Implementing Recursive Filters with Large Ratio of Sample Rate to Bandwidth

fred harris
San Diego State University
fred.harris@sdsu.edu

Wade Lowdermilk
BAE Systems
wade.lowdermilk@baesystems.com

ABSTRACT

The poles of a recursive filter shift their position when the polynomial coefficients are quantized and represented with fixed bit width approximations. This sensitivity is quite severe for high-order low-bandwidth filters. At best the root shift may cause significant deviation in spectral response, and at worst is responsible for instability in many filter designs. Narrowband filters also exhibit large numerical gain which lead to extended bit width internal registers and extended width multipliers. We address techniques to implement high order very low-bandwidth recursive lowpass filters without the brute force requirement for extended precision coefficients and registers.

1. INTRODUCTION

A compact description of a recursive digital filter is the list of its poles and zeros, the denominator and numerator roots which for insight are often presented graphically as in figure 1. An equivalent description is the denominator and numerator polynomials forms by expanding the factored form as shown in eq-1. Without quantization, the two denominators of eq-1 are equivalent. With coefficient quantization, the coefficients (a_m) are replaced with approximate coefficients ($a_m+\Delta a_m$), which causes the roots to move from (p_m) to ($p_m+\Delta p_m$).

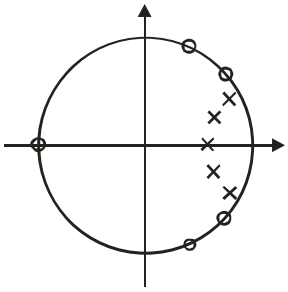

Figure 1. Pole-Zero Diagram of 5-th Order IIR Filter

$$H(Z) = \frac{\prod_{k=1}^{N}(Z - z_k)}{\prod_{k=1}^{N}(Z - p_k)} = \frac{\sum_{n=1}^{N} b_n Z^{N-n}}{\sum_{n=1}^{N} a_n Z^{N-n}} \qquad (1)$$

Traditional sensitivity analysis tells us how much change Δp_m we expect the root p_m to exhibit due to the change Δa_m of the coefficient a_m. Equation 2 shows the sensitivity coefficient and eq-3 reminds us the total shift is the sum of the shifts due to each coefficient change.

$$S_{a_k}^{p_m} \doteq \frac{\Delta p_m}{\Delta a_k} = -\frac{(p_m)^k}{\prod_{\substack{n=1 \\ n\neq m}}^{N}(p_m - p_n)} \qquad (2)$$

$$\Delta p_k = \sum_{n=1}^{N} \Delta a_n S_{a_n}^{p_k} \qquad (3)$$

To first order, the m-th root moves the reciprocal of the denominator of eq-2 which is seen to be the product of the distances between the m-th root and the remaining roots of the polynomial. Thus if we have 5 roots, and the distance from a selected root to each of his four companions is on the order of 0.1, the expected root shift is on the order of 10,000 times the change in the coefficient. It is for this reason we avoid designs with multiple roots in the same polynomial. When we unpack the polynomial to form a cascade of first and second order filters as shown in figure 3, we obtain significantly lower sensitivities. A sensitivity of 10 is manageable for two roots of a second order polynomial separated by 0.1.

The second implementation consideration related to the number of roots in a single IIR filter is the processing gain or bit growth between input and internal registers. Digital filters grow their pass band, in contrast to analog filters which attenuate their stop band. This growth, called processing gain or numerical gain, is proportional to the ratio of sample rate to filter bandwidth. For FIR filters the proportionality factor is 1, while for IIR filters this factor is 0.37 times the bandwidth ratio all raised to the number of poles in the filter. As an example, while a FIR filter with normalized two-sided bandwidth 0f 0.01 has a peak gain of 100 (40 dB), Butterworth IIR filters of the same bandwidth with 1-through-4 poles exhibit gains of approximately 32 (30 dB), 1000 (60 dB) and 32,000 (90 dB) and 1,000,000 (120 dB) respectively. This is illustrated in the unscaled frequency responses of figure 2. To accommodate these gains, the IIR filter must use extended precision state registers and extended precision multipliers. This gain must be controlled in a fixed point machine.

