Nonlinear Signal Processing
ELEG 833

Gonzalo R. Arce

Department of Electrical and Computer Engineering
University of Delaware
arce@ee.udel.edu

Fall 2008

Outline

Example: Adaptive Equalization

The LMS algorithm will be used to equalize a linear dispersive channel.

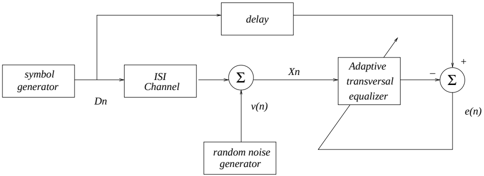

Figure: Block diagram of an adaptive equalization experiment.

The symbol generator provides a test signal $D_n = \pm 1$ applied to the real valued channel $h_n = [0.2194, 1.0000, 0.2194]$, the output of the channel is corrupted by Gaussian noise $\nu(n)$ which has zero mean and variance $\sigma_\nu^2 = 0.001$

The equalizer has 11 taps, and since the channel response is symmetric (around $n = 1$), the equalizer must be also symmetric (around $n = 5$). The channel input is then delayed by $\Delta = 1 + 5 = 6$ to provide the desired response.
The correlation matrix of the 11-tap inputs of the equalizer will be given by:

$$\mathbf{R} = \begin{bmatrix} r(0) & r(1) & r(2) & 0 & \cdots & 0 \\ r(1) & r(0) & r(1) & r(2) & \cdots & 0 \\ r(2) & r(1) & r(0) & r(1) & \cdots & 0 \\ 0 & r(2) & r(1) & r(0) & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & r(0) \end{bmatrix}$$

where:

$$\begin{aligned} r(0) &= h_0^2 + h_1^2 + h_2^2 + \sigma_\nu^2 &= 1.0973 \\ r(1) &= h_0 h_1 + h_1 h_2 &= 0.4388 \\ r(2) &= h_0 h_2 &= 0.0481 \end{aligned}$$

and the vector p will be given by:

$$\mathbf{p} = E\{D(n)\mathbf{X}(n)\} = [0,\ 0,\ 0,\ 0,\ h_2,\ h_1,\ h_0,\ 0,\ 0,\ 0,\ 0]$$

Then the Wiener solution will be:

$$\begin{aligned} \mathbf{W}_0 &= \mathbf{R}^{-1}\mathbf{p} \\ &= [-0.0006,\ 0.0031,\ -0.0136,\ 0.0590,\ -0.2561,\ 1.1110, \\ &\quad -0.2561,\ 0.0590,\ -0.0136,\ 0.0031,\ -0.0006] \end{aligned} \tag{1}$$

The eigenvalues of the correlation matrix **R** are:

$$\lambda = [0.3339,\ 0.3890,\ 0.4842,\ 0.6219,\ 0.8016,\ 1.0172,\ 1.2565,\ 1.5005,\ 1.7263,\ 1.9097,\ 2.0295]$$

so the LMS algorithm will converge if we choose:

$$\mu \leq \frac{2}{\lambda_{max}} = 0.9860$$

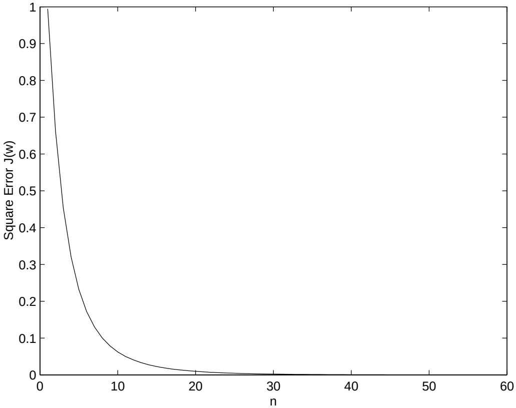

Figure: Steepest descent learning curve, single realization ($\mu = 0.09$)

$$
\begin{aligned}
\mathbf{W}(n+1) &= \mathbf{W}(n) + \mu[\mathbf{p} - \mathbf{R}\mathbf{W}(n)] \\
J(\mathbf{W}(n)) &= \sigma_d^2 - 2\mathbf{W}(n)^T\mathbf{p} + \mathbf{W}(n)^T\mathbf{R}\mathbf{W}(n)
\end{aligned}
$$

