FSAN815/ELEG815: Foundations of Statistical Learning

Gonzalo R. Arce

Chapter 7: Least Squares & RLS

Fall 2014

Course Objectives & Structure

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/$\sim$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))

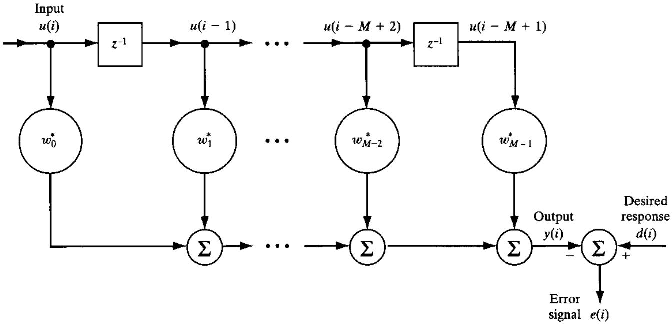
Motivation: Develop a general method for optimally adjusting parameters to model observed data

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: $i = 1, 2, \cdots, N$.

- 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

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

Let

$$\begin{aligned} \mathbf{w} &= [w_0, w_1, \cdots, w_{M-1}]^T \qquad \text{[weight vector]} \\ \mathbf{x}(i) &= [x(i), x(i-1), \cdots, x(i-M+1)]^T, M \leq i \leq N \quad \text{[obs. vect.]} \end{aligned}$$

The error at time i is

$$e(i) = d(i) - \mathbf{w}^H \mathbf{x}(i)$$

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

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

$$\begin{aligned}
\boldsymbol{\varepsilon}^H &= [e(M), e(M+1), \cdots, e(N)] \qquad \text{[error vector]} \\
\mathbf{d}^H &= [d(M), d(M+1), \cdots, d(N)] \qquad \text{[desired vector]}
\end{aligned}$$

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

$$\begin{aligned}
\hat{\mathbf{d}}^H &= [\hat{d}(M), \hat{d}(M+1), \cdots, \hat{d}(N)] \\
&= [\mathbf{w}^H\mathbf{x}(M), \mathbf{w}^H\mathbf{x}(M+1), \cdots, \mathbf{w}^H\mathbf{x}(N)] \\
&= \mathbf{w}^H[\mathbf{x}(M), \mathbf{x}(M+1), \cdots, \mathbf{x}(N)] \\
&= \mathbf{w}^H\mathbf{A}^H
\end{aligned}$$

where

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

is the observation data matrix

Expanding the data matrix

$$\begin{aligned}\mathbf{A}^H &= [\mathbf{x}(M), \mathbf{x}(M+1), \cdots, \mathbf{x}(N)] \\ &= \begin{bmatrix} x(M) & x(M+1) & \cdots & x(N) \\ x(M-1) & x(M) & \cdots & x(N-1) \\ \vdots & \vdots & \ddots & \vdots \\ x(1) & x(2) & \cdots & x(N-M+1) \end{bmatrix}\end{aligned}$$

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

Combining all the above:

$$\begin{aligned}\text{Filter output vector:} \quad & \hat{\mathbf{d}}^H = \mathbf{w}^H \mathbf{A}^H \\ \text{Desired output vector:} \quad & \mathbf{d}^H \\ \text{Error vector:} \quad & \boldsymbol{\varepsilon}^H = \mathbf{d}^H - \hat{\mathbf{d}}^H = \mathbf{d}^H - \mathbf{w}^H \mathbf{A}^H\end{aligned}$$

Note: All incorporate samples for $M \leq i \leq N$

The sum of the squared estimate errors can now be written as

$$\begin{aligned}\varepsilon(\mathbf{w}) &= \sum_{i=M}^{N} |e(i)|^2 \\ &= \boldsymbol{\varepsilon}^H \boldsymbol{\varepsilon} \\ &= (\mathbf{d}^H - \mathbf{w}^H \mathbf{A}^H)(\mathbf{d} - \mathbf{A}\mathbf{w}) \\ &= \mathbf{d}^H \mathbf{d} - \mathbf{d}^H \mathbf{A}\mathbf{w} - \mathbf{w}^H \mathbf{A}^H \mathbf{d} + \mathbf{w}^H \mathbf{A}^H \mathbf{A}\mathbf{w}\end{aligned}$$

Minimizing with respect to **w**,

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

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

$$\Rightarrow \mathbf{A}^H \mathbf{A}\hat{\mathbf{w}} = \mathbf{A}^H \mathbf{d} \qquad \text{[Deterministic normal equation]}$$

Note: $\mathbf{A}$ is not generally square, and thus not invertible, but $\mathbf{A}^H\mathbf{A}$ is square and generally invertible

$$\begin{aligned} \mathbf{A}^H\mathbf{A}\hat{\mathbf{w}} &= \mathbf{A}^H\mathbf{d} \\ \Rightarrow \hat{\mathbf{w}} &= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{d} \end{aligned}$$

The deterministic normal equation can be rearranged as

$$\begin{aligned} \mathbf{A}^H\mathbf{A}\hat{\mathbf{w}} - \mathbf{A}^H\mathbf{d} &= \mathbf{0} \\ \mathbf{A}^H(\mathbf{A}\hat{\mathbf{w}} - \mathbf{d}) &= \mathbf{0} \qquad [\text{or using } \boldsymbol{\varepsilon}_{\min} = \mathbf{d} - \mathbf{A}\hat{\mathbf{w}}] \\ \mathbf{A}^H\boldsymbol{\varepsilon}_{\min} &= \mathbf{0} \end{aligned}$$

