ADAPTIVE FILTERING
OUTLINE

* APPLICATIONS OF ADAPTIVE FILTERS
* ADAPTIVE DIRECT FORM F.I.R. FILTERS (LMS, RLS algorithms)
* ADAPTIVE LATTICE - LADDER FILTERS
* PERFORMANCE COMPARISON
ADAPTIVE FILTERS

The statistical characteristics of the signals to be filtered are either unknown a priori or, slowly time variant (non-stationary signals)

* Adaptive beamforming [Widrow et al. (1967)]
* Adaptive noise canceling [Widrow et al. (1975), Hsu and Giordano (1978), Ketsum and Proakis (1982)]
* System modeling/identification, Echo cancellation, Speech coding, etc.
REFERENCES

A few key references from the extensive literature on adaptive filtering are:


* J. Proakis et al., "Advanced Digital Signal Processing" (Macmillan, 1992)

FIR FILTER for adaptive filtering

\[ \text{Input} \]

\[ 2^{-1} \quad 2^{-1} \quad 2^{-1} \quad 2^{-1} \]

\[ h(0) \quad h(1) \quad h(2) \quad h(3) \quad h(4) \]

\[ \times \quad \times \quad \times \quad \times \quad \times \]

\[ \text{Coefficient adjustment} \]

\[ \Sigma \rightarrow \text{Output} \]

**FIGURE 1** Direct-form adaptive FIR filter.

* Stability of the filter depends on coefficient adjustment algorithm.
* IIR filters suffer from stability problems more often.
* Direct form and lattice form FIR filter structures are common.
OPTIMIZATION CRITERIA

* Very important for efficient adjustment of filter coefficients

* Criterion must be a meaningful measure of filter performance and result in practically realizable algorithms

* The least-squares (LS) and mean-square error (MSE) criteria result in quadratic performance index with a single minimum. They are used widely in practice.
FIGURE 2 Application of adaptive filtering to system identification.

\[ \hat{d}(n) = \sum_{k=0}^{M-1} h(k) x(n-k) \]

* \( e(n) = y(n) - \hat{d}(n) \) is an error sequence

* Select \( \xi h(k) \) for \( k=0, ..., M-1 \) to minimize \( \sum_{n=0}^{N-1} |e(n)|^2 \) (LS criterion)
**ADAPTIVE CHANNEL EQUALIZATION**

![Diagram of adaptive channel equalization](image)

**FIGURE 3** Application of adaptive filtering to adaptive channel equalization.

\[
x(n) = \sum_{k=0}^{\infty} a(k) \cdot q(\eta-k) + w(n) = a(n) + \sum_{k=0}^{\infty} a(k) q(\eta-k) + w(n)
\]

true symbol

intersymbol interference (ISI)
Assume that the equalizer filter is an FIR filter with \( M \) adjustable coefficients \( \{ h(n) \}_{n=0}^{M-1} \).

Output of equalizer:
\[
\hat{d}(n) = \sum_{k=0}^{M-1} h(k) x(n-k)
\]

Form the error:
\[
e(n) = d(n) - \hat{d}(n)
\]

where: \( d(n) = a(n+D) \) is the desired true value and \( D \) accounts for delay in the channel.

(Note: \( d(n) = \hat{a}(n) \) after initial convergence is achieved)

Select \( \{ h(n) \}_{n=0}^{M-1} \) to minimize
\[
\sum_{n=0}^{N-1} |e(n)|^2
\]
FIGURE 3. Block diagram model of a digital communication system that used echo cancellers in the modems.

* Echoes due to impedance mismatch between Hybrid A and the channel (near-end echoes)
* Echoes due to impedance mismatch at Hybrid B (far-end echoes)
**FIGURE 4.** Symbol-rate echo canceller.

- **MODEL 4**

\[
\hat{S}_A(n) = \sum_{k=0}^{N-1} h(k) a(n-k),
\]

* **Error:** 
\[
e(n) = d(n) - [r_A(n) - \hat{S}_A(n)],
\]

Minimize 
\[
\sum_{n=0}^{N-1} |e(n)|^2
\]
SUPPRESSION OF NARROWBAND INTERFERENCE IN A WIDEBAND SIGNAL (1)

\[ |V(f)| = |X(f)| + |W(f)| \]

**Figure 5.** Strong narrowband interference \( X(f) \) in a wideband signal \( W(f) \).

\[ v(n) = x(n) + w(n) \leftarrow \text{not highly correlated.} \]

\[ \uparrow \text{Highly correlated} \quad (\text{x(n) and w(n) uncorrelated}) \]
Suppression of Narrowband Interference in a Wideband Signal (2)

\[ v(n) = w(n) + x(n) \]

\[ e(n) = \hat{w}(n) \]

\[ v(n-D) \]

\[ z^{-D} \]

\[ z^{-1} \]

\[ h(0) \]

\[ h(1) \]

\[ h(2) \]

\[ h(M-1) \]

\[ \sum_{k=0}^{M-1} h(k) v(n-D-k) \]

Form the error

\[ e(n) = v(n) - \hat{x}(n) \]

Obtain \( \{h(k)\} \) for \( k=0, \ldots, M-1 \) by minimizing the LS criterion

\[ \sum_{n=0}^{N-1} |e(n)|^2 \]

Fig. 6.
ADAPTIVE DIRECT-FORM FIR FILTERS

From previous examples we observe a common framework in adaptive filter applications.

