Adaptive IIR filters*

Douglas L. Jones

This work is produced by OpenStax-CNX and licensed under the
Creative Commons Attribution License 1.0†

Adaptive IIR filters are attractive for the same reasons that IIR filters are attractive: many fewer coefficients may be needed to achieve the desired performance in some applications. However, it is more difficult to develop stable IIR algorithms, they can converge **very** slowly, and they are susceptible to local minima. Nonetheless, adaptive IIR algorithms are used in some applications (such as low frequency noise cancellation) in which the need for IIR-type responses is great. In some cases, the exact algorithm used by a company is a tightly guarded trade secret.

Most adaptive IIR algorithms minimize the **prediction** error, to linearize the estimation problem, as in deterministic or block linear prediction.

$$y_k = \sum_{n=1}^{L} v_n^k y_{k-n} + \sum_{n=0}^{L} w_n^k x_{k-n}$$

Thus the coefficient vector is

$$W_k = \begin{pmatrix} v_1^k \\ v_2^k \\ \vdots \\ v_L^k \\ w_0^k \\ w_1^k \\ \vdots \\ w_L^k \end{pmatrix}$$

*Version 1.1: May 11, 2004 11:39 am -0500
†http://creativecommons.org/licenses/by/1.0

and the "signal" vector is

$$U_k = \begin{pmatrix} y_{k-1} \\ y_{k-2} \\ \vdots \\ y_{k-L} \\ x_k \\ x_{k-1} \\ \vdots \\ x_{k-L} \end{pmatrix}$$

The error is

$$\epsilon_k = d_k - y_k = d_k - {W_k}^T U_k$$

An LMS algorithm can be derived using the approximation $E\left[{\epsilon_k}^2\right] = {\epsilon_k}^2$ or

$$\nabla_k = \frac{\partial {\epsilon_k}^2}{\partial W_k} = 2\epsilon_k \frac{\partial \epsilon_k}{\partial W_k} = 2\epsilon_k \begin{pmatrix} \frac{\partial \epsilon_k}{\partial v_1^k} \\ \vdots \\ \frac{\partial w_1^k}{\partial \epsilon_k} \\ \vdots \end{pmatrix} = -2\epsilon_k \begin{pmatrix} \frac{\partial y_k}{\partial v_1^k} \\ \vdots \\ \frac{\partial y_k}{\partial v_L^k} \\ \frac{\partial y_k}{\partial w_0^k} \\ \vdots \\ \frac{\partial y_k}{\partial w_L^k} \end{pmatrix}$$

Now

$$\frac{\partial y_k}{\partial v_i^k} = \frac{\partial \left(\sum_{n=1}^L v_n^k y_{k-n} + \sum_{n=0}^L w_n^k x_{k-n}\right)}{\partial v_i^k} = y_{k-n} + \sum_{n=1}^L v_n^k \frac{\partial y_{k-n}}{\partial v_i^k} + 0$$

$$\frac{\partial y_k}{\partial w_i^k} = \frac{\partial \left(\sum_{n=1}^L v_n^k y_{k-n} + \sum_{n=0}^L w_n^k x_{k-n}\right)}{\partial w_i^k} = \sum_{n=1}^L v_n^k \frac{\partial y_{k-n}}{\partial w_i^k} + x_{k-n}$$

Note that these are difference equations in $\frac{\partial y_k}{\partial v_i^k}$, $\frac{\partial y_k}{\partial w_i^k}$: call them $\alpha_i^k = \frac{\partial y_k}{\partial w_i^k}$, $\beta_i^k = \frac{\partial y_k}{\partial v_i^k}$, then $\nabla_k = \begin{pmatrix} \beta_1^k & \beta_2^k & \dots & \beta_L^k & \alpha_0^k & \dots \end{pmatrix}$ and the IIR LMS algorithm becomes

$$y_k = {W_k}^T U_k$$

$$\alpha_i^k = x_{k-i} + \sum_{j=1}^L v_j^k \alpha_i^{k-j}$$

$$\beta_i^k = y_{k-i} + \sum_{j=1}^L v_j^k \beta_i^{k-j}$$

$$\nabla_k = -2\epsilon_k \begin{pmatrix} \beta_1^k & \beta_2^k & \dots & \alpha_0^k & \alpha_1^k & \dots & \alpha_L^k \end{pmatrix}^T$$

and finally

$$W_{k+1} = W_k - U\nabla_k$$

where the μ may be **different** for the different IIR coefficients. Stability and convergence rate depends on these choices, of course. There are a number of variations on this algorithm.

Due to the slow convergence and the difficulties in tweaking the algorithm parameters to ensure stability, IIR algorithms are used only if there is an overriding need for an IIR-type filter.