The course provides an introduction to the mathematics of data analysis and a detailed overview of statistical models for inference and prediction.

Course Structure:
- Weekly lectures [notes: www.ece.udel.edu/~arce/Courses]
- Homework & computer assignments [30%]
- Midterm & Final examinations [70%]

Textbooks:
- Papoulis and Pillai, Probability, random variables, and stochastic processes.
- Hastie, Tibshirani and Friedman, The elements of statistical learning.
- Haykin, Adaptive Filter Theory.
Method of Least Squares (LS)

Definition (Method of Least Squares (LS))

Motivation: Develop a general method for optimally adjusting parameters to model observed data

Solution: Set the sum of squared residuals (errors) as the performance criteria and restrict the model to be linear

- The LS filtering method is a deterministic method
- Can be applied to linear and nonlinear systems
- LS corresponds to the ML criterion if the errors have a normal distribution
- The method is related to linear regression
- Optimization procedure results in a LS best fit for a filter over the observed (training) samples

Historical Note:
Gauss developed LS in 1795 at the age of 18
Consider the linear transversal filter

and a fixed number of observed samples: \( i = 1, 2, \ldots, N. \)

- \( M \) – the number of taps in the filter
- \( \{x(i)\} \) – input sequence
- \( \{d(i)\} \) – desired output sequence
Objective: Set the tap weights to minimize the sum of squared errors

$$\varepsilon (w) = \sum_{i=M}^{N} |e(i)|^2$$

Let

$$w = [w_0, w_1, \ldots, w_{M-1}]^T \quad \text{[weight vector]}$$

$$x(i) = [x(i), x(i-1), \ldots, x(i-M+1)]^T, M \leq i \leq N \quad \text{[obs. vect.]}$$

The error at time $i$ is

$$e(i) = d(i) - w^H x(i)$$

The full set of error values can be compiled into a vector
The Least Square Method

Define the $(N - M + 1) \times 1$ vectors:

$\mathbf{e}^H = [e(M), e(M+1), \cdots, e(N)]$  \hspace{1em} \text{[error vector]}

$\mathbf{d}^H = [d(M), d(M+1), \cdots, d(N)]$  \hspace{1em} \text{[desired vector]}

Denoting the filter output as $\hat{d}(i)$ and using vector form:

$\hat{\mathbf{d}}^H = [\hat{d}(M), \hat{d}(M+1), \cdots, \hat{d}(N)]$

$= [\mathbf{w}^H \mathbf{x}(M), \mathbf{w}^H \mathbf{x}(M+1), \cdots, \mathbf{w}^H \mathbf{x}(N)]$

$= \mathbf{w}^H [\mathbf{x}(M), \mathbf{x}(M+1), \cdots, \mathbf{x}(N)]$

$= \mathbf{w}^H \mathbf{A}^H$

where

$\mathbf{A}^H = [\mathbf{x}(M), \mathbf{x}(M+1), \cdots, \mathbf{x}(N)]$

is the observation data matrix
The Least Square Method

Expanding the data matrix

\[ A^H = [x(M), x(M+1), \ldots, x(N)] \]

\[ = \begin{bmatrix}
  x(M) & x(M+1) & \cdots & x(N) \\
  x(M-1) & x(M) & \cdots & x(N-1) \\
  \vdots & \vdots & \ddots & \vdots \\
  x(1) & x(2) & \cdots & x(N-M+1)
\end{bmatrix} \]

\[ \Rightarrow A^H \text{ is a } M \times (N - M + 1) \text{ rectangular toplitz matrix.} \]

Combining all the above:

Filter output vector: \[ \hat{d}^H = w^H A^H \]

Desired output vector: \[ d^H \]

Error vector: \[ \varepsilon^H = d^H - \hat{d}^H = d^H - w^H A^H \]

Note: All incorporate samples for \( M \leq i \leq N \)
The sum of the squared estimate errors can now be written as

$$\varepsilon(w) = \sum_{i=M}^{N} |e(i)|^2$$

$$= \varepsilon^H \varepsilon$$

$$= (d^H - w^H A^H)(d - Aw)$$

$$= d^H d - d^H Aw - w^H A^H d + w^H A^H Aw$$

Minimizing with respect to $w$,

$$\frac{\partial \varepsilon(w)}{\partial w} = -2A^H d + 2A^H Aw \quad (*)$$

Setting $(*)$ equal to zero gives the optimal LS weight $\hat{w}$

$$\Rightarrow A^H A\hat{w} = A^H d \quad [\text{Deterministic normal equation}]$$
Note: \( A \) is not generally square, and thus not invertible, but \( A^H A \) is square and generally invertible

\[
A^H A \hat{w} = A^H d
\]

\( \Rightarrow \hat{w} = (A^H A)^{-1} A^H d \)

