# Adaptive Filtering: Part II

Dr. Deepa Kundur

University of Toronto





## Discrete-Time Signals and Systems

**Reference:**

Section 13.2 of

John G. Proakis and Dimitris G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*, 4th edition, 2007.





## Background

Recall, the Wiener-Hopf equation:

$$\sum_{k=0}^{M-1} h(k)\gamma_{xx}(l-k) = \gamma_{dx}(l)$$

- There are many ways to determine the solution to the Wiener-Hopf equation.
- Our focus: recursive methods using gradient algorithms.





## Matrix-Vector Notation

**Q:** Why matrix-vector notation?

**A:** Because it gives a compact representation that enables better insight.

$$\begin{aligned}\sum_{k=0}^{M-1} h(k)\gamma_{xx}(l-k) &= \gamma_{dx}(l), \quad l = 0, 1, \ldots, M-1 \\ \mathbf{\Gamma}_M \mathbf{h}_M &= \boldsymbol{\gamma}_d\end{aligned}$$

# Matrix-Vector Notation

Convention:

matrix ≡ **CAPITAL BOLDFACE**
vector ≡ **lowercase boldface**

$\mathbf{a}_M \equiv M$-dimensional column vector
$\mathbf{A}_M \equiv M \times M$-dimensional matrix

$\mathbf{a}^t{}_M \equiv$ transpose of $\mathbf{a}_M$
$\mathbf{a}^*{}_M \equiv$ complex conjugate of $\mathbf{a}_M$
$\mathbf{a}^H{}_M \equiv$ conjugate transpose of $\mathbf{a}_M$





ASIDE:

$$\sum_{k=0}^{M-1} a(k)b(l-k) = c(l), \quad l = 0, 1, \ldots, M-1$$

$$\begin{aligned}
\sum_{k=0}^{M-1} a(k)b(0-k) &= c(0) \\
\sum_{k=0}^{M-1} a(k)b(1-k) &= c(1) \\
&\vdots \\
\sum_{k=0}^{M-1} a(k)b(M-1-k) &= c(M-1)
\end{aligned}$$





ASIDE (cont'd):

Let $\mathbf{c}_M = [c(0)\ c(1)\ \cdots\ c(M-1)]^t$.
Notice:

$$\begin{aligned}
c(l) &= \sum_{k=0}^{M-1} a(k)b(l-k) = [b(l)\ b(l-1) \cdots b(l-(M-1))] \begin{bmatrix} a(0) \\ a(1) \\ \cdots \\ a(M-1) \end{bmatrix} \\
&= a(0)b(l) + a(1)b(l-1) + \cdots + a(M-1)b(l-(M-1))
\end{aligned}$$

$$\mathbf{c}_M = \begin{bmatrix} c(0) \\ c(1) \\ \vdots \\ c(M-1) \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{M-1} a(k)b(0-k) \\ \sum_{k=0}^{M-1} a(k)b(1-k) \\ \vdots \\ \sum_{k=0}^{M-1} a(k)b(M-1-k) \end{bmatrix}$$





ASIDE (cont'd):

$$\begin{aligned}
\mathbf{c}_M &= \begin{bmatrix} c(0) \\ c(1) \\ \vdots \\ c(M-1) \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{M-1} a(k)b(0-k) \\ \sum_{k=0}^{M-1} a(k)b(1-k) \\ \vdots \\ \sum_{k=0}^{M-1} a(k)b(M-1-k) \end{bmatrix} \\
&= \begin{bmatrix} b(0) & b(0-1) & \cdots & b(0-(M-1)) \\ b(1) & b(1-1) & \cdots & b(1-(M-1)) \\ & & \vdots & \\ b(M-1) & b(M-1-1) & \cdots & b(M-1-(M-1)) \end{bmatrix} \begin{bmatrix} a(0) \\ a(1) \\ \vdots \\ a(M-1) \end{bmatrix} \\
&= \underbrace{\begin{bmatrix} b(0) & b(-1) & \cdots & b(-M+1) \\ b(1) & b(0) & \cdots & b(-M+2) \\ & & \vdots & \\ b(M-1) & b(M-2) & \cdots & b(0) \end{bmatrix}}_{=\mathbf{B}_M} \underbrace{\begin{bmatrix} a(0) \\ a(1) \\ \vdots \\ a(M-1) \end{bmatrix}}_{=\mathbf{a}_M} \\
&= \mathbf{B}_M\ \mathbf{a}_M
\end{aligned}$$

ASIDE (cont'd):