Observation: The LS orthogonality principle states that the estimate error $\boldsymbol{\varepsilon}_{\min}$ is orthogonal to the row vectors of the data matrix $\mathbf{A}^H$

Objective: Determine the minimum sum of squared errors ($e_{\min}$)

$$\begin{aligned} e_{\min} &= \boldsymbol{\varepsilon}_{\min}^H \boldsymbol{\varepsilon}_{\min} \\ &= (\mathbf{d}^H - \hat{\mathbf{w}}^H \mathbf{A}^H)(\mathbf{d} - \mathbf{A}\hat{\mathbf{w}}) \\ &= \mathbf{d}^H\mathbf{d} - \hat{\mathbf{w}}^H\mathbf{A}^H\mathbf{d} - \mathbf{d}^H\mathbf{A}\hat{\mathbf{w}} + \hat{\mathbf{w}}^H\mathbf{A}^H\mathbf{A}\hat{\mathbf{w}} \end{aligned}$$

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

$$\begin{aligned} e_{\min} &= \mathbf{d}^H\mathbf{d} - \underbrace{\hat{\mathbf{w}}^H\mathbf{A}^H\mathbf{d}}_{\hat{\mathbf{w}}^H\mathbf{A}^H\mathbf{A}\hat{\mathbf{w}}} - \mathbf{d}^H\mathbf{A}\hat{\mathbf{w}} + \hat{\mathbf{w}}^H\mathbf{A}^H\mathbf{A}\hat{\mathbf{w}} \\ &= \mathbf{d}^H\mathbf{d} - \mathbf{d}^H\mathbf{A}\hat{\mathbf{w}} \end{aligned}$$

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

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

Note that

$$\mathbf{d}^H\mathbf{d} = \sum_{i=M}^{N} |d(i)|^2 \qquad \text{[energy of desired response]}$$

Consider again the deterministic normal equation

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

Note that

$$\begin{aligned}
\mathbf{A}^H\mathbf{A} &= [\mathbf{x}(M), \mathbf{x}(M+1), \cdots, \mathbf{x}(N)]\begin{bmatrix} \mathbf{x}^H(M) \\ \mathbf{x}^H(M+1) \\ \vdots \\ \mathbf{x}^H(N) \end{bmatrix} \\
&= \sum_{i=M}^{N} \mathbf{x}(i)\mathbf{x}^H(i) \\
&= \mathbf{\Phi} \qquad [\text{time averaged correlation matrix, size } M \times M\,]
\end{aligned}$$

From $\mathbf{\Phi} = \sum_{i=M}^{N} \mathbf{x}(i)\mathbf{x}^H(i)$ it can be shown that:

1. $\mathbf{\Phi}$ is Hermitian
2. $\mathbf{\Phi}$ is nonnegative definite

   To prove this, note that for any $\mathbf{a}$

$$\begin{aligned} \mathbf{a}^H\mathbf{\Phi}\mathbf{a} &= \sum_{i=M}^{N} \mathbf{a}^H\mathbf{x}(i)\mathbf{x}^H(i)\mathbf{a} \\ &= \sum_{i=M}^{N} [\mathbf{a}^H\mathbf{x}(i)][\mathbf{a}^H\mathbf{x}(i)]^H \\ &= \sum_{i=M}^{N} |\mathbf{a}^H\mathbf{x}(i)|^2 \geq 0 \end{aligned}$$

3. From (1) and (2) we can prove that the eigenvalues of $\mathbf{\Phi}$ are real and nonnegative

The deterministic normal equation,

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

also employs

$$\begin{aligned}\mathbf{A}^H\mathbf{d} &= [\mathbf{x}(M), \mathbf{x}(M+1), \cdots, \mathbf{x}(N)]\begin{bmatrix} d^*(M) \\ d^*(M+1) \\ \vdots \\ d^*(N) \end{bmatrix} \\ &= \sum_{i=M}^{N} \mathbf{x}(i)d^*(i) \\ &= \boldsymbol{\theta} \qquad [\text{Time averaged cross-correlation vector}, \text{ size } M \times 1]\end{aligned}$$

Thus the deterministic normal equation, $\mathbf{A}^H\mathbf{A}\hat{\mathbf{w}} = \mathbf{A}^H\mathbf{d}$, reduces to

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

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

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

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

$$\begin{aligned} e_{\min} &= \mathbf{d}^H\mathbf{d} - \underbrace{\mathbf{d}^H\mathbf{A}}_{\boldsymbol{\theta}^H}\underbrace{(\mathbf{A}^H\mathbf{A})^{-1}}_{\mathbf{\Phi}^{-1}}\underbrace{\mathbf{A}^H\mathbf{d}}_{\boldsymbol{\theta}} \\ &= e_d - \boldsymbol{\theta}^H\mathbf{\Phi}^{-1}\boldsymbol{\theta} \end{aligned}$$

where e_d is the energy of desired signal

Consider again the orthogonality principle

$$\mathbf{A}^H \boldsymbol{\varepsilon}_{\min} = \mathbf{0}$$

Recall that $\hat{\mathbf{d}} = \mathbf{A}\hat{\mathbf{w}}$. Thus

$$\begin{aligned} \mathbf{A}^H \boldsymbol{\varepsilon}_{\min} &= \mathbf{0} \\ \Rightarrow \hat{\mathbf{w}}^H \mathbf{A}^H \boldsymbol{\varepsilon}_{\min} &= \hat{\mathbf{w}}^H \mathbf{0} \\ \Rightarrow \hat{\mathbf{d}}^H \boldsymbol{\varepsilon}_{\min} &= \mathbf{0} \end{aligned}$$

