

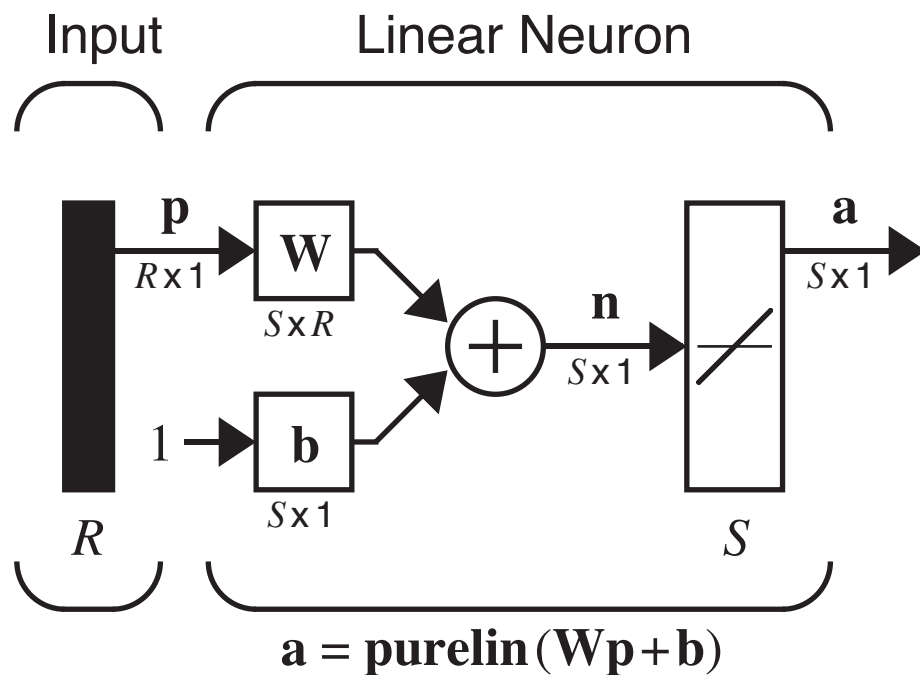
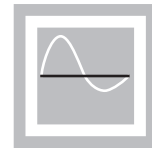


Widrow-Hoff Öğrenme

(Widrow-Hoff Learning)

En Küçük Ortalama Kareler Yönt.
(LMS Algorithm)

ADALINE Network

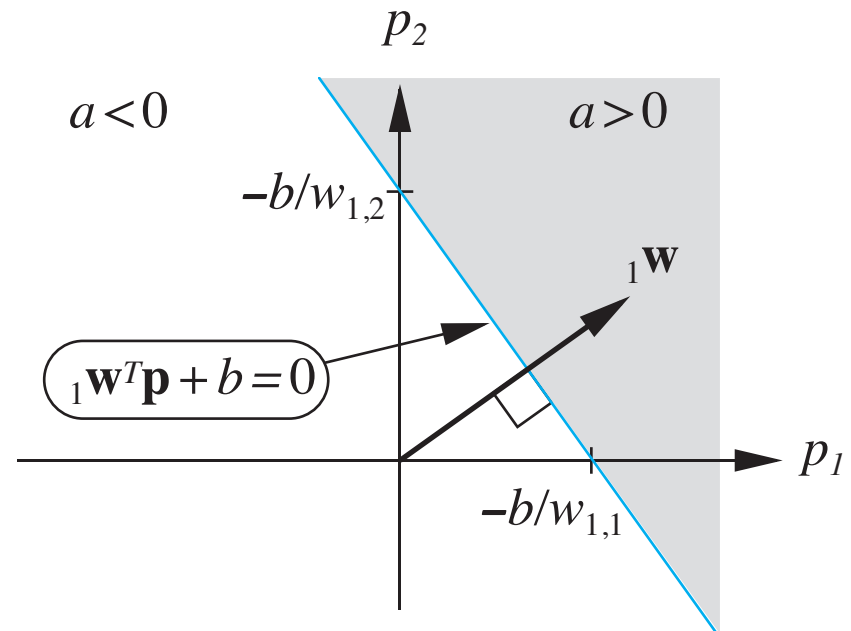
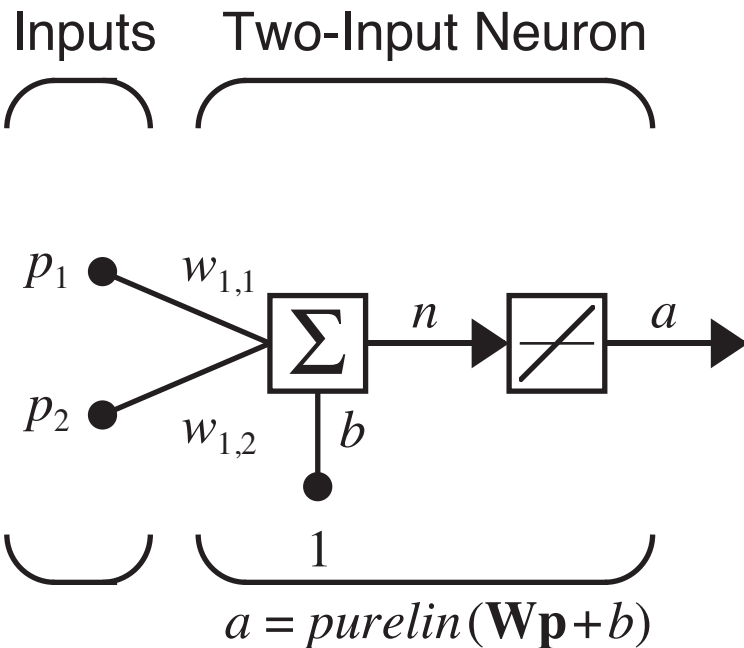
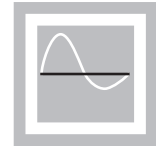


