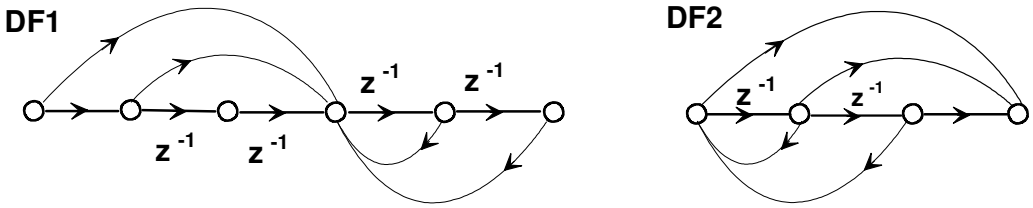
IIR Biquad Structures

A brief narrative , by John O'Donnell

Block diagram, and signal-flow graph descriptions of transfer functions date back to the early 1950s, when dynamic system simulations were done on analog computers. Much of the standard discussion is in "state-variable" textbooks of the mid 1960s, and is repeated in the Oppenheim & Schafer textbook of 1975. Of course, many dozens of derivative texts have presented the same material since then. Because it is well understood from numerical analysis results that a direct implementation of a high-order transfer function leads to severe coefficient sensitivity problems—even in floating-point arithmetic—attention has been focussed on implementations of a biquad section, with the actual transfer function realized as a cascade or parallel connection of biquad sections.

Using Oppenheim & Schafer's[1] terminology, the two simplest forms for a biquad section are the direct forms I and II—hereafter just DF1 and DF2. The DF1 is simply the cascade of the numerator and the denominator realized as separate second-order elements. The DF2 combines the numerator and denominator coefficients into a single second-order block. Because the DF2 uses the minimum number of delay elements (the output of each of which is a "state variable") it is called a canonic form. By contrast, the DF1 structure has twice the number of required delay elements and thus would be thought to be inferior to the DF2, and consequently of little importance.



For use with fixed-point arithmetic the canonic structures have the drawback that the dynamic range of their state variables is determined by the poles of the biquad. When the biquad has a pair of rather high Q complex conjugate poles—which is typical of sharp-cutoff filters— then the state variables possess a very large dynamic range for input signals which are in the normal input range--usually -1 to +1. It is not difficult to design digital filters of moderate performance which have one or more biquad sections

---

1 *Digital Signal Processing*, Prentice-Hall, 1975 [Sect. 4.3]

in which the state variables overflow the (-1,1) range by factors of 50 to 100. In building a filter with canonic biquad sections the user is forced into severe scaling-down of the input signal to avoid overflows and overflow-induced limit-cycle oscillations. The marketing groups of DSP chip manufacturers publicize the biquad timing for their chips using the canonic DF2 structure because it can be realized with the fewest instructions. However, for practical purposes the canonic DF2 structure is valueless for implementing high performance IIR filters in fixed-point arithmetic chips because of the poor signal-to-quantization-noise ratio which results from severe input signal scaling.

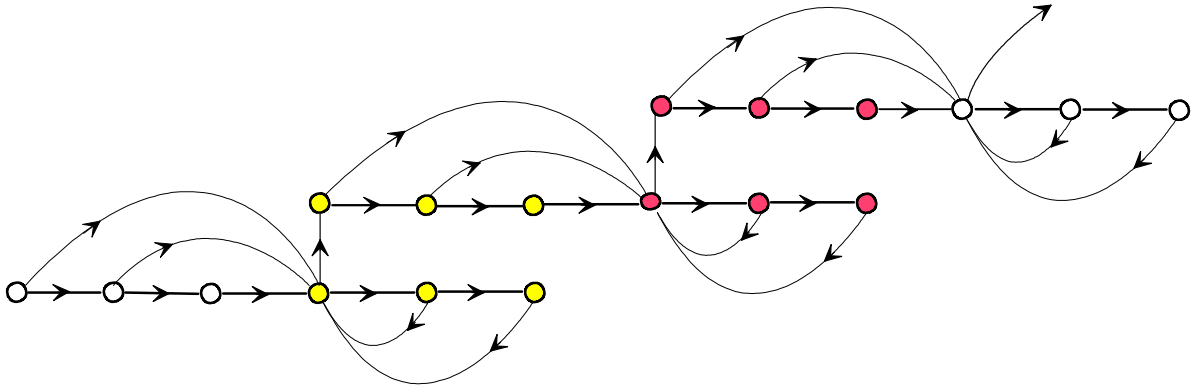
Realizing these defects in the canonic DF2 structure I looked at many other implementations of a biquad. In their DSP text, Roberts & Mullis[2] present a wide variety of structures, all based on state variable representations. Some structures provided lower sensitivity to coefficient quantization than DF1 and DF2, and other structures provided absolute, or at least relative, immunity to overflow-induced oscillations. What each of these new structures shared was a computational burden significantly greater than that of the canonic DF2.