Result: The minimum estimation error vector, $\boldsymbol{\varepsilon}_{\min}$, is orthogonal to the data matrix $\mathbf{A}^H$ and the LS estimate $\hat{\mathbf{d}}$

Objective: Analyze the Least Squares solution in terms of

- Bias – it 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

$$\begin{aligned} d(i) &= \sum_{k=0}^{M-1} w_{0k}^* x(i-k) + e_0(i) \\ &= \mathbf{w}_0^H \mathbf{x}(i) + e_0(i) \end{aligned}$$

$e_0(i)$ is the unobservable measurement error

$\Rightarrow$ $e_0(i)$ is white (uncorrelated) with zero mean and variance σ^2

Express the desired signal in vector form

$$\mathbf{d} = \mathbf{A}\mathbf{w}_0 + \boldsymbol{\varepsilon}_0$$

where $\boldsymbol{\varepsilon}_0^H = [e_0(M), e_0(M+1), \cdots, e_0(N)]$

Objective: Evaluate the bias of $\hat{\mathbf{w}}$

Recall that

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

Using $\mathbf{d} = \mathbf{A}\mathbf{w}_0 + \boldsymbol{\varepsilon}_0$ in the above

$$\begin{aligned}
\hat{\mathbf{w}} &= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{d} \\
&= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H(\mathbf{A}\mathbf{w}_0 + \boldsymbol{\varepsilon}_0) \\
&= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{A}\mathbf{w}_0 + (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\boldsymbol{\varepsilon}_0 \\
&= \mathbf{w}_0 + (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\boldsymbol{\varepsilon}_0 \qquad (*)
\end{aligned}$$

Note $\mathbf{A}$ is fixed. Thus taking the expectation of $(*)$ yields

$$\begin{aligned}
E\{\hat{\mathbf{w}}\} &= \mathbf{w}_0 + (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H E\{\boldsymbol{\varepsilon}_0\} \\
&= \mathbf{w}_0
\end{aligned}$$

Result: The LS estimate, $\hat{\mathbf{w}}$, is unbiased

Objective: Evaluate the covariance of $\hat{\mathbf{w}}$

Note that from $(*)$

$$
\begin{aligned}
\hat{\mathbf{w}} &= \mathbf{w}_0 + (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\boldsymbol{\varepsilon}_0 \\
\Rightarrow \hat{\mathbf{w}} - \mathbf{w}_0 &= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\boldsymbol{\varepsilon}_0
\end{aligned}
$$

Thus

$$
\begin{aligned}
\text{cov}[\hat{\mathbf{w}}] &= E\{(\hat{\mathbf{w}} - \mathbf{w}_0)(\hat{\mathbf{w}} - \mathbf{w}_0)^H\} \\
&= E\{(\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\boldsymbol{\varepsilon}_0\boldsymbol{\varepsilon}_0^H\mathbf{A}(\mathbf{A}^H\mathbf{A})^{-1}\} \\
&= \boldsymbol{\Phi}^{-1}\mathbf{A}^H\underbrace{E\{\boldsymbol{\varepsilon}_0\boldsymbol{\varepsilon}_0^H\}}_{\sigma^2\mathbf{I}}\mathbf{A}\boldsymbol{\Phi}^{-1} \\
&= \sigma^2\boldsymbol{\Phi}^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^{-1} = \sigma^2\boldsymbol{\Phi}^{-1} \qquad (\maltese_1)
\end{aligned}
$$

Result: The covariance of $\hat{\mathbf{w}}$ 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 $\hat{\mathbf{w}}$ is the Best Linear Unbiased Estimate (BLUE)

- Consider any linear unbiased estimate $\tilde{\mathbf{w}}$
- Note that $\tilde{\mathbf{w}}$ is a linear function of the observed date and can thus be written as

$$\tilde{\mathbf{w}} = \mathbf{B}\mathbf{d}$$

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

Substituting $\mathbf{d} = \mathbf{A}\mathbf{w}_0 + \boldsymbol{\varepsilon}_0$ into the above,

$$\begin{aligned}
\tilde{\mathbf{w}} &= \mathbf{B}\mathbf{A}\mathbf{w}_0 + \mathbf{B}\boldsymbol{\varepsilon}_0 \qquad (*) \\
\Rightarrow E\{\tilde{\mathbf{w}}\} &= \mathbf{B}\mathbf{A}\mathbf{w}_0 \\
\Rightarrow \mathbf{B}\mathbf{A} &= \mathbf{I} \qquad \text{[since } \tilde{\mathbf{w}} \text{ unbiased]}
\end{aligned}$$