$$\mathbf{a} = \text{purelin}(\mathbf{Wp} + \mathbf{b}) = \mathbf{Wp} + \mathbf{b}$$

$$a_i = \text{purelin}(n_i) = \text{purelin}({}_i\mathbf{w}^T \mathbf{p} + b_i) = {}_i\mathbf{w}^T \mathbf{p} + b_i$$

$${}_i\mathbf{w} = \begin{bmatrix} w_{i,1} \\ w_{i,2} \\ \vdots \\ w_{i,R} \end{bmatrix}$$

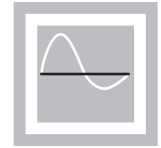
Two-Input ADALINE



$$a = \text{purelin}(n) = \text{purelin}({}_1\mathbf{w}^T \mathbf{p} + b) = {}_1\mathbf{w}^T \mathbf{p} + b$$

$$a = {}_1\mathbf{w}^T \mathbf{p} + b = w_{1,1}p_1 + w_{1,2}p_2 + b$$

Mean Square Error



Training Set:

$$\{\mathbf{p}_1, \mathbf{t}_1\}, \{\mathbf{p}_2, \mathbf{t}_2\}, \dots, \{\mathbf{p}_Q, \mathbf{t}_Q\}$$

Input: \mathbf{p}_q Target: \mathbf{t}_q

Notation:

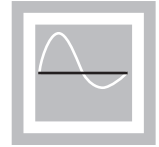
$$\mathbf{x} = \begin{bmatrix} 1 \\ \mathbf{w} \\ b \end{bmatrix} \quad \mathbf{z} = \begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \quad a = \mathbf{w}^T \mathbf{p} + b \quad \Rightarrow \quad a = \mathbf{x}^T \mathbf{z}$$

Mean Square Error:

$$F(\mathbf{x}) = E[e^2] = E[(t - a)^2] = E[(t - \mathbf{x}^T \mathbf{z})^2]$$



Error Analysis



$$F(\mathbf{x}) = E[e^2] = E[(t - a)^2] = E[(t - \mathbf{x}^T \mathbf{z})^2]$$

$$F(\mathbf{x}) = E[t^2 - 2t\mathbf{x}^T \mathbf{z} + \mathbf{x}^T \mathbf{z} \mathbf{z}^T \mathbf{x}]$$

$$F(\mathbf{x}) = E[t^2] - 2\mathbf{x}^T E[t\mathbf{z}] + \mathbf{x}^T E[\mathbf{z} \mathbf{z}^T] \mathbf{x}$$

$$F(\mathbf{x}) = c - 2\mathbf{x}^T \mathbf{h} + \mathbf{x}^T \mathbf{R} \mathbf{x}$$

$$c = E[t^2] \quad \mathbf{h} = E[t\mathbf{z}] \quad \mathbf{R} = E[\mathbf{z} \mathbf{z}^T]$$

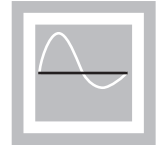
The mean square error for the ADALINE Network is a quadratic function:

$$F(\mathbf{x}) = c + \mathbf{d}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x}$$

$$\mathbf{d} = -2\mathbf{h} \quad \mathbf{A} = 2\mathbf{R}$$



Durağan (Stationary) Point



Hessian Matrix:

$$\mathbf{A} = 2\mathbf{R}$$

The correlation matrix \mathbf{R} must be at least positive semidefinite. If there are any zero eigenvalues, the performance index will either have a weak minimum or else no stationary point, otherwise there will be a unique global minimum \mathbf{x}^* .