The deterministic normal equation can be rearranged as

\[
A^H A \hat{w} - A^H d = 0
\]

\[
A^H (A \hat{w} - d) = 0 \quad \text{[or using } \varepsilon_{\text{min}} = d - A \hat{w}] \]

\[
A^H \varepsilon_{\text{min}} = 0
\]

Observation: The LS orthogonality principle states that the estimate error \( \varepsilon_{\text{min}} \) is orthogonal to the row vectors of the data matrix \( A^H \)
**Objective:** Determine the minimum sum of squared errors ($e_{\text{min}}$)

$$
e_{\text{min}} = e_{\text{min}}^H e_{\text{min}}$$

$$= (d^H - \hat{w}^H A^H)(d - A\hat{w})$$

$$= d^H d - \hat{w}^H A^H d - d^H A\hat{w} + \hat{w}^H A^H A\hat{w}$$

Utilizing the normal equations $\hat{w}^H A^H d = \hat{w}^H A^H A\hat{w}$

$$e_{\text{min}} = d^H d - \hat{w}^H A^H d - d^H A\hat{w} + \hat{w}^H A^H A\hat{w}$$

$$= d^H d - d^H A\hat{w}$$

or using $\hat{w} = (A^H A)^{-1} A^H d$

$$e_{\text{min}} = d^H d - d^H A(A^H A)^{-1} A^H d$$  \text{(*)}$$

Note that

$$d^H d = \sum_{i=M}^{N} |d(i)|^2$$  \text{[energy of desired response]}
Consider again the deterministic normal equation

\[ A^H A \hat{w} = A^H d \]

Note that

\[ A^H A = [x(M), x(M+1), \ldots, x(N)] \begin{bmatrix} x^H(M) \\ x^H(M+1) \\ \vdots \\ x^H(N) \end{bmatrix} \]

\[ = \sum_{i=M}^{N} x(i)x^H(i) \]

\[ = \Phi \quad [\text{time averaged correlation matrix, size } M \times M] \]
From $\Phi = \sum_{i=M}^{N} x(i)x^H(i)$ it can be shown that:

1. $\Phi$ is Hermitian
2. $\Phi$ is nonnegative definite

To prove this, note that for any $a$

$$a^H\Phi a = \sum_{i=M}^{N} a^H x(i)x^H(i)a$$

$$= \sum_{i=M}^{N} [a^H x(i)][a^H x(i)]^H$$

$$= \sum_{i=M}^{N} |a^H x(i)|^2 \geq 0$$

3. From (1) and (2) we can prove that the eigenvalues of $\Phi$ are real and nonnegative
The deterministic normal equation,

$$A^H A \hat{w} = A^H d$$

also employs

$$A^H d = \begin{bmatrix} x(M), x(M+1), \ldots, x(N) \end{bmatrix} \begin{bmatrix} d^*(M) \\ d^*(M+1) \\ \vdots \\ d^*(N) \end{bmatrix}$$

$$= \sum_{i=M}^{N} x(i) d^*(i)$$

$$= \theta \quad \text{[Time averaged cross-correlation vector, size } M \times 1 \text{]}$$
Thus the deterministic normal equation, $A^H A \hat{w} = A^H d$, reduces to

$$\Phi \hat{w} = \theta$$

$\Phi$ is usually positive definite (always positive semi-definite) $\Rightarrow$ the solution is well defined

$$\hat{w} = \Phi^{-1} \theta \quad \text{[LS optimal weight vector]}$$

Also, recall from (*) that $e_{\text{min}}$ can be expressed as

$$e_{\text{min}} = d^H d - A^H \Phi^{-1} A d$$

$$= e_d - \theta^H \Phi^{-1} \theta$$

where $e_d$ is the energy of desired signal
Consider again the orthogonality principle

\[ A^H \epsilon_{\text{min}} = 0 \]

Recall that \( \hat{d} = A\hat{w} \). Thus

\[ A^H \epsilon_{\text{min}} = 0 \Rightarrow \hat{w}^H A^H \epsilon_{\text{min}} = \hat{w}^H 0 \]
\[ \Rightarrow \hat{d}^H \epsilon_{\text{min}} = 0 \]

**Result:** The minimum estimation error vector, \( \epsilon_{\text{min}} \), is orthogonal to the data matrix \( A^H \) and the LS estimate \( \hat{d} \)
Objective: Analyze the Least Squares solution in terms of
  - Bias – is the LS solution unbiased?
  - BLUE – is the LS solution the Best Linear Unbiased Estimate?
Assumption: Take the true underlying system to be a linear

\[ d(i) = \sum_{k=0}^{M-1} w_0^* x(i-k) + e_0(i) \]
\[ = w_0^H x(i) + e_0(i) \]