Thus $\mathbf{B}\mathbf{A} = \mathbf{I}$ and $(*) \Rightarrow$

$$\tilde{\mathbf{w}} = \mathbf{w}_0 + \mathbf{B}\boldsymbol{\varepsilon}_0$$

Rearranging $\tilde{\mathbf{w}} = \mathbf{w}_0 + \mathbf{B}\boldsymbol{\varepsilon}_0$,

$$\begin{aligned}
\tilde{\mathbf{w}} - \mathbf{w}_0 &= \mathbf{B}\boldsymbol{\varepsilon}_0 \\
\Rightarrow \mathrm{cov}[\tilde{\mathbf{w}}] &= E\{(\tilde{\mathbf{w}} - \mathbf{w}_0)(\tilde{\mathbf{w}} - \mathbf{w}_0)^H\} \\
&= E\{\mathbf{B}\boldsymbol{\varepsilon}_0\boldsymbol{\varepsilon}_0^H\mathbf{B}^H\} \\
&= \sigma^2\mathbf{B}\mathbf{B}^H \qquad (\maltese_2)
\end{aligned}$$

Now define

$$\begin{aligned}
\Psi &= \mathbf{B} - (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H \\
\Rightarrow \Psi\Psi^H &= [\mathbf{B} - \boldsymbol{\Phi}^{-1}\mathbf{A}^H][\mathbf{B}^H - \mathbf{A}\boldsymbol{\Phi}^{-1}] \\
&= \mathbf{B}\mathbf{B}^H - \underbrace{\mathbf{B}\mathbf{A}}_{\mathbf{I}}\boldsymbol{\Phi}^{-1} - \boldsymbol{\Phi}^{-1}\underbrace{\mathbf{A}^H\mathbf{B}^H}_{\mathbf{I}} + \underbrace{\boldsymbol{\Phi}^{-1}\mathbf{A}^H\mathbf{A}\boldsymbol{\Phi}^{-1}}_{\boldsymbol{\Phi}^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^{-1}} \\
&= \mathbf{B}\mathbf{B}^H - \boldsymbol{\Phi}^{-1} - \boldsymbol{\Phi}^{-1} + \boldsymbol{\Phi}^{-1} \\
&= \mathbf{B}\mathbf{B}^H - \boldsymbol{\Phi}^{-1} \\
&= \mathbf{B}\mathbf{B}^H - (\mathbf{A}^H\mathbf{A})^{-1}
\end{aligned}$$

Observation: The diagonal elements at $\Psi\Psi^H$ must be ≥ 0

Thus $\Psi\Psi^H = \mathbf{B}\mathbf{B}^H - (\mathbf{A}^H\mathbf{A})^{-1} \Rightarrow$

$$\begin{aligned} \text{diag}[\mathbf{B}\mathbf{B}^H] &\geq \text{diag}[(\mathbf{A}^H\mathbf{A})^{-1}] \\ \Rightarrow \text{diag}[\sigma^2\mathbf{B}\mathbf{B}^H] &\geq \text{diag}[\sigma^2(\mathbf{A}^H\mathbf{A})^{-1}] \qquad (*) \end{aligned}$$

But recall from $(\maltese_1)$ and $(\maltese_2)$ that

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

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

$$\text{variance}[\tilde{w}_i] \geq \text{variance}[\hat{w}_i] \quad i = 1, 2, \cdots, M$$

Thus the weights in $\hat{\mathbf{w}}$ have lower variance than any other linear estimates

Result: The LS estimate $\hat{\mathbf{w}}$ 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

$$\begin{aligned}\hat{\mathbf{w}} &= (\mathbf{A}^H\mathbf{A})^{-1}\mathbf{A}^H\mathbf{d} \\ &= \boldsymbol{\Phi}^{-1}\boldsymbol{\theta}\end{aligned}$$

where

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

- $(\mathbf{A}^H\mathbf{A})$ is $M \times M$ and inversion requires $O(M^3)$ multiplications and additions

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

- 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

$$\begin{aligned} e(i) &= d(i) - \mathbf{w}^H(n)\mathbf{x}(i) \\ \mathbf{x}(i) &= [x(i), x(i-1), \cdots, x(i-M+1)]^T \\ \mathbf{w}(n) &= [w_0(n), w_1(n), \cdots, w_{M-1}(n)]^T \end{aligned}$$

$\Rightarrow$ $\beta(n,i) \in (0,1]$ 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

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

where now

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

$$\boldsymbol{\theta}(n) = \sum_{i=1}^{n} \lambda^{n-i} \mathbf{x}(i) d^*(i)$$

The normal equation terms can be updated recursively,

$$\begin{aligned}
\mathbf{\Phi}(n) &= \sum_{i=1}^{n} \lambda^{n-i} \mathbf{x}(i)\mathbf{x}^H(i) \\
&= \lambda \underbrace{\left[\sum_{i=1}^{n-1} \lambda^{(n-1)-i} \mathbf{x}(i)\mathbf{x}^H(i)\right]}_{\mathbf{\Phi}(n-1)} + \mathbf{x}(n)\mathbf{x}^H(n) \\
&= \lambda \mathbf{\Phi}(n-1) + \mathbf{x}(n)\mathbf{x}^H(n)
\end{aligned}$$

Similarly

$$\begin{aligned}
\boldsymbol{\theta}(n) &= \sum_{i=1}^{n} \lambda^{n-i} \mathbf{x}(i) d^*(i) \\
&= \lambda \left[\sum_{i=1}^{n-1} \lambda^{(n-1)-i} \mathbf{x}(i) d^*(i)\right] + \mathbf{x}(n) d^*(n) \\
&= \lambda \boldsymbol{\theta}(n-1) + \mathbf{x}(n) d^*(n)
\end{aligned}$$