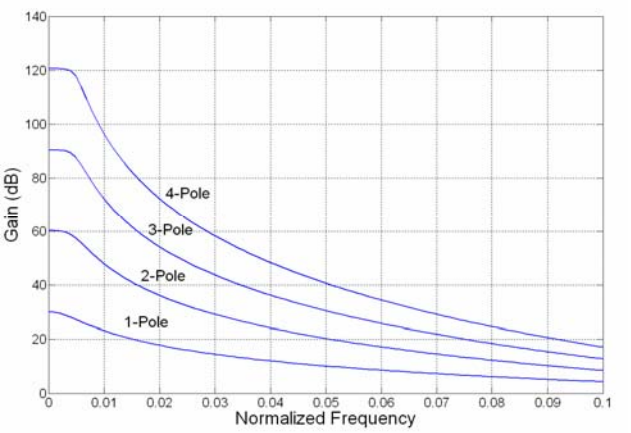

Figure 2. Recursive Gain to Internal State for Butterworth Filters of Orders 1 through 4 for Double sided BW = 0.01

The gain of an IIR low-pass filter, shown in equation 4, is seen to be the inverse of the product of the distance from each pole in the polynomial to the Z=1 test point on the unit circle.

$$H(0)\big|_{Poles} = \left.\frac{1}{\prod_{k=1}^{N}(Z - p_k)}\right|_{Z=1} = \frac{1}{\prod_{k=1}^{N}(1 - p_k)} \tag{4}$$

To control the undesired attributes of an IIR filter, coefficient sensitivity and processing gain, we unpack the denominator polynomial and implement IIR filters as a cascade of first and second order sub filters with gain scaling between stages. This unpacking is shown in figure 3.

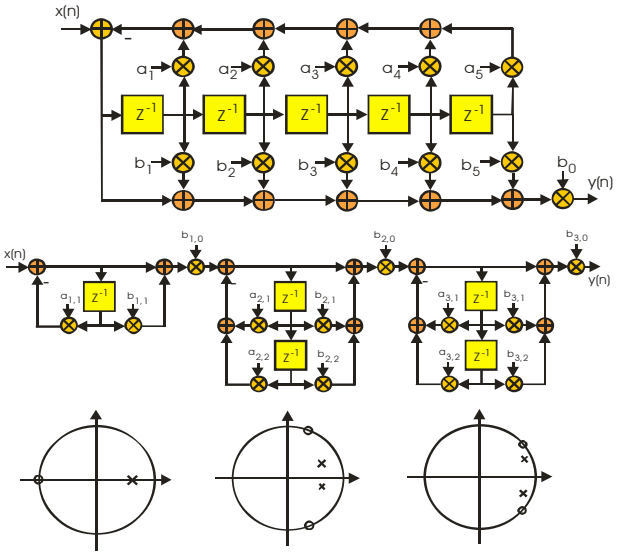

Figure 3. Unpacking A Single Stage 5-th Order Filter into a Cascade of First and Second Order Filters.

We now examine the sensitivity of root locations of a second order polynomial to quantization of its two coefficients. In eq-5 we show the polynomial formed from its factored form with roots x ± jy.

$$\begin{aligned} p(z) &= (z - x - jy)(z - x + jy) \\ &= z^2 + (-2x)z + (x^2 + y^2) \end{aligned} \tag{5}$$

Equation 6 shows us the standard representation of a second order polynomial.

$$p(z) = z^2 + a_1 z + a_2 \tag{6}$$

Equating corresponding terms from eq-5 and eq-6 we determine how the roots of the polynomial are related to its coefficients. This relationship is shown in eq-7.

$$\begin{aligned} &\text{Real part root:} \quad x = -\frac{a_1}{2} \\ &\text{Distance from origin: } R = \sqrt{x^2 + y^2} = \sqrt{a_2} \end{aligned} \tag{7}$$

The relationships shown in eq-7 suggest a graphical method to determine the root locations directly from the coefficients and offers insight into the coefficient sensitivity. Referring to figure 4, we see that the roots lie at the intersection of the line $x = -a_1/2$ and the arc of radius $\sqrt{a_2}$. We can see that when the roots are very close, the distance $-a_1/2$, and the radii $\sqrt{a_2}$ are nearly the same size and that the arc and the line are almost parallel where they meet. We also know that parallel lines don't meet hence the intersection of the line and arc for low bandwidth filters will be difficult to control when the coefficients a_1 and a_2 are quantized. This is demonstrated in figure 5 which identifies the possible root locations due to quantized line-arc intersections for 8-bit a_1 and a_2 coefficients. Notice the sparseness of roots in the region near Z=1, corresponding to low bandwidth low pass filters.