Therefore,

$$\sum_{k=0}^{M-1} a(k)b(l-k) = c(l), \quad l = 0, 1, \ldots, M-1$$

is equivalent to

$$\mathbf{c}_M = \mathbf{B}_M \, \mathbf{a}_M$$





ASIDE (cont'd):

Consider

$$\mathbf{B}_5 = \begin{bmatrix} b(0) & b(-1) & b(-2) & b(-3) & b(-4) \\ b(1) & b(0) & b(-1) & b(-2) & b(-3) \\ b(2) & b(1) & b(0) & b(-1) & b(-2) \\ b(3) & b(2) & b(1) & b(0) & b(-1) \\ b(4) & b(3) & b(2) & b(1) & b(0) \end{bmatrix}$$

Recall,

- Toeplitz matrix: each descending diagonal from left to right is constant.
- $\mathbf{B}_M$ is a Toeplitz matrix.





ASIDE (cont'd):

In our adaptive filtering case, we have

$$\mathbf{\Gamma}_M \mathbf{h}_M = \gamma_d$$

where the matrix $\mathbf{\Gamma}_M$ for $M = 5$ is given by

$$\mathbf{\Gamma}_5 = \begin{bmatrix} \gamma_{xx}(0) & \gamma_{xx}(-1) & \gamma_{xx}(-2) & \gamma_{xx}(-3) & \gamma_{xx}(-4) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(-1) & \gamma_{xx}(-2) & \gamma_{xx}(-3) \\ \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(-1) & \gamma_{xx}(-2) \\ \gamma_{xx}(3) & \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(-1) \\ \gamma_{xx}(4) & \gamma_{xx}(3) & \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) \end{bmatrix}$$





ASIDE (cont'd):

Recall,

$$\begin{aligned} \gamma_{xx}(m) &= E[x(n)x^*(n-m)] \\ &= E[x(n+D)x^*(n-m+D)] \quad \text{due to stationarity of } x(n) \\ &= E[x^*(n-m+D)x(n+D)] \\ &= E[x^*(n)x(n+m)] \quad \text{for } D = m \\ &= E[(x(n)x^*(n+m))^*] \\ &= (E[x(n)x^*(n+m)])^* \\ &= (\gamma_{xx}(-m))^* = \gamma_{xx}^*(-m) \end{aligned}$$

Therefore,

$$\boxed{\gamma_{xx}(m) = \gamma_{xx}^*(-m)}$$

ASIDE (cont'd):
Therefore,

$$\mathbf{\Gamma}_5 = \begin{bmatrix} \gamma_{xx}(0) & \gamma_{xx}(-1) & \gamma_{xx}(-2) & \gamma_{xx}(-3) & \gamma_{xx}(-4) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(-1) & \gamma_{xx}(-2) & \gamma_{xx}(-3) \\ \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(-1) & \gamma_{xx}(-2) \\ \gamma_{xx}(3) & \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma_{xx}(-1) \\ \gamma_{xx}(4) & \gamma_{xx}(3) & \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) \end{bmatrix}$$

$$= \begin{bmatrix} \gamma_{xx}(0) & \gamma^*{}_{xx}(1) & \gamma^*{}_{xx}(2) & \gamma^*{}_{xx}(3) & \gamma^*{}_{xx}(4) \\ \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma^*{}_{xx}(1) & \gamma^*{}_{xx}(2) & \gamma^*{}_{xx}(3) \\ \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma^*{}_{xx}(1) & \gamma^*{}_{xx}(2) \\ \gamma_{xx}(3) & \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) & \gamma^*{}_{xx}(1) \\ \gamma_{xx}(4) & \gamma_{xx}(3) & \gamma_{xx}(2) & \gamma_{xx}(1) & \gamma_{xx}(0) \end{bmatrix}$$

Note: $\mathbf{\Gamma}_5^t = \mathbf{\Gamma}_5^*$ and more generally $\mathbf{\Gamma}_M^t = \mathbf{\Gamma}_M^*$.
Therefore, $\mathbf{\Gamma}_M$ is a Hermitian autocorrelation matrix.





# Wiener-Hopf Equations

Thus, the Wiener-Hopf equations can be represented as:

$$\mathbf{\Gamma}_M \mathbf{h}_M = \gamma_d$$

where

- $\mathbf{h}_M$ denotes the vector of adaptive filter coefficients
- $\gamma_d$ is an $M \times 1$ crosscorrelation vector
- $\mathbf{\Gamma}_M$ is an $M \times M$ Hermitian autocorrelation matrix





# Wiener-Hopf Equations

Therefore, the solution for the optimum filter coefficients can be obtained as:

$$\begin{aligned} \mathbf{\Gamma}_M \mathbf{h}_M &= \gamma_d \\ \mathbf{h}_{opt} &= \mathbf{\Gamma}_M^{-1} \gamma_d \end{aligned}$$





# The LMS Algorithm

- Solution to the Wiener-Hopf equations can be conducted in numerous ways.
- Our focus: recursive methods that determine minimum of a function of several variables.
- Good news:
  - Our optimization function is the MSE $\mathcal{E}_M$ that is convex in $\mathbf{h}_M$.
  - There is a unique minimum, which can be found through iterative search.