Aside: *Matrix inversion lemma*: If

$$\underbrace{\mathbf{A}}_{M\times M} = \underbrace{\mathbf{B}^{-1}}_{M\times M} + \underbrace{\mathbf{C}}_{M\times L}\underbrace{\mathbf{D}^{-1}}_{L\times L}\underbrace{\mathbf{C}^{H}}_{L\times M}$$

where $\mathbf{A}, \mathbf{B}, \mathbf{D}$ are positive definite (non-singular), then

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

Apply the lemma to

$$\mathbf{\Phi}(n) = \lambda\mathbf{\Phi}(n-1) + \mathbf{x}(n)\mathbf{x}^{H}(n)$$

Accordingly, set

$$\begin{array}{llll} \mathbf{A} = \mathbf{\Phi}(n) & [M \times M] & \mathbf{B}^{-1} = \lambda\mathbf{\Phi}(n-1) & [M \times M] \\ \mathbf{C} = \mathbf{x}(n) & [M \times 1] & \mathbf{D} = 1 & [1 \times 1] \end{array}$$

Utilizing

$$\begin{array}{ll} \mathbf{A} = \mathbf{\Phi}(n) & \mathbf{B}^{-1} = \lambda \mathbf{\Phi}(n-1) \\ \mathbf{C} = \mathbf{x}(n) & \mathbf{D} = 1 \end{array}$$

and

$$\mathbf{A}^{-1} = \mathbf{B} - \mathbf{BC}[\mathbf{D} + \mathbf{C}^H\mathbf{BC}]^{-1}\mathbf{C}^H\mathbf{B} \qquad (*)$$

we get

$$[\mathbf{D} + \mathbf{C}^H\mathbf{BC}]^{-1} = [1 + \lambda^{-1}\mathbf{x}^H(n)\mathbf{\Phi}^{-1}(n-1)\mathbf{x}(n)]^{-1}$$

which is a scalar. Thus evaluating $(*)$ yields

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

To simplify the result, let $\mathbf{P}(n) = \mathbf{\Phi}^{-1}(n)$ and

$$\underbrace{\mathbf{k}(n)}_{\text{Gain vector}} = \frac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{x}(n)}{1 + \lambda^{-1}\mathbf{x}^H(n)\mathbf{P}(n-1)\mathbf{x}(n)}$$

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

$$\begin{aligned}
\mathbf{\Phi}^{-1}(n) &= \lambda^{-1}\mathbf{\Phi}^{-1}(n-1) - \frac{\lambda^{-2}\mathbf{\Phi}^{-1}(n-1)\mathbf{x}(n)\mathbf{x}^H(n)\mathbf{\Phi}^{-1}(n-1)}{1+\lambda^{-1}\mathbf{x}^H(n)\mathbf{\Phi}^{-1}(n-1)\mathbf{x}(n)} \\
\Rightarrow \mathbf{P}(n) &= \lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^H(n)\mathbf{P}(n-1) \qquad (*)
\end{aligned}$$

Also, the gain vector can be simplified as

$$\begin{aligned}
\mathbf{k}(n) &= \frac{\lambda^{-1}\mathbf{P}(n-1)\mathbf{x}(n)}{1+\lambda^{-1}\mathbf{x}^H(n)\mathbf{P}(n-1)\mathbf{x}(n)} \qquad \text{[multiply by denom.]} \\
\Rightarrow \mathbf{k}(n) &= \lambda^{-1}\mathbf{P}(n-1)\mathbf{x}(n) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^H(n)\mathbf{P}(n-1)\mathbf{x}(n) \\
&= \underbrace{[\lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^H(n)\mathbf{P}(n-1)]}_{=\mathbf{P}(n) \text{ from } (*)}\mathbf{x}(n) \\
&= \mathbf{P}(n)\mathbf{x}(n) = \mathbf{\Phi}^{-1}(n)\mathbf{x}(n) \qquad (**)
\end{aligned}$$

We must now derive an update for the tap weight vector. Recall,

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

Using the recursion $\boldsymbol{\theta}(n) = \lambda\boldsymbol{\theta}(n-1) + \mathbf{x}(n)d^*(n)$ in the above

$$\hat{\mathbf{w}}(n) = \lambda\mathbf{P}(n)\boldsymbol{\theta}(n-1) + \mathbf{P}(n)\mathbf{x}(n)d^*(n) \qquad (***)$$