\( e_0(i) \) is the unobservable measurement error
\[ \Rightarrow e_0(i) \text{ is white (uncorrelated) with zero mean and variance } \sigma^2 \]
Express the desired signal in vector form

\[ d = Aw_0 + \epsilon_0 \]
where \( \epsilon_0^H = [e_0(M), e_0(M+1), \ldots, e_0(N)] \)
Objective: Evaluate the bias of $\hat{w}$

Recall that

$$\hat{w} = (A^H A)^{-1} A^H d$$

Using $d = Aw_0 + \epsilon_0$ in the above

$$\hat{w} = (A^H A)^{-1} A^H (Aw_0 + \epsilon_0)$$

$$= (A^H A)^{-1} A^H Aw_0 + (A^H A)^{-1} A^H \epsilon_0$$

$$= w_0 + (A^H A)^{-1} A^H \epsilon_0 \quad (\ast)$$

Note $A$ is fixed. Thus taking the expectation of $(\ast)$ yields

$$E\{\hat{w}\} = w_0 + (A^H A)^{-1} A^H E\{\epsilon_0\}$$

$$= w_0$$

Result: The LS estimate, $\hat{w}$, is unbiased
Objective: Evaluate the covariance of \( \hat{w} \)

Note that from (\(^\star\))

\[
\hat{w} = w_0 + (A^H A)^{-1} A^H \epsilon_0
\]

\[
\Rightarrow \hat{w} - w_0 = (A^H A)^{-1} A^H \epsilon_0
\]

Thus

\[
\text{cov}[\hat{w}] = E\{(\hat{w} - w_0)(\hat{w} - w_0)^H\}
\]

\[
= E\{(A^H A)^{-1} A^H \epsilon_0 \epsilon_0^H A (A^H A)^{-1}\}
\]

\[
= \Phi^{-1} A^H E\{\epsilon_0 \epsilon_0^H\} A \Phi^{-1}
\]

\[
= \sigma^2 I \left( \Phi^{-1} \Phi \right) = \sigma^2 \Phi^{-1}
\]

Result: The covariance of \( \hat{w} \) is proportional to: (1) the variance of the measurement noise and (2) the inverse of the time average correlation matrix.
Objective: Show that the LS estimate $\hat{w}$ is the Best Linear Unbiased Estimate (BLUE)

Consider any linear unbiased estimate $\tilde{w}$

Note that $\tilde{w}$ is a linear function of the observed date and can thus be written as

$$\tilde{w} = Bd$$

where $B$ is a $M \times (N - M + 1)$ matrix

Substituting $d = Aw_0 + \varepsilon_0$ into the above,

$$\tilde{w} = BAw_0 + B\varepsilon_0 \quad (*)$$

$$\Rightarrow E\{\tilde{w}\} = BAw_0$$

$$\Rightarrow BA = I \quad [\text{since } \tilde{w} \text{ unbiased}]$$

Thus $BA = I$ and (*) $\Rightarrow$

$$\tilde{w} = w_0 + B\varepsilon_0$$
Rearranging $\tilde{w} = w_0 + B\epsilon_0$,

$$\tilde{w} - w_0 = B\epsilon_0$$

$$\Rightarrow \text{cov}[\tilde{w}] = E\{(\tilde{w} - w_0)(\tilde{w} - w_0)^H\}$$

$$= E\{B\epsilon_0\epsilon_0^H B^H\}$$

$$= \sigma^2 BB^H \quad (\star_2)$$

Now define

$$\psi = B - (A^H A)^{-1} A^H$$

$$\Rightarrow \psi\psi^H = [B - \Phi^{-1} A^H][B^H - A\Phi^{-1}]$$

$$= BB^H - BA\Phi^{-1} - \Phi^{-1} A^H B^H + \Phi^{-1} A^H A\Phi^{-1}$$

$$= BB^H - \Phi^{-1} - \Phi^{-1} + \Phi^{-1}$$

$$= BB^H - \Phi^{-1}$$

$$= BB^H - (A^H A)^{-1}$$
Observation: The diagonal elements at $\psi\psi^H$ must be $\geq 0$

Thus $\psi\psi^H = \mathbf{BB}^H - (\mathbf{A}^H\mathbf{A})^{-1}$ $\Rightarrow$

$$\text{diag}[\mathbf{BB}^H] \geq \text{diag}[(\mathbf{A}^H\mathbf{A})^{-1}]$$
$$\Rightarrow \text{diag}[\sigma^2\mathbf{BB}^H] \geq \text{diag}[\sigma^2(\mathbf{A}^H\mathbf{A})^{-1}] \quad (\ast)$$

