WIKIPEDIA

Digital biquad filter

In signal processing, a **digital biquad filter** is a second order recursive linear filter, containing two poles and two zeros. "Biquad" is an abbreviation of "*biquadratic*", which refers to the fact that in the Z domain, its transfer function is the ratio of two quadratic functions:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{a_0 + a_1 z^{-1} + a_2 z^{-2}}$$

The coefficients are often normalized such that a_0 = 1:

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

High-order IIR filters can be highly sensitive to quantization of their coefficients, and can easily become unstable. This is much less of a problem with first and second-order filters; therefore, higher-order filters are typically implemented as serially-cascaded biquad sections (and a first-order filter if necessary). The two poles of the biquad filter must be inside the unit circle for it to be stable. In general, this is true for all discrete filters i.e. all poles must be inside the unit circle in the Z-domain for the filter to be stable.

Contents

Implementation

Direct form 1

The most straightforward implementation is the direct form 1, which has the following difference equation:

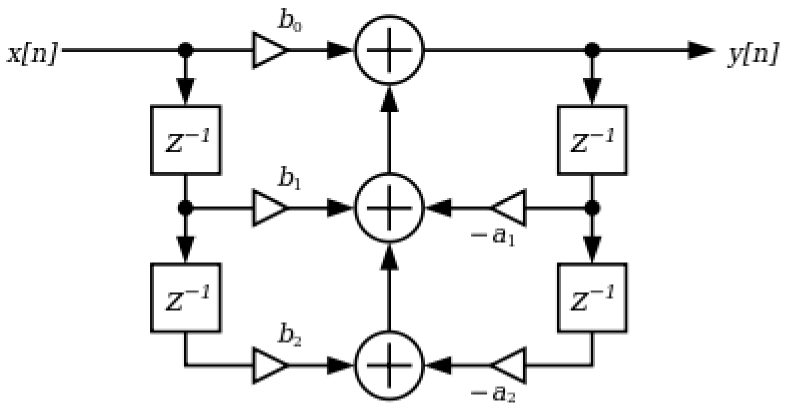
$$y[n] = \frac{1}{a_0}\left(b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] - a_1 y[n-1] - a_2 y[n-2]\right)$$

or, if normalized:

$$y[n] = b_0 x[n] + b_1 x[n-1] + b_2 x[n-2] - a_1 y[n-1] - a_2 y[n-2]$$

Here the b_0, b_1 and b_2 coefficients determine zeros, and a_1, a_2 determine the position of the poles.

Flow graph of biquad filter in direct form 1:



When these sections are cascaded for filters of order greater than 2, efficiency of implementation can be improved by noticing the z^{-1} delay of a section output is cloned in the next section input. Two storage delay components may be eliminated between sections.

Direct form 2

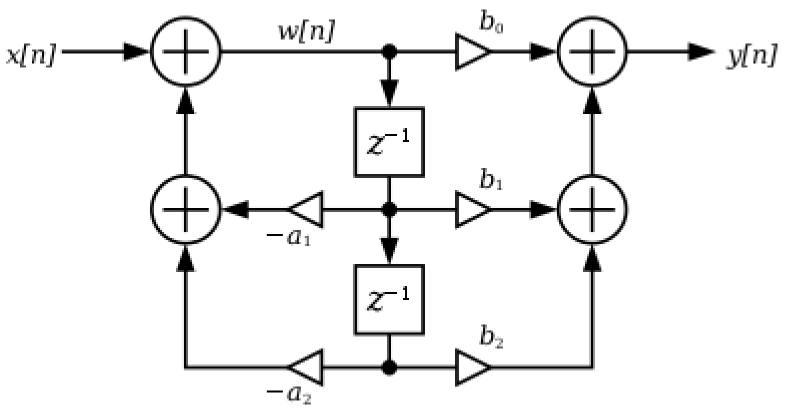
The direct form 2 implements the same normalized transfer function as direct form 1, but in two parts:

$$y[n] = b_0 w[n] + b_1 w[n-1] + b_2 w[n-2],$$

and using the difference equation:

$$w[n] = x[n] - a_1 w[n-1] - a_2 w[n-2].$$

Flow graph of biquad filter in direct form 2:



The direct form 2 implementation only needs *N* delay units, where *N* is the order of the filter – potentially half as much as direct form 1. This structure is obtained by reversing the order of the numerator and denominator sections of direct Form 1, since they are in fact two linear systems, and the commutativity property applies. Then, one will notice that there are two columns of delays (z^{-1}) that tap off the center net, and these can be combined since they are redundant, yielding the implementation as shown.

