Statistical Digital Signal Processing

Week 13 Adaptive Filtering: FIR Adaptive Filter

Lecturer: ***Zelalem Hailu (Assistant Prof.)***

Addis Ababa Science and Technology University,
Addis Ababa, Ethiopia

Previous Topic (Week-12)

Parametric Spectrum Estimation

- Basics of Parametric Spectrum Estimation
- Parametric Spectrum Estimation Steps
- Autoregressive (AR) Spectrum Estimation
- Moving Average (MA) Spectrum Estimation
- Autoregressive Moving Average (ARMA) Spectrum Estimation

Lecture Learning Outcomes

1. Explain the fundamental concepts and applications of adaptive filtering, including the need for adaptive systems in nonstationary signal processing environments
2. Analyze the structure and operation of FIR adaptive filters, and formulate the adaptive filtering problem using performance criteria such as the mean square error (MSE)
3. Describe and apply the Steepest Descent algorithm for adaptive filter coefficient optimization
4. Develop and evaluate the Least Mean Squares (LMS) algorithm, and assess the effects of step size, convergence rate, and stability on adaptive filter performance
5. Design and analyze adaptive linear predictors using LMS-based adaptation

Week 13: FIR Adaptive Filter

Outline

- Adaptive Filtering: Introduction
- FIR Adaptive Filter
- The Steepest Descent Adaptive Filter
- The LMS Algorithm
- Adaptive Linear Prediction

Adaptive Filtering: Introduction

- In different problems we have seen so far such as signal modeling, Wiener filtering, and spectrum estimation with an assumption of stationary random signals
- However, the signals associated with most applications are non-stationary random signals
- Therefore, the techniques we have seen so far would not be appropriate for the problem involving non-stationary signals
- One way of handling such problem is to process these non-stationary process in blocks over interval for which the process is approximately stationary
- However, This approach has limitation in effectiveness due to:
  - For rapidly varying process, the interval over which the signal assumed to be stationary will be too short to ensure good accuracy
  - Imposing of incorrect model on the data such as piecewise stationarity

Adaptive Filtering: Introduction

- **Better Approach:** Start by assuming the signal is nonstationary from the beginning
- As a starting point for the new approach, let’s consider the FIR Wiener filtering problem with an assumption of non-stationary process:

$$\hat{d}(n) = \sum_{k=0}^{p} w(k)x(n-k) \qquad (1)$$

- Recalling the previous lecture, if $d(n)$ and $x(n)$ are jointly wide-sense stationary process, the estimation error is given by:

$$e(n) = d(n) - \hat{d}(n) \qquad (2)$$

- Then the filter coefficient which minimize the mean square error, $E\ |e(n)|^2$ can be found by solving the following Wiener-Hopf equations:

$$\mathbf{R}_x\mathbf{w} = \mathbf{r}_{dx} \qquad (3)$$

Adaptive Filtering: Introduction

- However, for the context that $d(n)$ and $x(n)$ are non-stationary, the filter coefficients that minimize $E\ |e(n)|^2$ will depend on n
- The filter will also be **shit varying (Adaptive)** filter and $\hat{d}(n)$ will be given by:

$$\hat{d}(n) = \sum_{k=0}^{p} w_n(k)x(n-k) \qquad (4)$$

Where: $w_n(k)$ is the value of the k^{th} filter coefficient at time n

- Writing eq(4) in vector notation:

$$\hat{d}(n) = \mathbf{w}_n^T \mathbf{x}(n) \qquad (5)$$

Where: $\mathbf{w}_n = \left[w_n(0), w_n(1), \ldots, w_n(p) \right]^T$

$$\mathbf{x}(n) = \left[x(n), x(n-1), \ldots, x(n-p) \right]^T$$

Adaptive Filtering: Introduction

- In adaptive filter design, it is required to find the set of optimum filter coefficients $w_n(k)$ for $k = 0,1,\ldots,p$ and for each n which makes the design challenging
- However, the design may be simplified if we relax $\mathbf{w}_n$ and use coefficient update equation having a form [1]:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \Delta\mathbf{w}_n \qquad (6)$$