$$\nabla F(\mathbf{x}) = \nabla \left(c + \mathbf{d}^T \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{A} \mathbf{x} \right) = \mathbf{d} + \mathbf{A} \mathbf{x} = -2\mathbf{h} + 2\mathbf{R} \mathbf{x}$$

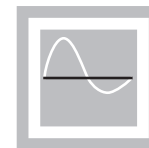
$$-2\mathbf{h} + 2\mathbf{R} \mathbf{x} = \mathbf{0}$$

If \mathbf{R} is positive definite:

$$\mathbf{x}^* = \mathbf{R}^{-1} \mathbf{h}$$



Approximate Steepest Descent



Approximate mean square error (one sample):

$$\hat{F}(\mathbf{x}) = (t(k) - a(k))^2 = e^2(k)$$

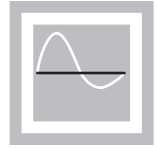
Approximate (stochastic) gradient:

$$\hat{\nabla} F(\mathbf{x}) = \nabla e^2(k)$$

$$[\nabla e^2(k)]_j = \frac{\partial e^2(k)}{\partial w_{1,j}} = 2e(k) \frac{\partial e(k)}{\partial w_{1,j}} \quad j = 1, 2, \dots, R$$

$$[\nabla e^2(k)]_{R+1} = \frac{\partial e^2(k)}{\partial b} = 2e(k) \frac{\partial e(k)}{\partial b}$$

Approximate Gradient Calculation



$$\frac{\partial e(k)}{\partial w_{1,j}} = \frac{\partial [t(k) - a(k)]}{\partial w_{1,j}} = \frac{\partial}{\partial w_{1,j}} [t(k) - (\mathbf{w}_1^T \mathbf{p}(k) + b)]$$

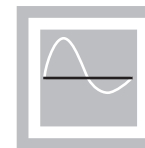
$$\frac{\partial e(k)}{\partial w_{1,j}} = \frac{\partial}{\partial w_{1,j}} \left[t(k) - \left(\sum_{i=1}^R w_{1,i} p_i(k) + b \right) \right]$$

$$\frac{\partial e(k)}{\partial w_{1,j}} = -p_j(k) \qquad \frac{\partial e(k)}{\partial b} = -1$$

$$\hat{\nabla} F(\mathbf{x}) = \nabla e^2(k) = -2e(k)\mathbf{z}(k)$$



LMS Algorithm



$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha \nabla F(\mathbf{x}) \big|_{\mathbf{x} = \mathbf{x}_k}$$

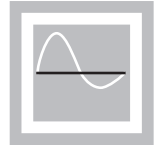
$$\mathbf{x}_{k+1} = \mathbf{x}_k + 2\alpha e(k) \mathbf{z}(k)$$

$${}_1\mathbf{w}(k+1) = {}_1\mathbf{w}(k) + 2\alpha e(k) \mathbf{p}(k)$$

$$b(k+1) = b(k) + 2\alpha e(k)$$



Multiple-Neuron Case



$${}_i\mathbf{w}(k+1) = {}_i\mathbf{w}(k) + 2\alpha e_i(k)\mathbf{p}(k)$$

$$b_i(k+1) = b_i(k) + 2\alpha e_i(k)$$

Matrix Form:

$$\mathbf{W}(k+1) = \mathbf{W}(k) + 2\alpha \mathbf{e}(k)\mathbf{p}^T(k)$$

$$\mathbf{b}(k+1) = \mathbf{b}(k) + 2\alpha \mathbf{e}(k)$$



Yakınsama Analizi

(Analysis of Convergence)



$$\mathbf{x}_{k+1} = \mathbf{x}_k + 2\alpha e(k)\mathbf{z}(k)$$

$$E[\mathbf{x}_{k+1}] = E[\mathbf{x}_k] + 2\alpha E[e(k)\mathbf{z}(k)]$$

$$E[\mathbf{x}_{k+1}] = E[\mathbf{x}_k] + 2\alpha\{E[t(k)\mathbf{z}(k)] - E[(\mathbf{x}_k^T \mathbf{z}(k))\mathbf{z}(k)]\}$$

$$E[\mathbf{x}_{k+1}] = E[\mathbf{x}_k] + 2\alpha\{E[t_k \mathbf{z}(k)] - E[(\mathbf{z}(k)\mathbf{z}^T(k))\mathbf{x}_k]\}$$

$$E[\mathbf{x}_{k+1}] = E[\mathbf{x}_k] + 2\alpha\{\mathbf{h} - \mathbf{R}E[\mathbf{x}_k]\}$$

$$E[\mathbf{x}_{k+1}] = [\mathbf{I} - 2\alpha\mathbf{R}]E[\mathbf{x}_k] + 2\alpha\mathbf{h}$$

For stability, the eigenvalues of this matrix must fall inside the unit circle.