While engaged in an effort to devise a scaling strategy for filter implementation I looked at the properties of the DF1: because of its use of two "tapped delay lines", instead of one as for DF2, all of the "state variables" are actually sample values of the input and output of the biquad. Clearly, if the biquad gain could be set so that a bounded input produced a bounded output then all of the state variables would also be bounded. Because of this attractive feature of DF1 it became a candidate for serious consideration, despite its apparent inefficiency relative to the canonic DF2. In seeking to incorporate the DF1 into my scaling strategy I realized that a substantial part of what appeared to be redundancy in the DF1 could be eliminated by merging the *zeros delay line* of a biquad section with the *poles delay line* of the preceding section. This has given rise to what I call the merged-biquad structure. The efficiency of the merged-biquad structure, relative to the DF2 canonic form, is thus dependent on the order of the filter: the more biquad sections that are cascaded, the less important the extra two delay elements become. Note that both direct forms have the same number of multiplications so that any speed difference would be associated with the accessing of the two extra storage locations for the merged-biquad form.

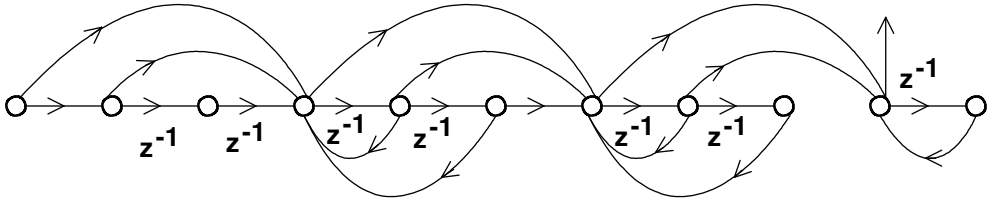
---

[2] *Digital Signal Processing*, Addison Wesley, 1987 [Chapters 8 & 9]

The diagram below shows the "normal" cascading of three DF1 biquads. It is drawn such that the overlap between the zeros delay lines and the poles delay lines is made



obvious. All that is needed is to merge the matching delay line sections in order to arrive at the merged-biquad configuration (shown below for a 5th-order filter).



Gain scaling of the biquads in an IIR filter design is an integral part of DISPRO. The procedure which is implemented is almost always successful in obtaining a gain coefficient for each biquad so that the output of each biquad will be bounded when the input to the filter is a sinusoid of amplitude 1.0 and of any frequency. In some cases it is not possible for the algorithm used in DISPRO to find a set of gains which will achieve "0 dB gain" at all frequencies for all biquad outputs. Usually the response peaking is small, of the order of 6 dB and less, so that only a modest scaling down of the signal is required. Clearly, in order to realize the benefits of this bounded-response scaling, it is necessary to implement the filter using the merged-biquad structure. In DISPRO the complete time behavior of an IIR filter implemented with either the canonic or merged-biquad form can be computed for floating-point and almost every possible configuration of fixed-point arithmetic. Any overflows or saturations are flagged on a display of the actual numerical values for all quantities computed in the biquad equations. In addition, for floating-point arithmetic, a graphical display shows the behavior of all appropriate variables.