LECTURES ON

NEURAL NETWORKS

BY

PROF. BERNARD WIDROW

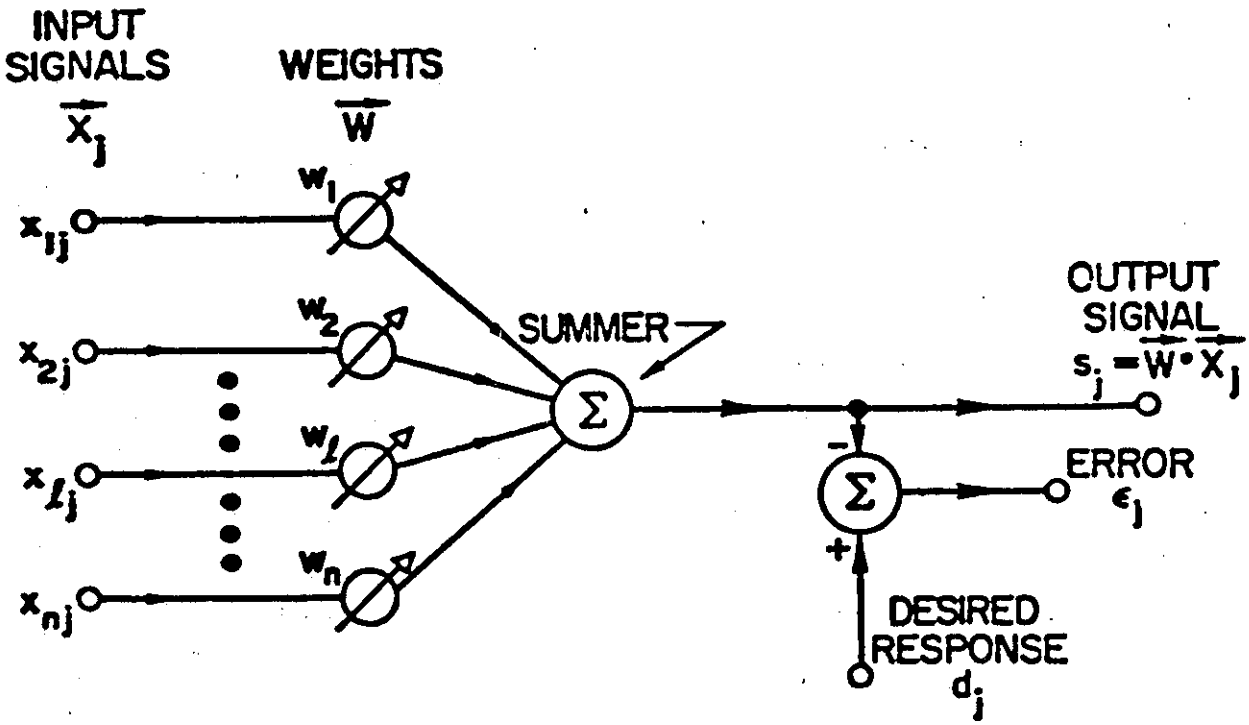
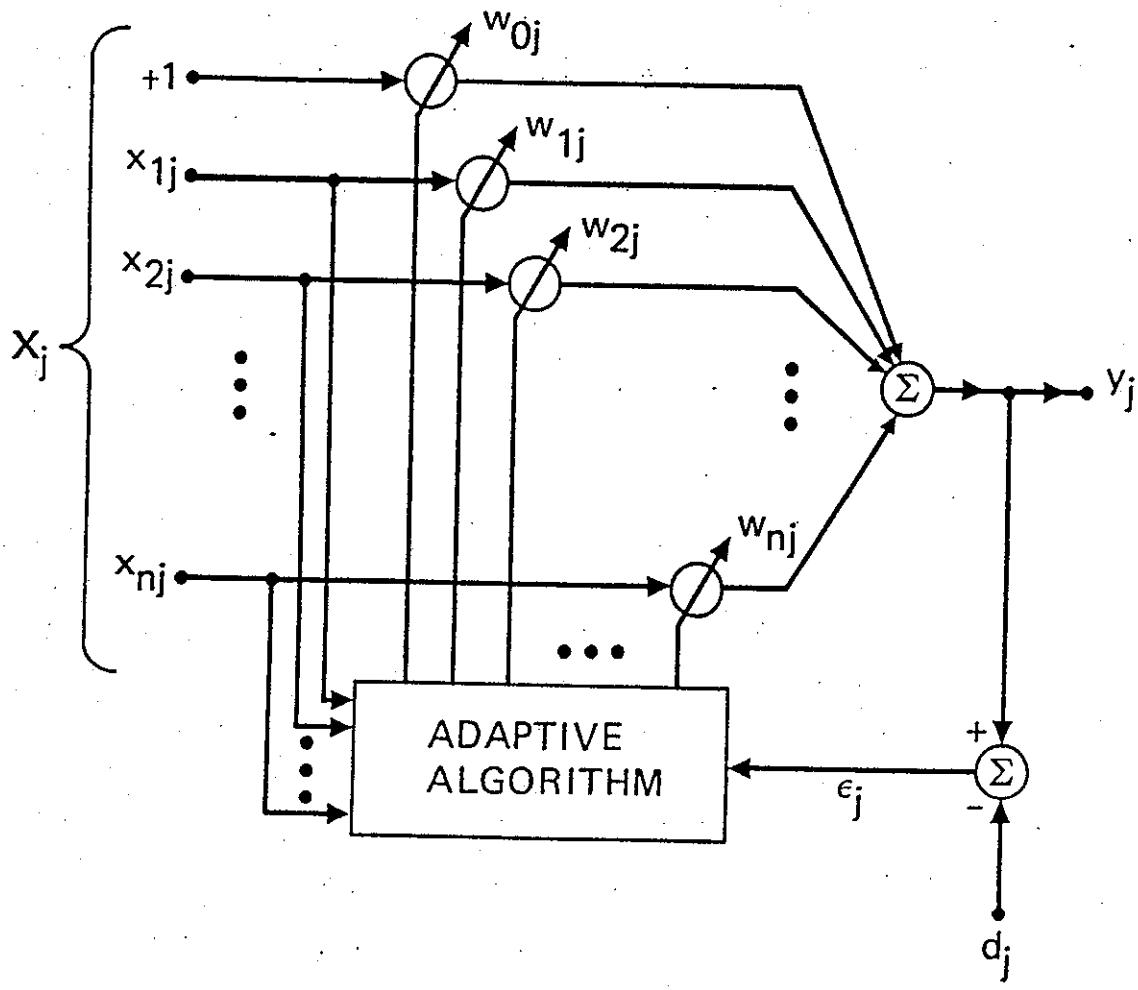
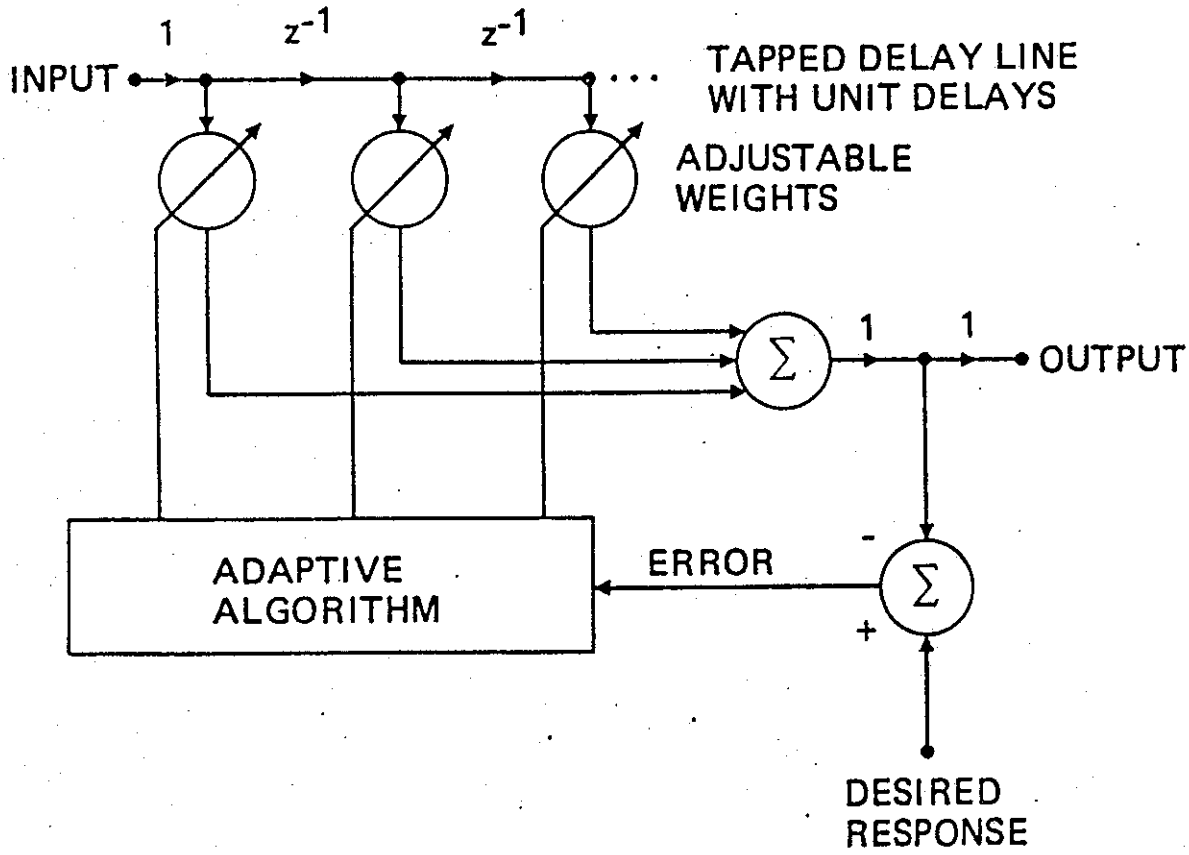


Fig. 2. Adaptive linear combinatorial system.


w_{0j}
+1
w_{1j}
x_{1j}
w_{2j}
x_{2j}
x_j
y_j
w_{nj}
x_{nj}
ADAPTIVE
ALGORITHM
+
ϵ_j
−
d_j


1
z^{-1}
z^{-1}
INPUT
TAPPED DELAY LINE
WITH UNIT DELAYS
ADJUSTABLE
WEIGHTS
1
1
OUTPUT
ADAPTIVE
ALGORITHM
ERROR
-
+
DESIRED
RESPONSE

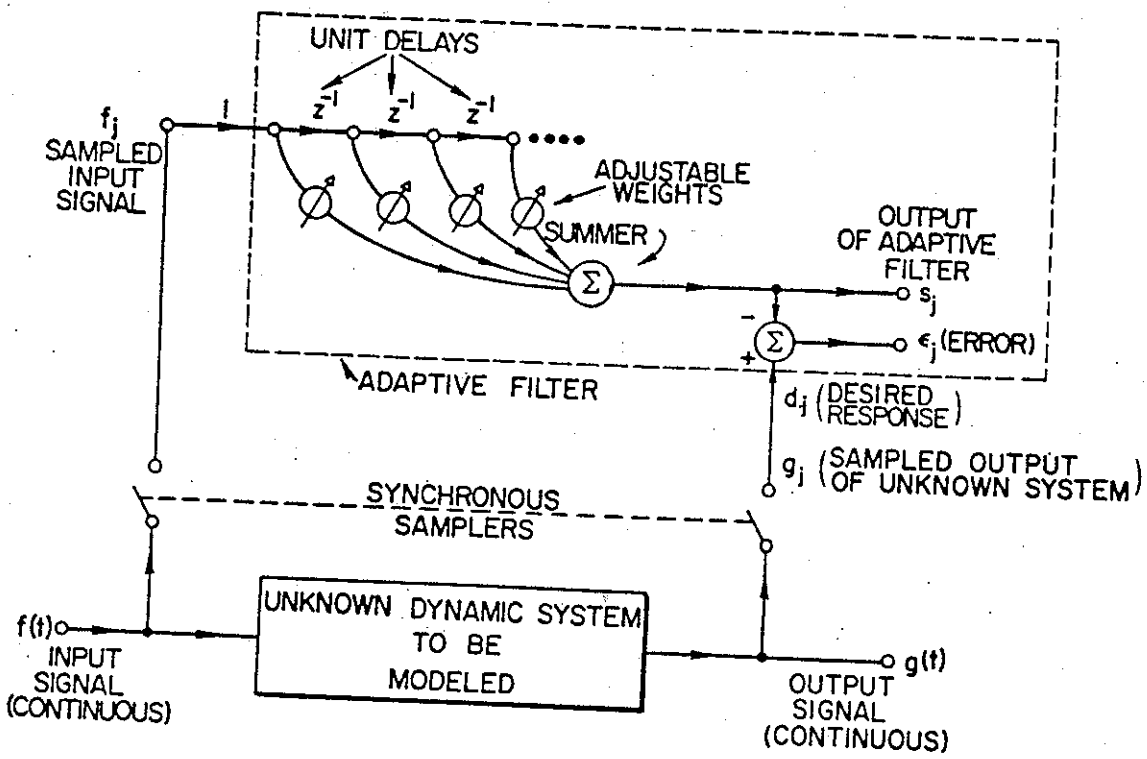
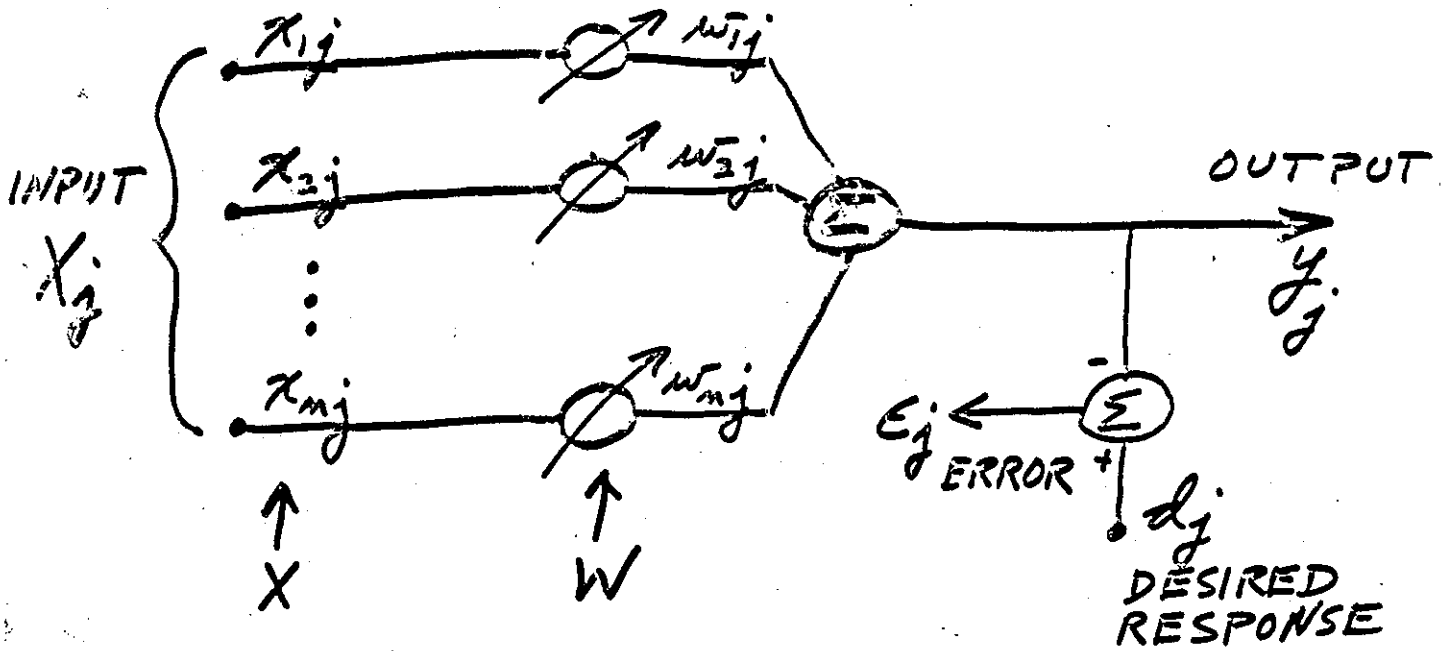


Fig. 1. Modeling an unknown system by a discrete adaptive filter.



$$y_j = X_j^T W = W^T X_j$$

$$\epsilon_j = d_j - X_j^T W$$

$$\epsilon_j^2 = d_j^2 - 2 d_j X_j^T W + W^T X_j X_j^T W$$

$$MSE = \xi \triangleq E[\epsilon_j^2] = E[d_j^2] - 2P^T W + W^T R W$$

$$\nabla = \frac{\partial \xi}{\partial W} = -2P + 2RW$$

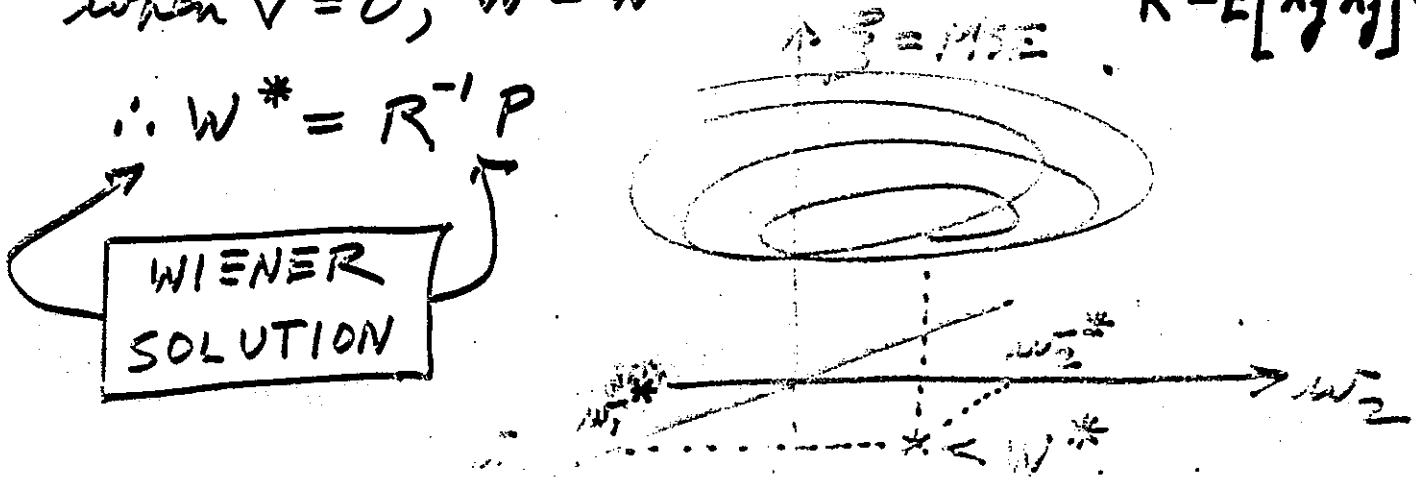
when $\nabla = 0$, $W = W^*$

$$\therefore W^* = R^{-1} P$$

WIENER SOLUTION

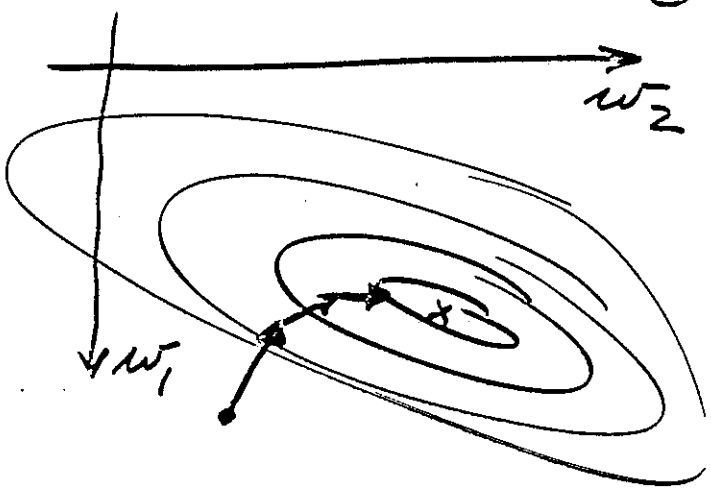
$$P \triangleq E\{d_j X_j\} \quad (n \times 1)$$

$$R \triangleq E[X_j X_j^T] \quad (n \times n)$$



STEEPEST DESCENT:

$$W_{j+1} = W_j - \mu \nabla_j$$



TRUE GRADIENT:

$$\nabla = \frac{\partial \xi}{\partial W} = -2P + 2RW$$

A GRADIENT ESTIMATE:

$$\hat{\nabla} = \frac{\partial \epsilon_j^2}{\partial W} \neq \frac{\partial E[\epsilon_j^2]}{\partial W} \triangleq \nabla$$

now, $\epsilon_j = d_j - X_j^T W$

$$\hat{\nabla} = 2\epsilon_j \frac{\partial \epsilon_j}{\partial W} = 2\epsilon_j(-X_j) = -2\epsilon_j X_j$$