Using the update $(*)$

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

in the first $\mathbf{P}(n)$ term of $(***)$

$$\begin{aligned}\hat{\mathbf{w}}(n) &= \lambda\mathbf{P}(n)\boldsymbol{\theta}(n-1) + \mathbf{P}(n)\mathbf{x}(n)d^*(n) \\ &= \lambda[\lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^H(n)\mathbf{P}(n-1)]\boldsymbol{\theta}(n-1) \\ &\quad + \mathbf{P}(n)\mathbf{x}(n)d^*(n)\end{aligned}$$

$$
\begin{aligned}
\hat{\mathbf{w}}(n) &= \lambda[\lambda^{-1}\mathbf{P}(n-1) - \lambda^{-1}\mathbf{k}(n)\mathbf{x}^H(n)\mathbf{P}(n-1)]\boldsymbol{\theta}(n-1) \\
&\quad + \mathbf{P}(n)\mathbf{x}(n)d^*(n) \\
&= \underbrace{\mathbf{P}(n-1)\boldsymbol{\theta}(n-1)}_{\hat{\mathbf{w}}(n-1)} - \mathbf{k}(n)\mathbf{x}^H(n)\underbrace{\mathbf{P}(n-1)\boldsymbol{\theta}(n-1)}_{\hat{\mathbf{w}}(n-1)} \\
&\quad + \mathbf{P}(n)\mathbf{x}(n)d^*(n) \\
&= \hat{\mathbf{w}}(n-1) - \mathbf{k}(n)\mathbf{x}^H(n)\hat{\mathbf{w}}(n-1) + \underbrace{\mathbf{P}(n)\mathbf{x}(n)}_{=\mathbf{k}(n) \text{ from } (**)} d^*(n) \\
&= \hat{\mathbf{w}}(n-1) - \mathbf{k}(n)[\mathbf{x}^H(n)\hat{\mathbf{w}}(n-1) - d^*(n)] \\
&= \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\alpha^*(n)
\end{aligned}
$$

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

Observation: Difference between $e(n)$ and $\alpha(n)$:

$e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{x}(n) \Rightarrow$ a posteriori error

$\alpha(n) = d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{x}(n) \Rightarrow$ a priori error

RLS Algorithm Summary

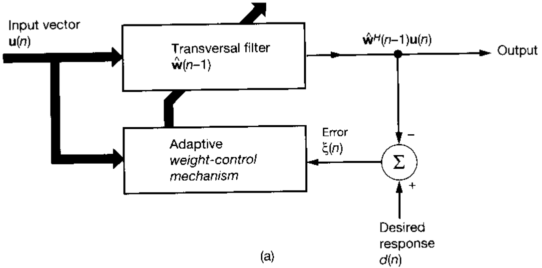
1. Given a new sample $x(n)$, update the gain vector

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

2. Update the innovation: $\alpha(n) = d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{x}(n)$
3. Update the tap weight vector: $\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\alpha^*(n)$
4. Update inverse correlation matrix

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

Initial Conditions: $\hat{\mathbf{w}}(0) = \mathbf{0}$ and $\mathbf{\Phi}(0) = \delta\mathbf{I}$, where δ is a small positive constant, $\delta \approx 0.01\sigma_x^2$.



Algorithm Comparison: RLS and LMS algorithm terms:

Entity	RLS	LMS
Error	$\alpha(n) = d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{x}(n)$ (a priori error)	$e(n) = d(n) - \hat{\mathbf{w}}^H(n)\mathbf{x}(n)$ (a posteriari error)
Weight Update	$\hat{\mathbf{w}}(n) = \hat{\mathbf{w}}(n-1) + \mathbf{k}(n)\alpha^*(n)$	$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{x}(n)e^*(n)$
Gain of error update	$\left(\dfrac{\lambda^{-1}\mathbf{P}(n-1)}{1+\lambda^{-1}\mathbf{x}^H(n)\mathbf{P}(n-1)\mathbf{x}(n)}\right)\mathbf{x}(n)$	$(\mu)\mathbf{x}(n)$

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

Complexity		
Stage	$O_\times$	O_+
(1)	M	$M-1$
(2)	0	1
(3)	$M+1$	M
Total complexity per iteration	$O_\times(2M+1)$	$O_+(2M)$

Case 2 – The LS algorithm: Algorithm solves

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

and has stages:

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

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

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

Complexity		
Stage	$O_\times$	O_+
(1) numerator (1) denominator (1) division	M^2 $\underline{M^2} + M$ M	$M(M-1)$ $\underline{M(M-1)} + M$
(2)	M	M
(3)	M	M
(4)	$\underline{M^2} + M^2$	$\underline{M(M-1)} + M^2$
Total complexity per iteration	$O_\times(3M^2 + 4M)$	$O_+(3M^2 + M)$

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

$$d(n) = \mathbf{w}_0^H \mathbf{x}(n) + e_0(n)$$

   where $e_0(n)$ is white with variance σ^2.

2. $\lambda = 1$ and $n \geq M$.

Then

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

where

$$\mathbf{\Phi}(n) = \sum_{i=1}^{n} \mathbf{x}(i)\mathbf{x}^H(i) \quad \text{and} \quad \boldsymbol{\theta}(n) = \sum_{i=1}^{n} \mathbf{x}(i)d^*(i)$$

Substituting $d^*(n) = \mathbf{x}^H(n)\mathbf{w}_0 + e_0^*(n)$ into $\boldsymbol{\theta}(n)$

$$\begin{aligned}
\boldsymbol{\theta}(n) &= \sum_{i=1}^{n} \mathbf{x}(i)[\mathbf{x}^H(i)\mathbf{w}_0 + e_0^*(i)] \\
&= \sum_{i=1}^{n} \mathbf{x}(i)\mathbf{x}^H(i)\mathbf{w}_0 + \sum_{i=1}^{n} \mathbf{x}(i)e_0^*(i) \\
&= \boldsymbol{\Phi}(n)\mathbf{w}_0 + \sum_{i=1}^{n} \mathbf{x}(i)e_0^*(i)
\end{aligned}$$