→ Given,

* the observed (received) data samples \( x(n) \),
* an FIR digital filter with unknown coefficients \( h(n) \) \( n = 0, \ldots, M-1 \)
* A desired response for the filter \( d(n) \)

→ Form the error quantity: \( e(n) = d(n) - \sum_{k=0}^{M-1} h(k) \cdot x(n-k) \)

→ Minimize a function of \( e(n) \) with respect to the \( \{ h(k) \} \).
   (Minimization should be carried out adaptively to accommodate changing signal conditions).
MINIMUM MEAN SQUARE ERROR CRITERION (MMSE)

Minimize the MSE function:

\[ J(h_M) = \mathbb{E} \{ |e(n)|^2 \} \]

where, \( h_M = [h(0), ..., h(M-1)]^T \),

\[ e(n) = d(n) - \sum_{k=0}^{M-1} h(k) x(n-k) \]

**Solution:** (\( J(\cdot) \) is a quadratic function of \( h_M \))

\[
\left[ \begin{array}{c}
\sum_{k=0}^{M-1} h(k) R_{xx}(e-k) = r_{dx}(e) \end{array} \right] , \ e=0,1,...,M-1
\]

where:

\[ R_{xx}(m) = \mathbb{E} \{ x(n)x^*(n-m) \} \], \( r_{dx}(m) = \mathbb{E} \{ d(n)x^*(n-m) \} \]
MMSE criterion (2)

In matrix form the solution is written as:

$$\mathbf{R}_m \cdot \mathbf{h}_m = \mathbf{\sigma}_d$$

(Wiener-Hopf Equation)

where:

$$\mathbf{R}_m = \begin{bmatrix} R(0) & R(-1) & \cdots & R_{xx}(-M+1) \\ R_{xx}(1) & R_{xx}(0) & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ R_{xx}(M-1) & \cdots & R_{xx}(0) \end{bmatrix}, \quad \mathbf{h}_m = \begin{bmatrix} h(0) \\ h(1) \\ \vdots \\ h(M-1) \end{bmatrix}, \quad \mathbf{\sigma}_d = \begin{bmatrix} \sigma_{dx}(0) \\ \sigma_{dx}(1) \\ \vdots \\ \sigma_{dx}(M-1) \end{bmatrix}$$

Autocorrelation Matrix (MxM), Filter coefficients (Mx1), Crosscorrelation Vector (Mx1)
\textbf{MMSE Criterion (3)}

* Optimum solution:  \( h_{\text{opt}} = B_M^{-1} \cdot R_d \)

* Minimum mean square error:

\[
J_{\text{min}} = J(h_{\text{opt}}) = E[\|d(n)\|^2] - \sum_{k=0}^{N-1} h_{\text{opt}}(k) R_{dx*}(k)
\]

\[
= 6_d^2 - R_d^H B_M^{-1} R_d
\]

\[
\begin{bmatrix}
* \text{ denotes conjugation} \\
H \text{ denotes conjugate transpose}
\end{bmatrix}
\]

* \( B_M \) is Hermitian and Toeplitz: Efficient solutions exist.
LEAST SQUARES CRITERION (LS)

Minimize the least-squares function

$$J_{ls}(h_m) = \frac{1}{2} \sum_{n=0}^{N-1} |e(n)|^2$$

where, $h_m$ and $e(n)$ are as before.

**SOLUTION:** ($J_{ls}(...)$ is also a quadratic function of $h_m$)

$$[ \sum_{k=0}^{M-1} h(k) \cdot \hat{R}_{xx}(l-k) = \hat{r}_{dx}(l+D), \quad l=0,1,...,M-1 ]$$

where:

$$\hat{R}^{(m)}_{xx} = \frac{1}{N} \sum_{n=0}^{N-1} x(n) x^*(n-m) \quad \hat{r}^{(m)}_{dx} = \frac{1}{N} \sum_{n=0}^{N-1} d(n)x^*(n-m)$$
**MMSE and LS Criteria**

* The solutions obtained from both criteria are similar in form.

* In MMSE, the true statistical autocorrelation and cross-correlation are employed. The optimum (Wiener) filter coefficients are obtained.