- $\Delta\mathbf{w}_n$ is the ***correction*** applied to $\mathbf{w}_n$ at time n to set a new coefficients $\mathbf{w}_{n+1}$ at time $n+1$
- The update equation given on eq(6) is the heart of adaptive filter design
- The design of adaptive filter also involves defining how the correction, $\Delta\mathbf{w}_n$, is to be formed as shown in the next figure, figure(1)

Adaptive Filtering: Introduction

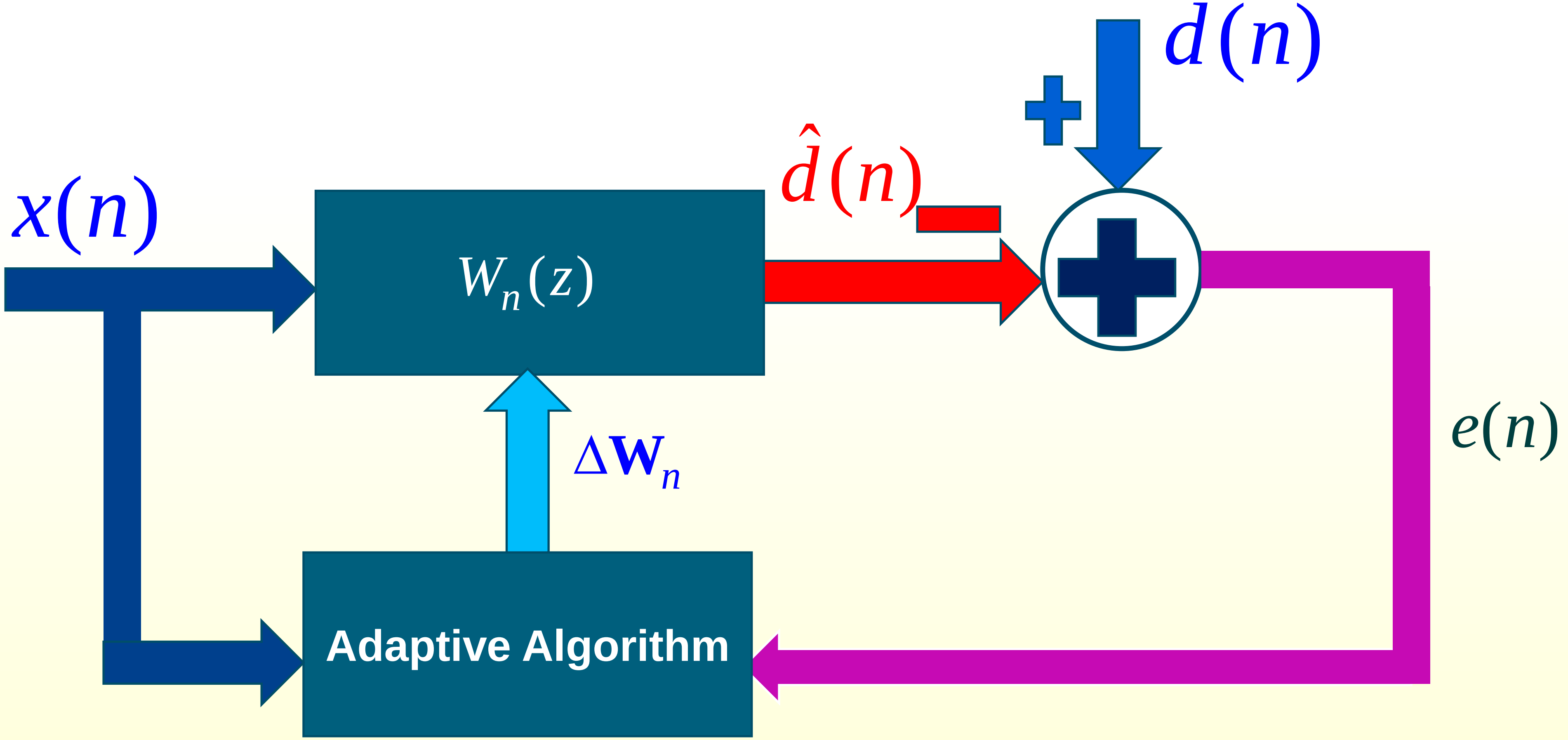

Figure 1: Illustration of adaptive filtering

Adaptive Filtering: Introduction

- The key component in adaptive filter is the algorithm which defines the set of rules that the correction $\Delta\mathbf{w}_n$ is formed
- Adaptive filter should have the following important characteristics:

