

Adaptive Filtering

Here we present an approach to signal filtering using an *adaptive filter* that is in some sense self-designing (really self-optimizing). The adaptive filter described here bases its own "design" (its internal adjustment settings) upon *estimated* (measured) statistical characteristics of input and output signals. The statistics are not measured explicitly and then used to design the filter; rather, the filter design is accomplished in a single process by a recursive algorithm that automatically updates the system adjustments with the arrival of each new data sample. How do we build such system?

A set of stationary input signals is weighted and summed to form an output signal. The input signals in the set are assumed to occur *simultaneously* (this is just for the general framework, we'll see then how to apply this method in order to model an autoregressive model!) and discretely in time. The j_{th} set of input signals is designated by the vector $\mathbf{X}^T(j) = [x_1(j), x_2(j), \dots, x_n(j)]$, the set of weights is designated by the vector $\mathbf{W}^T(j) = [w_1(j), w_2(j), \dots, w_n(j)]$, the j_{th} output signal is:

$$y(j) = \sum_{l=1}^n w_l(j)x_l(j) \quad (1)$$

This can be written in matrix form as:

$$y(j) = \mathbf{W}^T(j)\mathbf{X}(j) = \mathbf{X}^T(j)\mathbf{W}(j) \quad (2)$$

Denoting the desired response for the j_{th} set of input signals as $d(j)$, the error at the j_{th} time is:

$$\epsilon(j) = d(j) - y(j) = d(j) - \mathbf{W}^T(j)\mathbf{X}(j) \quad (3)$$

The square of this error is:

$$\epsilon^2(j) = d^2(j) - 2d(j)\mathbf{X}^T(j)\mathbf{W}(j) + \mathbf{W}^T(j)\mathbf{X}(j)\mathbf{X}^T(j)\mathbf{W}(j) \quad (4)$$

Which can be rewritten as

$$E[\epsilon^2(j)] = d^2(j) - 2\Phi^T(x, d)\mathbf{W}(j) + \mathbf{W}^T(j)\Phi(x, x)\mathbf{W}(j) \quad (5)$$

where the vector of cross-correlation between the input signals and the desired response is defined as

$$\Phi(x, d) = E \begin{bmatrix} x_1(j)d(j) \\ x_2(j)d(j) \\ \vdots \\ x_n(j)d(j) \end{bmatrix} \quad (6)$$

and where the correlation matrix of the input signals is defined as

$$E[\mathbf{X}(j)\mathbf{X}^T(j)] = E \begin{bmatrix} x_1(j)x_1(j) & x_1(j)x_2(j) & \dots \\ x_2(j)x_1(j) & x_1(j)x_2(j) & \dots \\ \vdots & \vdots & \ddots \\ \vdots & \vdots & \ddots \\ \vdots & \vdots & \ddots \\ & & x_n(j)x_n(j) \end{bmatrix} = \Phi(x, x) \quad (7)$$

It may be observed that for stationary input signals, the mean-square error is precisely a second-order function of the weights. The mean-square-error performance function may be visualized as a *bowl* shaped surface, a parabolic function of the weight variables. The adaptive process has the job of continually seeking the "bottom of the bowl". A means of accomplishing this by the well-known method of steepest descent is discussed below.

In the nonstationary case, the bottom of the bowl *may be moving*, while the orientation and curvature of the bowl may be changing. The adaptive process has to track the bottom of the bowl when inputs are non-stationary. It will be assumed that the input and desired-response signals are stationary. Here we are concerned with transient phenomena that take place when a system is adapting to an unknown stationary input process, and in addition, it is concerned with steady-state behaviour after the adaptive transients die out.

The method of steepest descent uses gradients of the performance surface in seeking its minimum. The gradient at any point on the performance surface may be obtained by differentiating the mean-square-error function with respect to the weight vector.

The gradient is

$$\begin{aligned} \frac{\partial}{\partial \mathbf{W}(j)} (d^2(j) - 2\Phi^T(x, d)\mathbf{W}(j) + \mathbf{W}^T(j)\Phi(x, x)\mathbf{W}(j)) = \\ \underbrace{\nabla[\bar{\epsilon}^2(j)]}_{N \times 1} = \underbrace{-2\Phi(x, d)}_{N \times 1} + \underbrace{2\Phi(x, x)\mathbf{W}(j)}_{N \times 1} \end{aligned} \quad (8)$$

Note that

$$\begin{aligned} \frac{\partial}{\partial \mathbf{W}(j)} (-2\Phi^T(x, d)\mathbf{W}(j)) &= \underbrace{-2\Phi(x, d)}_{N \times 1} \\ \frac{\partial}{\partial \mathbf{W}(j)} (\mathbf{W}^T(j)\Phi(x, x)\mathbf{W}(j)) &= \underbrace{2\Phi(x, x)\mathbf{W}(j)}_{N \times 1} \end{aligned} \quad (9)$$

To find the "optimal" weight vector \mathbf{W}_{LMS} that yields the least mean-square error, set the gradient to zero. Accordingly:

$$\begin{aligned} \underbrace{\Phi(x, d)}_{N \times 1} &= \underbrace{\Phi(x, x)}_{N \times N} \underbrace{\mathbf{W}_{LMS}}_{N \times 1} \\ \mathbf{W}_{LMS} &= \underbrace{\Phi^{-1}(x, x)}_{N \times N} \underbrace{\Phi(x, d)}_{N \times 1} \end{aligned} \quad (10)$$

The equation above is the *Wiener-Hopf equation* in matrix form.