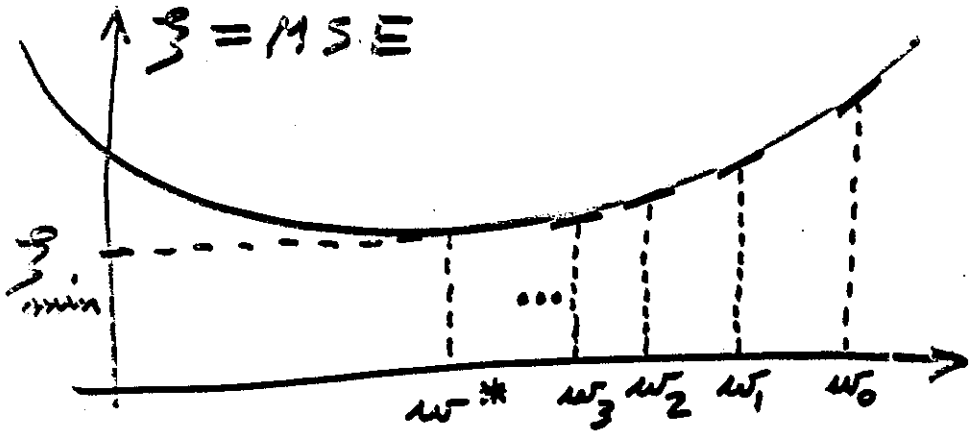
$$E[\hat{\nabla}] = -2E[\epsilon_j X_j] = -2E[d_j X_j - X_j X_j^T W] = -2P + 2RW = \nabla$$

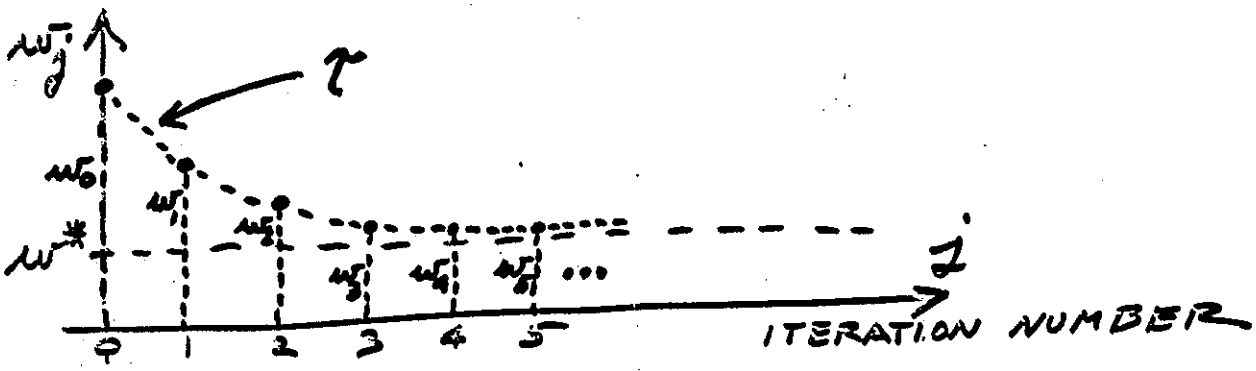
$\therefore$ grad. estimate is unbiased!

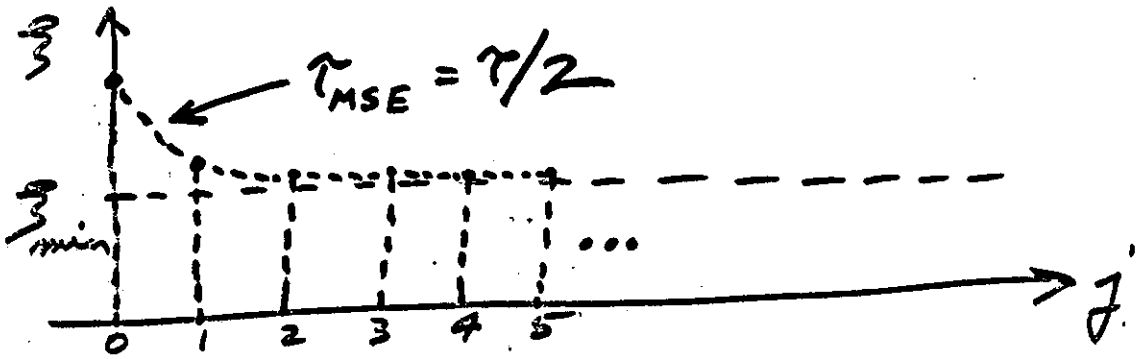
LMS ALGORITHM (Least mean square error) OF WIDROW and HOFF (1959):

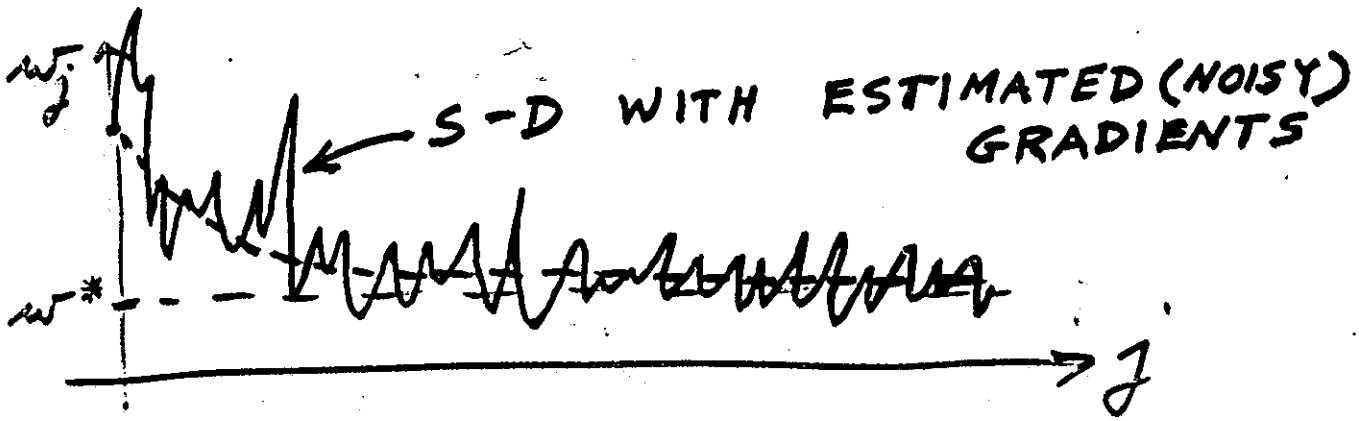
$$W_{j+1} = W_j - \mu \hat{\nabla}_j \leftarrow \text{S-D using estimated gradient}$$

$$\boxed{W_{j+1} = W_j + 2\mu \epsilon_j X_j \quad \text{and} \quad \epsilon_j = d_j - X_j^T W_j} \leftarrow \text{LMS}$$


$\mathcal{E} = MSE$
$\mathcal{E}_{min}$
w^*
w_3
w_2
w_1
w_0


w_j
τ
w_0
w^*
j
0
1
2
3
4
5
ITERATION NUMBER


$\mathcal{E}$
$\tau_{MSE} = \tau/2$
$\mathcal{E}_{min}$
0
1
2
3
4
5
j


w_j
S-D WITH ESTIMATED (NOISY) GRADIENTS
w^*
j

CONDITION FOR CONVERGENCE (STABILITY) OF THE MEAN OF THE WEIGHT VECTOR:

$$\lim_{j\to\infty}\left[I - 2\mu\Lambda\right]^j = 0, \text{ ie}$$

$$\frac{1}{\lambda_{max}} > \mu > 0.$$

SUFFICIENT (BUT NOT NECESSARY), EASY TO APPLY, CONVERGENCE OF THE MEAN:

$$\frac{1}{\text{TRACE } R} > \mu > 0$$

NOTE: CONVERGENCE OF THE MEAN OF THE WEIGHT VECTOR DOES NOT GUARANTEE CONVERGENCE OF ITS VARIANCE OF OTHER MOMENT.

TIME CONSTANTS OF LMS ALGORITHM

$$\tau_p = \frac{1}{2\mu\lambda_p}$$

$$\tau_{p_{MSE}} = \frac{1}{4\mu\lambda_p}$$

MISADJUSTMENT:

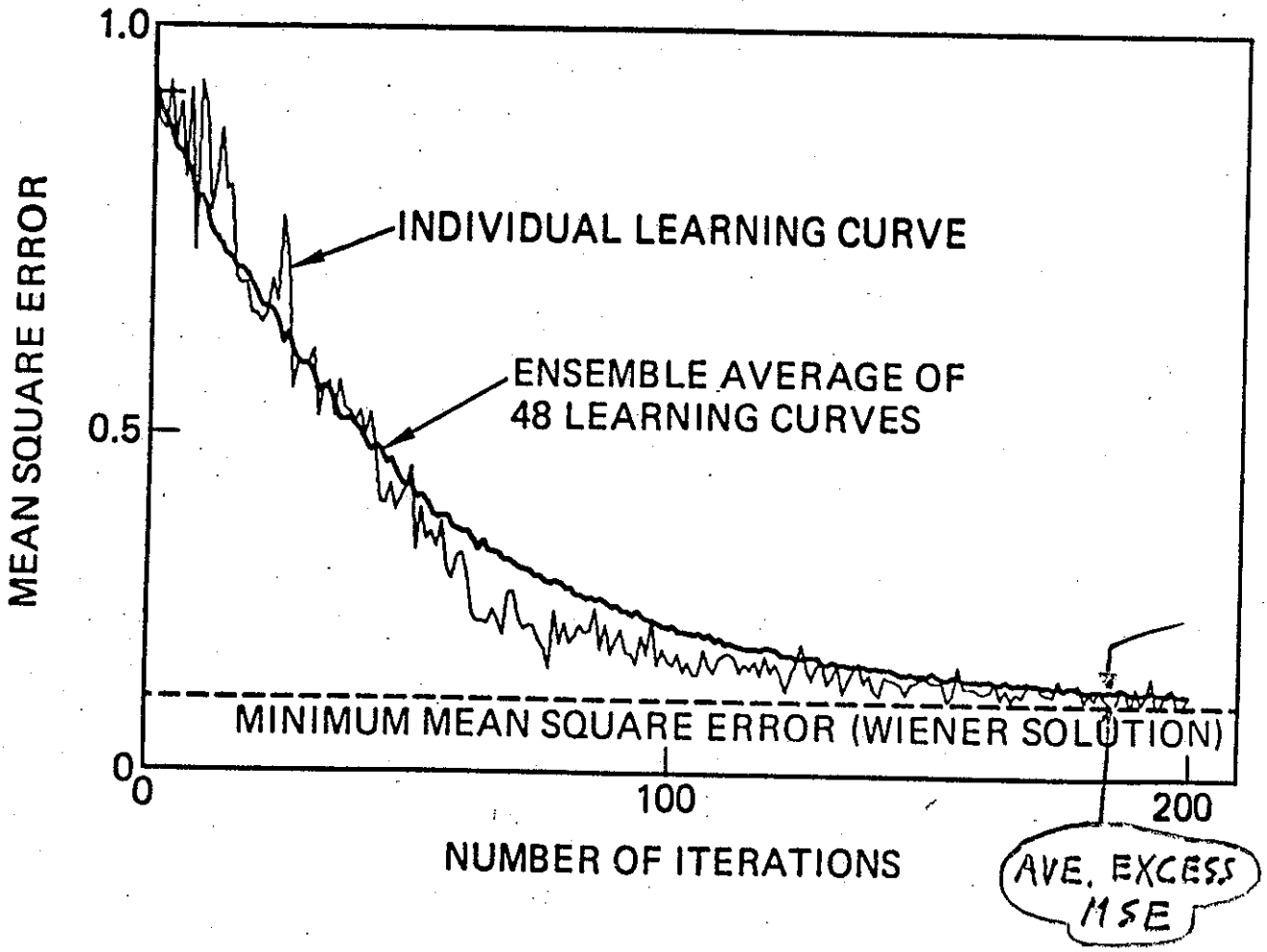
$$M \triangleq \frac{\text{AVE EXCESS MSE}}{\text{MINIMUM MSE}} = \mu\,\text{TRACE}R = \frac{1}{2}\sum_{p=1}^{n}\frac{1}{\tau_p} = \frac{n}{2}\left(\frac{1}{\tau_p}\right)_{AVE}$$

$$M = \frac{n}{4}\left(\frac{1}{\tau_{p_{MSE}}}\right)_{AVE} = \frac{n}{4}\frac{1}{\tau_{MSE}}$$ ← WHEN ALL λ'S ARE EQUAL

EXAMPLE: Let $M = 10\%$ be an "OK" misadj; and let $n = 10$ weights. The question is, how big does τ_{MSE} need to be, and what is the settling time of the adaptive process?

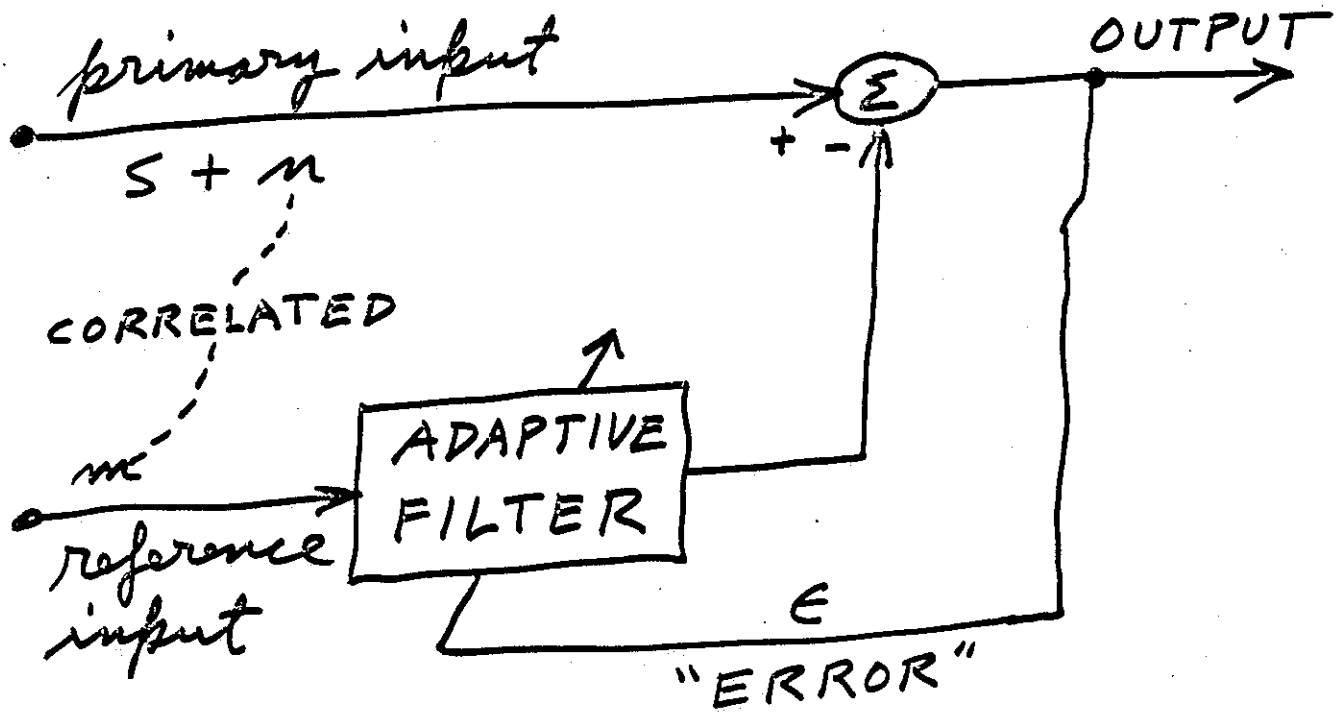
$$M = 0.1 = \frac{10}{4}\frac{1}{\tau_{MSE}} \quad \therefore \tau_{MSE} = 25 \text{ iterations}$$

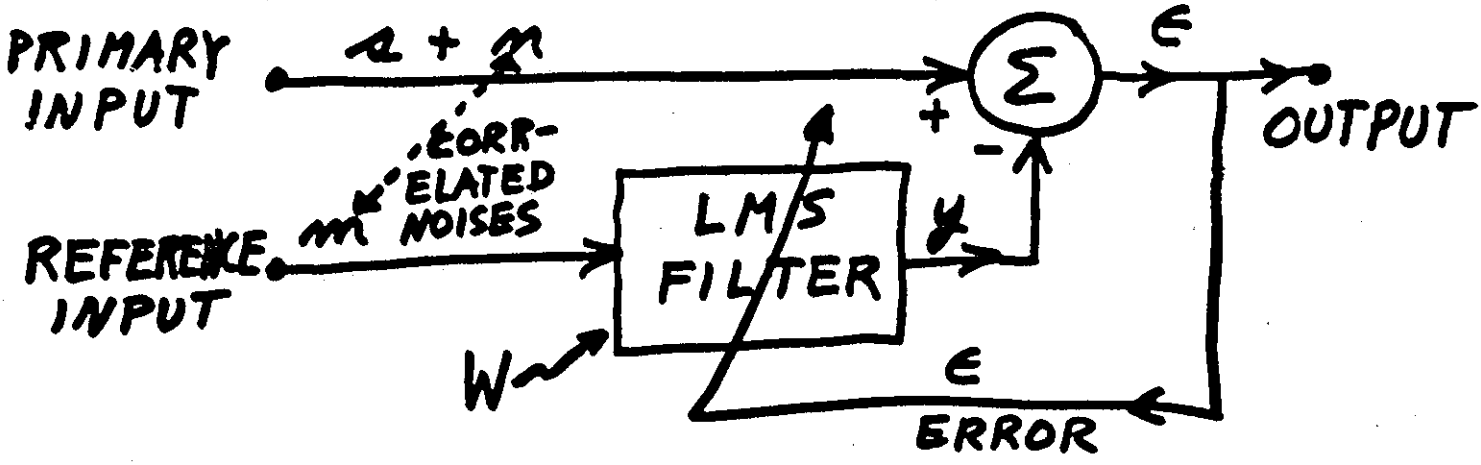
Settling time $\simeq$ 100 iterations = 10X filter length.