But recall from $(\ast_1)$ and $(\ast_2)$ that

$$\text{cov}[^\w] = \sigma^2(\mathbf{A}^H\mathbf{A})^{-1} \quad \text{and} \quad \text{cov}[^\tilde{w}] = \sigma^2\mathbf{BB}^H$$

Utilizing these results in $(\ast) \Rightarrow$

$$\text{variance}[^\tilde{w}_i] \geq \text{variance}[^\w_i] \quad i = 1, 2, \ldots, M$$

Thus the weights in $^\w$ have lower variance than any other linear estimates

Result: The LS estimate $^\w$ is unbiased and has the smallest weight variance $\Rightarrow$ it is the Best Linear Unbiased Estimate (BLUE)
Definition (Recursive Least Squares (RLS))

Motivation: LS requires solving

\[ \hat{w} = (A^H A)^{-1} A^H d \]

\[ = \Phi^{-1} \theta \]

where

\[ \Phi = \sum_{i=M}^{N} x(i)x^H(i) \quad \text{and} \quad \theta = \sum_{i=M}^{N} x(i)d^*(i) \]

- \((A^H A)\) is \(M \times M\) and inversion requires \(O(M^3)\) multiplications and additions.

Approach: Suppose the LS optimal weights are known at time \(n\), \(\hat{w}(n)\). As time evolves, find the new estimate, \(\hat{w}(n+1)\), in terms of \(\hat{w}(n)\).

- Employ the matrix inversion lemma to reduce the number of computations.
Let the observation sequence be \( x(1), x(2), \cdots, x(n) \)

\( \Rightarrow \) Assume \( x(l) = 0 \) for \( l \leq 0 \)

Define the error as

\[
\varepsilon(n) = \sum_{i=1}^{n} \beta(n, i)|e(i)|^2
\]

where

\[
e(i) = d(i) - w^H(n)x(i)
\]

\[
x(i) = [x(i), x(i-1), \cdots, x(i-M+1)]^T
\]

\[
w(n) = [w_0(n), w_1(n), \cdots, w_{M-1}(n)]^T
\]

\( \Rightarrow \) \( \beta(n, i) \in (0, 1] \) is a forgetting factor used in non–stationary statistics cases
A commonly used forgetting factor is the exponential forgetting factor

$$\beta(n, i) = \lambda^{n-i} \quad i = 1, 2, \cdots, n, \quad \lambda \in (0, 1]$$

Thus,

$$\varepsilon(n) = \sum_{i=1}^{n} \lambda^{n-i} |e(i)|^2$$

The LS solution is given by the deterministic normal equation

$$\Phi(n)\hat{w}(n) = \theta(n)$$

where now

$$\Phi(n) = \sum_{i=1}^{n} \lambda^{n-i} x(i)x^H(i)$$

$$\theta(n) = \sum_{i=1}^{n} \lambda^{n-i} x(i)d^*(i)$$
The normal equation terms can be updated recursively,

\[ \Phi(n) = \sum_{i=1}^{n} \lambda^{n-i} x(i)x^H(i) \]

\[ = \lambda \left[ \sum_{i=1}^{n-1} \lambda^{(n-1)-i} x(i)x^H(i) \right] + x(n)x^H(n) \]

\[ = \lambda \Phi(n-1) + x(n)x^H(n) \]

Similarly

\[ \theta(n) = \sum_{i=1}^{n} \lambda^{n-i} x(i)d^*(i) \]

\[ = \lambda \left[ \sum_{i=1}^{n-1} \lambda^{(n-1)-i} x(i)d^*(i) \right] + x(n)d^*(n) \]

\[ = \lambda \theta(n-1) + x(n)d^*(n) \]
Aside: **Matrix inversion lemma:** If

\[
\begin{bmatrix}
A
\end{bmatrix}_{M \times M} = \begin{bmatrix}
B^{-1}
\end{bmatrix}_{M \times M} + \begin{bmatrix}
C D^{-1} C^H
\end{bmatrix}_{M \times L \times L \times M}
\]

where \( A, B, D \) are positive definite (non-singular), then

\[
A^{-1} = B - B C [D + C^H B C]^{-1} C^H B
\]

Apply the lemma to

\[
\Phi(n) = \lambda \Phi(n - 1) + x(n) x^H(n)
\]