Thus

$$\begin{aligned}
\hat{\mathbf{w}}(n) &= \boldsymbol{\Phi}^{-1}(n)\boldsymbol{\theta}(n) \\
&= \boldsymbol{\Phi}^{-1}(n)[\boldsymbol{\Phi}(n)\mathbf{w}_0 + \sum_{i=1}^{n} \mathbf{x}(i)e_0^*(i)] \\
&= \mathbf{w}_0 + \boldsymbol{\Phi}^{-1}(n)\sum_{i=1}^{n} \mathbf{x}(i)e_0^*(i) \qquad (*)
\end{aligned}$$

Note that $E\{A\} = E\{E\{A|B\}\}$. Thus

$$\begin{aligned} \hat{\mathbf{w}}(n) &= \mathbf{w}_0 + \boldsymbol{\Phi}^{-1}(n)\sum_{i=1}^{n}\mathbf{x}(i)e_0^*(i) \\ \Rightarrow E\{\hat{\mathbf{w}}(n)\} &= \mathbf{w}_0 + E\{E\{\boldsymbol{\Phi}^{-1}(n)\sum_{i=1}^{n}\mathbf{x}(i)e_0^*(i)|x(i), i = 1, 2, \cdots, n\}\} \\ &= \mathbf{w}_0 + E\{\boldsymbol{\Phi}^{-1}(n)\sum_{i=1}^{n}\mathbf{x}(i)E\{e_0^*(i)\}\} = \mathbf{w}_0 \end{aligned}$$

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

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

$$\hat{\mathbf{w}}(n) = \mathbf{w}_0 + \mathbf{\Phi}^{-1}(n)\sum_{i=1}^{n}\mathbf{x}(i)e_0^*(i)$$

which gives

$$\boldsymbol{\varepsilon}(n) = \hat{\mathbf{w}}(n) - \mathbf{w}_0 = \mathbf{\Phi}^{-1}(n)\sum_{i=1}^{n}\mathbf{x}(i)e_0^*(i)$$

Thus the weight error correlation matrix is

$$\begin{aligned}
\mathbf{K}(n) &= E\{\boldsymbol{\varepsilon}(n)\boldsymbol{\varepsilon}^H(n)\} \\
&= E\{\mathbf{\Phi}^{-1}(n)\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{x}(i)e_0^*(i)e_0(j)\mathbf{x}^H(j)\right)\mathbf{\Phi}^{-1}(n)\}
\end{aligned}$$

Again using $E\{A\} = E\{E\{A|B\}\}$ yields

$$
\begin{aligned}
\mathbf{K}(n) &= E\left\{\boldsymbol{\Phi}^{-1}(n)\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbf{x}(i)\underbrace{E\{e_0^*(i)e_0(j)\}}_{\sigma^2\delta(i-j)}\mathbf{x}^H(j)\right)\boldsymbol{\Phi}^{-1}(n)\right\} \\
&= \sigma^2 E\left\{\boldsymbol{\Phi}^{-1}(n)\left(\sum_{i=1}^{n}\mathbf{x}(i)\mathbf{x}^H(i)\right)\boldsymbol{\Phi}^{-1}(n)\right\} \\
&= \sigma^2 E\{\boldsymbol{\Phi}^{-1}(n)\boldsymbol{\Phi}(n)\boldsymbol{\Phi}^{-1}(n)\} \\
&= \sigma^2 E\{\boldsymbol{\Phi}^{-1}(n)\}
\end{aligned}
$$

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

$$E\{\boldsymbol{\Phi}^{-1}(n)\} = \frac{1}{n-M-1}\mathbf{R}^{-1} \quad n > M+1$$

Using $\mathbf{K}(n) = \frac{\sigma^2}{n-M-1}\mathbf{R}^{-1}$ and the trace

$$
\begin{aligned}
E\{||\boldsymbol{\varepsilon}(n)||^2\} &= E\{\boldsymbol{\varepsilon}^H(n)\boldsymbol{\varepsilon}(n)\} \\
&= E\{\text{trace}[\boldsymbol{\varepsilon}^H(n)\boldsymbol{\varepsilon}(n)]\} \\
&= E\{\text{trace}[\boldsymbol{\varepsilon}(n)\boldsymbol{\varepsilon}^H(n)]\} \\
&= \text{trace}\, E\{\boldsymbol{\varepsilon}(n)\boldsymbol{\varepsilon}^H(n)\} \\
&= \text{trace}[\mathbf{K}(n)] \\
&= \frac{\sigma^2}{n-M-1}\text{trace}[\mathbf{R}^{-1}] \\
&= \frac{\sigma^2}{n-M-1}\sum_{i=1}^{M}\frac{1}{\lambda_i} \qquad n > M+1
\end{aligned}
$$

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

$$\begin{aligned}
\alpha(n) &= d(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{x}(n) \\
&= d(n) - \hat{d}_0(n) + \hat{d}_0(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{x}(n) \\
&= e_0(n) + \mathbf{w}_0^H\mathbf{x}(n) - \hat{\mathbf{w}}^H(n-1)\mathbf{x}(n) \\
&= e_0(n) - \boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)
\end{aligned}$$

