algorithm:**

The applicability of the RLS algorithm requires that we initiate the recursion of Equation (3.26) by choosing a starting value

P(0) that assures the nonsingularity of the correlation matrix $\Phi(n)$:

$$P(0) = \delta^{-1} I \;, \quad \delta = \text{a small positive constant}$$

$$W(0) = 0$$



Mean-square error as a function of the number of iterations for a decision feedback equalizer
For least mean square: $\mu = 0.03$; for recursive least squares: $\lambda = 0.99, \delta = 10^{-9}$.

| — ⏞ ⏟

Σ Π Ω Γ Φ Λ Χ Θ Δ a^2 x^2

α β γ κ ρ π ∘ ˋ ˙

π σ ψ ε ρ ξ ζ… ˆ ˜ ∴ ≒ ⊥ ∞

η τ ω μ λ ν δ ∮ ∫

≠ ≦ ≧ E_s ∫ ± — ^ ^

ln ≒ ÷ ∩ ∪ ⊥

∼ √ → ← ∗ ▽ ‖ ‖

★ ⊕ § ⁀ ↓ ↑ #