Accordingly, set

\[
\begin{align*}
A &= \Phi(n) & [M \times M] \\
C &= x(n) & [M \times 1] \\
B^{-1} &= \lambda \Phi(n - 1) & [M \times M] \\
D &= 1 & [1 \times 1]
\end{align*}
\]
Utilizing
\[
\begin{align*}
A &= \Phi(n) & B^{-1} &= \lambda \Phi(n - 1) \\
C &= x(n) & D &= 1
\end{align*}
\]
and
\[
A^{-1} = B - BC[D + C^HBC]^{-1}C^HB
\]
we get
\[
[D + C^HBC]^{-1} = [1 + \lambda^{-1}x^H(n)\Phi^{-1}(n - 1)x(n)]^{-1}
\]
which is a scalar. Thus evaluating (*) yields
\[
\Phi^{-1}(n) = \lambda^{-1}\Phi^{-1}(n - 1) - \frac{\lambda^{-2}\Phi^{-1}(n - 1)x(n)x^H(n)\Phi^{-1}(n - 1)}{1 + \lambda^{-1}x^H(n)\Phi^{-1}(n - 1)x(n)}
\]
To simplify the result, let \( P(n) = \Phi^{-1}(n) \) and
\[
\underbrace{k(n)}_{\text{Gain vector}} = \frac{\lambda^{-1}P(n - 1)x(n)}{1 + \lambda^{-1}x^H(n)P(n - 1)x(n)}
\]
Using $P(n) = \Phi^{-1}(n)$ and $k(n) = \frac{\lambda^{-1}P(n-1)x(n)}{1 + \lambda^{-1}x^H(n)P(n-1)x(n)}$:

\[
\Phi^{-1}(n) = \lambda^{-1}\Phi^{-1}(n-1) - \frac{\lambda^{-2}\Phi^{-1}(n-1)x(n)x^H(n)\Phi^{-1}(n-1)}{1 + \lambda^{-1}x^H(n)\Phi^{-1}(n-1)x(n)}
\]

\[
\Rightarrow P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}k(n)x^H(n)P(n-1) \quad (*)
\]

Also, the gain vector can be simplified as

\[
k(n) = \frac{\lambda^{-1}P(n-1)x(n)}{1 + \lambda^{-1}x^H(n)P(n-1)x(n)} \quad \text{[multiply by denom.]} \]

\[
\Rightarrow k(n) = \lambda^{-1}P(n-1)x(n) - \lambda^{-1}k(n)x^H(n)P(n-1)x(n)
\]

\[
= [\lambda^{-1}P(n-1) - \lambda^{-1}k(n)x^H(n)P(n-1)]x(n)
\]

\[
= P(n) \text{ from (*)}
\]

\[
= P(n)x(n) = \Phi^{-1}(n)x(n) \quad (**)\\
\]
We must now derive an update for the tap weight vector. Recall,

\[ \hat{w}(n) = \Phi^{-1}(n)\theta(n) = P(n)\theta(n) \]

Using the recursion \( \theta(n) = \lambda \theta(n-1) + x(n)d^*(n) \) in the above

\[ \hat{w}(n) = \lambda P(n)\theta(n-1) + P(n)x(n)d^*(n) \quad (*** \quad \text{using update} \quad \star \quad \star \quad \star) \]

Using the update (\( \star \))

\[ P(n) = \lambda^{-1}P(n-1) - \lambda^{-1}k(n)x^H(n)P(n-1) \]

in the first \( P(n) \) term of (***)

\[
\hat{w}(n) = \lambda P(n)\theta(n-1) + P(n)x(n)d^*(n) \\
= \lambda [\lambda^{-1}P(n-1) - \lambda^{-1}k(n)x^H(n)P(n-1)]\theta(n-1) \\
+ P(n)x(n)d^*(n)
\]
\[ \hat{w}(n) = \lambda [\lambda^{-1} P(n-1) - \lambda^{-1} k(n)x^H(n)P(n-1)] \theta(n-1) + P(n)x(n)d^*(n) \]
\[ = P(n-1) \theta(n-1) - k(n)x^H(n)P(n-1) \theta(n-1) + P(n)x(n)d^*(n) \]
\[ = \hat{w}(n-1) - k(n)x^H(n)\hat{w}(n-1) + P(n)x(n)d^*(n) \]
\[ = \hat{w}(n-1) - k(n)[x^H(n)\hat{w}(n-1) - d^*(n)] \]
\[ = \hat{w}(n-1) + k(n)\alpha^*(n) \]

where \( \alpha(n) = d(n) - \hat{w}^H(n-1)x(n) \)

**Observation:** Difference between \( e(n) \) and \( \alpha(n) \):
\[ e(n) = d(n) - \hat{w}^H(n)x(n) \Rightarrow a \text{ posteriori error} \]
\[ \alpha(n) = d(n) - \hat{w}^H(n-1)x(n) \Rightarrow a \text{ priori error} \]
### RLS Algorithm Summary