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

$$\begin{aligned}
J_\alpha(n) &= E\{|\alpha(n)|^2\} \\
&= E\{[e_0^*(n) - \mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)][e_0(n) - \boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)]\} \\
&= E\{|e_0(n)|^2\} - E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)e_0(n)\} \\
&\quad - E\{\varepsilon^H(n-1)\mathbf{x}(n)e_0^*(n)\} + E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)\}
\end{aligned}$$

To analyze $J_\alpha(n)$, consider each term individually

$$
\begin{aligned}
J_\alpha(n) &= E\{|e_0(n)|^2\} - E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)e_0(n)\} \\
&\quad - E\{\varepsilon^H(n-1)\mathbf{x}(n)e_0^*(n)\} + E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)\}
\end{aligned}
$$

Term: $E\{|e_0(n)|^2\}$.
Clearly,

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

Term: $E\{\varepsilon^H(n-1)\mathbf{x}(n)e_0^*(n)\}$.
By the independence theorem, $\boldsymbol{\varepsilon}(n-1)$ is independent of $\mathbf{x}(n)$ and $e_0(n)$. Thus,

$$
\begin{aligned}
E\{\varepsilon^H(n-1)\mathbf{x}(n)e_0^*(n)\} &= E\{\varepsilon^H(n-1)\}E\{\mathbf{x}(n)e_0^*(n)\} \\
&= 0
\end{aligned}
$$

where the final result is due to the orthogonality principle.

Term: $E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)e_0(n)\} \rightarrow 0$ by similar arguments

$$
\begin{aligned}
J_\alpha(n) &= E\{|e_0(n)|^2\} - E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)e_0(n)\} \\
&\quad - E\{\varepsilon^H(n-1)\mathbf{x}(n)e_0^*(n)\} + E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)\}
\end{aligned}
$$

Term: $E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)\}$

$$
\begin{aligned}
E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)\} &= E\{\mathrm{trace}[\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)]\} \\
&= E\{\mathrm{trace}[\mathbf{x}(n)\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)]\}
\end{aligned}
$$

Invoking the independence theorem

$$
\begin{aligned}
&E\{\mathbf{x}^H(n)\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)\mathbf{x}(n)\} \\
&\qquad = \mathrm{trace}[E\{\mathbf{x}(n)\mathbf{x}^H(n)\}E\{\boldsymbol{\varepsilon}(n-1)\boldsymbol{\varepsilon}^H(n-1)\}] \\
&\qquad = \mathrm{trace}[\mathbf{RK}(n-1)]
\end{aligned}
$$

Utilizing $\mathbf{K}(n-1) = \frac{\sigma^2}{n-M-2}\mathbf{R}^{-1}$ and substituting back each of the components

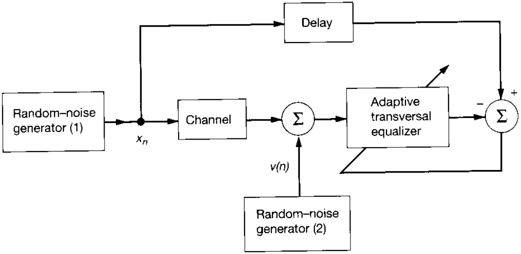
$$\begin{aligned} J_\alpha(n) &= \sigma^2 + \mathrm{trace}[\mathbf{RK}(n-1)] \\ &= \sigma^2 + \frac{M\sigma^2}{n-M-2} \quad n > M+1 \end{aligned}$$

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 $\mathbf{\Phi}(n)$

Example

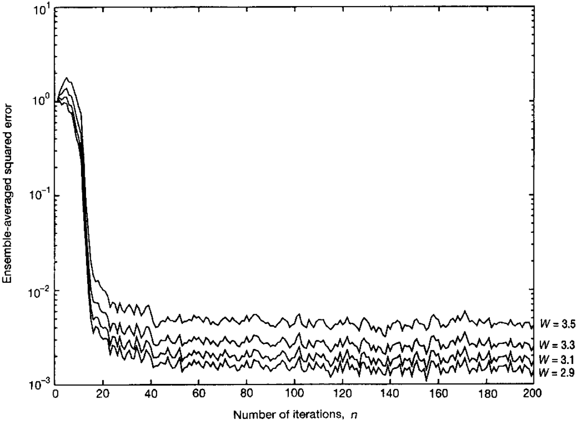
Consider again the channel equalization problem



where

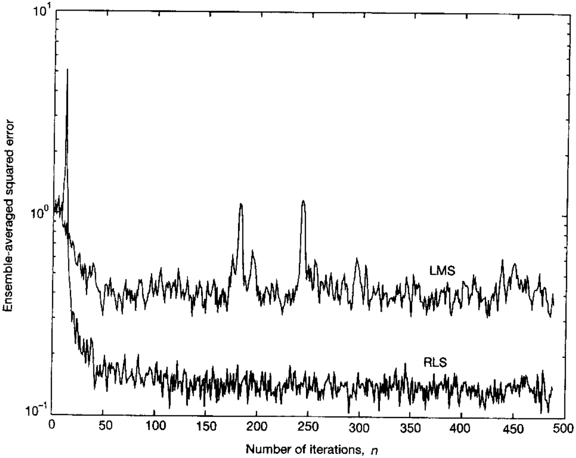
$$h_n = \begin{cases} \frac{1}{2}[1+\cos(\frac{2\pi}{W}(n-1))] & n=1,2,3 \\ 0 & \text{otherwise} \end{cases}$$

- 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