However it is also possible to evaluate the optimal vector *iteratively*, where in each step we change the vector *proportionally to the negative of the gradient vector*.

$$\mathbf{W}_{j+1} = \mathbf{W}_j - \mu \nabla_j \quad (11)$$

where μ is a scalar that controls the stability and rate of convergence of the algorithm. It is easy to demonstrate that

$$\nabla[\bar{\epsilon}^2(j)] = -2\Phi(x, d) + 2\Phi(x, x)\mathbf{W}(j) = -2(d(j) - \mathbf{W}^T(j)\mathbf{X}(j))\mathbf{X}_j = -2\epsilon_j\mathbf{X}_j \quad (12)$$

And the optimal weight vector is estimated as

$$\bar{\mathbf{W}}_{j+1} = \bar{\mathbf{W}}_j + 2\mu\epsilon_j\mathbf{X}_j \quad (13)$$

It's important to observe that in this case we don't have to compute neither the *autocorrelation* neither the *cross-correlation* matrices, which is really computationally convenient!

Moreover the equation above can be visualized in a block diagram as you can see in this figure:

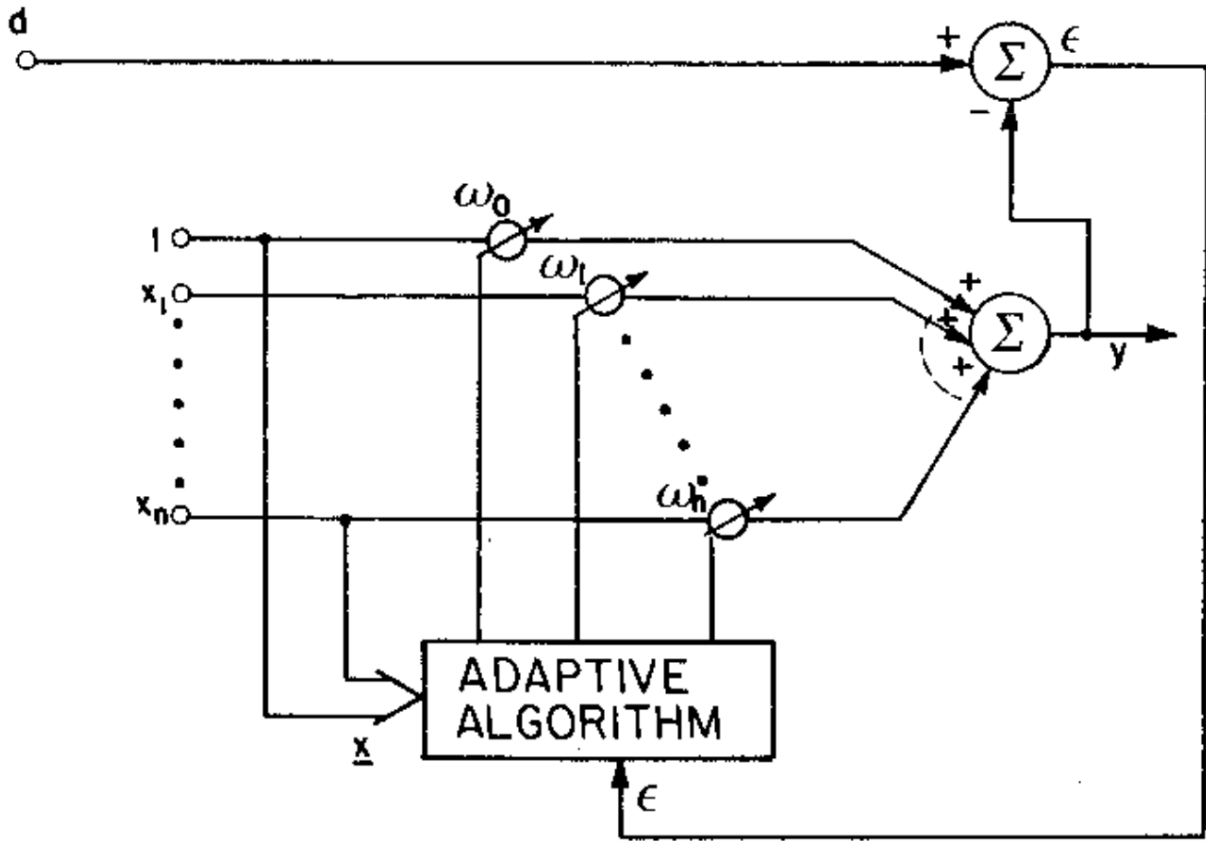


FIGURE 4. Adaptive linear combiner.