1.0
0.5
0
0
100
200
MEAN SQUARE ERROR
INDIVIDUAL LEARNING CURVE
ENSEMBLE AVERAGE OF
48 LEARNING CURVES
MINIMUM MEAN SQUARE ERROR (WIENER SOLUTION)
NUMBER OF ITERATIONS
AVE. EXCESS
MSE

CONVENTIONAL FILTERING

ADAPTIVE NOISE CANCELLING





$$\epsilon = s + n - y$$

$$\epsilon^2 = [s + (n-y)]^2 = s^2 + (n-y)^2 + 2s(n-y)$$

$$E[\epsilon^2] = E[s^2] + E[(n-y)^2] + 2E[s(n-y)]$$

$$E[\epsilon^2] = E[s^2] + E[(n-y)^2].$$

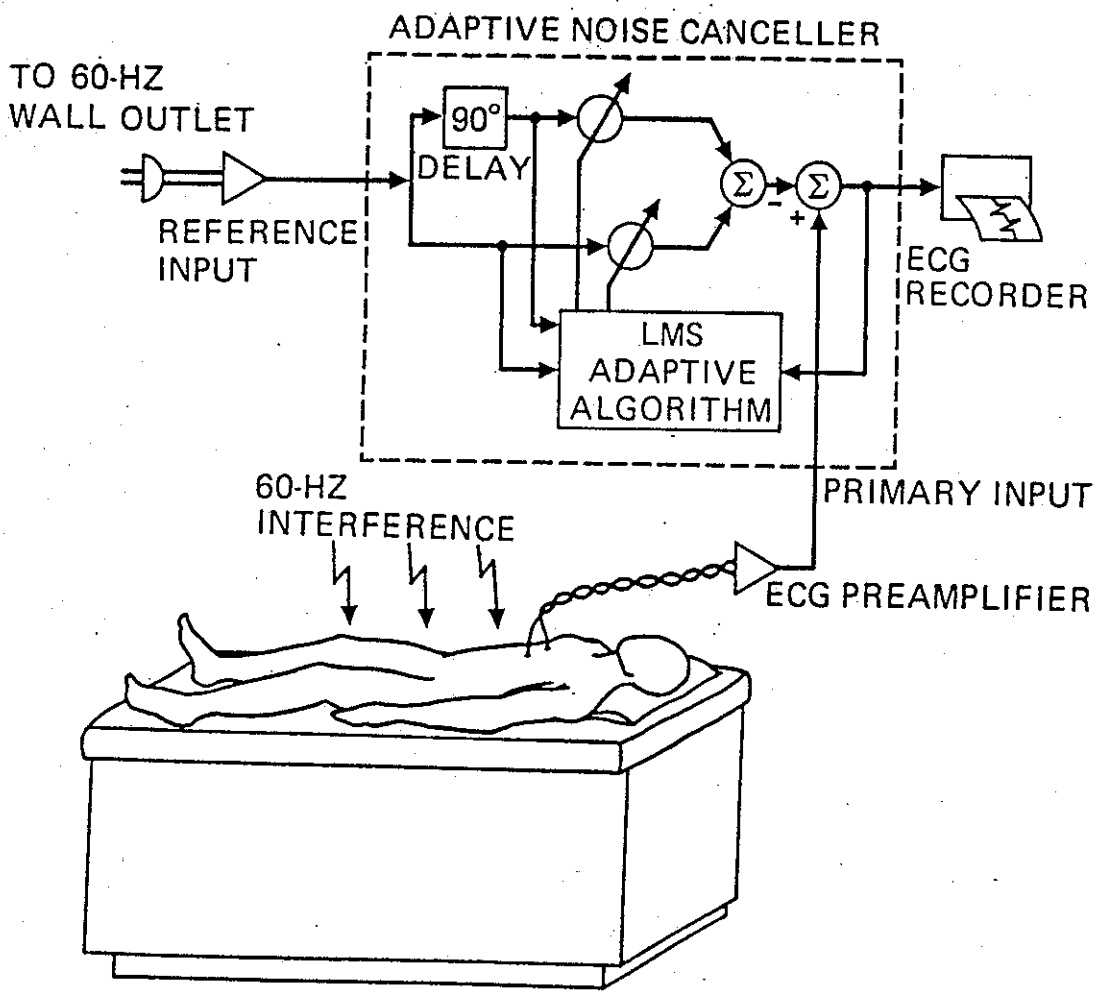
$$\min_W E[\epsilon^2] = E[s^2] + \min_W [E(n-y)^2], \text{ THUS}$$

(1) MINIMIZING $E[\epsilon^2]$ MINIMIZES $E[(n-y)^2]$,

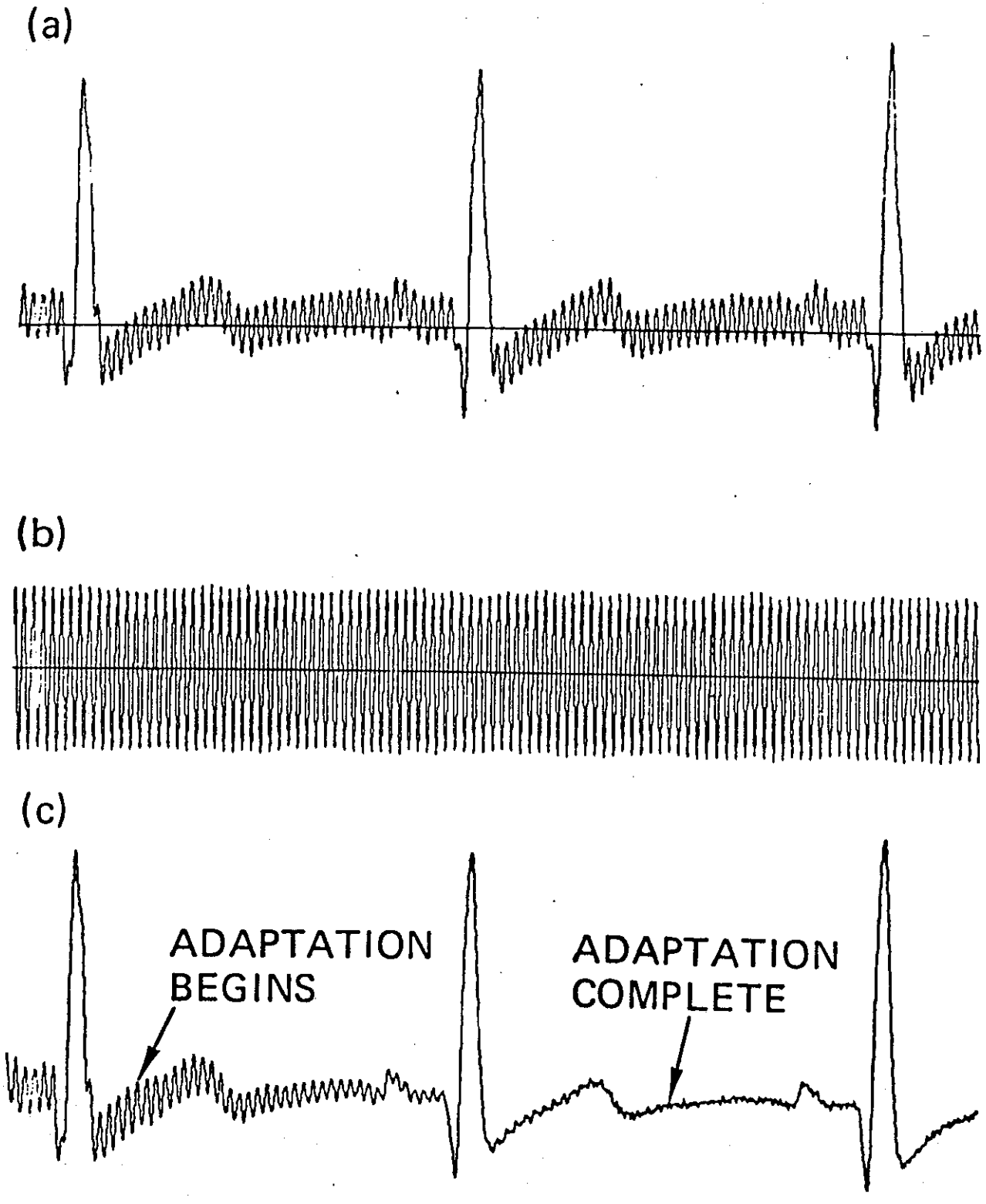
(2) CAUSES y TO BE BEST LEAST SQUARES ESTIMATE OF n,