* In LS, estimates of the autocorrelation and cross-correlation are used. The underlying assumption is that the observed data sequence is stationary and ergodic. Estimates of the optimum filter coefficients are obtained.
Consider the recursive algorithm

\[ h_{m}(n+1) = h_{m}(n) + \frac{1}{2} \mu(n) D(n) \quad , \quad n=0,1, \ldots \]

where:
- \( h_{\eta}(n) \) the vector of filter coefficients at iteration \( \eta \)
- \( \mu(n) \) is a step size at iteration \( \eta \)
- \( D(n) \) iteration vector at iteration \( \eta \).

**SPECIAL CASE:** Steepest-descent methods (gradient methods)

\[
D(n) = - \frac{d \mathcal{S}(h_{m}(n))}{d h_{m}(n)} = 2 \left[ r_{d} - B_{m} h_{m}(n) \right].
\]

Thus:

\[
\left[ h_{m}(n+1) = h_{m}(n) + \mu(n) \left[ r_{d} - B_{m} h_{m}(n) \right] \right] \quad , \quad n=0,1, \ldots
\]
By substituting, \( \mathbf{\Sigma}_d = \mathbb{E}\{d(n)X_M^*(n)\} \), \( \mathbf{B}_M = \mathbb{E}\{X_M^*(n)X_M^T(n)\} \)

where \( X_M(n) = [x(n), x(n-1), \ldots, x(n-M+1)]^T \)

we obtain:

\[
\hat{h}_M(n+1) = \hat{h}_M(n) + \mu(n) \cdot \mathbb{E}\{X_M^*(n) [d(n) - X_M^T(n) \hat{h}_M(n)] e(n) \}
\]

Thus:

\[
\hat{h}_M(n+1) = \hat{h}_M(n) + \mu(n) \cdot \mathbb{E}\{e(n)X_M^*(n)\}, \quad n=0,1,\ldots
\]

* It can be shown that the above algorithm converges provided \( \mu(n) \) is properly chosen.

* Adaptation stops when \( \mathbb{E}\{e(n)X_M^*(n)\} = 0 \) (orthogonality principle)
The Least-Mean-Squares (LMS) Algorithm

* In practice, the term $E\sum e(n)X^*_M(n)$ is replaced by an estimate.

* A simple unbiased estimate is obtained by dropping the expectation operation.

* In practice, the step size $\mu(n)$ is fixed to a constant value $\mu > 0$.

Thus,

\[
LMS: \quad h_M(n+1) = h_M(n) + \mu \cdot e(n) \cdot X^*_M(n), \quad n=0,1,\ldots
\]

(Various variations of the LMS algorithm exist in the literature.)
PROPERTIES OF THE LMS ALGORITHM

- The rate of convergence depends on the following:
  1) Step size $\mu$: The higher the value of $\mu$, the faster the convergence. The higher the value of $\mu$, the higher the final mean square error achieved by the algorithm.
  2) Eigenvalue spread of $B_m$: The larger the eigenvalue spread, the slower the convergence.

- There is a trade-off between convergence speed and final mean square error.

- The algorithm is stable provided
  \[ 0 < \mu < \frac{2}{\lambda_{\text{max}}} \]
  where $\lambda_{\text{max}}$ is the largest eigenvalue of $B_m$. 
Properties of the LMS Algorithm (2)

- In practice choose: \( 0 < \mu < \frac{1}{\left(X_M^T X_M\right)^{-1} \sum_{k=0}^{N} |x(n-k)|^2} \)

- In nonstationary signal environments (slowly time varying), the final mean-square error achieved is

\[ J_{\text{total}}(n) = J_{\text{min}}(n) + J_\mu(n) + J_e(n) \]

- \( J_\mu(n) \): Gradient noise error
- \( J_e(n) \): Lag error
FIGURE 9. Learning curves for the LMS algorithm applied to an adaptive equalizer of length $M = 11$ and a channel with eigenvalue spread $\lambda_{\text{max}}/\lambda_{\text{min}} = 11$. 

- $\mu = 0.045$
- $\mu = 0.09$
- $\mu = 0.115$
**SUMMARY OF THE LMS ALGORITHM.**

**Parameters:**  
- \( M = \text{number of taps} \)  
- \( \mu = \text{step size} \)  
\[ 0 < \mu < \frac{2}{\sum_{i=0}^{M} |x(n-i)|^2} \]

**Initial Conditions:**  
\( h_M(0) = 0 = [0, 0, \ldots, 0]^T \)

**Data:**  
\( X_M(n) = [x(n), x(n-1), \ldots, x(n-M+1)]^T \)

\( d(n) : \text{desired response.} \)

\( h_M(n) = [h(0,n), h(1,n), \ldots, h(M-1,n)]^T \)