  I. In stationary context, the adaptive filter should produce correction, $\Delta\mathbf{w}_n$, that can converge $\mathbf{w}_n$ to the solution of Wiener-Hopf equation as:

$$\lim_{n\to\infty} \mathbf{w}_n = \mathbf{R}_x^{-1}\mathbf{r}_{dx} \qquad (7)$$

  II. The estimation of the signal statistics, $r_x(k)$ and $r_{dx}(k)$, should be built into the adaptive filter. Hence, it is not necessary to know those statistics

  III. The filter should adapt the change in statistics and track the solution as it evolves in time for the case of non-stationary signal

FIR Adaptive Filter

- FIR filters are commonly used in adaptive filtering applications due to:
  - Capability of easily controlling stability by ensuring that the filter coefficients are bounded
  - Prescence of simple and efficient algorithms to adjust the filter coefficients
  - The convergence and stability performance of those algorithms are well known
  - FIR adaptive filters performance often well enough to satisfy the design criteria
- As illustrated in the next figure, figure (2), the estimation of FIR adaptive filter for estimating $d(n)$ from related signal $x(n)$ is :

$$\hat{d}(n) = \sum_{k=0}^{p} w_n(k)x(n-k) = \mathbf{w}_n^T \mathbf{x}(n) \qquad (8)$$

FIR Adaptive Filter

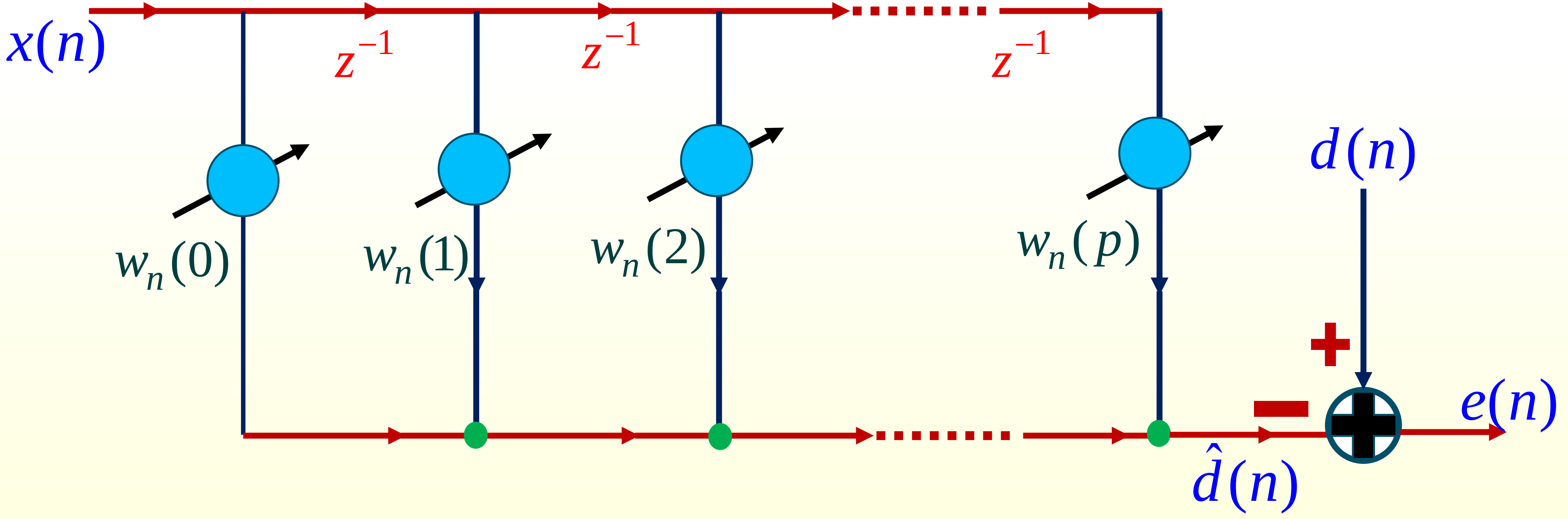

Figure 2: FIR adaptive filter

FIR Adaptive Filter

- Assuming $x(n)$ and $d(n)$ are non stationary process, our goal is to find $\mathbf{w}_n$ at time n which minimizes the mean square error:

$$\xi(n) = E\ |e(n)|^2 \qquad (9)$$

Where:

$$e(n) = d(n) - \hat{d}(n) = d(n) - \sum_{l=0}^{p} w_n(l)x(n-l) = d(n) - \mathbf{w}_n^T \mathbf{x}(n) \qquad (10)$$