(3) " ϵ " " " " " " OF s,

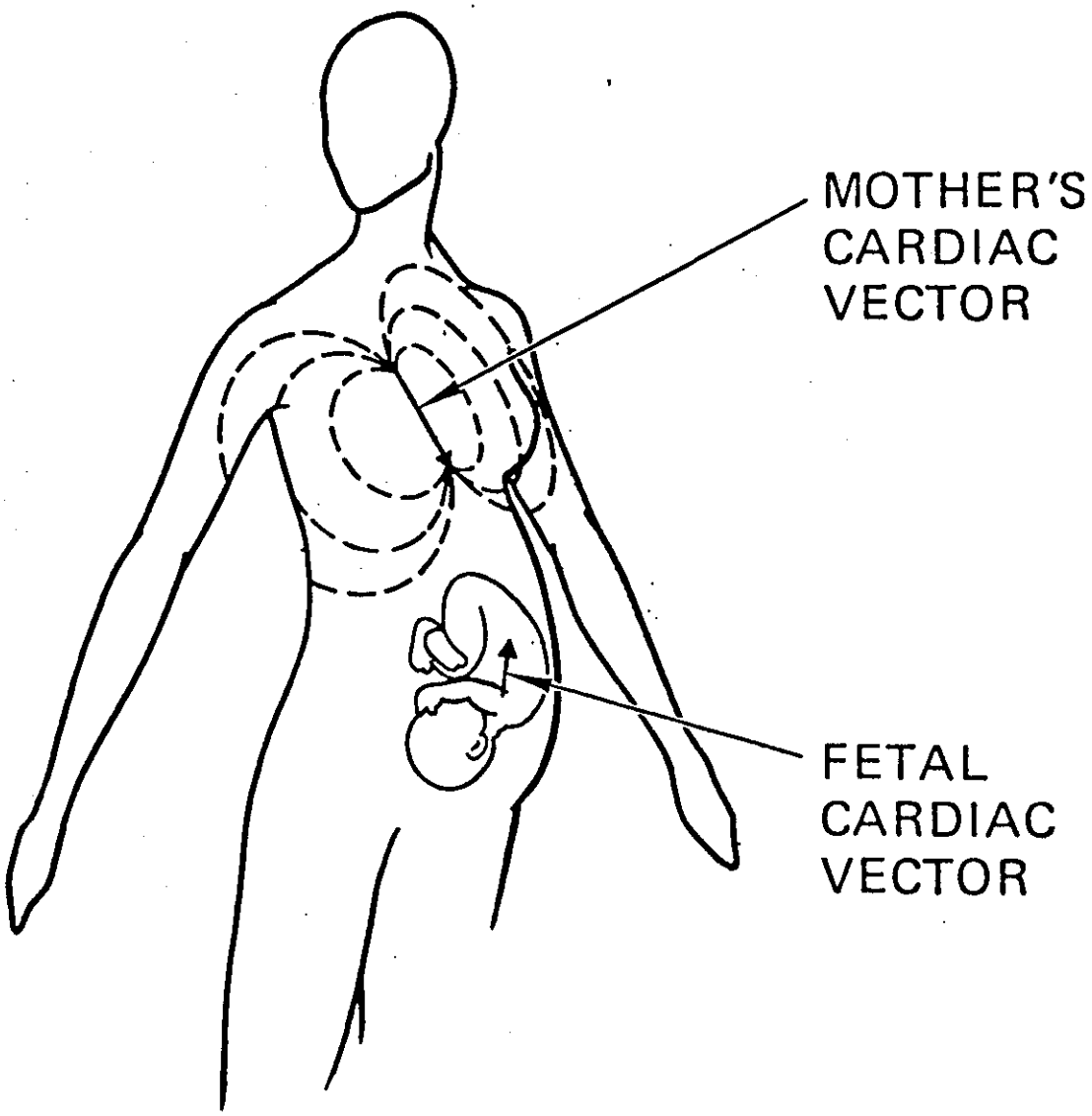
(4) MAXIMIZES OUTPUT SNR WITHOUT SIGNAL DISTORTION



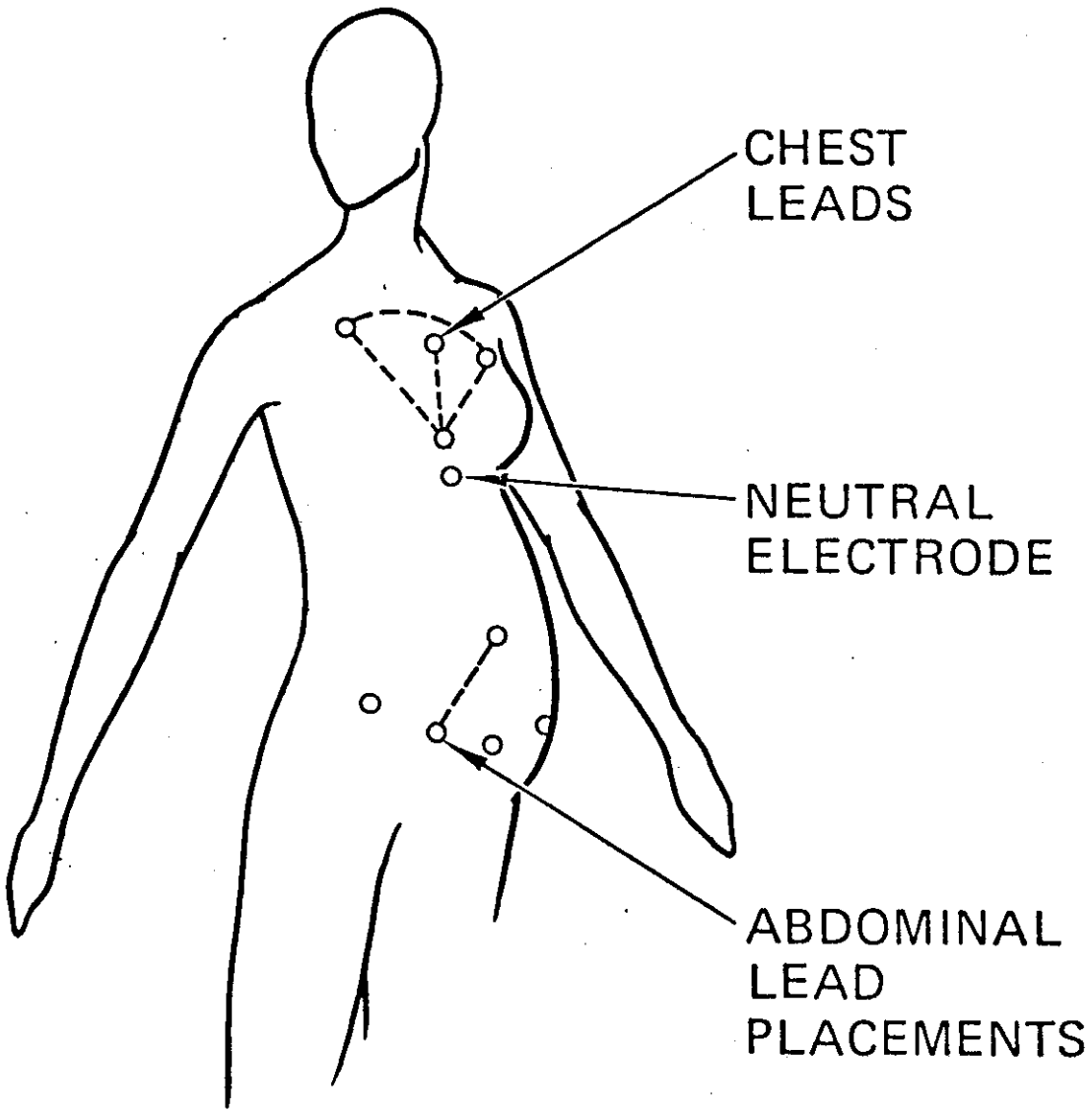
CANCELLING 60Hz INTERFERENCE IN THE EKG

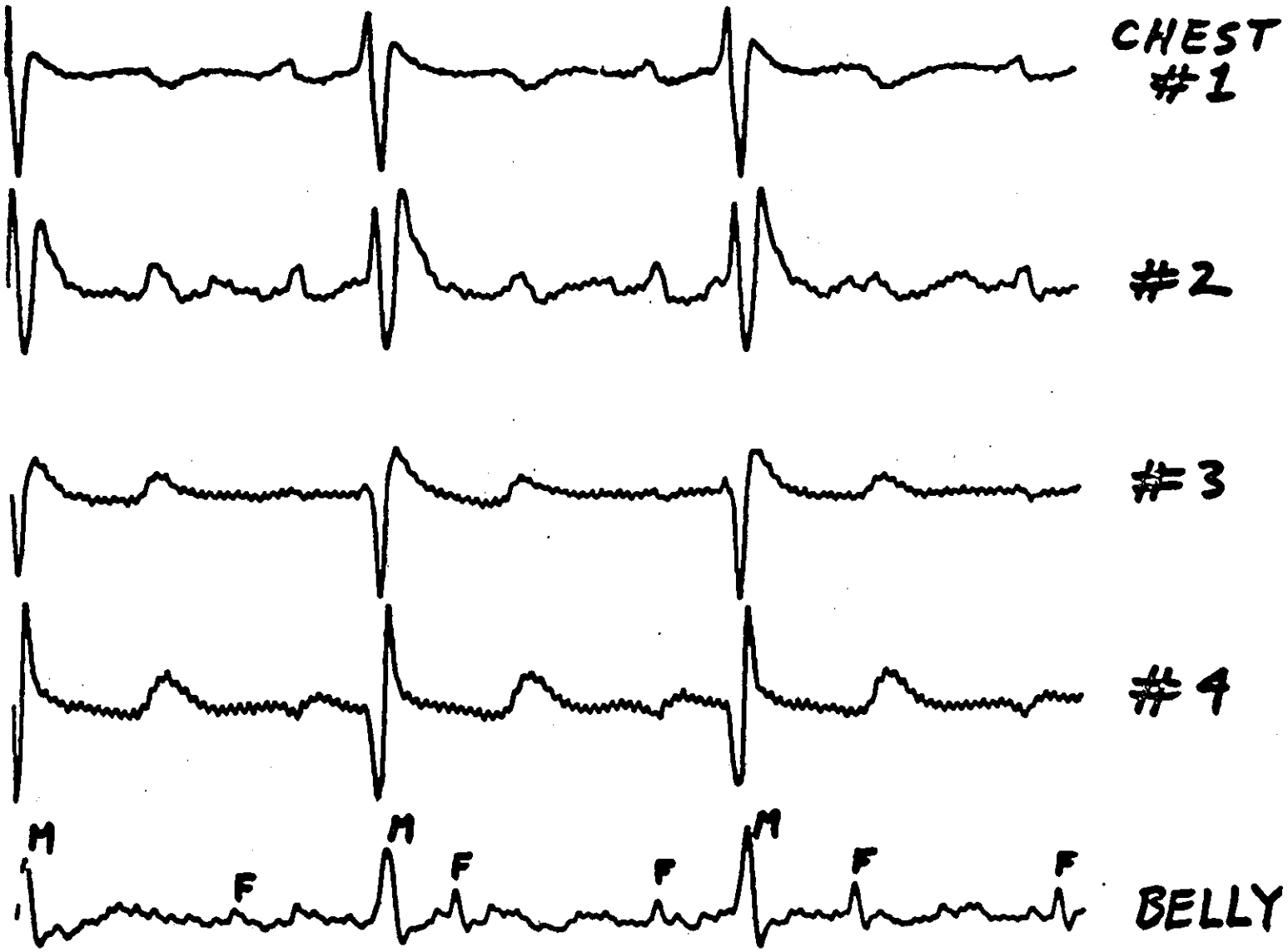

(a)
(b)
(c)
ADAPTATION
BEGINS
ADAPTATION
COMPLETE

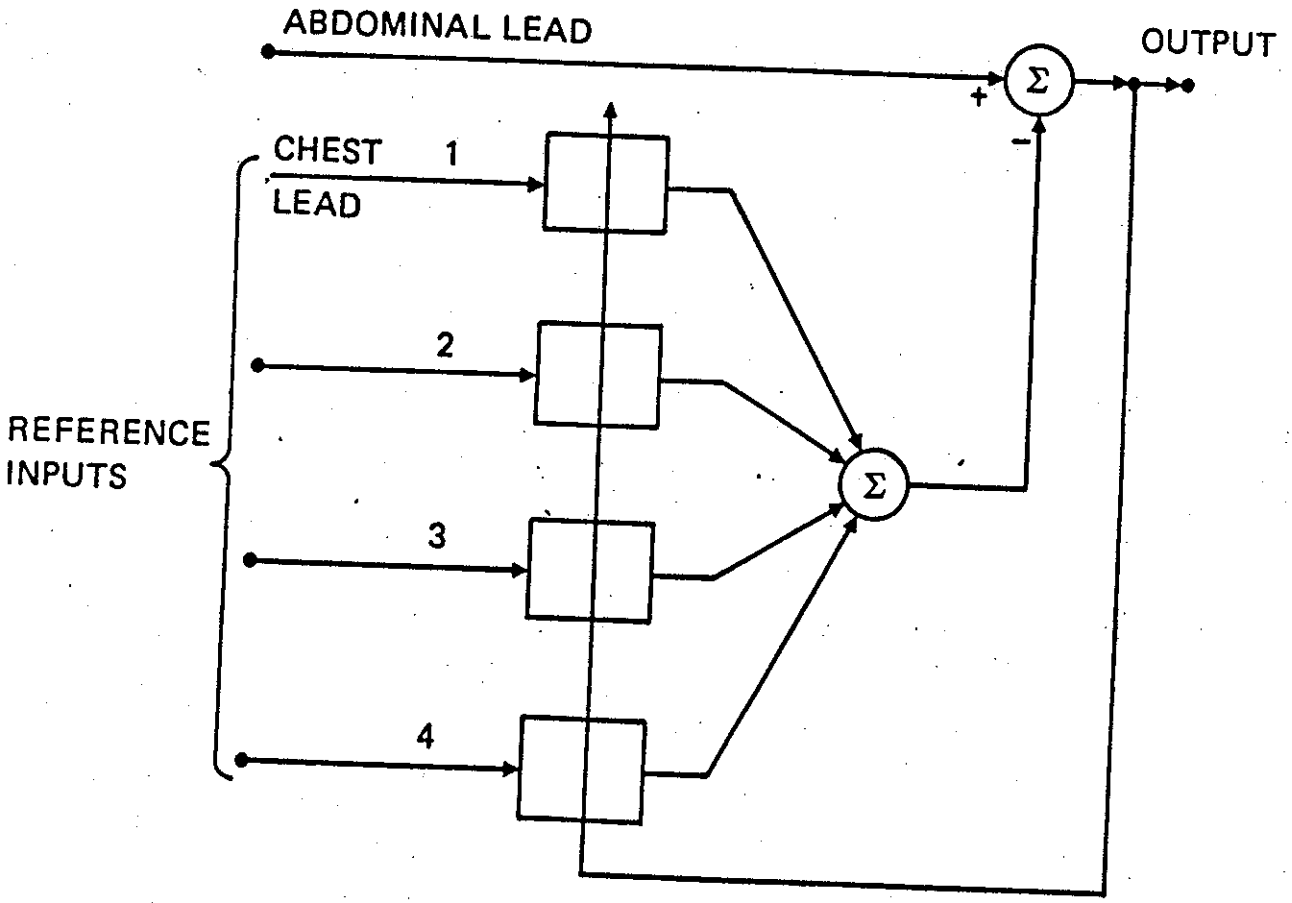
(a)

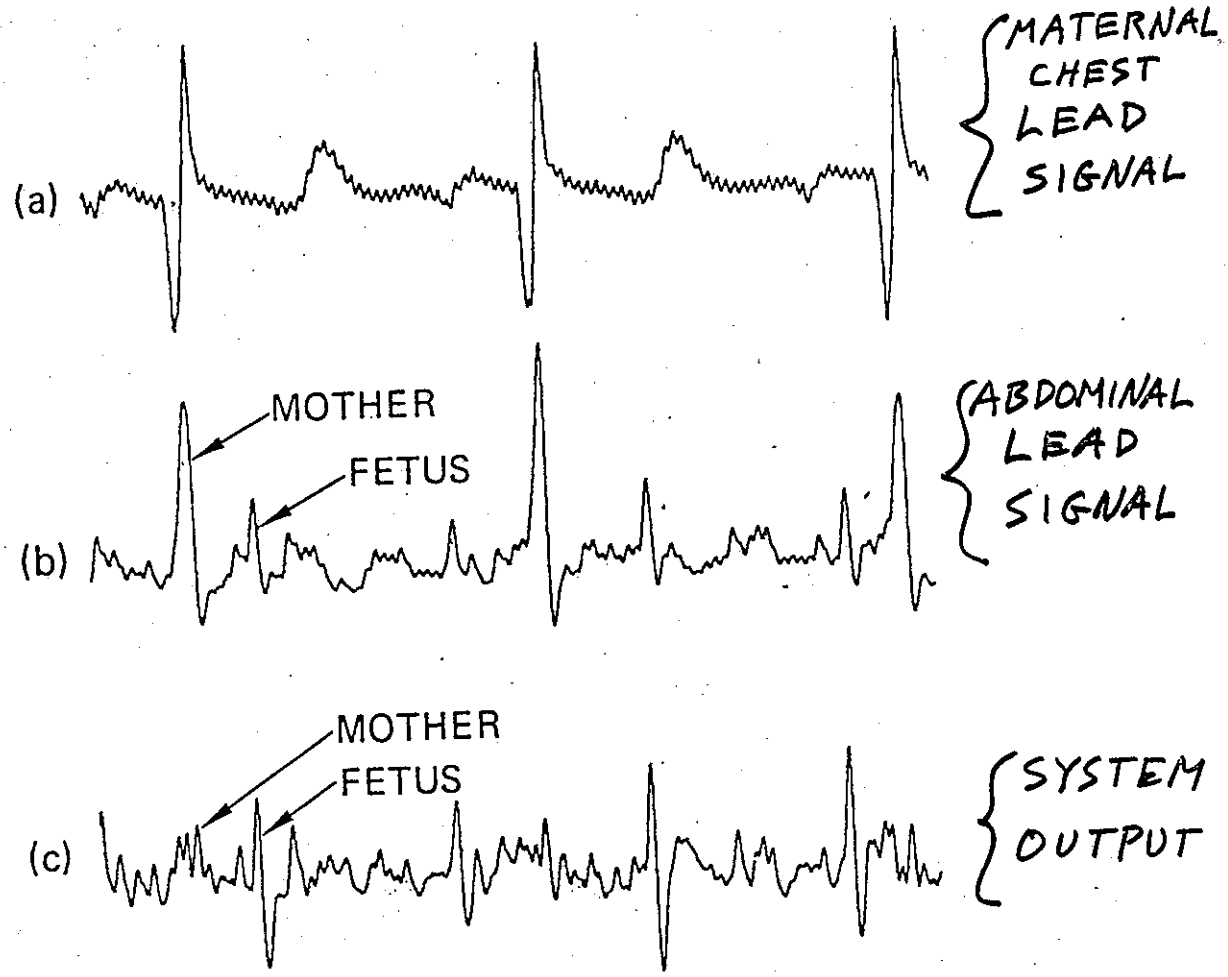


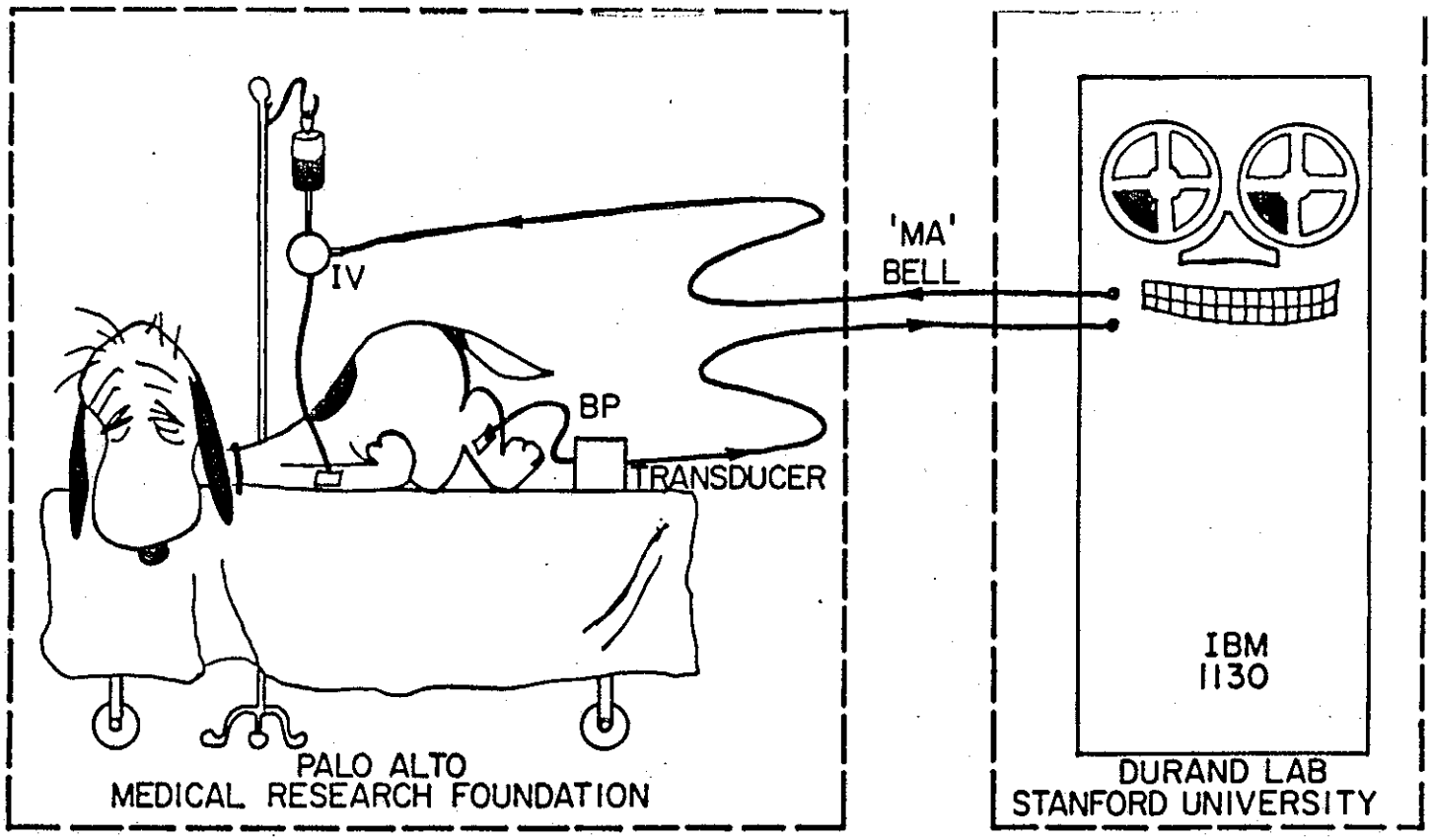
(b)




CHEST #1
#2
#3
#4
M
F
M
F
F
M
F
F
BELLY


ABDOMINAL LEAD
OUTPUT
Σ
+
−
CHEST
LEAD
1
2
3
4
REFERENCE
INPUTS
Σ


(a)
MATERNAL CHEST LEAD SIGNAL
MOTHER
FETUS
(b)
ABDOMINAL LEAD SIGNAL
MOTHER
FETUS
(c)
SYSTEM OUTPUT


IV
'MA'
BELL
BP
TRANSDUCER
IBM
1130
PALO ALTO
MEDICAL RESEARCH FOUNDATION
DURAND LAB
STANFORD UNIVERSITY

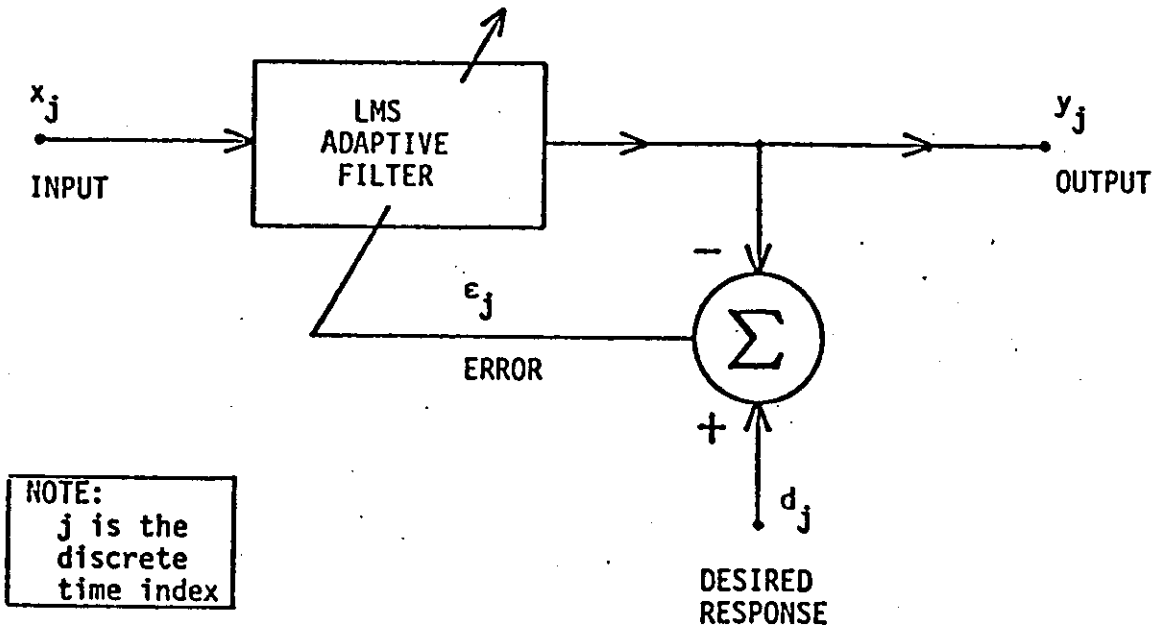


Fig. 1. Symbolic Representation of an Adaptive Transversal Filter Adapted by the LMS Algorithm.

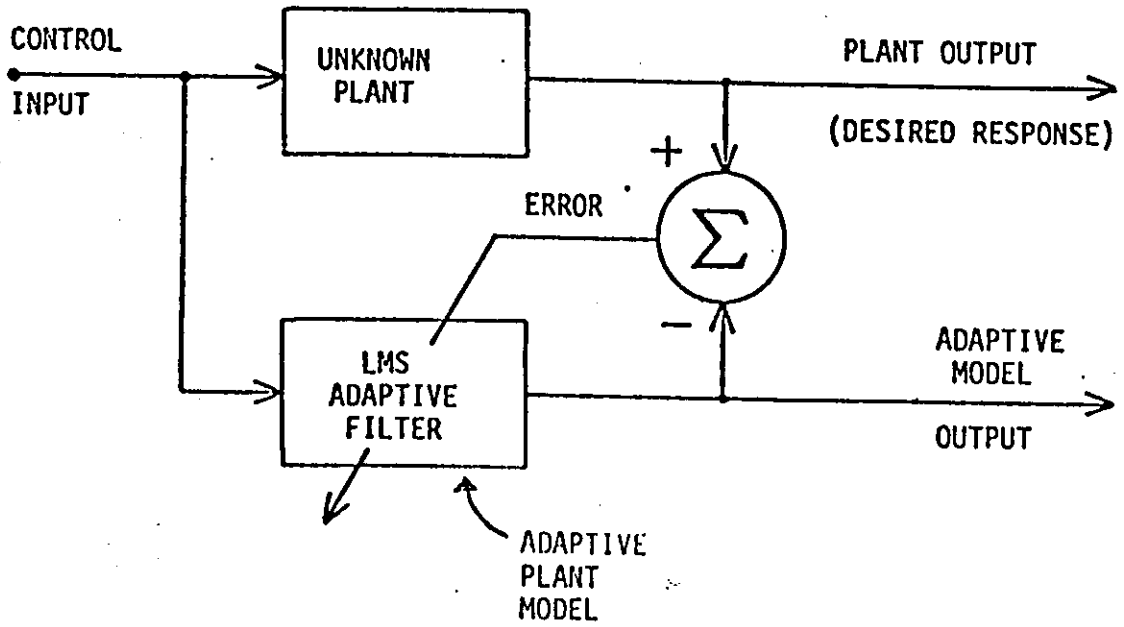


Fig. 2. Modeling an Unknown Plant by Means of an Adaptive Transversal Filter.

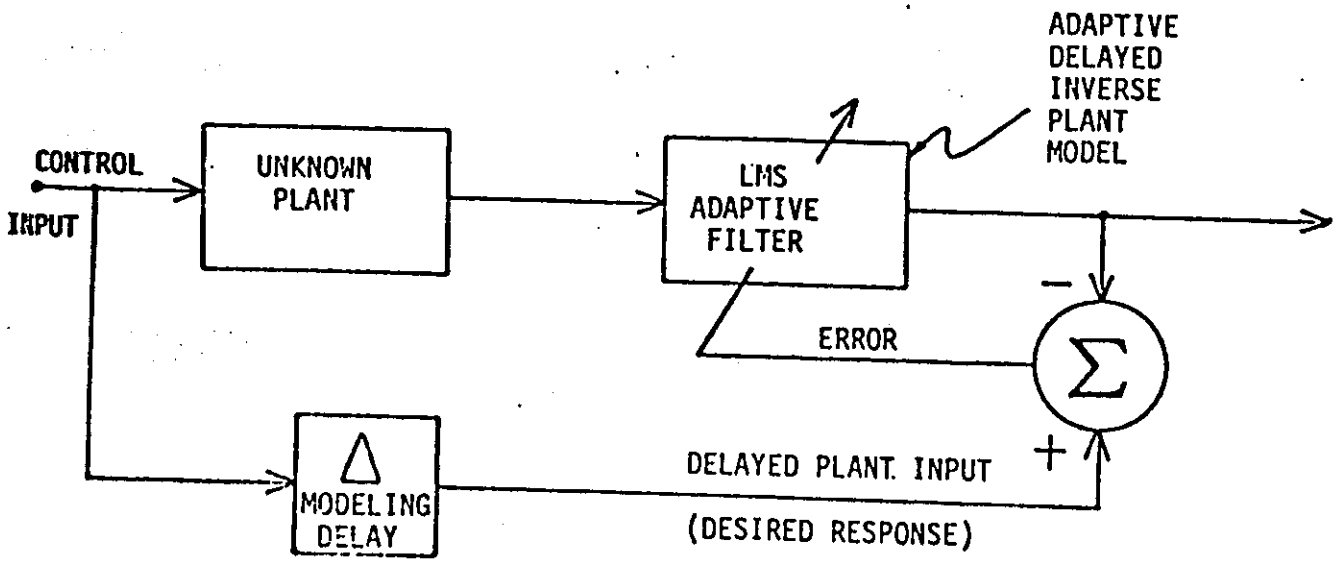


Fig. 4. Delayed Inverse Modeling of an Unknown Plant.