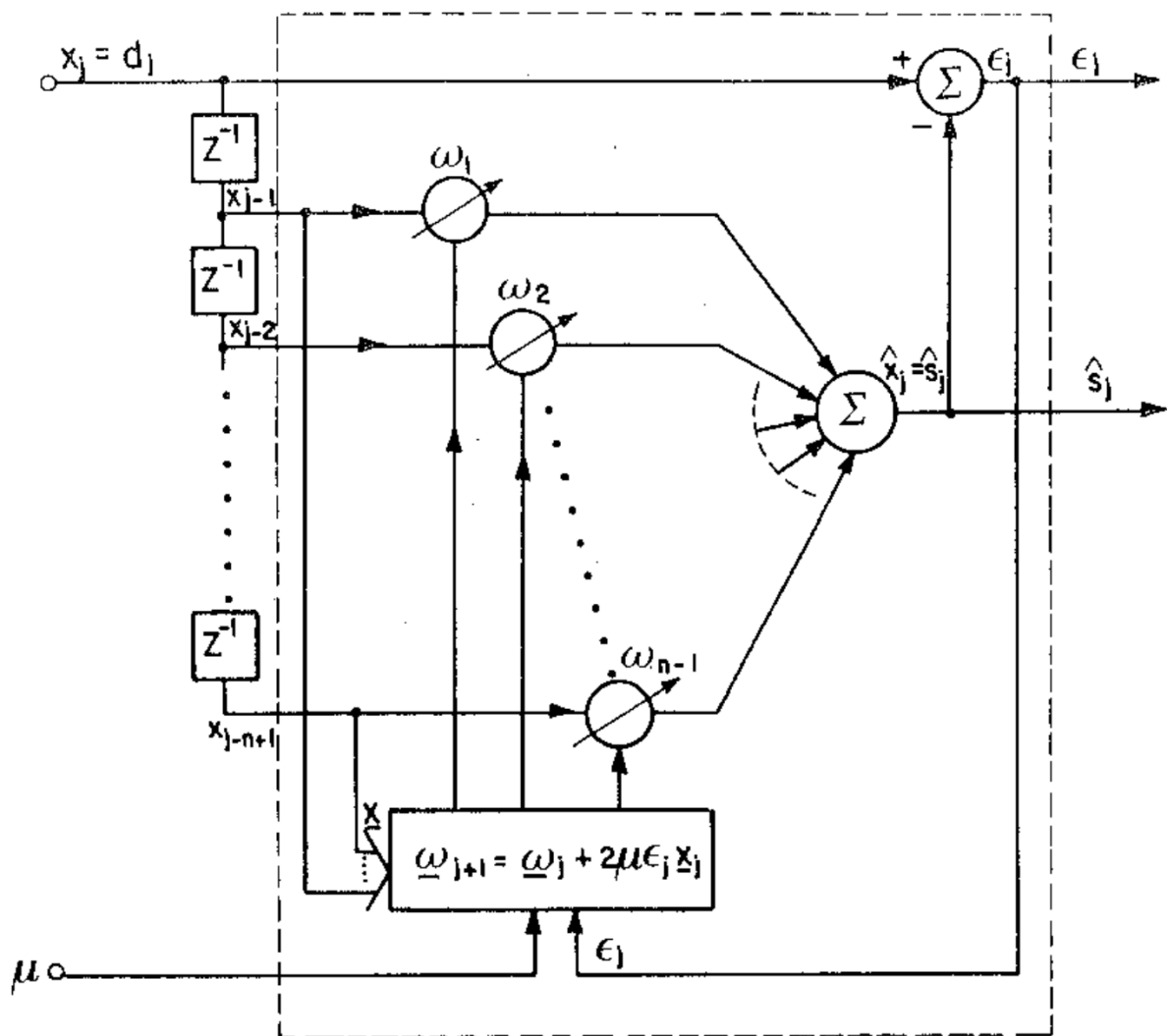
A necessary and sufficient condition for convergence is:

$$0 < \mu < \lambda_{max}^{-1} \quad (14)$$

where λ_{max} is the largest eigenvalue of the correlation matrix $\Phi(x, x)$.

But more importantly we can use the *adaptive filter* in order to model an *Autoregressive model* !

Our target (which whas d_j) becomes x_j and our inputs become $x_{j-1}, x_{j-2}, \dots, x_{j-n+1}$.



It is common practice to add a new input $x_0 = 1$ in order to associate it with a learnable bias w_0

Let's talk now about another important topic: *Adaptive Noise Cancelling*.

The concept of the adaptive noise canceller is as follows. An adaptive estimate $n_0(j)$, denoted by $\hat{n}_0(j)$, is calculated by the adaptive LMS filter. As shown before this filter is an adaptive AR filter estimating the unknown filter $H_2^{-1}(z^{-1})H_1(z^{-1})$ by means of the reference input $n_r(j)$ and the error. Adaptive noise cancelling filters have been extensively used in biomedical signal processing and many other applications.

For further details check the file `cohen-biomedical-signal-processing.pdf` at page 151