- Similar to what we did for FIR Wiener filter, the minimization problem for eq(9) can be done by taking the partial derivative of $\xi(n)$ with respect to $w_n^*(k)$ as:

$$\frac{\partial \xi(n)}{\partial w_n^*(k)} = E\ e(n)x^*(n-k)\ = 0\ \ ;\ k = 0,1,\ldots,p \qquad (11)$$

FIR Adaptive Filter

- Substituting eq(10) in eq(11) for $e(n)$:

$$E\left\{\left[d(n)-\sum_{k=0}^{p} w_n(l)x(n-l)\right]x^*(n-k)\right\}=0 \quad ; k=0,1,2,\dots,p \qquad (12)$$

- Eq(12) can be rearranged as:

$$\sum_{l=0}^{p} w_n(l)E\ \ x(n-l)x^*(n-k)\ \ = E\ \ d(n)x^*(n-k) \quad ; k=0,1,2,\dots,p \qquad (13)$$

- Unlike FIR Wiener filter, the solution of Eq(13) depends on n
- Eq(13) can also be written in vector notation as:

$$\mathbf{R}_x(n)\mathbf{w}_n=\mathbf{r}_{dx}(n) \qquad (14)$$

FIR Adaptive Filter

- The $\mathbf{R}_x(n)$ in eq(14) is the following a $(p+1)\times(p+1)$ Hermitian matrix of autocorrelation:

$$\mathbf{R}_x(n) = \begin{bmatrix} E\{x(n)x^*(n)\} & E\{x(n-1)x^*(n)\} & \cdots & E\{x(n-p)x^*(n)\} \\ E\{x(n)x^*(n-1)\} & E\{x(n-1)x^*(n-1)\} & \cdots & E\{x(n-p)x^*(n-1)\} \\ \vdots & \vdots & \cdots & \vdots \\ E\{x(n)x^*(n-p)\} & E\{x(n-1)x^*(n-p)\} & \cdots & E\{x(n-p)x^*(n-p)\} \end{bmatrix} \quad (15)$$

- And $\mathbf{r}_{dx}(n)$:

$$\mathbf{r}_{dx}(n) = \left[E\{d(n)x^*(n)\}, E\{d(n)x^*(n-1)\}, \ldots, E\{d(n)x^*(n-p)\}\right]^T \quad (16)$$

The Steepest Descent Adaptive Filter

- While designing FIR adaptive filter, the goal is to find $\mathbf{w}_n$ at time n which minimizes the quadratic function $\xi(n) = E\,|e(n)|^2$
- As we have seen previously, $\mathbf{w}_n$ which minimizes $\xi(n)$ can be found by setting the partial derivative of $\xi(n)$ with respect to $w_n^*(k)$ equals to zero
- However, there is another iterative approach called ***Steepest Descent Method*** to search the solution
- The steepest descent method works as follows:

Let $\mathbf{w}_n$ be an estimate of the vector which minimizes $\xi(n)$ at time n

⬇

At time $n+1$ a new estimate will be formed by adding a correction factor on $\mathbf{w}_n$ that is designed to bring $\mathbf{w}_n$ closer to the **desired solution**

The Steepest Descent Adaptive Filter

- The correction is made by moving a step size μ in the direction of maximum descent on the quadratic error surface
- Let's consider the following quadratic function $\xi(n)$ for two real valued coefficients $w(0)$ and $w(1)$

$$\xi(n) = 6 - 6w(0) - 4w(1) + 6\left[w^2(0) + w^2(1)\right] + 6w(0)w(1) \qquad (17)$$

- The three dimensional plot of eq(17) is given on figure (3a)
- As shown also on figure (3b), the contour of constant error, when projected on $w(0) - w(1)$ plane, it forms a set of concentric ellipses
- The direction of steepest descent at any point corresponds to the path a marble would naturally follow if placed on the surface of the quadratic bowl