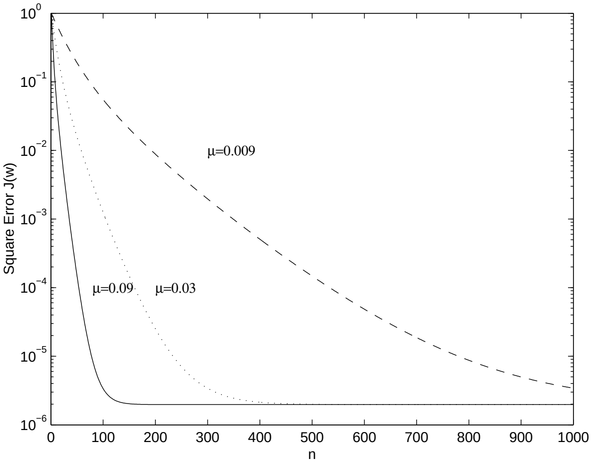

Figure: Steepest descent learning curves for different values of μ. Logarithmic plot

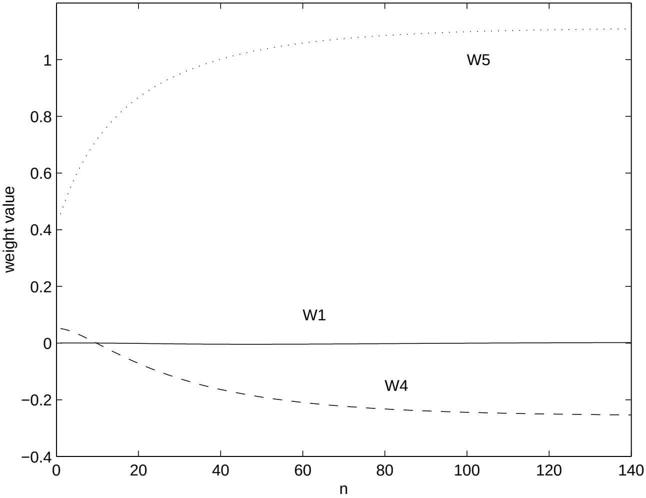

Figure: Convergence of the weights in the steepest descent algorithm

The final weights resulting from the steepest descent algorithm were:

$$W_{0.09} = [-0.0007,\ 0.0032,\ -0.0137,\ 0.0594,\ -0.2571,\ 1.1127,\ -0.2571,\ 0.0594,\ -0.0137,\ 0.0032,\ -0.0007]$$

$$W_{0.03} = [-0.0007,\ 0.0032,\ -0.0137,\ 0.0594,\ -0.2571,\ 1.1126,\ -0.2571,\ 0.0594,\ -0.0137,\ 0.0032,\ -0.0007]$$

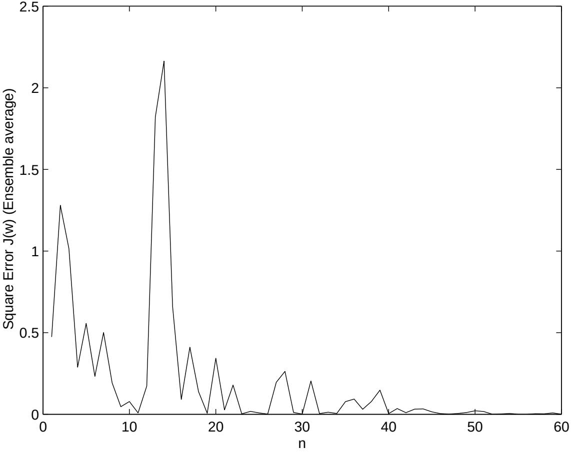

FIGURE: LMS learning curve, single realization ($\mu = 0.09$)