The disadvantage is that direct form 2 increases the possibility of arithmetic overflow for filters of high Q or resonance.[1] It has been shown that as Q increases, the round-off noise of both direct form topologies increases without bounds.[2] This is because, conceptually, the signal is first passed through an all-pole filter (which normally boosts gain at the resonant frequencies) before the result of that is saturated, then passed through an all-zero filter (which often attenuates much of what the all-pole half amplifies).

The direct form 2 implementation is called the canonical form, because it uses the minimal amount of delays, adders and multipliers, yielding in the same transfer function as the direct form 1 implementation.

Transposed direct forms

Each of the two direct forms may be transposed by reversing the flow graph without altering the transfer function. Branch points are changed to summers and summers are changed to branch points.[3] These provide modified implementations that accomplish the same transfer function which can be mathematically significant in a real-world implementation where precision may be lost in state storage.

The difference equations for transposed direct form 2 are:

$$y[n] = b_0 x[n] + s_1[n-1],$$

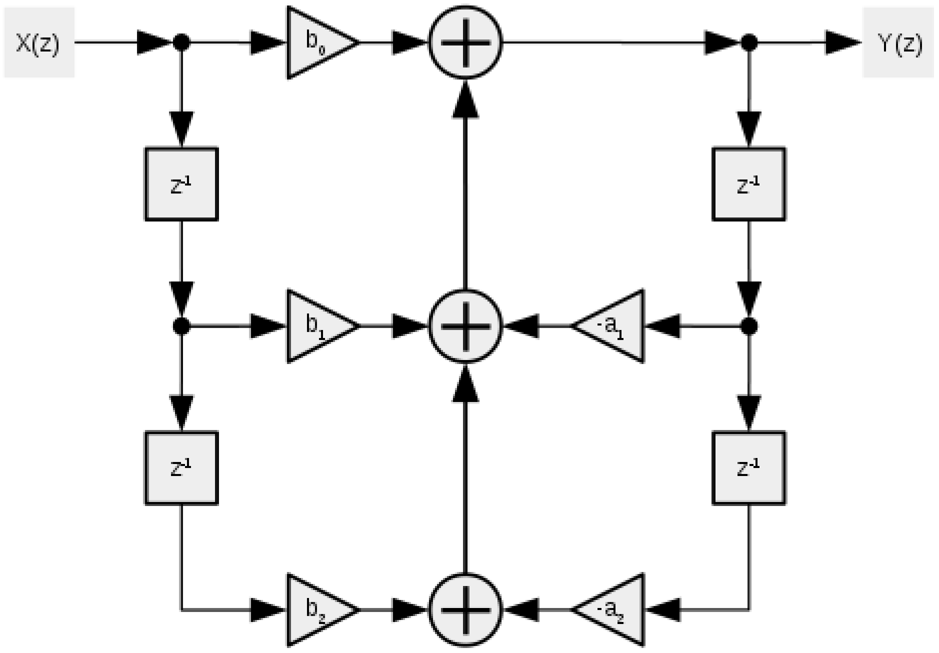
where

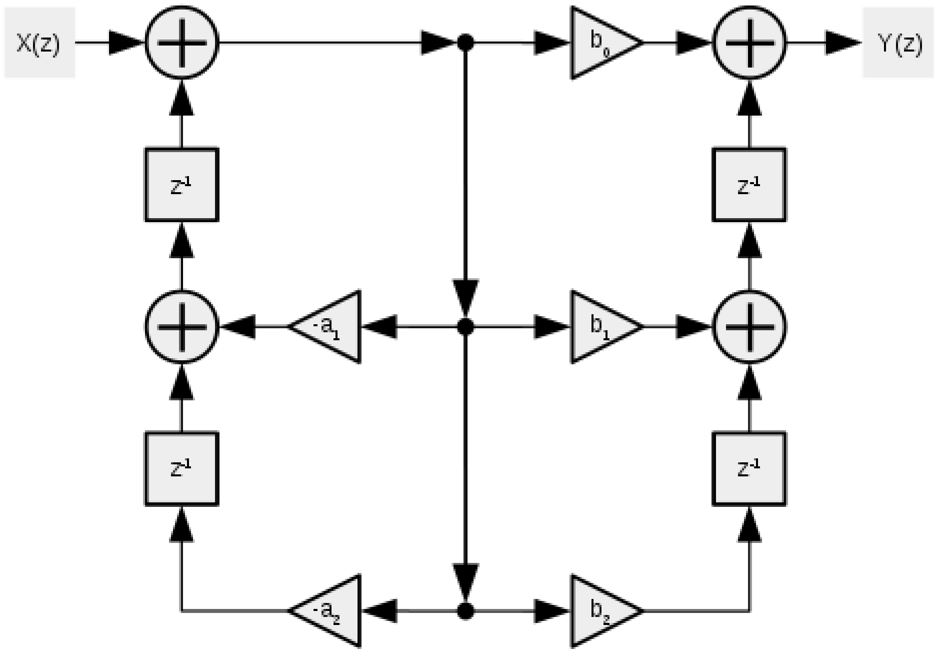
$$s_1[n] = s_2[n-1] + b_1 x[n] - a_1 y[n]$$