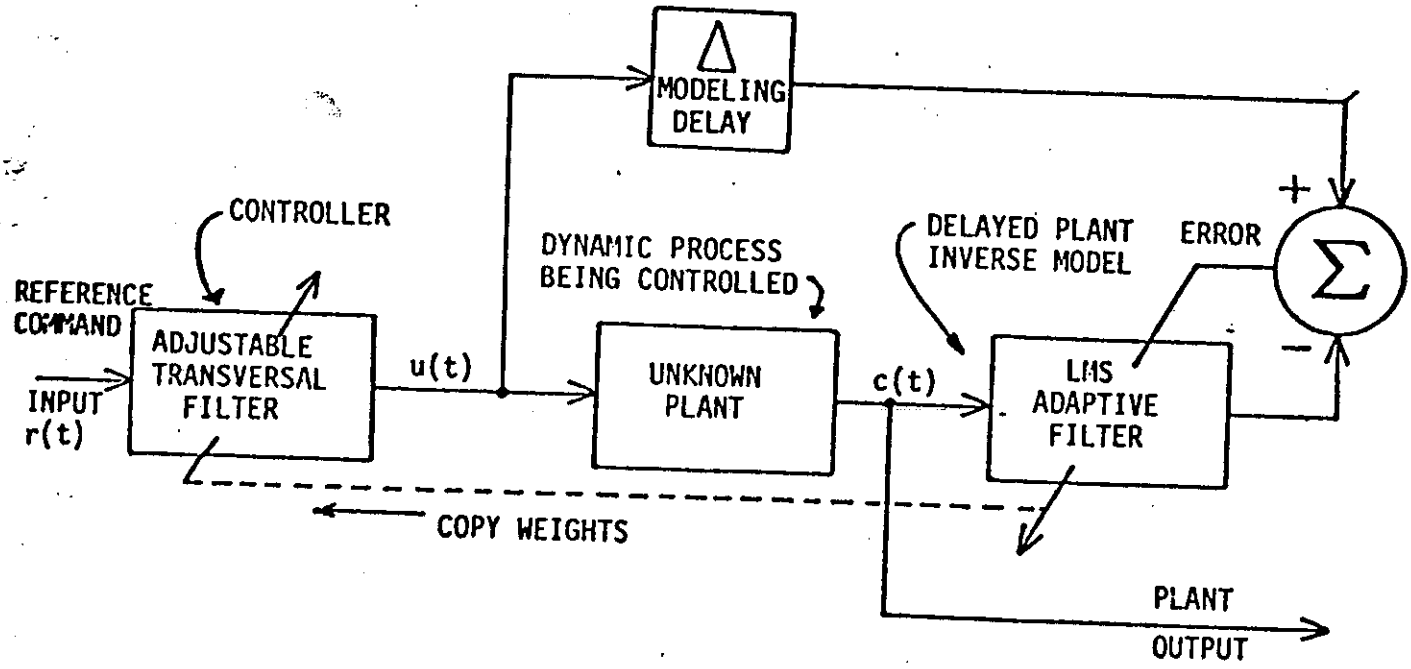


Fig. 5. Adaptive Inverse Model Control System.

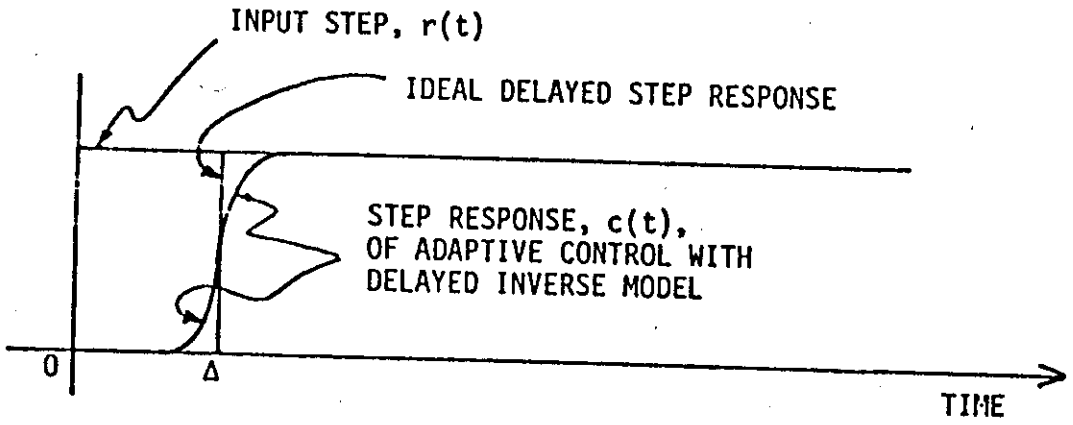


Fig. 6. Comparison of Ideal Step Response vs. Adaptive Delayed Inverse Model Control System Step Response.

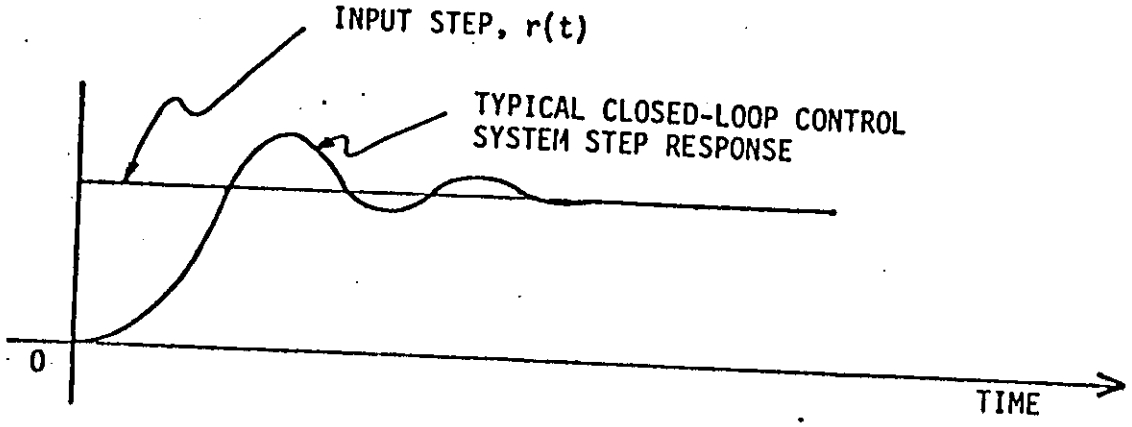


Fig. 7. Step Response of a Conventional Closed-Loop Control System.

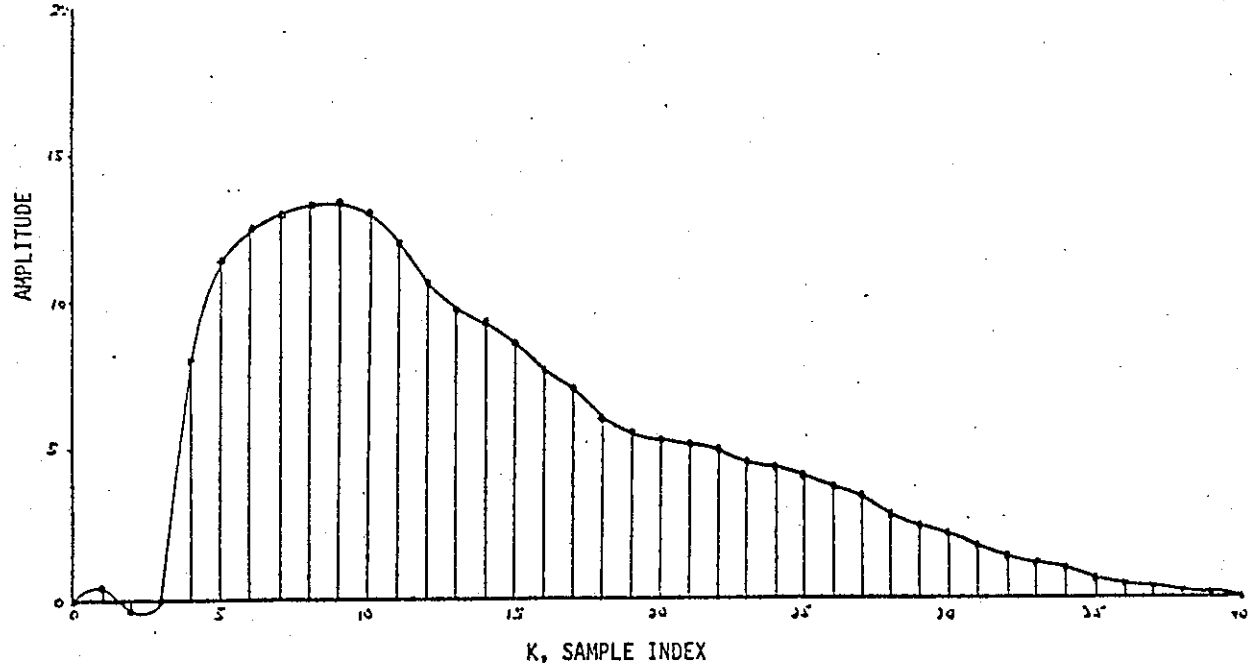


Fig. 2.5. A Discrete Impulse Response Whose Inverse Will Be Sought.

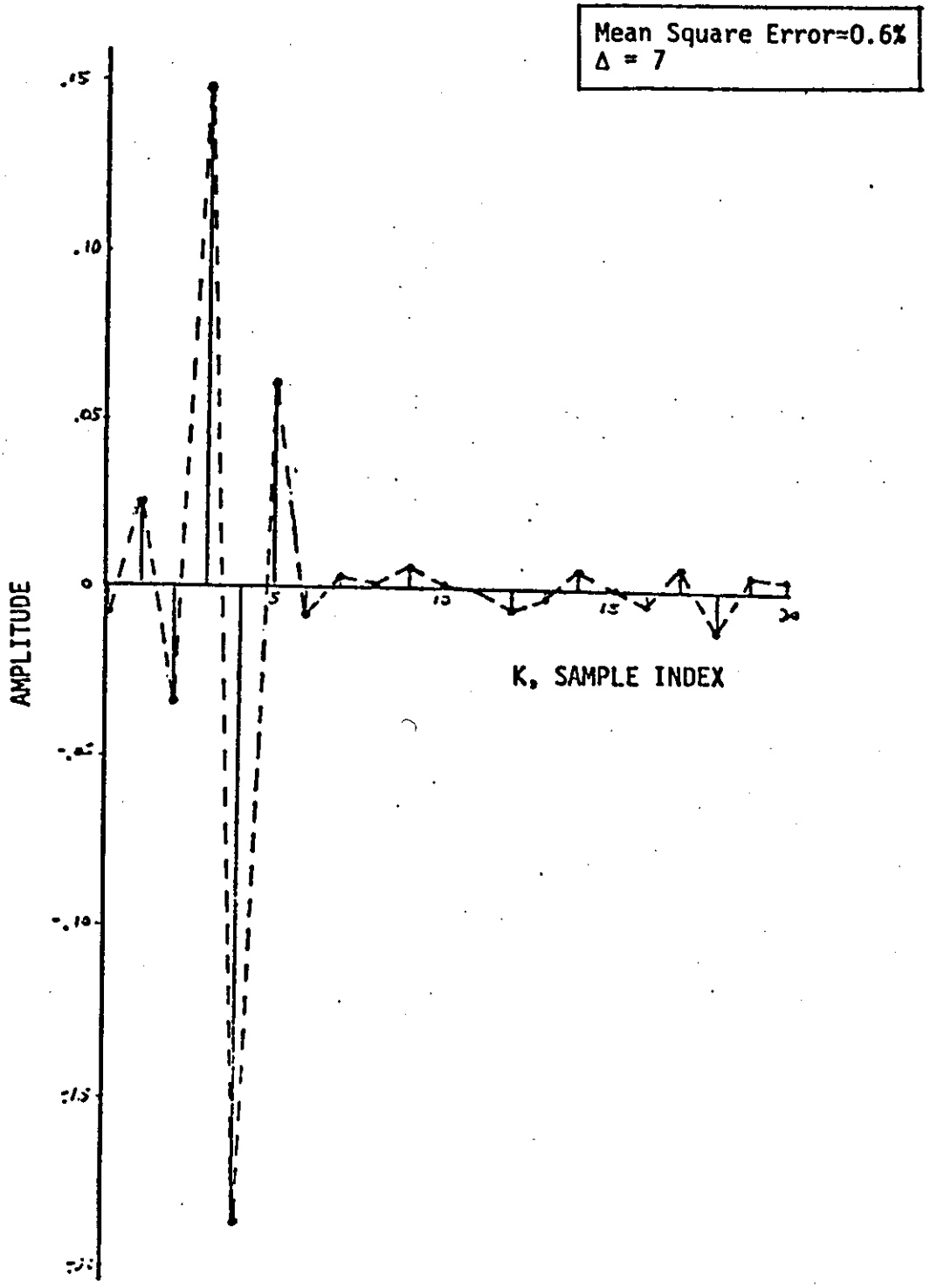


Fig. 2.7. A 21 Weight Delayed Approximate Inverse Impulse Response (7 Units of Delay).

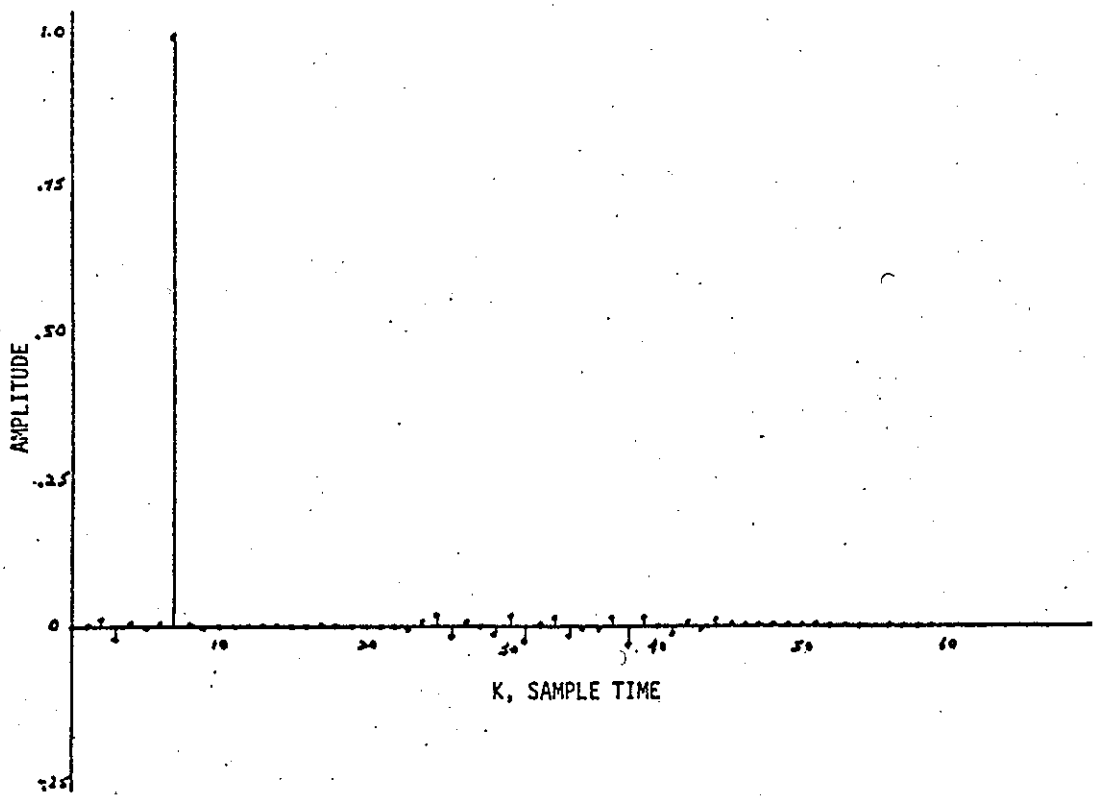


Fig. 2.8. Convolution of the Channel Impulse Response with Its Approximate Inverse

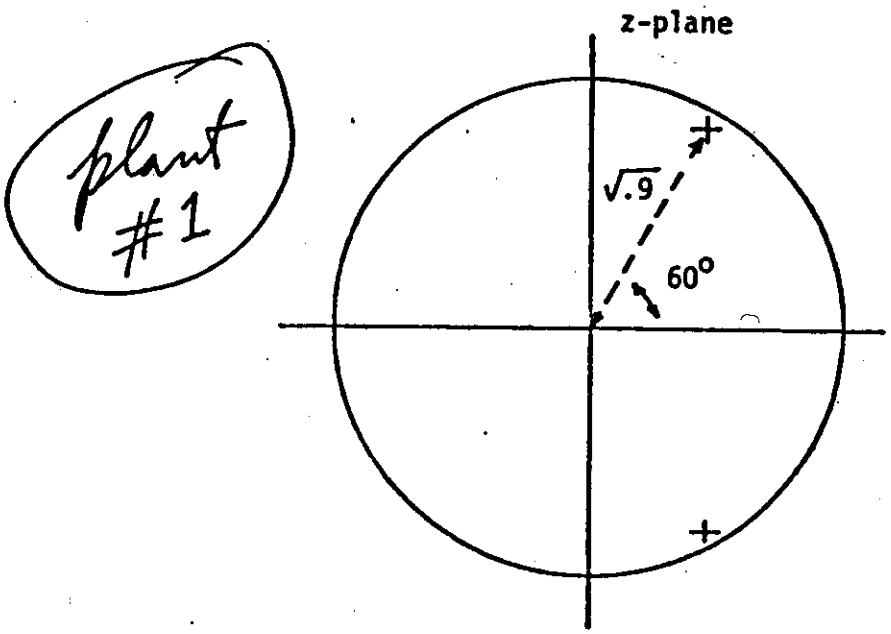


Fig. 10. Pole Locations of Plant #1.

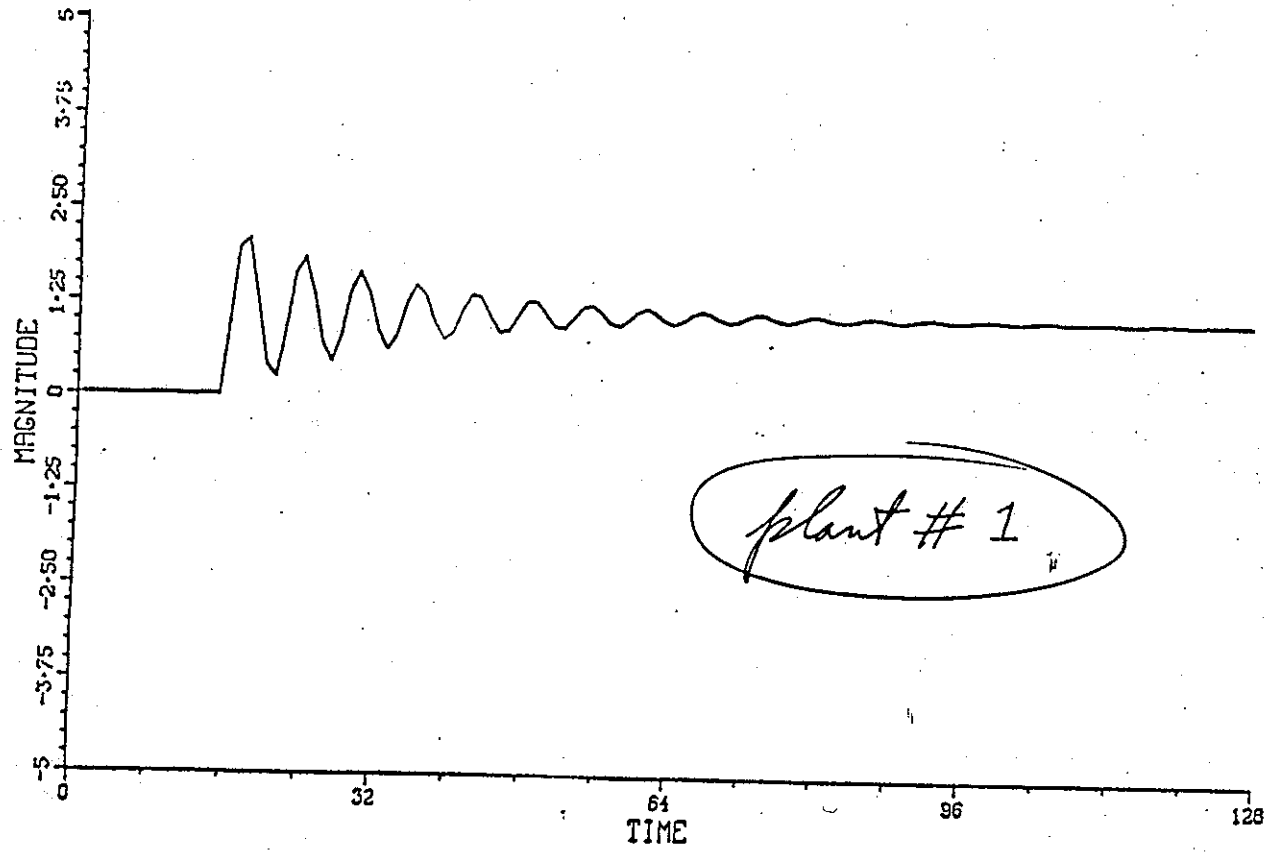


Fig. 11, Step Response of Plant #1.