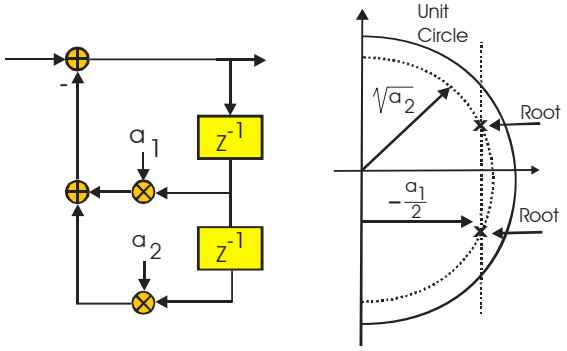

Figure 4. Locus of Second-Order Roots from Polynomial Coefficients

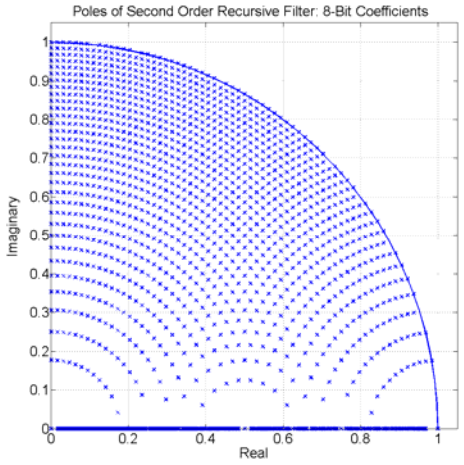

Figure 5. Possible Roots of Second Order Polynomial with 8-bit Quantized Real Coefficients.

One approach to raising the density of pole positions near Z=1 is to limit filters to a cascade of first order polynomials. This requires the use of complex coefficients to realize complex roots. The structure of the single complex pole filter, recognized as the normal filter, is shown in figure 6.

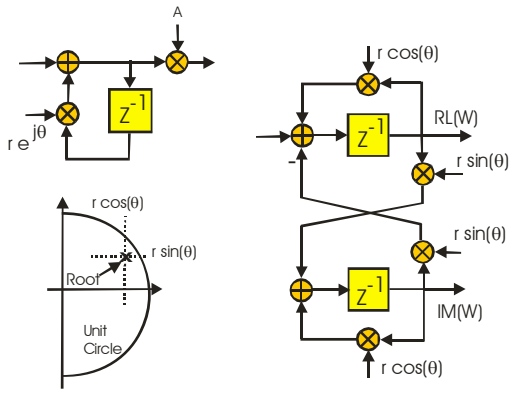

Figure 6. Locus of First-Order Roots from Complex Polynomial Coefficient

The root location for a first order polynomial is the negative of the single coefficient. This coefficient has quantized values of the real and imaginary components and the quantized component identifies the location of quantized roots. Since the real and imaginary components lie on orthogonal Cartesian coordinates, the quantized root grid coincides with the same grid. This is demonstrated in figure 7 which identifies the possible root locations due to quantized x and y components for 7-bit Real and Imaginary coefficient components. Notice the uniformity of root distribution enable filters in the region corresponding to low bandwidth low pass filters. The penalty we pay for using the complex coefficient filter is that it takes 4-multiplies to form the single pole. Incidentally, the conjugate pole does not have to be formed in a second filter.