and

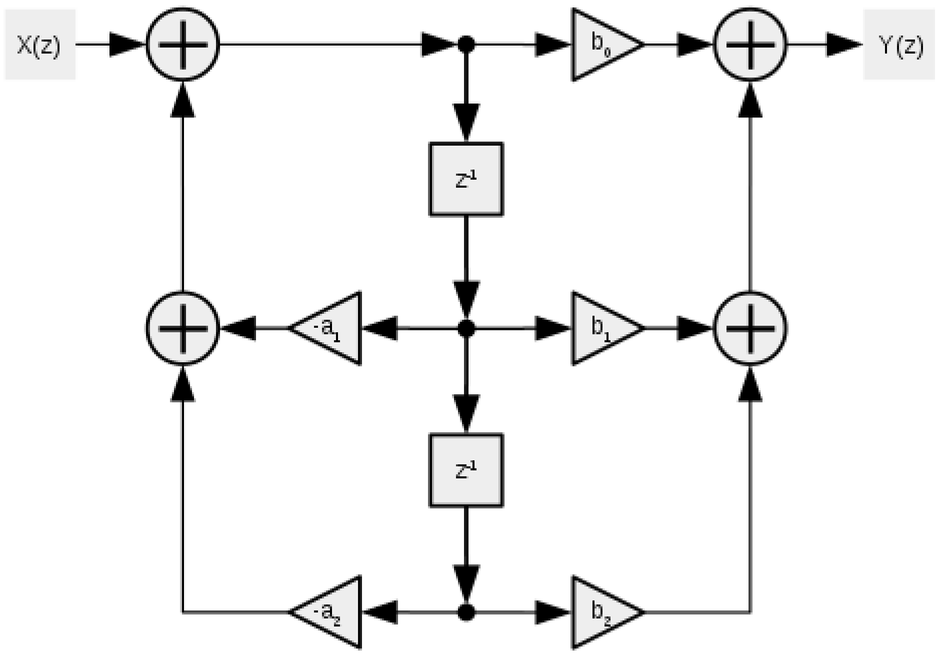
$$s_2[n] = b_2 x[n] - a_2 y[n].$$

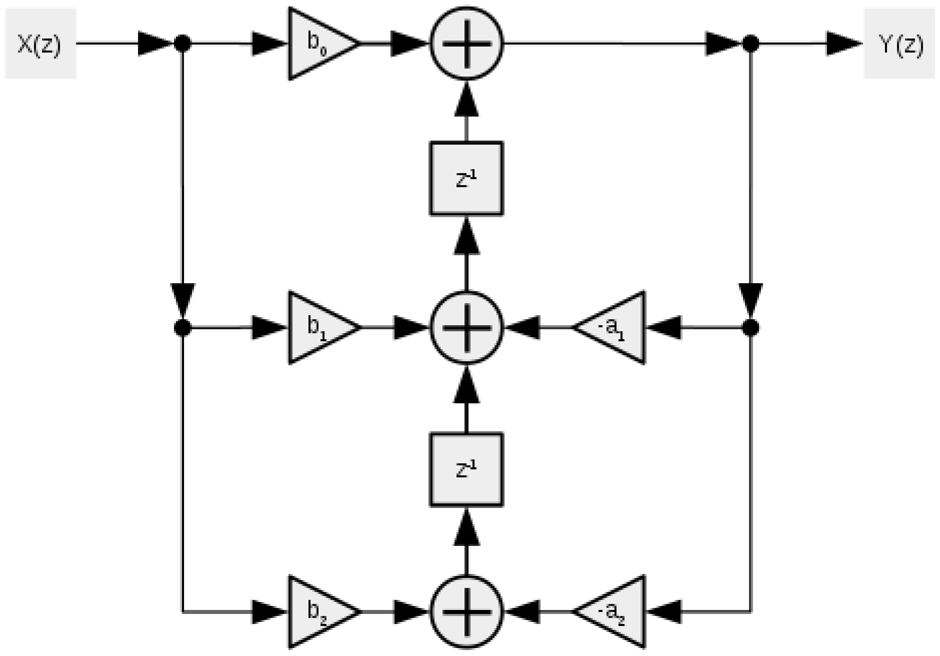
Transposed direct form 1

The direct form 1  is

transposed into 

Transposed direct form 2

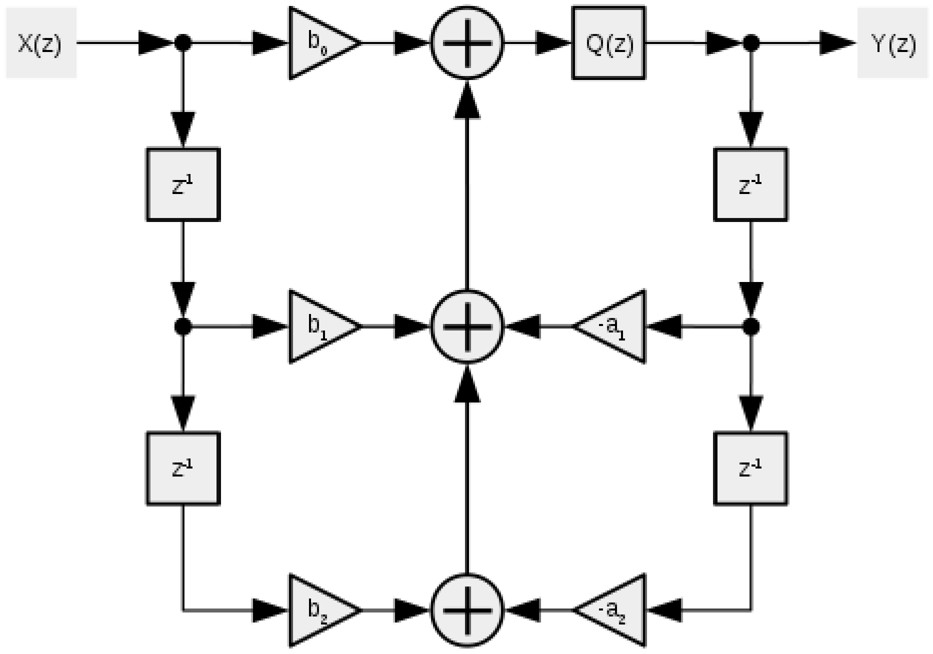
The direct form 2  is

transposed into 

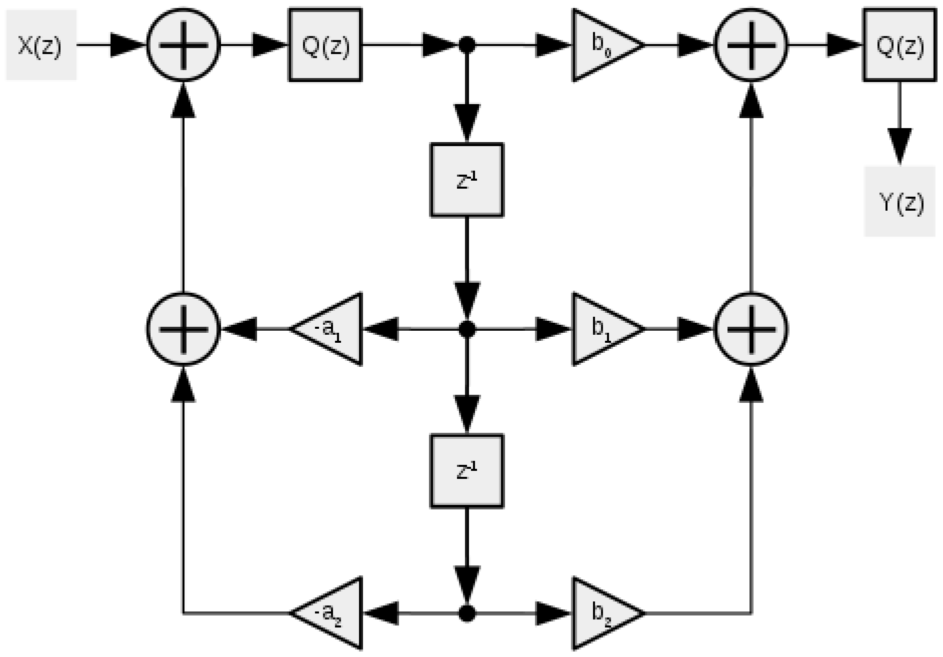
Quantizing noise

When a sample of n bits is multiplied by a coefficient of m bits, the product has n+m bits. These products are typically accumulated in a DSP register, the addition of five products may need 3 overflow bits; this register is often large enough to hold n+m+3 bits. The z^{-1} is implemented by storing a value for one sample time; this storage register is usually n bits, the accumulator register is rounded to fit n bits, and this introduced quantizing noise.