# The LMS Algorithm

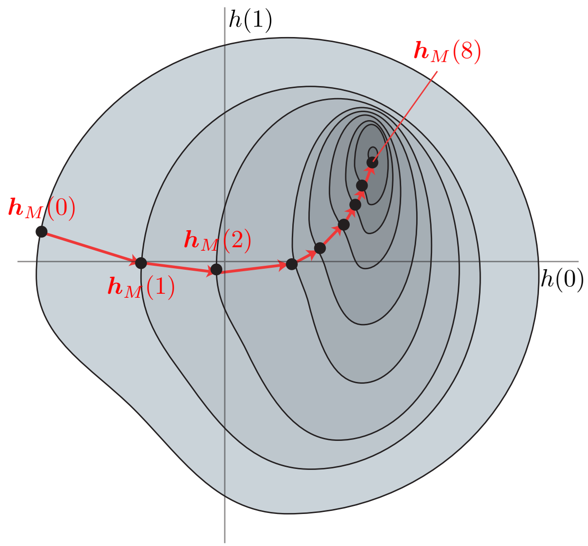
- Assume that $\mathbf{\Gamma}_M$ and $\gamma_d$ are known.
- $\mathcal{E}_M$ is a known function of coefficients $\mathbf{h}_M$.
- Recursive algorithms in search of the minimum of $\mathcal{E}_M$ have the form:

$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \frac{1}{2}\Delta(n)\mathbf{S}(n)$$

where
  - $n = \{0, 1, 2, \ldots\}$: iteration number
  - $\mathbf{h}_M(n)$: vector of filter coefficients at the $n$th iteration
  - $\Delta(n)$: scalar step size at the $n$th iteration
  - $\mathbf{S}(n)$: direction vector for adaptation at the $n$th iteration








- $M = 2$
- contour lines shown for $\mathcal{E}_M$ constant
- darker shade $\equiv$ smaller $\mathcal{E}_M$
- vectors represent $\frac{1}{2}\Delta(n)\mathbf{S}(n)$





# Steepest Descent Approach

Update law for iteration $n+1$:

$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \frac{1}{2}\Delta(n)\mathbf{S}(n)$$

- $\mathbf{S}(n)$ selected to be the negative of the gradient vector:

$$\begin{aligned}\mathbf{S}(n) &= -\frac{d\mathcal{E}_M}{d\mathbf{h}_M(n)} = -\left[\frac{d\mathcal{E}_\mathcal{M}}{dh_n(0)}\ \frac{d\mathcal{E}_\mathcal{M}}{dh_n(1)}\ \cdots\ \frac{d\mathcal{E}_\mathcal{M}}{dh_n(M-1)}\right]^t \\ &= -2\left[\mathbf{\Gamma}_M\mathbf{h}_M(n) - \gamma_d\right]\end{aligned}$$





# Steepest Descent Approach

Thus, the steepest descent approach leads to the following update law for $n = 0, 1, 2, \ldots$:

$$\begin{aligned}\mathbf{h}_M(n+1) &= \mathbf{h}_M(n) + \frac{1}{2}\Delta(n)\mathbf{S}(n) \\ &= \mathbf{h}_M(n) - \frac{1}{2}\Delta(n)2\left[\mathbf{\Gamma}_M\mathbf{h}_M(n) - \gamma_d\right] \\ &= \mathbf{h}_M(n) - \Delta(n)\mathbf{\Gamma}_M\mathbf{h}_M(n) + \Delta(n)\gamma_d \\ &= \left[\mathbf{I} - \Delta(n)\mathbf{\Gamma}_M\right]\mathbf{h}_M(n) + \Delta(n)\gamma_d\end{aligned}$$

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

$$\mathbf{h}_M(n+1) = [\mathbf{I} - \Delta(n)\mathbf{\Gamma}_M]\,\mathbf{h}_M(n) + \Delta(n)\boldsymbol{\gamma}_d$$

- $\mathbf{\Gamma}_M$: $M \times M$ autocorrelation matrix
- $\boldsymbol{\gamma}_d$: $M \times 1$ crosscorrelation vector
- Note: convergence $\mathbf{h}_M(n) \to \mathbf{h}_{opt}$ as $n \to \infty$ possible provided $\sum_{n=0}^{\infty} |\Delta(n)| < \infty$ and $\Delta(n) \to 0$ for $n \to \infty$.