**Computation:**  
For \( n = 0, 1, 2, \ldots \) compute

\[ e(n) = d(n) - X_M^T(n) h_M(n) \]

\[ h_M(n+1) = h_M(n) + \mu \cdot e(n) \cdot X_M^*(n) \]


Given an FIR filter with coefficients 

\[ h_M(n) = [h(0,n), h(1,n), \ldots, h(M-1,n)]^T \]

and the data vector 

\[ \hat{X}_M(n) = [x(n), x(n-1), \ldots, x(n-M+1)]^T \]

Suppose we observe the vectors: \( \hat{X}_M(l), l=0,1,2,\ldots,n \) and we wish to determine the filter coefficients vector \( h_M(n) \) that minimizes the weighted sum of magnitude-squared errors:

\[
\mathcal{E}_M = \sum_{l=0}^{n} w^{n-l} |e_m(l,n)|^2
\]

where, \( e_m(l,n) = d(l) - \hat{d}(l) \) and \( w \) is a forgetting factor.
RLS ESTIMATION (2)

Minimization of $E_M$ with respect to the $h_M(n)$ yields

$$B_M(n) \cdot h_M(n) = D_M(n)$$

where,

$$B_M(n) = \sum_{l=0}^{n} w^{n-l} X_M^*(l) X_M^T(l)$$

$$D_M(n) = \sum_{l=0}^{n} w^{n-l} X_M^*(n) \cdot d(l)$$

**Solution:**

$$\begin{bmatrix} h_M(n) = B_M^{-1} D_M(n) \end{bmatrix}$$
RLS ESTIMATION (3)

Suppose we have the optimum solution at time \( n-1 \) and we wish to compute \( \tilde{h}(n) \). Recursive solution

\[ R_{M}(n) = w R_{M}(n-1) + X_{M}^{*}(n) X_{M}^{T}(n) \]

\[ D_{M}(n) = w D_{M}(n-1) + X_{M}^{*}(n) \cdot d(n) \]

* TIME UPDATE EQUATIONS

* MATRIX INVERSION LEMMA

\[ R^{-1}_{M}(n) = \frac{1}{w} \left[ R^{-1}_{M}(n-1) - \frac{R^{-1}_{M}(n-1) \cdot X_{M}^{*}(n) X_{M}^{T}(n) R^{-1}_{M}(n-1)}{w + X_{M}^{T}(n) R^{-1}_{M}(n-1) X_{M}^{*}(n)} \right] \]
Let $P_M(n) = R_M^{-1}(n)$, and

$$K_M(n) = \frac{P_M(n-1) \Delta_M(n)}{w^T X_M(n) P_M(n-1) X_M^T(n)}$$

(Kalman Gain vector)

Then,

$$h_M(n) = P_M^{-1}(n) \cdot D_M(n) = P_M(n) \cdot D_M(n) = \ldots \ldots$$

$$= \frac{P_M(n-1) D_M(n-1)}{h_M(n-1)} + K_M(n) \left[ d(n) - X_M^*(n) \cdot h_M(n) \right]$$

$$e_M(n)$$

or

$$h_M(n) = h_M(n-1) + K_M(n) e_M(n)$$
It can be shown that $K_M(n) = P_M(n)X_M^*(n)$.

By substituting into RLS recursion equation,

$$h_M(n) = h_M(n-1) + P_M(n)X_M^*(n)e_M(n)$$

Note that for the LMS algorithm we found

$$h_M(n) = h_M(n-1) + \mu X_M^*(n)e_M(n)$$
**RLS Algorithm**

1) Compute the filter output:
   \[ \hat{d}(n) = X_M^T(n) h_M(n-1) \]

2) Compute the error:
   \[ e_M(n) = d(n) - \hat{d}(n) \]

3) Compute the gain vector:
   \[ K_M(n) = \frac{P_M(n-1) X_M^*(n)}{w^T X_M^T(n) P_M(n) X_M^*(n)} \]

4) Update the inverse of the autocorrelation matrix:
   \[ [P_M(n-1) - K_M(n) X_M^*(n) P_M(n-1)] \]

5) Update the coefficient vector of the filter:
   \[ h_M(n) = h_M(n-1) + K_M(n) e_M(n) \]
FIGURE 10 Learning curves for RLS algorithm and LMS algorithm for adaptive equalizer of length $M = 11$. The eigenvalue spread of the channel is $\lambda_{\text{max}}/\lambda_{\text{min}} = 11$. The step size for the LMS algorithm is $\Delta = 0.02$. (From Digital Communication by John G. Proakis. © 1983 by McGraw-Hill Book Company.)
RLS algorithm

Limitations: → Complexity \( \sim M^3 \)

→ Stability of the algorithm requires high precision arithmetic (24 bits or more)
   (Computation of \( P_m(n) \))

Potential Solution:

Use a square-root RLS algorithm based on LDU decomposition of \( P_m(n) \) or \( P_m(n) \).