In the direct form 1 arrangement, there is a single quantizing/rounding function Q(z):



In the direct form 2 arrangement, there also is a quantizing/rounding function for an intermediate value. In a cascade, the value may not need rounding between stages, but the final output may need rounding.



Fixed point DSP usually prefers the non transposed forms and has an accumulator with a large number of bits, and is rounded when stored in main memory. Floating point DSP usually prefers the transposed form, each multiplication and potentially each addition are rounded; the additions are higher precision result, when both operands have similar magnitude.

See also

- Biquad filter
- Digital filter

References

1. J. O. Smith III, Direct Form II (http://ccrma.stanford.edu/~jos/filters/Direct_Form_II.html)
2. L. B. Jackson, "On the Interaction of Roundoff Noise and Dynamic Range in Digital Filters," *Bell Sys. Tech. J.*, vol. 49 (1970 Feb.), reprinted in *Digital Signal Process*, L. R. Rabiner and C. M. Rader, Eds. (IEEE Press, New York, 1972).
3. "Transposed Direct-Forms" (https://ccrma.stanford.edu/~jos/fp/Transposed_Direct_Forms.html).

External links

- Cookbook formulae for audio EQ biquad filter coefficients (http://shepazu.github.io/Audio-EQ-Cookbook/audio-eq-cookbook.html)
- Biquad filter (http://peabody.sapp.org/class/350.838/lab/biquad/)
- JOS BiQuad section (http://ccrma.stanford.edu/~jos/filters/BiQuad_Section.html)
- https://ccrma.stanford.edu/~jos/fp/Transposed_Direct_Forms.html
- WikiBook on digital signal processing (https://en.wikibooks.org/wiki/Digital_Signal_Processing/IIR_Filter_Design#Chain_of_Second_Order_Sections)
- Matched Second Order Digital Filters (https://www.vicanek.de/articles/BiquadFits.pdf)