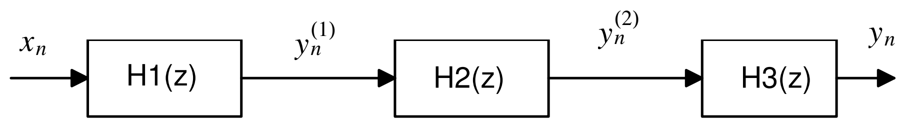
Merged-Biquad Implementation for IIR Filter

by John O'Donnell

The filter coefficients are available for a cascade form. Let's consider, specifically, a 5th order filter.

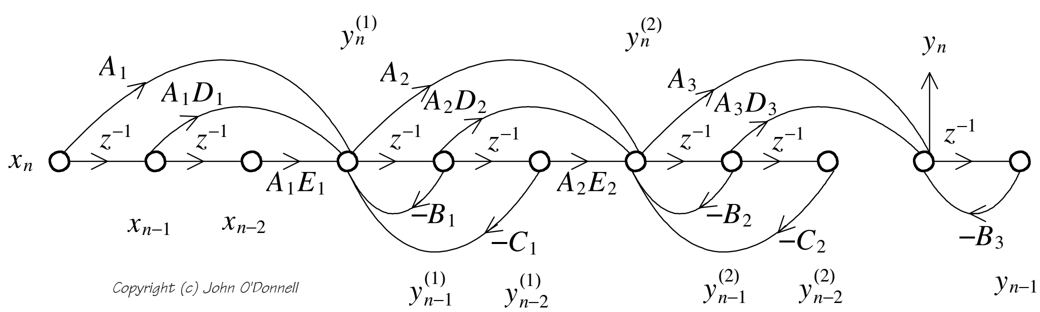
$$A_1 \frac{1 + D_1 z^{-1} + E_1 z^{-2}}{1 + B_1 z^{-1} + C_1 z^{-2}} \times A_2 \frac{1 + D_2 z^{-1} + E_2 z^{-2}}{1 + B_2 z^{-1} + C_2 z^{-2}} \times A_3 \frac{1 + D_3 z^{-1}}{1 + B_3 z^{-1}}$$

which is implemented in the cascade form



We assume that A_1, A_2, and A_3 have been scaled so that the frequency responses $|H_1(e^{j\omega})|$, $|H_2(e^{j\omega})|$, and $|H_3(e^{j\omega})|$ where $H(z) = H_1(z)\, H_2(z)\, H_3(z)$, have a peak ≤ 0 dB. With this precondition we can realize H(z) using a structure in which all "state" variables are present and past values of x_n, $y_n^{(1)}$, $y_n^{(2)}$, and y_n. Because of the scaling of the A_i coefficients, none of these quantities will overflow if the input is a full-scale sinusoid. If the input is an arbitrary sum of sinusoids—with $|x_n| \le$ full scale—then phase or delay distortion in the filter may cause a slight amount of overflow.

To realize the advantages of this scaling, each biquad is implemented in a form in which the zeros—the feedforward paths—are before the poles—the feedback paths. The following signal-flow diagram, and difference equations, describe what we call the *merged-biquad* structure.



In the signal-flow diagram above, we have indicated not only the coefficients which are used to multiply the node variables, but we have also included the identifier for each variable at each node. Note that the coefficients A_i and D_i have been combined as the products $A_i D_i$. From the diagram we can write the difference equations which will be used to program the filter.

$$
\left.\begin{aligned}
y_n^{(1)} &= A_1 x_n + A_1 D_1 x_{n-1} + A_1 E_1 x_{n-2} - B_1 y_{n-1}^{(1)} - C_1 y_{n-2}^{(1)} \\
y_n^{(2)} &= A_2 y_n^{(1)} + A_2 D_2 y_{n-1}^{(1)} + A_2 E_2 y_{n-2}^{(1)} - B_2 y_{n-1}^{(2)} - C_2 y_{n-2}^{(2)} \\
y_n &= A_3 y_n^{(2)} + A_3 D_3 y_{n-1}^{(2)} \qquad\qquad\qquad - B_3 y_{n-1}
\end{aligned}\right\} \text{Compute}
$$

$$
\left.\begin{aligned}
& x_{n-1} \rightarrow x_{n-2}\,,\; x_n \rightarrow x_{n-1} \\
& y_{n-1}^{(1)} \rightarrow y_{n-2}^{(1)}\,,\; y_n^{(1)} \rightarrow y_{n-1}^{(1)} \\
& y_{n-1}^{(2)} \rightarrow y_{n-2}^{(2)}\,,\; y_n^{(2)} \rightarrow y_{n-1}^{(2)} \\
& y_n \rightarrow y_{n-1}
\end{aligned}\right\} \text{Update the State Variables}
$$

Although we have presented the difference equations, and the updating relationships, as a group representing the specifics of this 5th-order filter example, in actual practice the merged-biquad equations would probably be embedded in a subprogram which would be called for each biquad section in the cascade. If that is done then particular attention must be paid to the updating sequence: state variable quantities for section i-1 are used by section i, so that either all state variables should be updated only after the computation for each section is done, or the subprogram should be written so that when the computation for section i is done then the state variables for section i-1 are updated.

In the DISPRO filter design software there is a scaling procedure which attempts to compute the values of the A_i so as to have the output variables, $y^{(i)}$, bounded by ± 1 for a unit amplitude sinusoid at any frequency between 0 and $F_S/2$. In effect, the frequency response of each section in the cascade has a peak of 0 dB. This scaling is almost always achievable, but, on the occasions that it is not, the peak is usually less than 6 dB, requiring a scaling of the input by 0.5, or just one bit. For an arbitrary signal input the actual output signal levels may exceed the ±1 range due to delay distortion in the filter; this is usually less than 2 dB and can easily be accommodated in the signal processing chain.