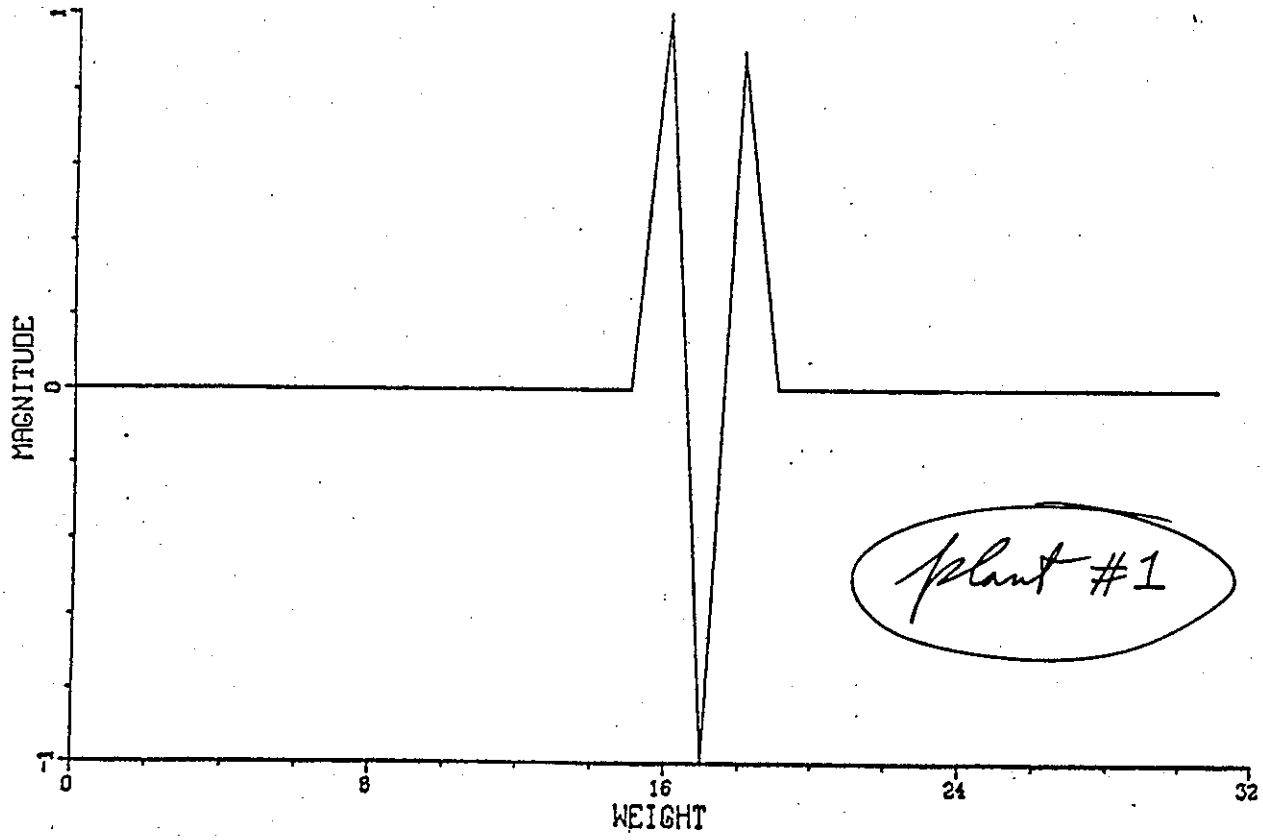


Fig. 12. Impulse Response of Inverse of Plant #1.

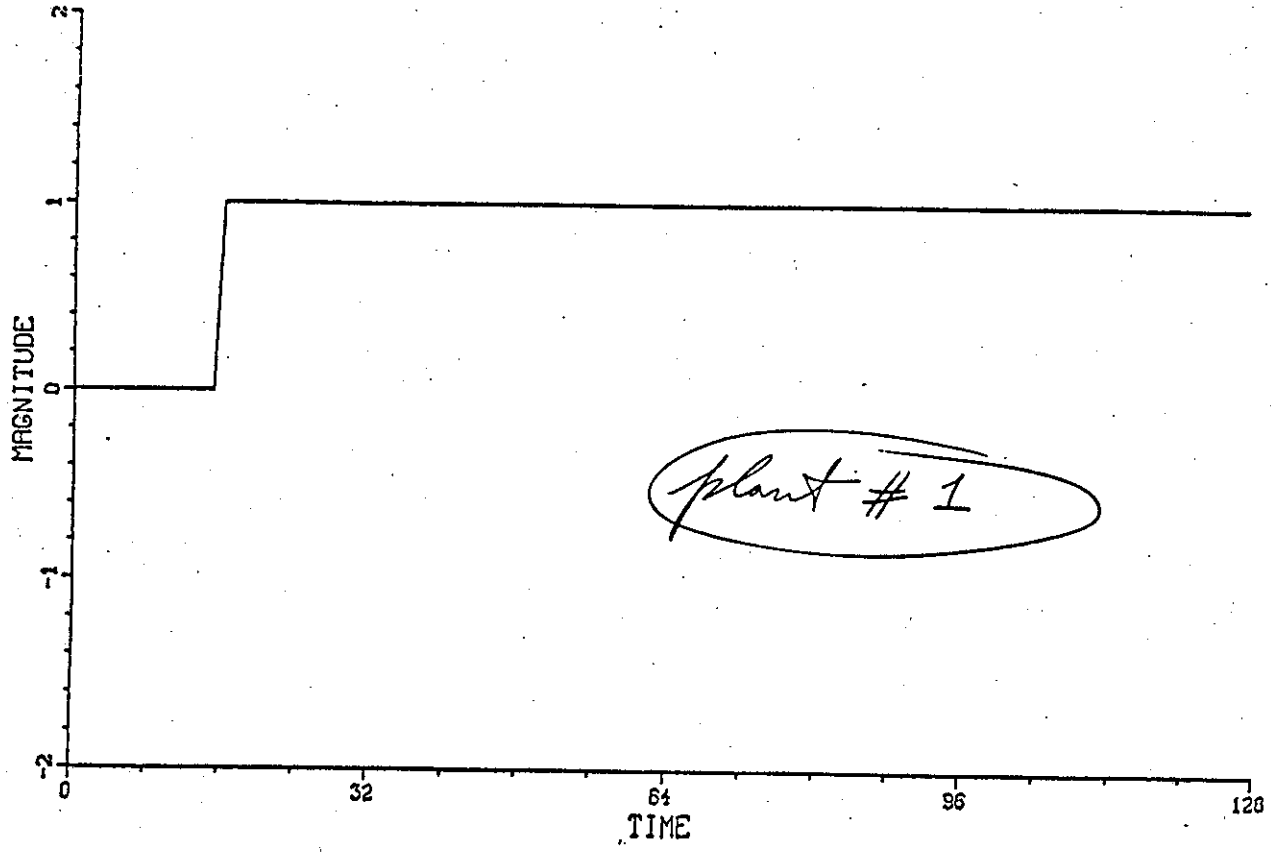


Fig. 13. Step Response of Inverse Model Control with Plant #1.

Proc. IEEE, vol. 78, no. 9, pp 1415-1441, Sept., 1990

30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation

Bernard Widrow Michael A. Lehr

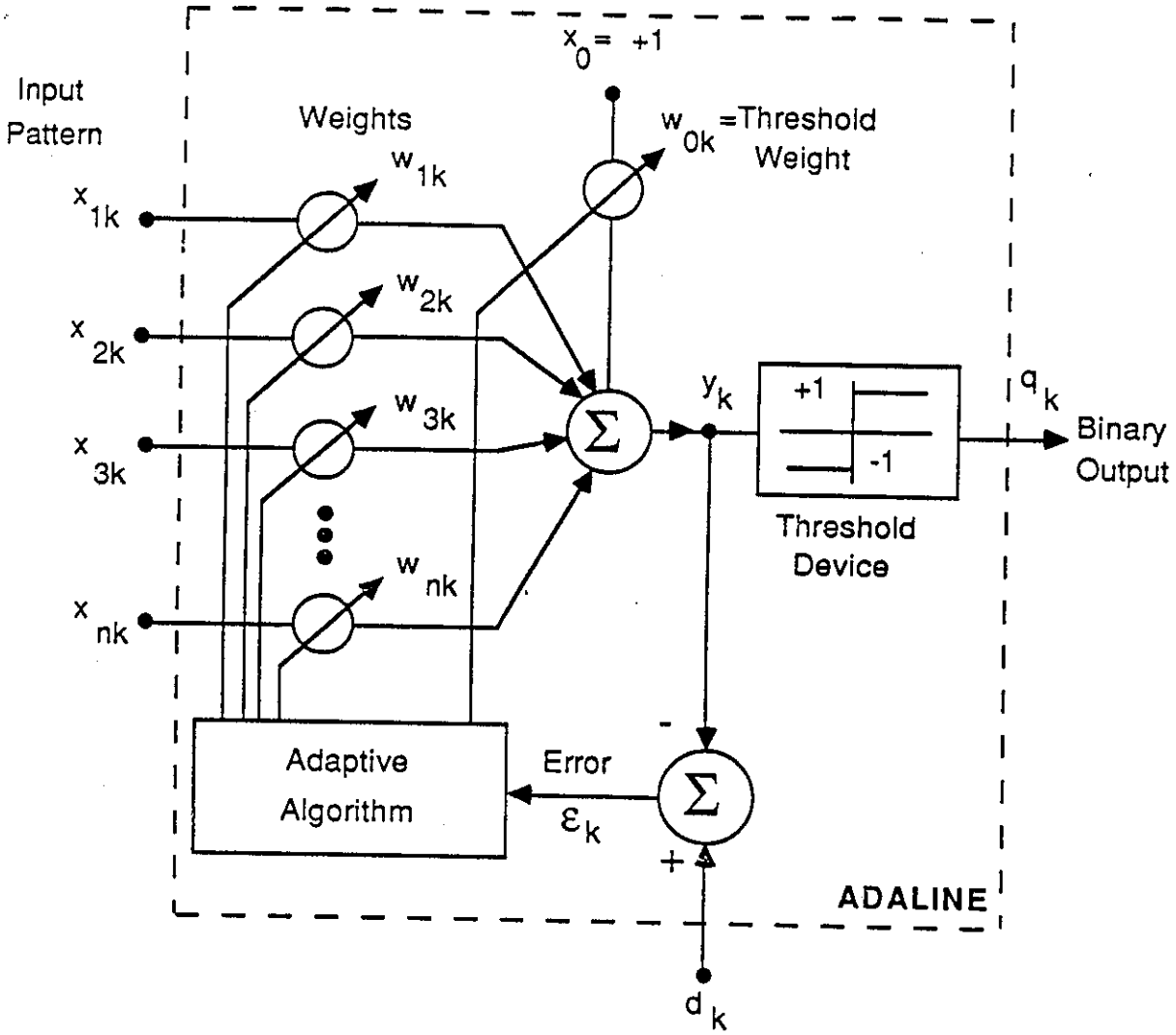
Stanford University Department of Electrical Engineering,

Stanford, CA 94305-4055

Abstract

Fundamental developments in feedforward artificial neural networks from the past thirty years are reviewed. The central theme of this paper is a description of the history, origination, operating characteristics, and basic theory of several supervised neural network training algorithms including the Perceptron rule, the LMS algorithm, three Madaline rules, and the backpropagation technique. These methods were developed independently, but with the perspective of history they can all be related to each other. The concept which underlies these algorithms is the "minimal disturbance principle," which suggests that during training it is advisable to inject new information into a network in a manner which disturbs existing information to the smallest extent possible.

37a



$$X_k = [x_{0k}, x_{1k}, x_{2k}, \ldots, x_{nk}]^T$$
$$= [+1, x_{1k}, x_{2k}, \ldots, x_{nk}]^T$$

$$W_k = [w_{0k}, w_{1k}, w_{2k}, \ldots, w_{nk}]^T$$

$$y_k = X_k^T W_k = W_k^T X_k$$

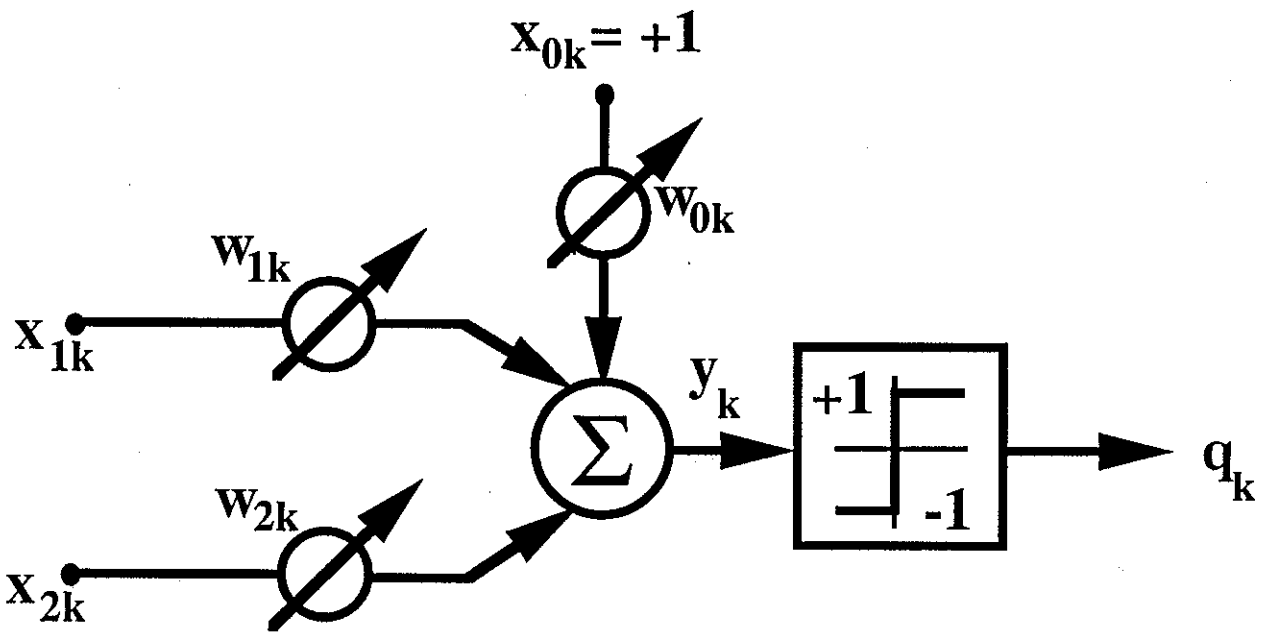
$$\varepsilon_k = d_k - y_k$$

$_k$ is time index

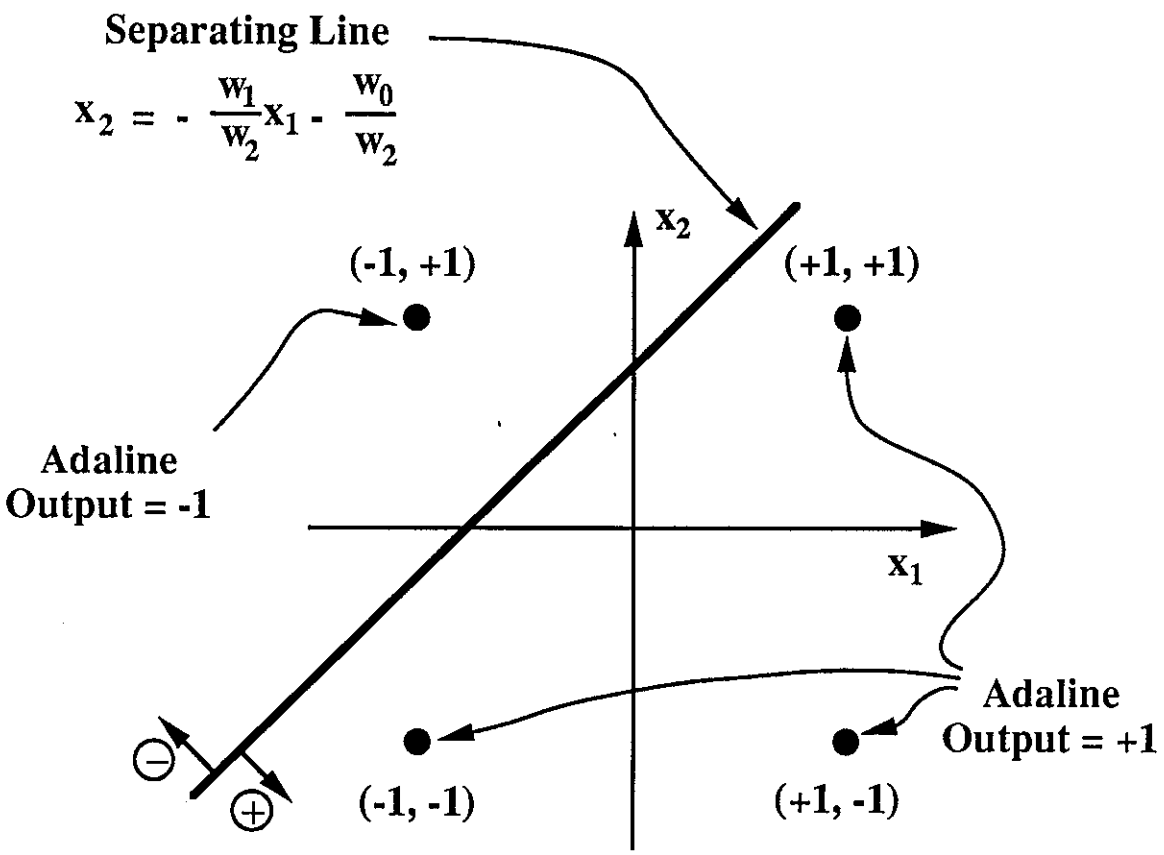
^{T} denotes vector transpose

An adaptive linear neuron (ADALINE).

$$
\begin{aligned}
y &= x_1 w_1 + x_2 w_2 + w_0 = 0 \\
x_2 &= -\frac{w_1}{w_2} x_1 - \frac{w_0}{w_2}
\end{aligned}
$$