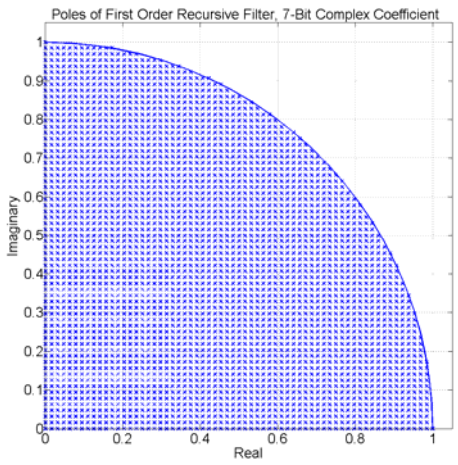

Figure 7. Possible Roots of First Order Polynomial with 7-bit Quantized Complex Coefficient.

We obtain the real output sequence as the scaled imaginary component, via its residue, of the single pole filter.

2. IMPROVED ROOT LOCATION DENSITY

It is possible to move roots into the vacant region of figure 5, near DC, without requiring extra bits or a different architecture which would require more multiplies per sample. The clue of how this is done can be seen by examining figure 8.

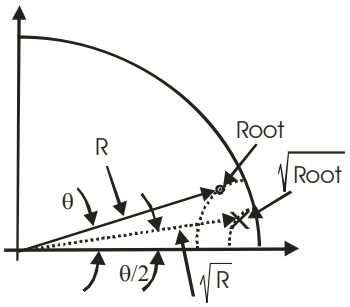

Figure 8. Root in Region Accessible with bit Width and Square-Root of Same Root.

What we see here is a root on the contour boundary of figure 5 denoted as R exp(jθ) and its square root $\sqrt{R}$ exp(jθ/2). From equation 8 we see the square root of the initial root is half the distance to the circle with half the angle from the x-axis. To first order, the square root operator forms a root half way to the Z=1 point. We note a second square root would generate image roots half way again to the Z=1 point.

$$\begin{aligned}\sqrt{R\,e^{j\theta}} &= \sqrt{R}\,e^{j\theta/2} \\ &= \sqrt{1-\Delta R}\,e^{j\theta/2} \\ &\cong (1-\Delta R/2)e^{j\theta/2}\end{aligned} \qquad (8)$$

Figure 9 illustrates the root images under the square root operation and figure 10 shows the root images under the 4-th root operation. Compare these figures to figure 5 and note the migration of the roots onto the Z=1 neighborhood.

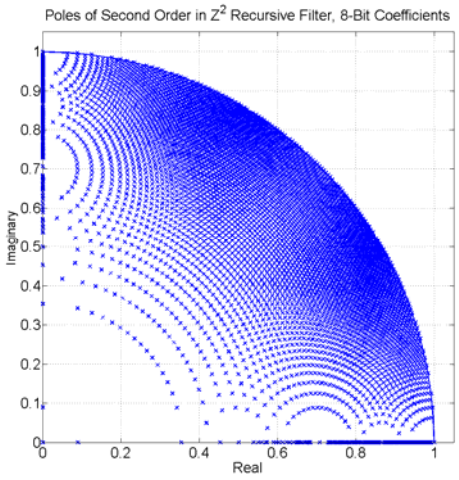

Figure 9. Root Locations from (Roots) $^{1/2}$ of Second Order Polynomial with 8-bit Quantized Real Coefficients.

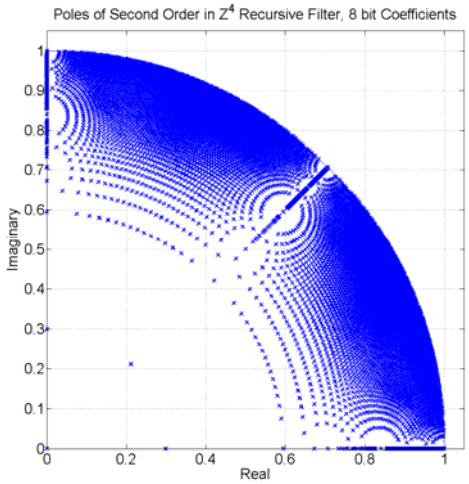

Figure 10. Root Locations from $(\text{Roots})^{1/4}$ of Second Order Polynomial with 8-bit Quantized Real Coefficients.