The Steepest Descent Adaptive Filter

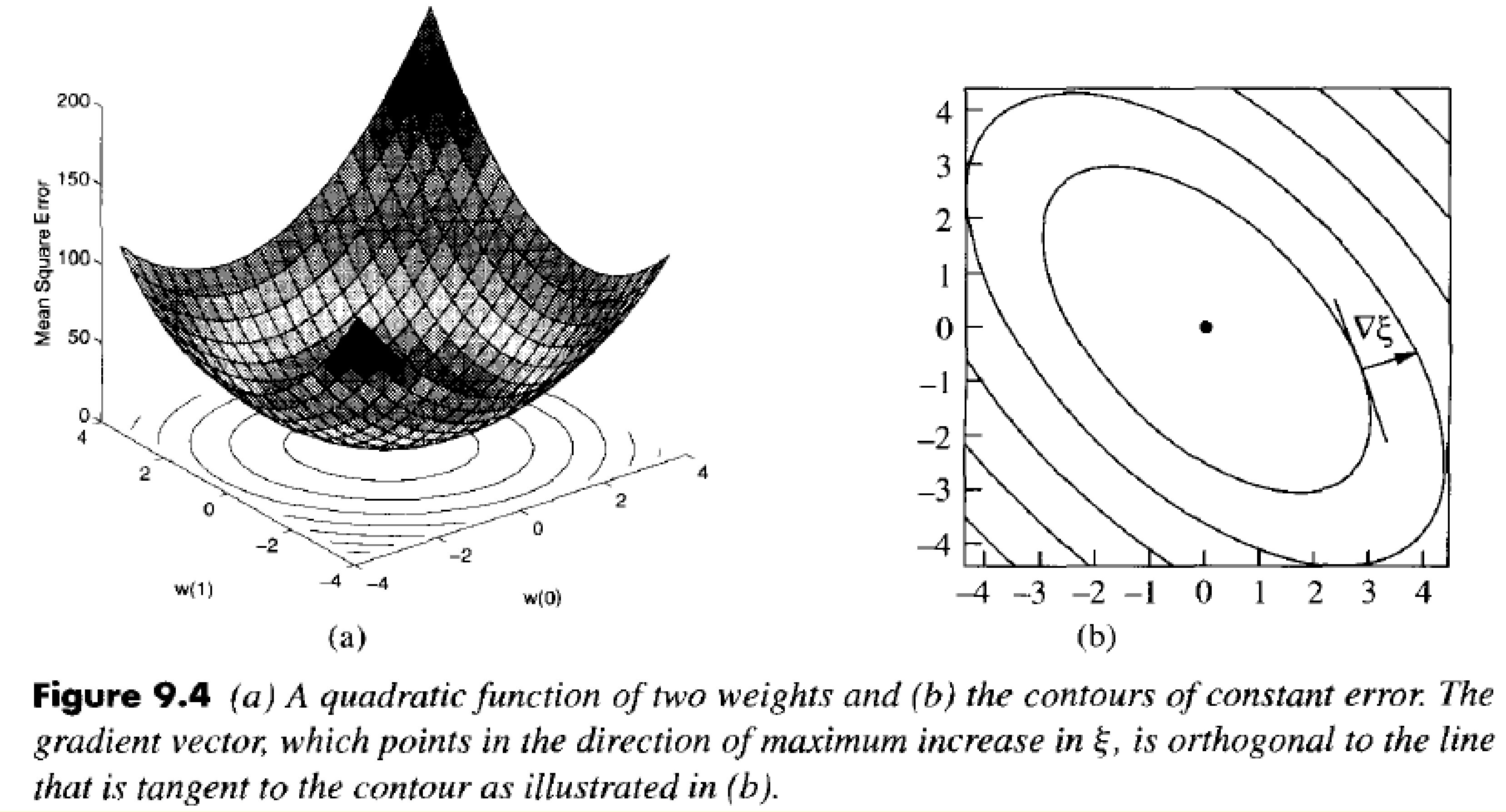

Figure 3: (a) Quadratic function of the two weights (b) the contours of constant error

Source: “Adaptive Filtering”, CSDN. https://i-blog.csdnimg.cn/blog_migrate/1b968b18828b1955c3dbb37083d639f1.png#pic_center