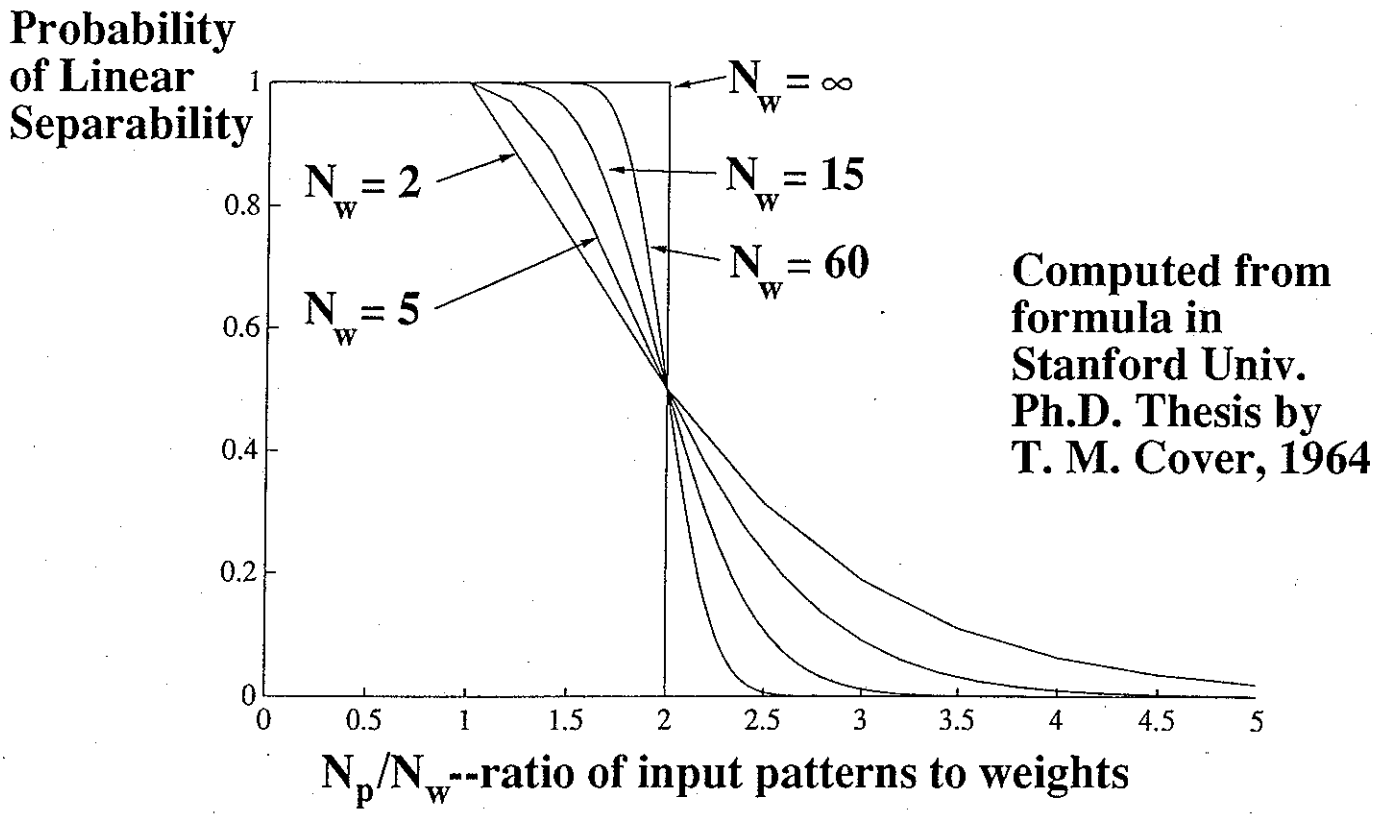


A two-Input Adaline

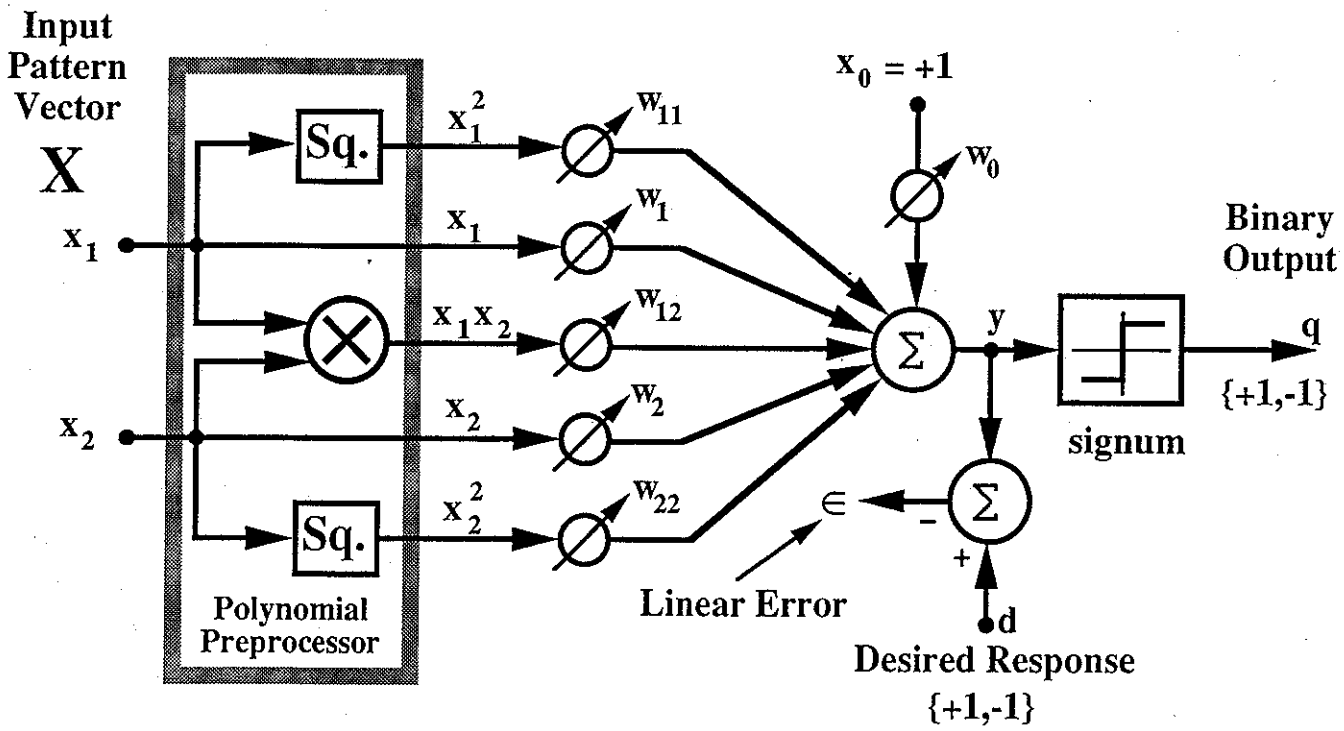


Separating line in pattern space

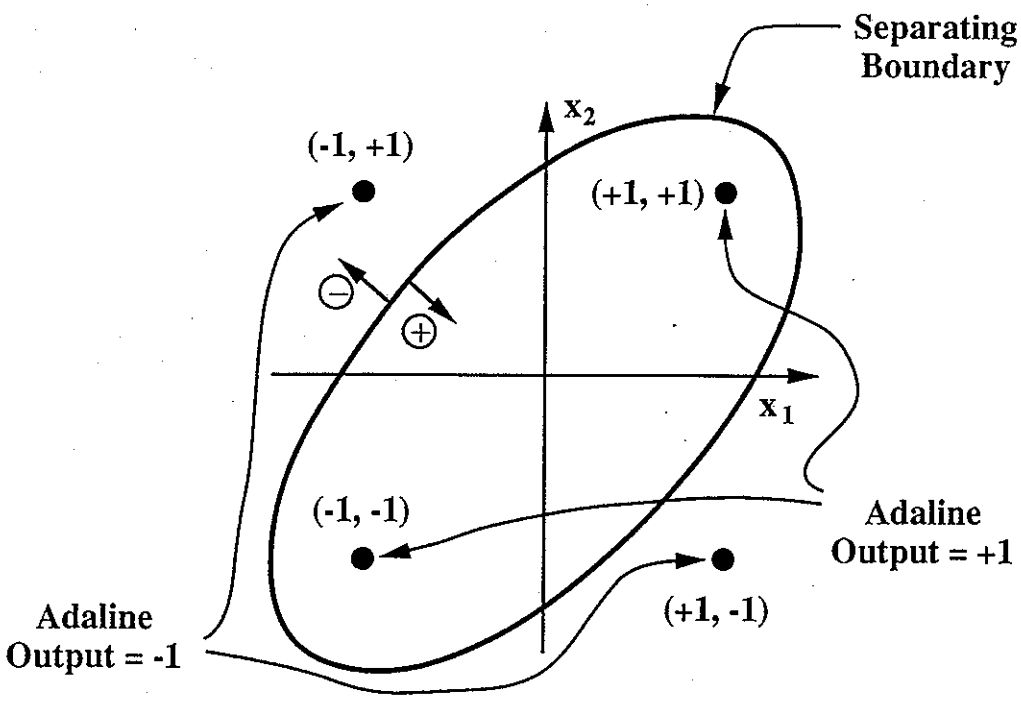
Adaline Capacity



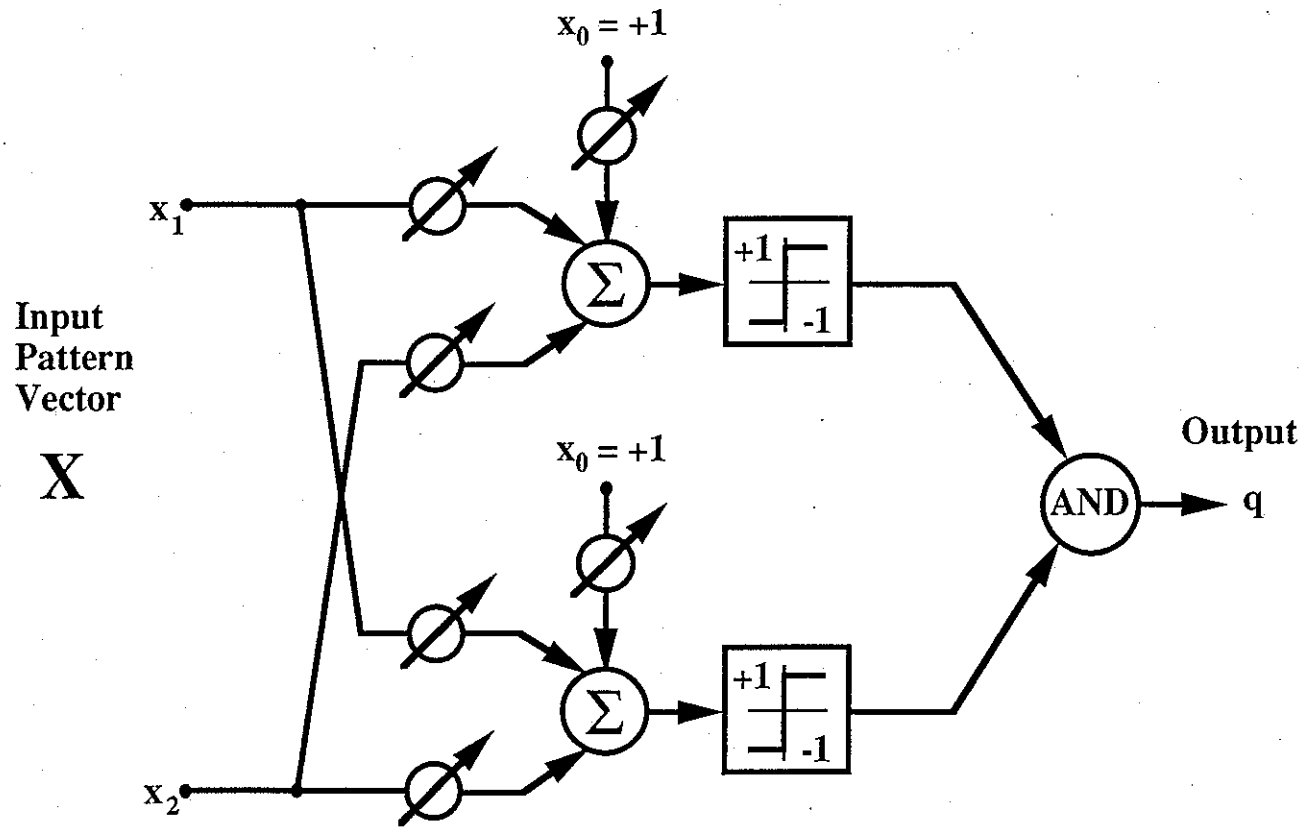
Computed from formula in Stanford Univ. Ph.D. Thesis by T. M. Cover, 1964



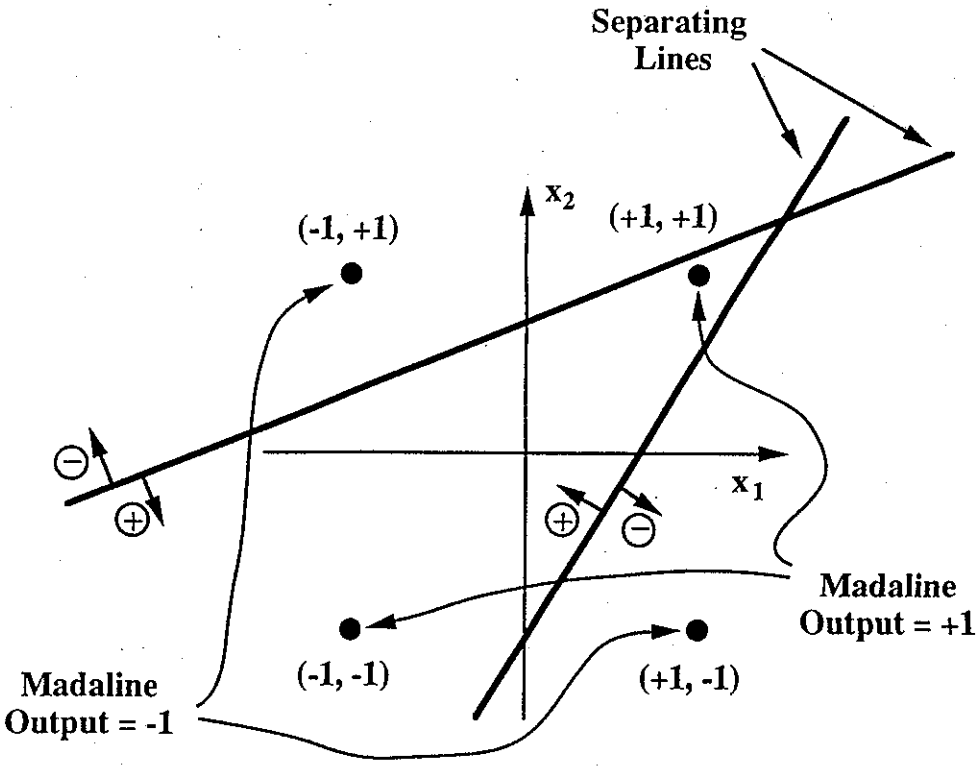
Adaline with polynomial preprocessor



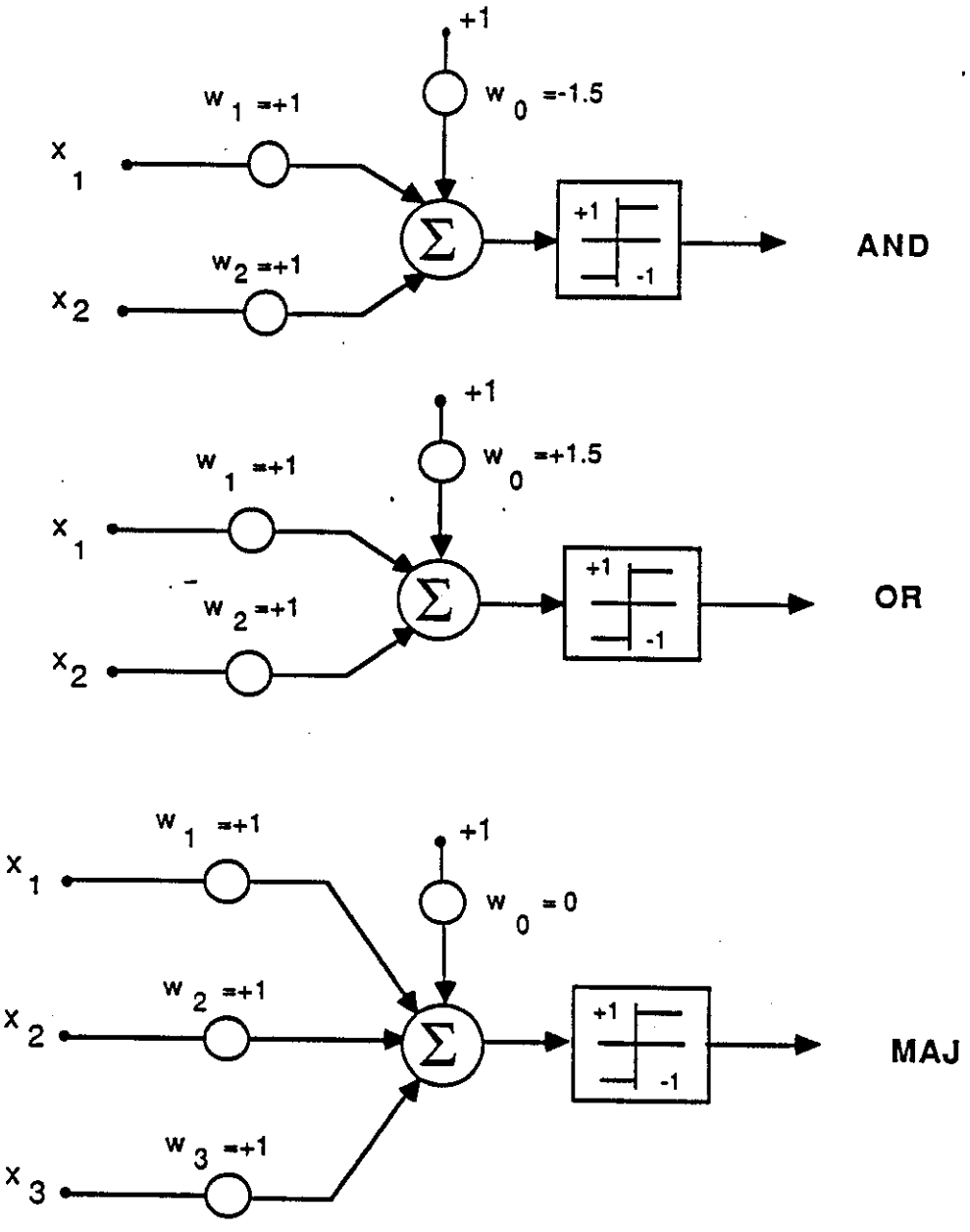
An elliptical separating boundary for the Exclusive NOR Function



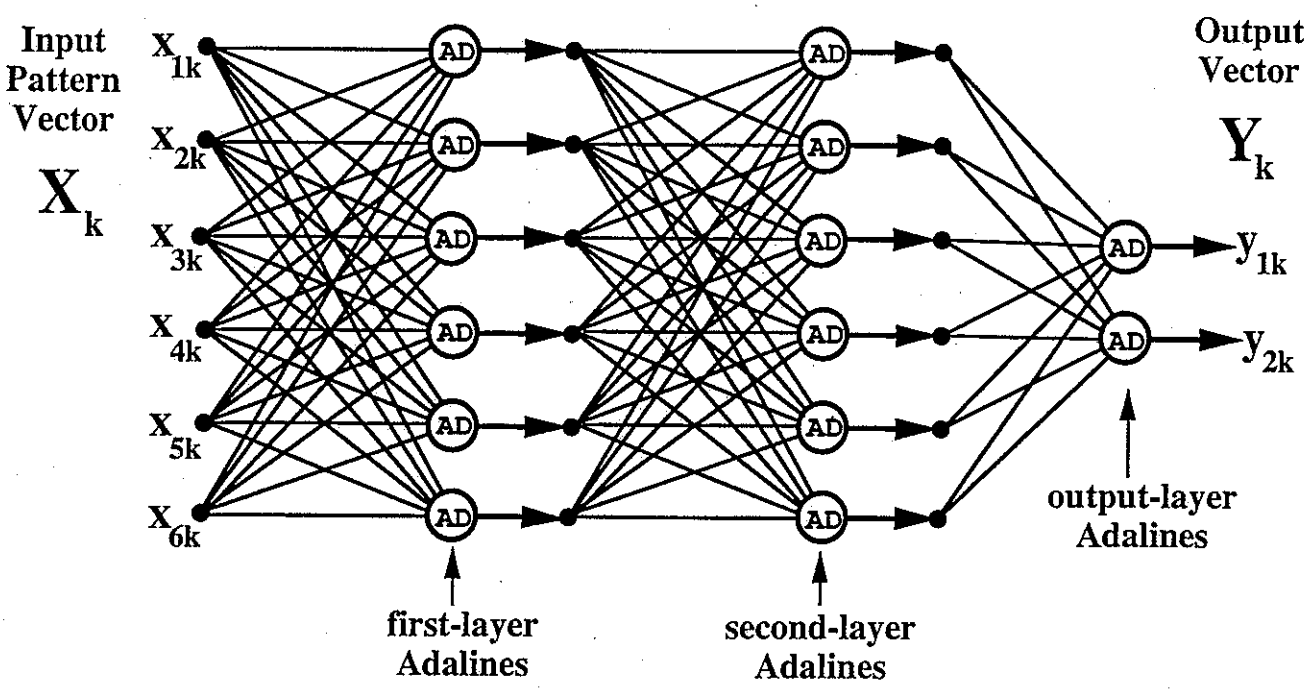
A two-Adaline form of Madaline I



Separating lines for the two-element Madaline



A neuronal implementation of AND, OR, and MAJ logic functions.