Conditions for Stability



$$|eig([\mathbf{I} - 2\alpha\mathbf{R}])| = |1 - 2\alpha\lambda_i| < 1$$

(where λ_i is an eigenvalue of \mathbf{R})

Since $\lambda_i > 0$, $1 - 2\alpha\lambda_i < 1$.

Therefore the stability condition simplifies to

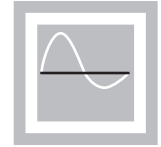
$$1 - 2\alpha\lambda_i > -1$$

$$\alpha < 1/\lambda_i \quad \text{for all } i$$

$$0 < \alpha < 1/\lambda_{max}$$



Steady State Response



$$E[\mathbf{x}_{k+1}] = [\mathbf{I} - 2\alpha\mathbf{R}]E[\mathbf{x}_k] + 2\alpha\mathbf{h}$$

If the system is stable, then a steady state condition will be reached.

$$E[\mathbf{x}_{ss}] = [\mathbf{I} - 2\alpha\mathbf{R}]E[\mathbf{x}_{ss}] + 2\alpha\mathbf{h}$$

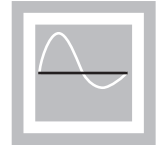
The solution to this equation is

$$E[\mathbf{x}_{ss}] = \mathbf{R}^{-1}\mathbf{h} = \mathbf{x}^*$$

This is also the strong minimum of the performance index.



Example



$$\text{Banana} \quad \left\{ \mathbf{p}_1 = \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}, \mathbf{t}_1 = [-1] \right\} \quad \text{Apple} \quad \left\{ \mathbf{p}_2 = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \mathbf{t}_2 = [1] \right\}$$

$$\mathbf{R} = E[\mathbf{p}\mathbf{p}^T] = \frac{1}{2}\mathbf{p}_1\mathbf{p}_1^T + \frac{1}{2}\mathbf{p}_2\mathbf{p}_2^T$$

$$\mathbf{R} = \frac{1}{2} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} \begin{bmatrix} -1 & 1 & -1 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

$$\lambda_1 = 1.0, \quad \lambda_2 = 0.0, \quad \lambda_3 = 2.0$$

$$\alpha < \frac{1}{\lambda_{\max}} = \frac{1}{2.0} = 0.5$$



Iteration One



Banana $a(0) = \mathbf{W}(0)\mathbf{p}(0) = \mathbf{W}(0)\mathbf{p}_1 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = 0$

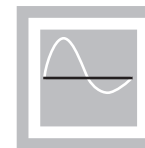
$$e(0) = t(0) - a(0) = t_1 - a(0) = -1 - 0 = -1$$

$$\mathbf{W}(1) = \mathbf{W}(0) + 2\alpha e(0)\mathbf{p}^T(0)$$

$$\mathbf{W}(1) = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} + 2(0.2)(-1) \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix}^T = \begin{bmatrix} 0.4 & -0.4 & 0.4 \end{bmatrix}$$



Iteration Two



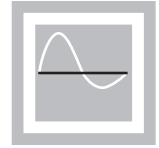
Apple $a(1) = \mathbf{W}(1)\mathbf{p}(1) = \mathbf{W}(1)\mathbf{p}_2 = \begin{bmatrix} 0.4 & -0.4 & 0.4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = -0.4$

$$e(1) = t(1) - a(1) = t_2 - a(1) = 1 - (-0.4) = 1.4$$

$$\mathbf{W}(2) = \begin{bmatrix} 0.4 & -0.4 & 0.4 \end{bmatrix} + 2(0.2)(1.4) \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}^T = \begin{bmatrix} 0.96 & 0.16 & -0.16 \end{bmatrix}$$



Iteration Three



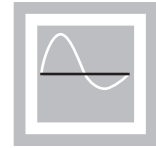
$$a(2) = \mathbf{W}(2)\mathbf{p}(2) = \mathbf{W}(2)\mathbf{p}_1 = \begin{bmatrix} 0.96 & 0.16 & -0.16 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ -1 \end{bmatrix} = -0.64$$