However, $\mathbf{\Gamma}_M$ and $\boldsymbol{\gamma}_d$ are unknown.

We will estimate them from the data!





Note:

$$\begin{aligned}
\frac{d\mathcal{E}_\mathcal{M}}{d\mathbf{h}_M(n)} &= 2\left[\mathbf{\Gamma}_M \mathbf{h}_M(n) - \boldsymbol{\gamma}_d\right] \\
&= 2\begin{bmatrix} \sum_{k=0}^{M-1} h_n(k)\gamma_{xx}(-k) - \gamma_{dx}(0) \\ \sum_{k=0}^{M-1} h_n(k)\gamma_{xx}(1-k) - \gamma_{dx}(1) \\ \vdots \\ \sum_{k=0}^{M-1} h_n(k)\gamma_{xx}(M-1-k) - \gamma_{dx}(M-1) \end{bmatrix} \\
&\approx 2\begin{bmatrix} \sum_{k=0}^{M-1} h_n(k)r_{xx}(-k) - r_{dx}(0) \\ \sum_{k=0}^{M-1} h_n(k)r_{xx}(1-k) - r_{dx}(1) \\ \vdots \\ \sum_{k=0}^{M-1} h_n(k)r_{xx}(M-1-k) - r_{dx}(M-1) \end{bmatrix} \\
&= 2\left[\sum_{k=0}^{M-1} h_n(k)x(n-k) - d(n)\right]\begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-M-1) \end{bmatrix}^*
\end{aligned}$$





Note:

$$\begin{aligned}
\frac{d\mathcal{E}_\mathcal{M}}{d\mathbf{h}_M(n)} &\approx 2\underbrace{\left[\sum_{k=0}^{M-1} h_n(k)x(n-k) - d(n)\right]}_{=-e(n)}\underbrace{\begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-M-1) \end{bmatrix}^*}_{\equiv \mathbf{X}_M^*(n)} \\
&= -2e(n)\mathbf{X}_M^*(n)
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mathbf{h}_M(n+1) &= \mathbf{h}_M(n) + \frac{1}{2}\Delta(n)\mathbf{S}(n) \\
&= \mathbf{h}_M(n) + \frac{1}{2}\Delta(n)\left[-\frac{d\mathcal{E}_\mathcal{M}}{d\mathbf{h}_M(n)}\right]
\end{aligned}$$

A practical update law is:

$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \Delta(n)e(n)\mathbf{X}_M^*(n)$$





$$\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \Delta(n)e(n)\mathbf{X}_M^*(n)$$

- The substitution of a time-average estimate for the gradient computation makes the update a stochastic-gradient-descent algorithm.
- Step size $\Delta(n)$ is typically set as a constant $\Delta$.
  - Easy to implement in software and/or hardware.
  - Allows tracking of time-varying statistics because $\Delta(n) \not\to 0$ as $n \to \infty$ (which is needed to guarantee convergence of the steepest-descent algorithm).

# The LMS Algorithm

Finally, the least-mean-squares algorithm is given by:

$$\boxed{\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \Delta e(n)\mathbf{X}_M^*(n)}$$





# Variations of the LMS Algorithm

Original LMS: $\boxed{\mathbf{h}_M(n+1) = \mathbf{h}_M(n) + \Delta\ e(n)\ \mathbf{X}_M^*(n)}$

$$\text{csgn}[x] = \begin{cases} 1+j & \text{if } \text{Re}(x) > 0 \text{ and } \text{Im}(x) > 0 \\ 1-j & \text{if } \text{Re}(x) > 0 \text{ and } \text{Im}(x) < 0 \\ -1+j & \text{if } \text{Re}(x) < 0 \text{ and } \text{Im}(x) > 0 \\ -1-j & \text{if } \text{Re}(x) < 0 \text{ and } \text{Im}(x) < 0 \end{cases}$$

$$\begin{aligned} \mathbf{h}_M(n+1) &= \mathbf{h}_M(n) + \Delta\ \text{csgn}[e(n)]\ \mathbf{X}_M^*(n) \\ \mathbf{h}_M(n+1) &= \mathbf{h}_M(n) + \Delta\ e(n)\ \text{csgn}[\mathbf{X}_M^*(n)] \\ \mathbf{h}_M(n+1) &= \mathbf{h}_M(n) + \Delta\ \text{csgn}[e(n)]\ \text{csgn}[\mathbf{X}_M^*(n)] \end{aligned}$$

■