A three-layer adaptive neural network

Principle of Minimal Disturbance

Adapt to reduce the output error for the current training pattern with minimal disturbance to the responses already learned.

α-LMS Algorithm

$$\epsilon_k \triangleq d_k - \mathbf{w}_k^T \mathbf{x}_k. \tag{1}$$

Changing the weights yields a corresponding change in the error:

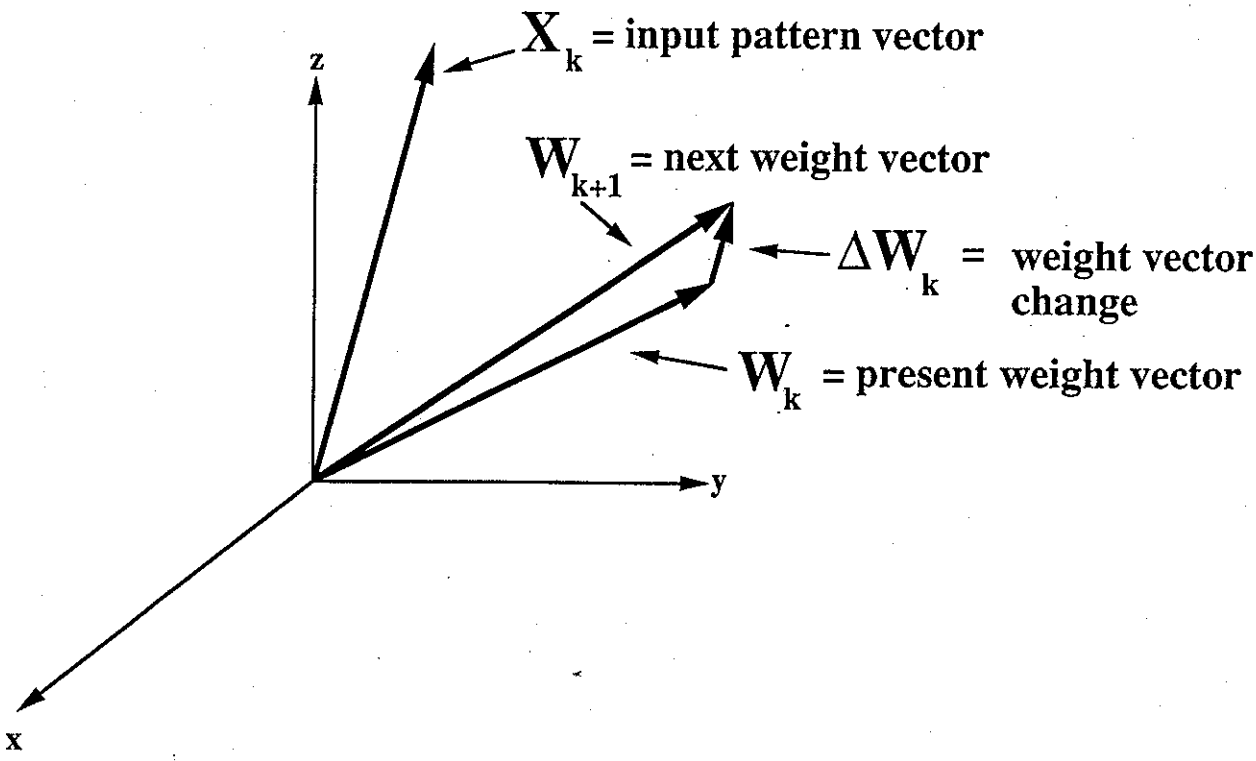
$$\Delta\epsilon_k = \Delta(d_k - \mathbf{w}_k^T \mathbf{x}_k) = -\mathbf{x}_k^T \Delta\mathbf{w}_k. \tag{2}$$

In accordance with the α-LMS rule, the weight change is as follows:

$$\Delta\mathbf{w}_k = \mathbf{w}_{k+1} - \mathbf{w}_k = \alpha \frac{\epsilon_k \mathbf{x}_k}{|\mathbf{x}_k|^2}. \tag{3}$$

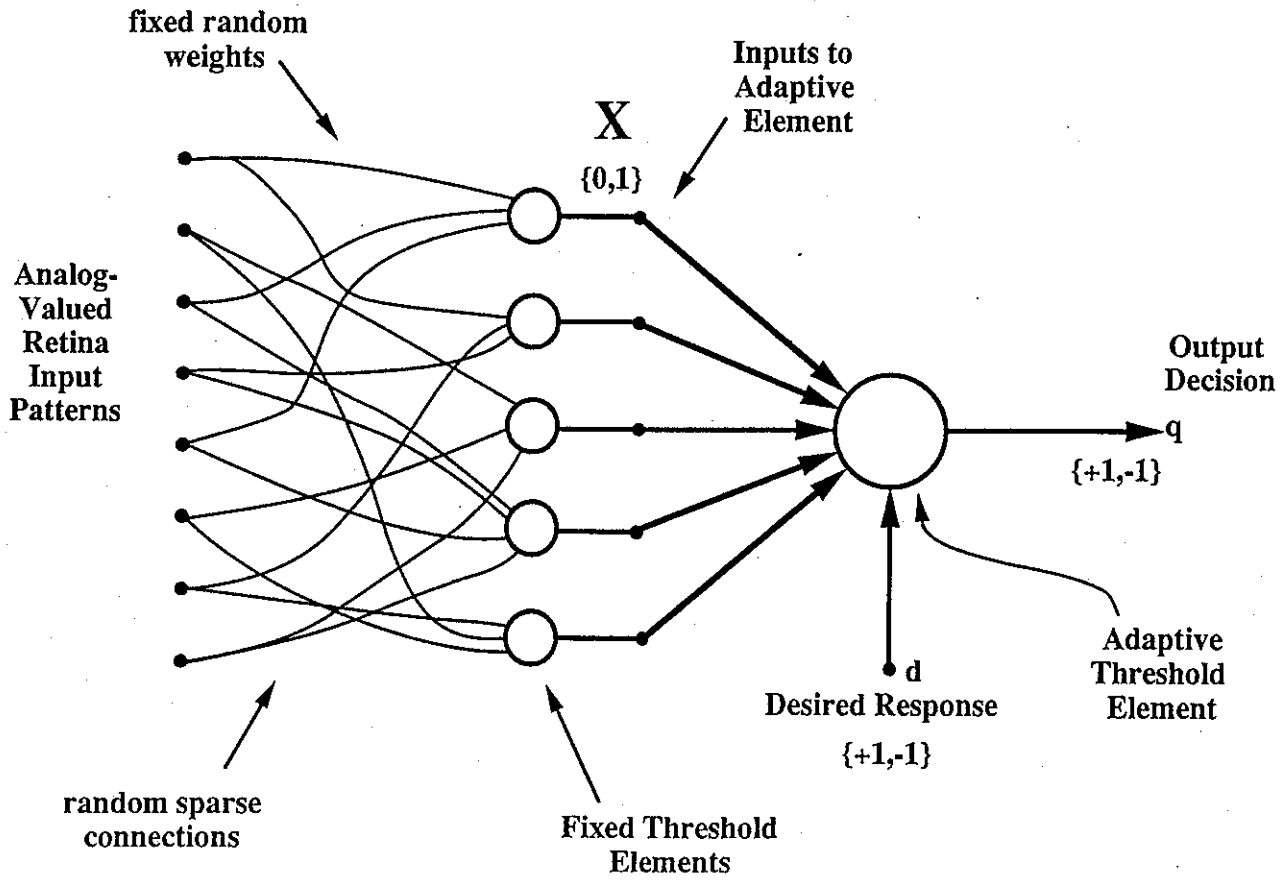
Combining Eqs. (2) and (3), we obtain

$$\Delta\epsilon_k = -\alpha \frac{\epsilon_k \mathbf{x}_k^T \mathbf{x}_k}{|\mathbf{x}_k|^2} = -\alpha\epsilon_k. \tag{4}$$

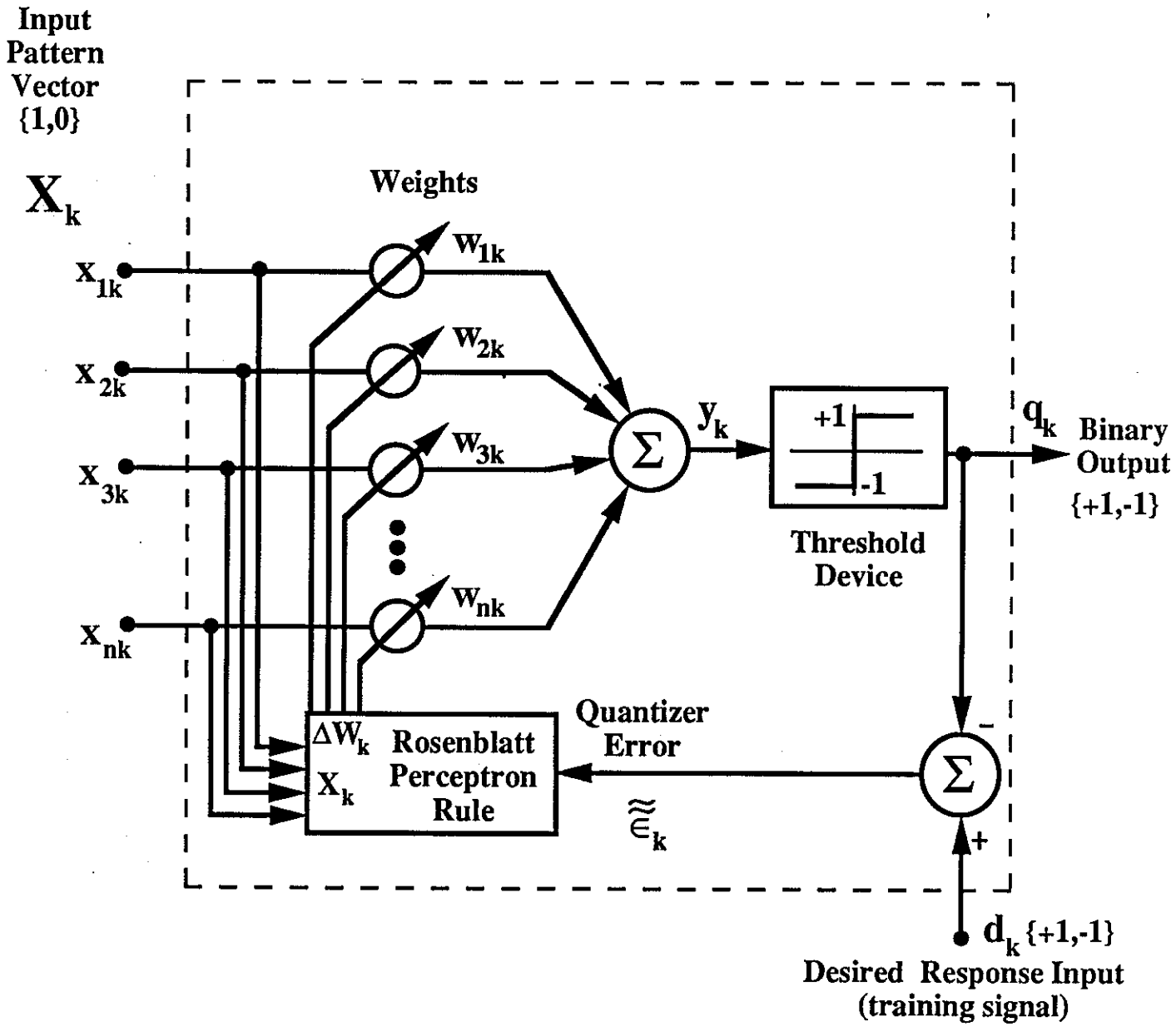


Weight correction by the LMS rule

Rosenblatt's Perceptron



Adaptive Threshold Element in the Perceptron



Perceptron Rule:

$$W_{k+1} = W_k + \mu \approx\approx\epsilon_k X_k$$

μ normally set to 1/2

Perceptron Rule

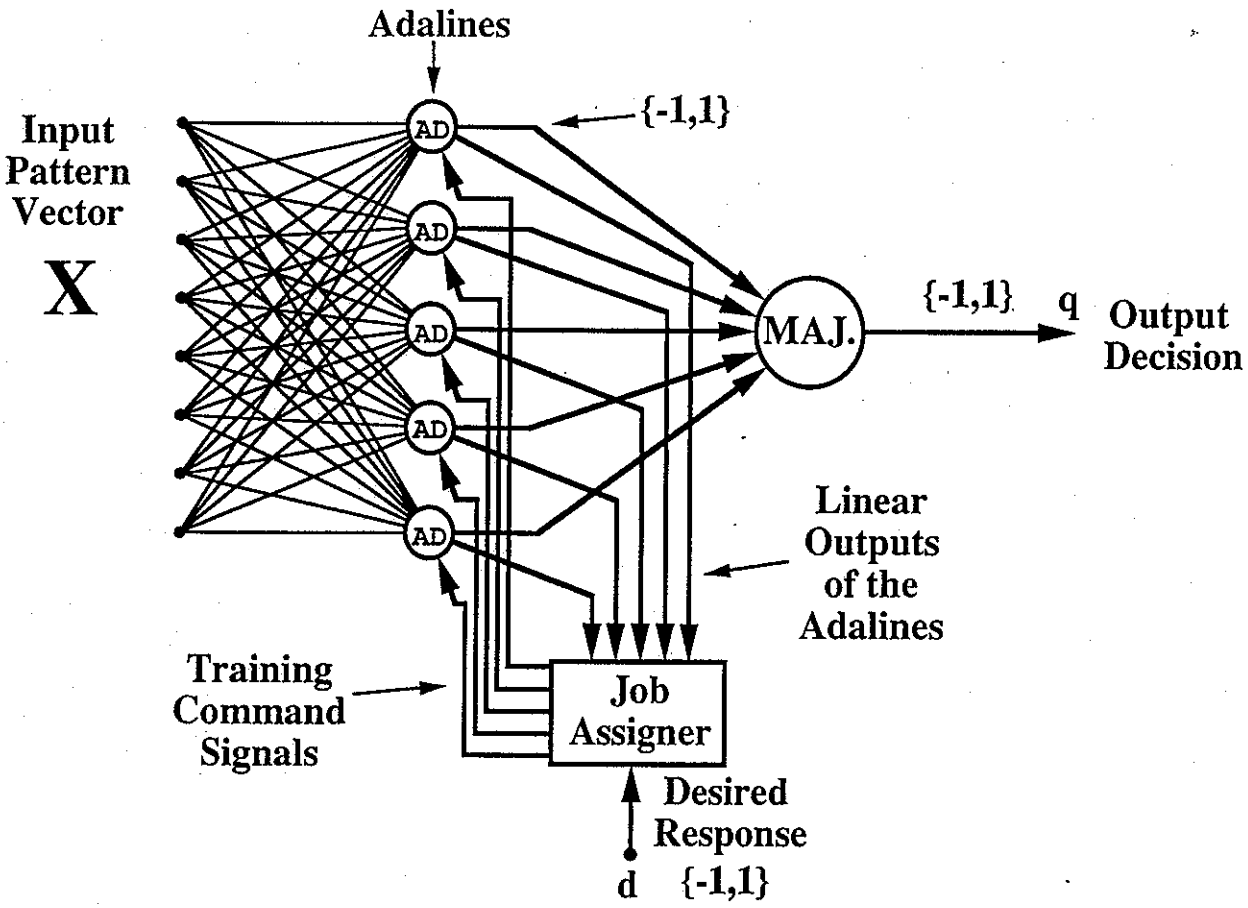
- If response is OK, do not adapt weights.
- Otherwise adapt weights by a fixed distance along the X-Vector to reduce error

Good Features

- Guaranteed to converge to solution if problem is linearly separable

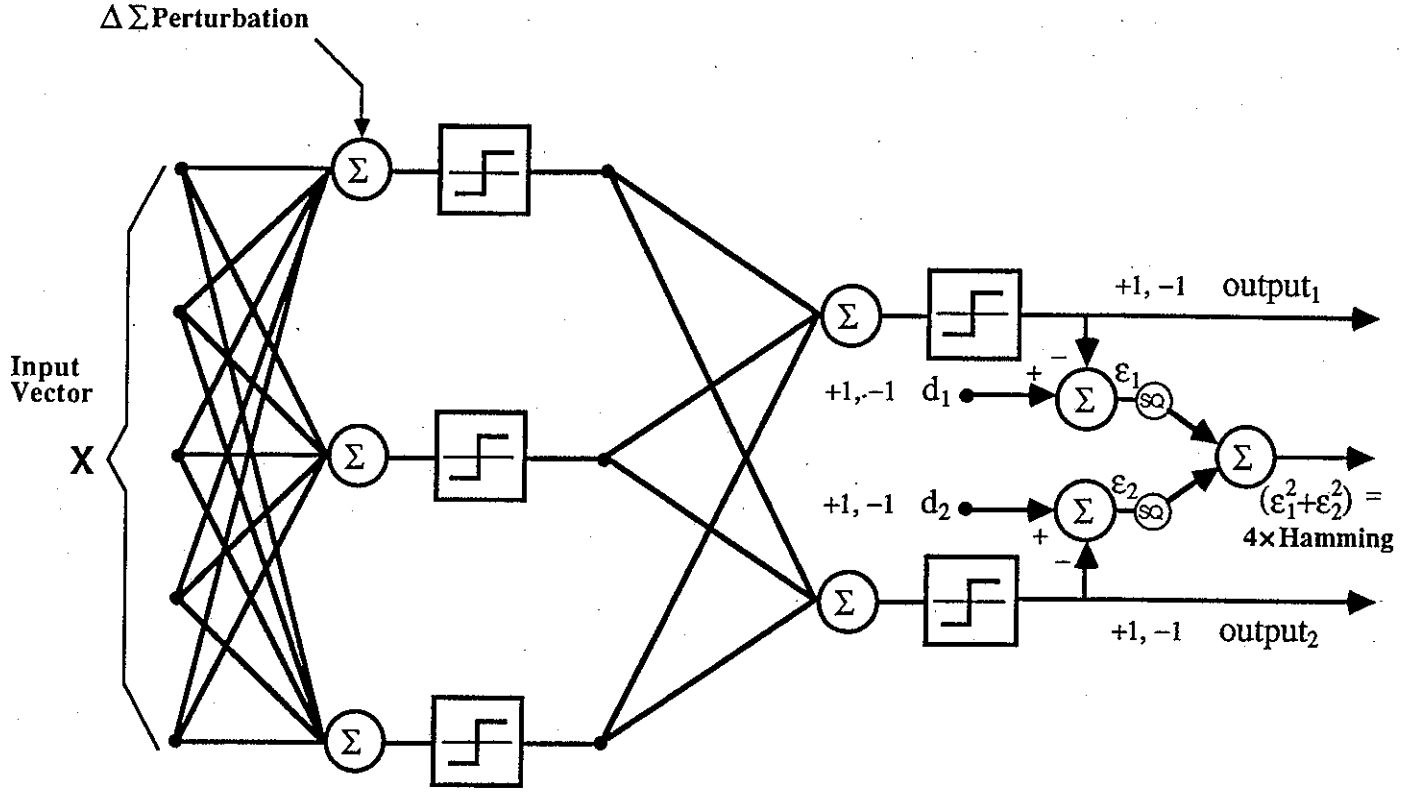
Bad Features

- Performs poorly if training set is not linearly separable.



A five-Adaline example of the MRI Architecture

MRII of B. Widrow & R. Winter



For each layer, beginning with layer 1:

Toggle output of neuron with sum closest to zero. If output Hamming error is reduced, adapt neuron. Repeat for neuron whose sum is next closest to zero, etc. Can also adapt two at a time, etc. Adaptation reduces Hamming error.