$$e(2) = t(2) - a(2) = t_1 - a(2) = -1 - (-0.64) = -0.36$$

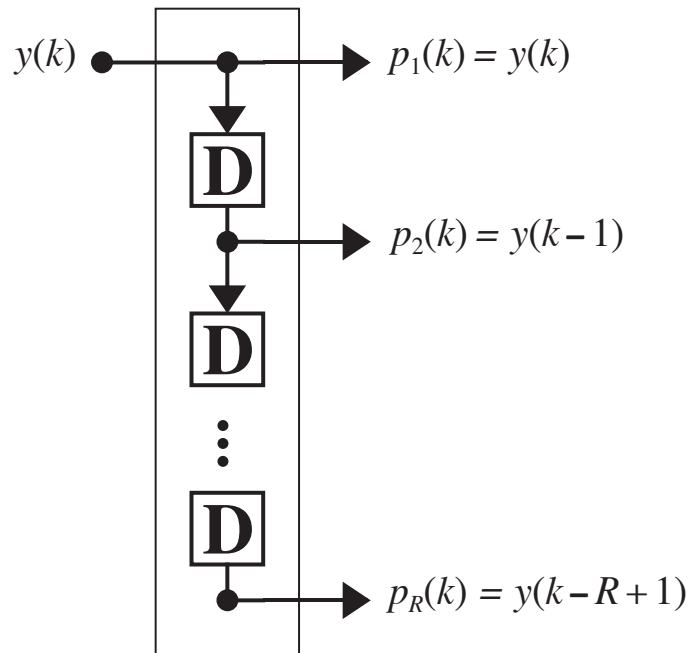
$$\mathbf{W}(3) = \mathbf{W}(2) + 2\alpha e(2)\mathbf{p}^T(2) = \begin{bmatrix} 1.1040 & 0.0160 & -0.0160 \end{bmatrix}$$