1. Given a new sample $x(n)$, update the gain vector

   $$k(n) = \frac{\lambda^{-1} P(n-1) x(n)}{1 + \lambda^{-1} x^H(n) P(n-1) x(n)}$$

2. Update the innovation: $\alpha(n) = d(n) - \hat{w}^H(n-1) x(n)$

3. Update the tap weight vector: $\hat{w}(n) = \hat{w}(n-1) + k(n) \alpha^*(n)$

4. Update inverse correlation matrix

   $$P(n) = \lambda^{-1} P(n-1) - \lambda^{-1} k(n) x^H(n) P(n-1)$$

**Initial Conditions:** $\hat{w}(0) = 0$ and $\Phi(0) = \delta I$, where $\delta$ is a small positive constant, $\delta \approx 0.01 \sigma_x^2$. 
Algorithm Comparison: RLS and LMS algorithm terms:

<table>
<thead>
<tr>
<th>Entity</th>
<th>RLS</th>
<th>LMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>$\alpha(n) = d(n) - \hat{w}^H(n-1)x(n)$ (a priori error)</td>
<td>$e(n) = d(n) - \hat{w}^H(n)x(n)$ (a posteriori error)</td>
</tr>
<tr>
<td>Weight Update</td>
<td>$\hat{w}(n) = \hat{w}(n-1) + k(n)\alpha^*(n)$</td>
<td>$w(n+1) = w(n) + \mu x(n)e^*(n)$</td>
</tr>
<tr>
<td>Gain of error update</td>
<td>$\left(\frac{\lambda^{-1}P(n-1)}{1 + \lambda^{-1}x^H(n)P(n-1)x(n)}\right)x(n)$</td>
<td>$(\mu)x(n)$</td>
</tr>
</tbody>
</table>
Objective: Compare the complexities (number of additions and multiplies) for the LMS, LS, and RLS algorithms.

- Assume the data is real and the filter is of size $M$

Case 1 – The LMS algorithm: Algorithm stages:

1. $\hat{d}(n) = w^T(n)x(n)$
2. $e(n) = d(n) - \hat{d}(n)$
3. $w(n+1) = w(n) + \mu x(n)e(n)$

<table>
<thead>
<tr>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage</td>
</tr>
<tr>
<td>(1)</td>
</tr>
<tr>
<td>(2)</td>
</tr>
<tr>
<td>(3)</td>
</tr>
<tr>
<td>Total complexity per iteration</td>
</tr>
</tbody>
</table>
Case 2 – The LS algorithm: Algorithm solves

$$\hat{w}(n) = \Phi^{-1}(n)\theta(n)$$

and has stages:

1. $$\Phi(n+1) = \Phi(n) + x(n+1)x^H(n+1)$$
2. $$\theta(n+1) = \theta(n) + x(n+1)d(n+1)$$
3. $$\hat{w}(n+1) = \Phi^{-1}(n+1)\theta(n+1)$$

<table>
<thead>
<tr>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage</td>
</tr>
<tr>
<td>(1)</td>
</tr>
<tr>
<td>(2)</td>
</tr>
<tr>
<td>(3)</td>
</tr>
</tbody>
</table>

Total complexity per iteration:

$$O_\times(M^3 + 2M^2 + M)$$

$$O_+(M^3 + 2M^2)$$
**Case 3 – The RLS algorithm:** Algorithm has stages (assuming $\lambda = 1$):

1. $k(n) = \frac{\lambda^{-1}P(n-1)x(n)}{1 + x^T(n)P(n-1)x(n)}$
2. $\alpha(n) = d(n) - \hat{w}^T(n-1)x(n)$
3. $\hat{w}(n) = \hat{w}(n-1) + k(n)\alpha(n)$
4. $P(n) = P(n-1) - k(n)x^T(n)P(n-1)$

**Note:** The operation $x^T(n)P(n-1)$ is repeated (but only performed once). Corresponding steps are underlined in the chart.

<table>
<thead>
<tr>
<th>Stage</th>
<th>$O_\times$</th>
<th>$O_+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) numerator</td>
<td>$M^2$</td>
<td>$M(M-1)$</td>
</tr>
<tr>
<td>(1) denominator</td>
<td>$M^2 + M$</td>
<td>$M(M-1) + M$</td>
</tr>
<tr>
<td>(1) division</td>
<td>$M$</td>
<td>$M$</td>
</tr>
<tr>
<td>(2)</td>
<td>$M$</td>
<td>$M$</td>
</tr>
<tr>
<td>(3)</td>
<td>$M$</td>
<td>$M$</td>
</tr>
<tr>
<td>(4)</td>
<td>$M^2 + M^2$</td>
<td>$M(M-1) + M^2$</td>
</tr>
<tr>
<td>Total complexity per iteration</td>
<td>$O_\times(3M^2 + 4M)$</td>
<td>$O_+(3M^2 + M)$</td>
</tr>
</tbody>
</table>
Objective: Analyze the RLS algorithm in terms of

- Bias
- Convergence in the mean; Convergence in the mean square
- Learning curve decay rate

Assumptions:

1. The desired signal is formed by the regression model

\[ d(n) = w_0^H x(n) + e_0(n) \]

where \( e_0(n) \) is white with variance \( \sigma^2 \).