The Steepest Descent Adaptive Filter

- The direction of the steepest decent can be expressed mathematically by the **Gradient**
- The gradient is the vector of partial derivative of $\xi(n)$ with respect to the coefficient $w(k)$
- For the quadratic function given on eq(17), the gradient becomes:

$$\nabla\xi(n) = \begin{bmatrix} \dfrac{\partial\xi(n)}{\partial w(0)} \\ \dfrac{\partial\xi(n)}{\partial w(1)} \end{bmatrix} = \begin{bmatrix} 12w(0) + 6w(1) - 6 \\ 12w(1) + 6w(0) - 4 \end{bmatrix} \qquad (18)$$

- As illustrated on figure (3b), the gradient is orthogonal to the tangent line to the contour of constant error at any vector $\mathbf{w}$
- However, the gradient vector points in the direction of steepest ascent
- Therefore, the direction of steepest decent points in the negative gradient direction

The Steepest Descent Adaptive Filter

- Then, the update equation for $\mathbf{w}_n$ becomes:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \nabla \xi(n) \qquad (19)$$

Where: μ is the step size

- The step size, μ , determines the rate at which the weight vector moves down the quadratic surface
- And μ should be positive number, otherwise, the weight vector will move in the direction of maximum ascent and would results in error increment
- From eq(19), it is also noted that, the choice of very small value of μ , the correction to $\mathbf{w}_n$ would very small and the movement down the quadratic surface becomes slow

The Steepest Descent Adaptive Filter

- On the contrary , for large μ the rate of decent increases
- But, there is an upper limit on how large the step size should be and if the step size is beyond that upper limit, the trajectory of $\mathbf{w}_n$ will be come unstable
- In general, the step of steepest decent algorithm can be summarized as follows:

1. Initialize the steepest decent algorithm with initial estimate $\mathbf{w}_0$
2. Evaluate $\nabla \xi(n)$ at the current estimate $\mathbf{w}_n$
3. Update the estimate at time n by adding a correction that is formed by taking a step size of μ in the negative gradient direction as:

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mu \nabla \xi(n)$$

4. Go back to (2) and repeat the process

The Steepest Descent Adaptive Filter

- If we assume $\mathbf{w}$ is complex, the gradient, $\nabla\xi(n)$, becomes the partial derivative of $E\left\{|e(n)|^2\right\}$ with respect to $\mathbf{w}^*$
- Hence,

$$\nabla\xi(n) = \nabla E\left\{|e(n)|^2\right\} = E\left\{\nabla|e(n)|^2\right\}$$
$$= E\left\{e(n)\nabla e^*(n)\right\}$$
$$= -E\left\{e(n)\mathbf{x}^*(n)\right\} \qquad (20)$$

- Then the weight update equation of the steepest decent algorithm becomes:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu E\left\{e(n)\mathbf{x}^*(n)\right\} \qquad (21)$$

The LMS Algorithm

- Recalled from the previous equation, the weight update equation of the steepest decent algorithm is:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu E\ \ e(n)\mathbf{x}^*(n) \qquad (22)$$

Unknown Term

- However, in practical scenario, the term $E\ \ e(n)\mathbf{x}^*(n)$ is generally unknown and this puts a practical limitation
- Therefore, $E\ \ e(n)\mathbf{x}^*(n)$ should be replaced by its estimate such as sample mean as follows:

$$\hat{E}\ \ e(n)\mathbf{x}^*(n) = \frac{1}{L}\sum_{l=0}^{L-1} e(n-l)\mathbf{x}^*(n-l) \qquad (23)$$

The LMS Algorithm

- Incorporating eq(23) in eq(22), the steepest decent weight update equation becomes:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu E\ \ e(n)\mathbf{x}^*(n)$$

$$= \mathbf{w}_n + \mu\left[\frac{1}{L}\sum_{l=0}^{L-1} e(n-l)\mathbf{x}^*(n-l)\right]$$

$$= \mathbf{w}_n + \frac{\mu}{L}\sum_{l=0}^{L-1} e(n-l)\mathbf{x}^*(n-l) \qquad (24)$$

- As special case, if we do one point sample mean ($L = 1$), eq(23) turns out to be :

$$\hat{E}\ \ e(n)\mathbf{x}^*(n)\ \ = e(n)\mathbf{x}^*(n) \qquad (25)$$

The LMS Algorithm

- Similarly, eq(24) will be simplified to:

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu e(n)\mathbf{x}^*(n) \qquad (26)$$

- Eq(26) is known as **LMS Algorithm**
- The updates of kth coefficient using LMS algorithm is given by:

$$w_{n+1}(k) = w_n(k) + \mu e(n)x^*(n-k) \qquad (27)$$

- The term $\mu e(n)$ in eq(27) can be computed once and the update of kth coefficient has only one addition and one multiplication which shows the simplicity of LMS Algorithm
- In general, the LMS adaptive filter having p+1 coefficients requires p+1 additions and p+1 multiplications to update the coefficients

The LMS Algorithm

- Furthermore, one addition is necessary to compute the error $e(n) = d(n) - y(n)$ and one multiplication is needed to form the product $\mu e(n)$
- Finally, $p + 1$ multiplications and p additions are necessary to calculate the output, $y(n)$, of the adaptive filter
- Therefore, a total of $2p + 3$ multiplications and $2p + 2$ additions per output point are required

LMS Algorithm

- Parameters: $\rightarrow$ $p = \text{ filter order}, \quad \mu = \text{step size}$
- Initialization: $\rightarrow$ $\mathbf{w}_0 = 0$
- Computation: $\rightarrow$ $\text{for } n = 0, 1, 2, \ldots$

$$(a)\ y(n) = \mathbf{w}_n^T \mathbf{x}(n)$$

$$(b)\ e(n) = d(n) - y(n)$$

$$(c)\ \mathbf{w}_{n+1} = \mathbf{w}_n + \mu e(n) \mathbf{x}^*(n)$$

Adaptive Linear Prediction Using LMS Algorithm

- Let's consider the following example to analyze adaptive linear prediction application using LMS algorithm
- Let $x(n)$ is AR(2) process given by:

$$x(n) = 1.2728x(n-1) - 0.81x(n-2) + v(n) \qquad (28)$$

Where: $v(n)$ is unit variance white noise

- From optimum filters topic, the optimum casual linear predictor for $x(n)$ is:

$$\hat{x}(n) = 1.2728x(n-1) - 0.8x(n-2) \qquad (29)$$

- However, in order to design this predictor (i.e., to know that the optimum predictor coefficients are 1.2728 and −0.81) it is necessary to know the autocorrelation sequence of x(n)

Adaptive Linear Prediction Using LMS Algorithm

- Therefore, suppose we consider an adaptive linear predictor of the form:

$$x(n) = w_n(1)x(n-1) - w_n(2)x(n-2) \qquad (30)$$

- From eq(27), the LMS algorithm will update $w_n(k)$ as follows:

$$w_{n+1}(k) = w_n(k) + \mu e(n)x^*(n-k) \qquad (31)$$

- If the step size μ is sufficiently small, then the coefficients $w_n(1)$ and $w_n(2)$ will converge in the mean to their optimum values, which are

$$w_n(1) = 1.2728 \qquad (32)$$

$$w_n(2) = -0.81 \qquad (33)$$

Adaptive Linear Prediction Using LMS Algorithm

- The prediction error is also given by:

$$e(n) = x(n) - \hat{x}(n) = \; 1.2728 - w_n(1) \;\; x(n-1) + \;\; -0.81 - w_n(2) \;\; x(n-2) + v(n) \quad (34)$$

- Therefore, when $w_n(1) = 1.2728$ and $w_n(2) = -0.81$, the error becomes:

$$e(n) = v(n) \qquad (35)$$

- Then the minimum mean square prediction error becomes:

$$\xi_{\min} = \sigma_v^2 = 1 \qquad (36)$$

Summary

- **Adaptive Filtering:**
  - ✓ Applicable for non stationary signal estimation and prediction
  - ✓ Wiener Filtering is not appropriate for non stationary signal context
- **FIR Adaptive Filters**
  - ✓ Non recessive filters and feed forward type
- **FIR Adaptive Filter Algorithms**
  - ✓ Steepest Decent Algorithm
  - ✓ LMS algorithm

References

[1] Monson H. Hayes, “Statistical Digital Signal Processing and Modeling’’, John Wiley and sons, Pp.494, 1996.

Contents Here

Thank You!