$$\mathbf{W}(\infty) = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

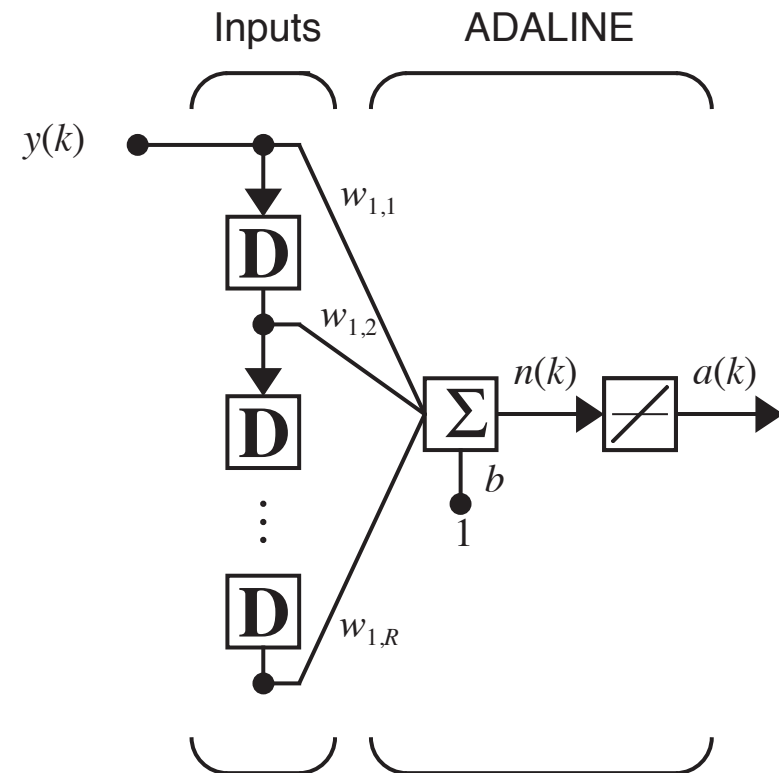
Adaptive Filtering



Tapped Delay Line



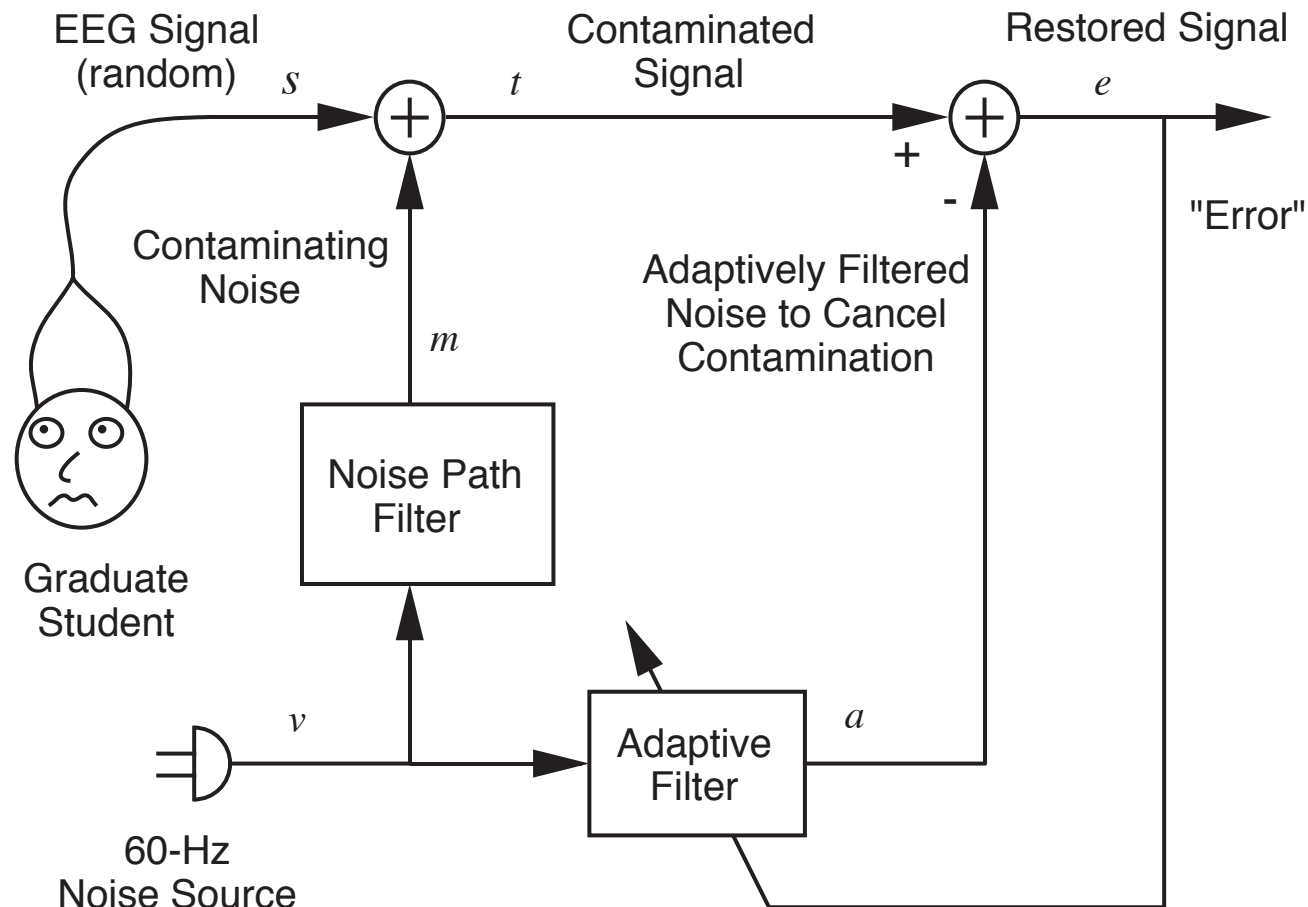
Adaptive Filter



$$a(k) = \text{purelin}(\mathbf{W}\mathbf{p}(k) + b)$$

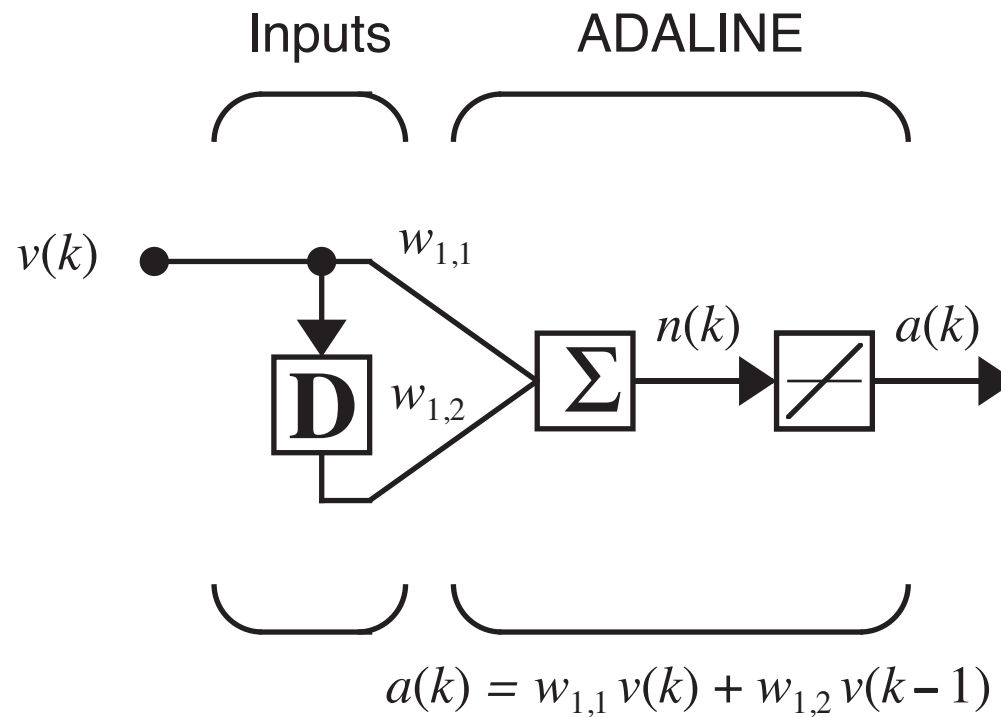
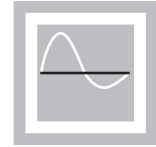
$$a(k) = \text{purelin}(\mathbf{W}\mathbf{p} + b) = \sum_{i=1}^R w_{1,i} y(k-i+1) + b$$