$$\mathbf{W}(n+1) = \mathbf{W}(n) + \mu \mathbf{X}(n)\ e(n)$$

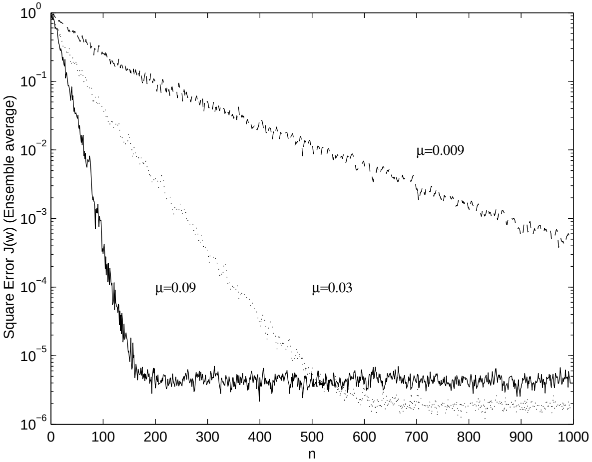

Figure: LMS learning curves for different values of μ (average). Logarithmic plot

The misadjustment for each of the previous plots will be

$$\begin{aligned} \mathcal{M} &= \frac{\mu}{2}\sum_{i=1}^{N}\lambda_i \\ &= 0.5427(\mu = 0.09) \\ &= 0.1809(\mu = 0.03) \\ &= 0.0543(\mu = 0.009) \end{aligned}$$

And the approximated floor errors for the plots that converged are:

$$5.273 \times 10^{-}6 \text{ for } (\mu = 0.09)$$
$$2.278 \times 10^{-}6 \text{ for } (\mu = 0.03)$$

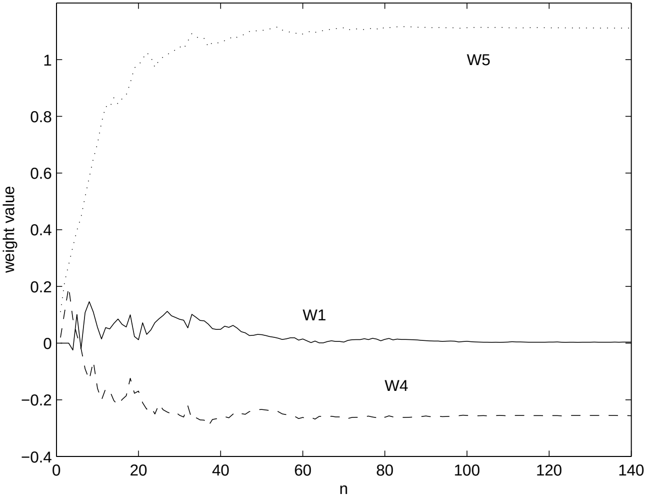

Figure: Convergence of the weights in the LMS algorithm

The final weights resulting from the LMS algorithm were:

$$\begin{aligned} W_{0.09} &= [-0.0004,\ 0.0029,\ -0.0128,\ 0.0593,\ -0.2571,\ 1.1133, \\ &\quad -0.2573,\ 0.0598,\ -0.0138,\ 0.0029,\ -0.0008] \end{aligned}$$

$$\begin{aligned} W_{0.03} &= [-0.0006,\ 0.0031,\ -0.0135,\ 0.0595,\ -0.2572,\ 1.1130, \\ &\quad -0.2571,\ 0.0597,\ -0.0138,\ 0.0031,\ -0.0007] \end{aligned}$$

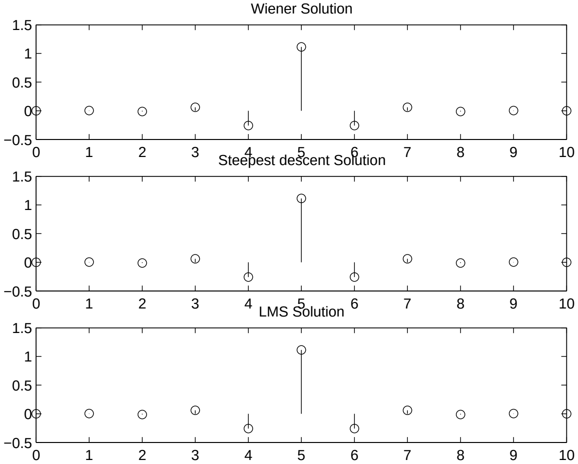

Figure: Comparison of the weights obtained with the three different methods