2. \( \lambda = 1 \) and \( n \geq M \).

Then

\[ \hat{w}(n) = \Phi^{-1}(n) \theta(n) \]

where

\[ \Phi(n) = \sum_{i=1}^{n} x(i)x^H(i) \quad \text{and} \quad \theta(n) = \sum_{i=1}^{n} x(i)d^*(i) \]
Substituting \( d^*(n) = x^H(n)w_0 + e_0^*(n) \) into \( \theta(n) \)

\[
\theta(n) = \sum_{i=1}^{n} x(i)[x^H(i)w_0 + e_0^*(i)]
\]

\[
= \sum_{i=1}^{n} x(i)x^H(i)w_0 + \sum_{i=1}^{n} x(i)e_0^*(i)
\]

\[
= \Phi(n)w_0 + \sum_{i=1}^{n} x(i)e_0^*(i)
\]

Thus

\[
\hat{w}(n) = \Phi^{-1}(n)\theta(n)
\]

\[
= \Phi^{-1}(n)[\Phi(n)w_0 + \sum_{i=1}^{n} x(i)e_0^*(i)]
\]

\[
= w_0 + \Phi^{-1}(n)\sum_{i=1}^{n} x(i)e_0^*(i) \quad (*)
\]
Note that $E\{A\} = E\{E\{A|B\}\}$. Thus

$$
\hat{w}(n) = w_0 + \Phi^{-1}(n) \sum_{i=1}^{n} x(i) e_0^*(i)
$$

$$
\Rightarrow E\{\hat{w}(n)\} = w_0 + E\{E\{\Phi^{-1}(n) \sum_{i=1}^{n} x(i) e_0^*(i)|x(i), i = 1, 2, \ldots, n}\}\}
$$

$$
= w_0 + E\{\Phi^{-1}(n) \sum_{i=1}^{n} x(i) E\{e_0^*(i)\}\} = w_0
$$

The above follows from the fact that $\Phi(n)$ and $e_0^*(i)$ are independent.

**Why?** $e_0(i)$ is independent of all observations and the $x(i)$ terms are given, uniquely defining $\Phi(n)$. $\Rightarrow$ independence of $\Phi(n)$ and $e_0^*(i)$.

**Result:** The RLS algorithm is **unbiased and convergent in the mean** for $n \geq M$.

**Question:** How does this compare to the LMS algorithm?
Next, consider the convergence in the mean square. Recall (*)

\[ \hat{w}(n) = w_0 + \Phi^{-1}(n) \sum_{i=1}^{n} x(i)e^*_0(i) \]

which gives

\[ \epsilon(n) = \hat{w}(n) - w_0 = \Phi^{-1}(n) \sum_{i=1}^{n} x(i)e^*_0(i) \]

Thus the weight error correlation matrix is

\[ K(n) = E\{\epsilon(n)\epsilon^H(n)\} = E\left\{ \Phi^{-1}(n) \left( \sum_{i=1}^{n} \sum_{j=1}^{n} x(i)e^*_0(i)e_0(j)x^H(j) \right) \Phi^{-1}(n) \right\} \]
Again using $E\{A\} = E\{E\{A|B\}\}$ yields

$$K(n) = E \left\{ \Phi^{-1}(n) \left( \sum_{i=1}^{n} \sum_{j=1}^{n} x(i) E \{ e_0^*(i) e_0(j) \} x^H(j) \right) \Phi^{-1}(n) \right\}$$

$$= \sigma^2 E \left\{ \Phi^{-1}(n) \left( \sum_{i=1}^{n} x(i) x^H(i) \right) \Phi^{-1}(n) \right\}$$

$$= \sigma^2 E \{ \Phi^{-1}(n) \Phi(n) \Phi^{-1}(n) \}$$

$$= \sigma^2 E \{ \Phi^{-1}(n) \}$$

**Note:** $\Phi^{-1}(n)$ has a Wishart distribution, the expectation of which is

$$E\{ \Phi^{-1}(n) \} = \frac{1}{n-M-1} R^{-1} \quad n > M + 1$$
Using $K(n) = \frac{\sigma^2}{n-M-1}R^{-1}$ and the trace

$$E\{||\epsilon(n)||^2\} = E\{\epsilon^H(n)\epsilon(n)\} = E\{\text{trace}[\epsilon^H(n)\epsilon(n)]\} = E\{\text{trace}[\epsilon(n)\epsilon^H(n)]\} = \text{trace}E\{\epsilon(n)\epsilon^H(n)\} = \text{trace}[K(n)]$$

$$= \frac{\sigma^2}{n-M-1}\text{trace}[R^{-1}]$$

$$= \frac{\sigma^2}{n-M-1} \sum_{i=1}^{M} \frac{1}{\lambda_i} \quad n > M + 1$$

**Results:**

- The weight vector MSE is initially proportional to $\sum_{i=1}^{M} \frac{1}{\lambda_i}$
- The weight vector converges linearly in the mean squared sense
Objective: Evaluate the RLS (error) learning curve