Example: Noise Cancellation

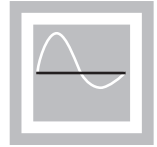


Adaptive Filter Adjusts to Minimize Error (and in doing this removes 60-Hz noise from contaminated signal)

Noise Cancellation Adaptive Filter



Correlation Matrix



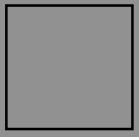
$$\mathbf{R} = [\mathbf{z}\mathbf{z}^T] \quad \mathbf{h} = E[t\mathbf{z}]$$

$$\mathbf{z}(k) = \begin{bmatrix} v(k) \\ v(k-1) \end{bmatrix}$$

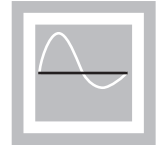
$$t(k) = s(k) + m(k)$$

$$\mathbf{R} = \begin{bmatrix} E[v^2(k)] & E[v(k)v(k-1)] \\ E[v(k-1)v(k)] & E[v^2(k-1)] \end{bmatrix}$$

$$\mathbf{h} = \begin{bmatrix} E[(s(k) + m(k))v(k)] \\ E[(s(k) + m(k))v(k-1)] \end{bmatrix}$$



Signals



$$v(k) = 1.2 \sin\left(\frac{2\pi k}{3}\right) \quad m(k) = 1.2 \sin\left(\frac{2\pi k}{3} - \frac{3\pi}{4}\right)$$

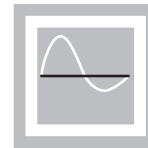
$$E[v^2(k)] = (1.2)^2 \frac{1}{3} \sum_{k=1}^3 \left(\sin\left(\frac{2\pi k}{3}\right) \right)^2 = (1.2)^2 0.5 = 0.72$$

$$E[v^2(k-1)] = E[v^2(k)] = 0.72$$

$$\begin{aligned} E[v(k)v(k-1)] &= \frac{1}{3} \sum_{k=1}^3 \left(1.2 \sin\frac{2\pi k}{3} \right) \left(1.2 \sin\frac{2\pi(k-1)}{3} \right) \\ &= (1.2)^2 0.5 \cos\left(\frac{2\pi}{3}\right) = -0.36 \end{aligned}$$

$$\mathbf{R} = \begin{bmatrix} 0.72 & -0.36 \\ -0.36 & 0.72 \end{bmatrix}$$

Stationary Point



$$E[(s(k) + m(k))v(k)] = E[s(k)v(k)] + E[m(k)v(k)]$$



$$0$$

$$E[m(k)v(k)] = \frac{1}{3} \sum_{k=1}^3 \left(1.2 \sin\left(\frac{2\pi k}{3} - \frac{3\pi}{4}\right) \right) \left(1.2 \sin\frac{2\pi k}{3} \right) = -0.51$$

$$E[(s(k) + m(k))v(k-1)] = E[s(k)v(k-1)] + E[m(k)v(k-1)]$$



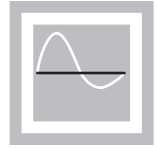
$$0$$

$$E[m(k)v(k-1)] = \frac{1}{3} \sum_{k=1}^3 \left(1.2 \sin\left(\frac{2\pi k}{3} - \frac{3\pi}{4}\right) \right) \left(1.2 \sin\frac{2\pi(k-1)}{3} \right) = 0.70$$

$$\mathbf{h} = \begin{bmatrix} E[(s(k) + m(k))v(k)] \\ E[(s(k) + m(k))v(k-1)] \end{bmatrix} \Rightarrow \mathbf{h} = \begin{bmatrix} -0.51 \\ 0.70 \end{bmatrix}$$

$$\mathbf{x}^* = \mathbf{R}^{-1} \mathbf{h} = \begin{bmatrix} 0.72 & -0.36 \\ -0.36 & 0.72 \end{bmatrix}^{-1} \begin{bmatrix} -0.51 \\ 0.70 \end{bmatrix} = \begin{bmatrix} -0.30 \\ 0.82 \end{bmatrix}$$

