

[Next](#) [Up](#) [Previous](#) [Contents](#)

Next: [4 Lattice Filters](#) **Up:** [2 Filter Structures supported](#) **Previous:** [2 Linear Combiner Filters](#) [Contents](#)

3 Recursive Filters

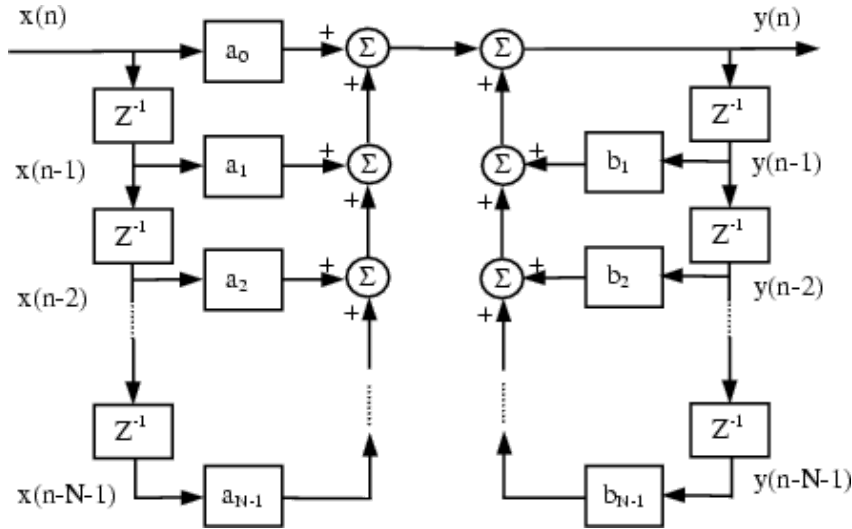


Figure 2.3: Recursive filter structure.

All filter structures mentioned so far are non-recursive structures that calculate the filter output from a linear combination of their input but do not make use of any feedback mechanism. Such non-recursive filters have impulse responses of limited duration and therefore known as Finite Impulse Response (FIR) filters. A filter structure that calculates its output as a linear combination of its current and previous input samples as well as previous samples of its output, such that shown in Fig. 2.3, is referred to as a recursive filter. Recursive filters usually have very long impulse response, therefore, they are referred to as Infinite Impulse Response (IIR) filters. The output $y(n]$ of an IIR filter is given by

$$y(n) = \sum_{i=0}^{N-1} a_i(n) \cdot x(n-i) + \sum_{j=1}^M b_j(n) \cdot y(n-j),$$

where $a_i(n); i = 0, 1, \dots, N-1$ are the feed-forward coefficients and $b_j(n); j = 1, 2, \dots, M$ are the feedback coefficients of the IIR filter. In vector notations, eq (2.5) can be written as

$$y(n) = \underline{a}(n)^T \cdot \underline{x}(n) + \underline{b}(n)^T \cdot \underline{y}(n).$$

where $\underline{a}(n) = [a_0(n) \ a_1(n) \ \dots \ a_{N-1}(n)]$ is the vector of feed-forward coefficients at time index n , $\underline{b}(n) = [b_1(n) \ b_2(n) \ \dots \ b_M(n)]$ is the vector of feedback coefficients at time index n , $\underline{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-N+1)]$, is the vector of current and past input samples, $\underline{y}(n) = [y(n-1) \ y(n-2) \ \dots \ y(n-M)]$, is the vector of past output samples, and $(\cdot)^T$ is the vector transpose operator. Besides calculating the filter output, an adaptive IIR filter must also update the $N+M$ filter coefficients to

optimize some performance function in the same manner as in the case of FIR adaptive filters. Adjusting the coefficients of an IIR filters, however, is complicated by two factors. The first is that the filter can run unstable very easily during adaptation if the filter poles shift outside the unit circle. The second is that the performance function to be optimized, in general, has many local minima which might lead to adjusting the filter coefficients to one of those minima and not to the desired global minimum. This is in contrast to the performance functions (the mean square error function for instance) usually encountered in adapting FIR filters which have a single global minimum. Despite those difficulties, recursive adaptive filters have found many practical applications, especially in control systems. In such applications, adaptive IIR filters offer great advantages when the physical system to be controlled or modeled is of a

recursive nature as is the case of adaptive control of mechanical systems, and in active vibration control systems. In such applications, an adaptive IIR filter of a few coefficients can result in much better performance than an FIR of a few thousand coefficients. The adaptive signal processing toolbox provides many IIR adaptive algorithms such as the Equation Error (see Section [6.2](#)), the Output Error (see Section [6.3](#)), and the Simple Hyperstable Adaptive Recursive Filter (see Section [6.4](#)). Furthermore, three algorithms for adapting second order IIR sections are provided, namely, SOIIR1, SOIIR2, and CSOIIR2 (see Sections [6.5](#), [6.6](#), and [6.1](#), respectively).

[Next](#) [Up](#) [Previous](#) [Contents](#)

Next: [4 Lattice Filters](#) **Up:** [2 Filter Structures supported](#) **Previous:** [2 Linear Combiner Filters](#) [Contents](#)