Recall the *a priori* estimation error

\[
\alpha(n) = d(n) - \hat{w}^H(n-1)x(n) \\
= d(n) - \hat{d}_0(n) + \hat{d}_0(n) - \hat{w}^H(n-1)x(n) \\
= e_0(n) + w_0^Hx(n) - \hat{w}^H(n-1)x(n) \\
= e_0(n) - \varepsilon^H(n-1)x(n)
\]

Now consider the MSE of \( \alpha(n) \)

\[
J_\alpha(n) = E\{ |\alpha(n)|^2 \} \\
= E\{ [e_0^*(n) - x^H(n)\varepsilon(n-1)][e_0(n) - \varepsilon^H(n-1)x(n)] \} \\
= E\{ |e_0(n)|^2 \} - E\{ x^H(n)\varepsilon(n-1)e_0(n) \} \\
- E\{ \varepsilon^H(n-1)x(n)e_0^*(n) \} + E\{ x^H(n)\varepsilon(n-1)\varepsilon^H(n-1)x(n) \}
\]

To analyze \( J_\alpha(n) \), consider each term individually
The Least Square Method
Performance Analysis

\[ J_\alpha(n) = E\{ |e_0(n)|^2 \} - E\{ x^H(n) \varepsilon(n-1) e_0(n) \} \]
\[ - E\{ \varepsilon^H(n-1) x(n) e_0^*(n) \} + E\{ x^H(n) \varepsilon(n-1) \varepsilon^H(n-1) x(n) \} \]

\textbf{Term:} \( E\{ |e_0(n)|^2 \} \).

Clearly,

\[ E\{ |e_0(n)|^2 \} = \sigma^2 \]

\textbf{Term:} \( E\{ \varepsilon^H(n-1) x(n) e_0^*(n) \} \).

By the independence theorem, \( \varepsilon(n-1) \) is independent of \( x(n) \) and \( e_0(n) \). Thus,

\[ E\{ \varepsilon^H(n-1) x(n) e_0^*(n) \} = E\{ \varepsilon^H(n-1) \} E\{ x(n) e_0^*(n) \} = 0 \]

where the final result is due to the orthogonality principle.

\textbf{Term:} \( E\{ x^H(n) \varepsilon(n-1) e_0(n) \} \rightarrow 0 \) by similar arguments.
\[ J_\alpha(n) = E\{|e_0(n)|^2\} - E\{x^H(n)e(n-1)e_0(n)\} - E\{e^H(n-1)x(n)e_0^*(n)\} + E\{x^H(n)e(n-1)e^H(n-1)x(n)\} \]

**Term:** \( E\{x^H(n)e(n-1)e^H(n-1)x(n)\} \)

\[
E\{x^H(n)e(n-1)e^H(n-1)x(n)\} = E\{\text{trace}[x^H(n)e(n-1)e^H(n-1)x(n)]\} \\
= E\{\text{trace}[x(n)x^H(n)e(n-1)e^H(n-1)]\} \\
= E\{\text{trace}[R K(n-1)]\}
\]
Utilizing $K(n-1) = \frac{\sigma^2}{n-M-2} R^{-1}$ and substituting back each of the components

$$J_\alpha(n) = \sigma^2 + \text{trace}[RK(n-1)]$$

$$= \sigma^2 + \frac{M\sigma^2}{n-M-2} \quad n > M + 1$$

**Results:**

- The ensemble average learning curve of the RLS converges in about $2M$ iterations, which is typically an order of magnitude faster than the LMS.
- $\lim_{n \to \infty} J_\alpha(n) = \sigma^2$ thus there is no excess MSE.
- Convergence of the RLS algorithm is independent of the eigenvalues of $\Phi(n)$. 
Example

Consider again the channel equalization problem

\[ h_n = \begin{cases} 
  \frac{1}{2} [1 + \cos \left( \frac{2\pi}{W} (n - 1) \right)] & n = 1, 2, 3 \\
  0 & \text{otherwise}
\end{cases} \]

- As before an 11-tap filter is used
- The SNR is 30dB and \( W \) is varied to control the eigenvalue spread
Observations:

- The RLS algorithm converges in about 20 iterations (twice the number of filter taps)
- The convergence (rate) is insensitive to the eigenvalue spread
Observations:

- The RLS algorithm converges faster than the LMS algorithm
- The RLS algorithm has lower steady state error than the LMS algorithm