Performance Index



$$F(\mathbf{x}) = c - 2\mathbf{x}^T \mathbf{h} + \mathbf{x}^T \mathbf{R} \mathbf{x}$$

$$c = E[t^2(k)] = E[(s(k) + m(k))^2]$$

$$c = E[s^2(k)] + 2E[s(k)m(k)] + E[m^2(k)]$$

$$E[s^2(k)] = \frac{1}{0.4} \int_{-0.2}^{0.2} s^2 ds = \frac{1}{3(0.4)} s^3 \Big|_{-0.2}^{0.2} = 0.0133$$

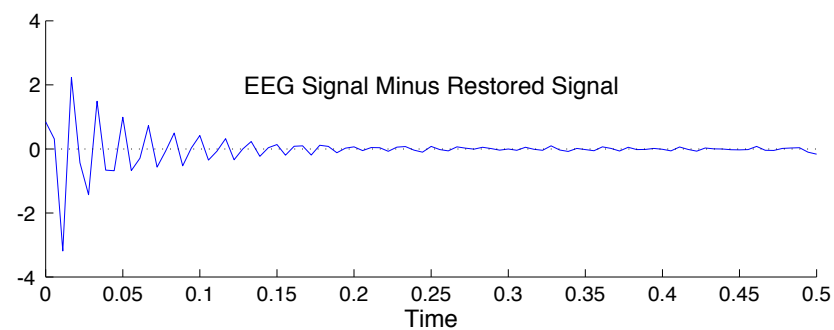
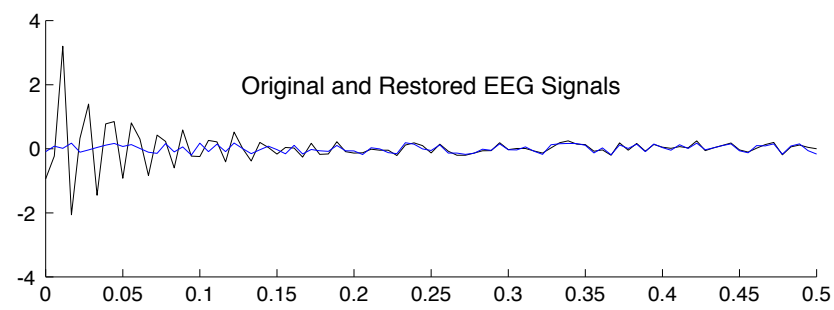
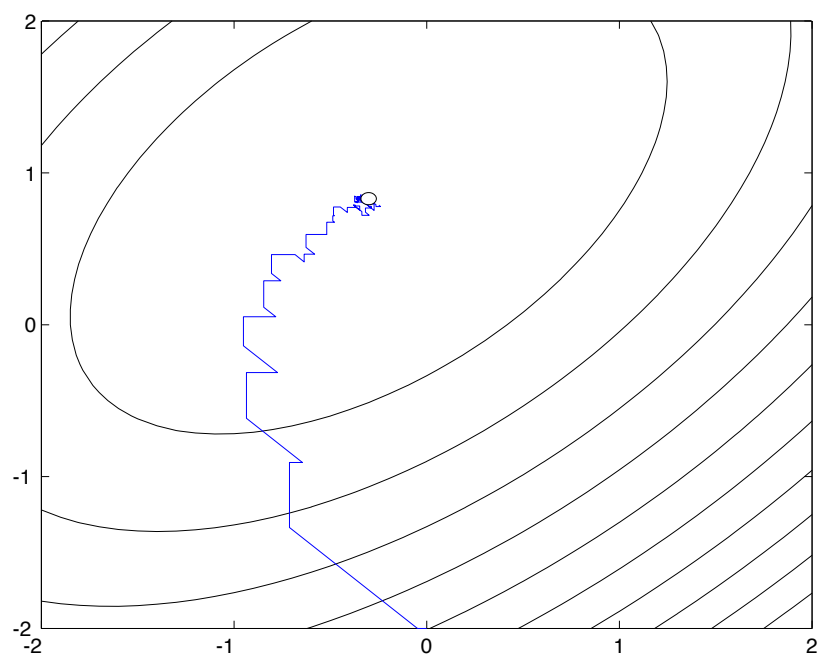
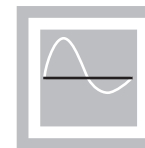
$$E[m^2(k)] = \frac{1}{3} \sum_{k=1}^3 \left\{ 1.2 \sin\left(\frac{2\pi}{3} - \frac{3\pi}{4}\right) \right\}^2 = 0.72$$

$$c = 0.0133 + 0.72 = 0.7333$$

$$F(\mathbf{x}^*) = 0.7333 - 2(0.72) + 0.72 = 0.0133$$



LMS Response



Echo Cancellation