We now know that the m-th root of the roots formed by a second order polynomial will give us access to roots near Z=1. The question then is; how do we form these roots? We accomplish this by zero-packing the impulse response of the prototype filter. This is akin to the zero-packing we associate with the iterated FIR (IFIR) filter process. The zero packing is implemented trivially by forming polynomials in Z^M, that is, replace each delay in the IIR filter with M delays. We know for instance that if we zero pack a time sequence for a filter 1–to-4 we obtain 4-replica spectra when traversing the unit circle. Similarly, a 1-to-M zero packing will result in M-fold replication of the prototype spectra. The M-fold spectral replication means there is an M-fold bandwidth reduction so, in anticipation, we design the prototype for M-times the desired bandwidth. The filter of course has an M-fold replica of the desired spectral response. A simple cascade of boxcar integrators (or CIC's) will suppress the undesired replicas.

3. EXAMPLE

As a specific example let us examine a 5-th order elliptic filter with two sided bandwidth 0.01 implemented with 12-bit coefficients. We used the standard MATLAB elliptic filter design call to obtain a scaled coefficient set. The roots of the prototype and its time and spectral response are shown in figures 11 and 12 respectively. We first examine the roots of the prototype filter.

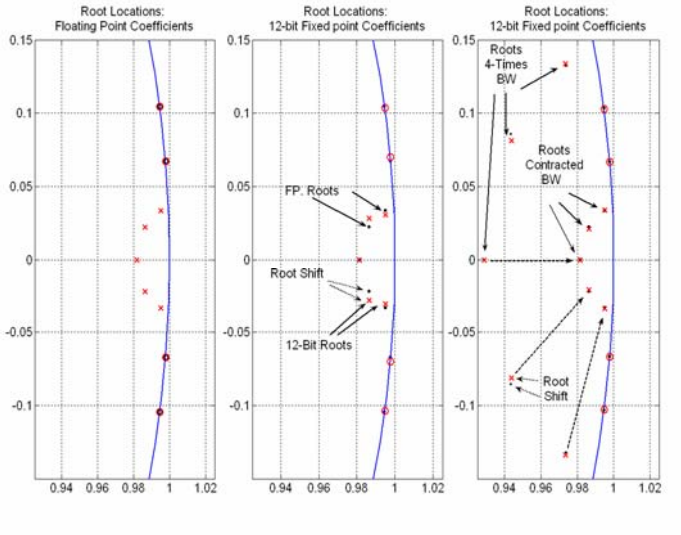

Figure 11. Roots of Prototype Filter with Floating Point coefficients, with 12-Bit Coefficients, and the 4-Times BW Filter with 12-Bit Coefficients and the 1-to-4 Zero Packed Filter

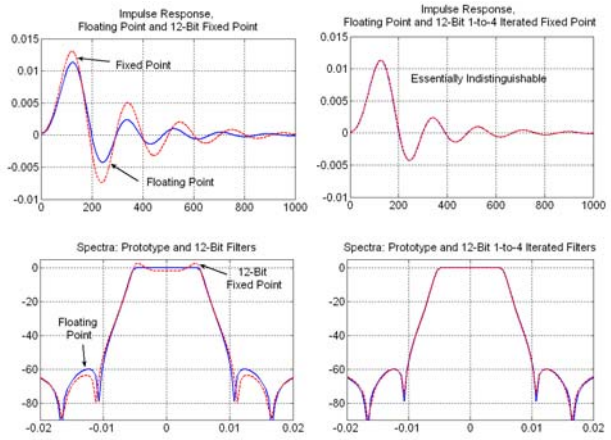

Figure 12. Impulse and Frequency Response of Prototype Filter with Floating Point coefficients and with 12-Bit Coefficients, and then of 4-Times BW 1-to-4 Zero-Packed Filter with 12-Bit Coefficients and Cascade CIC Filter

The left subplot in figure 11 presents the prototype roots and the center subplot presents the shifted roots due to the 12-bit representation of the filter. We note that the complex root pairs of the filter have shifted. The effect of this shift is seen in the time domain and in frequency domain shown in figure 12. The shifted poles have caused a 3-dB resonance peak at the band edge which is seen as a decreased damping factor damped sinusoid impulse response. The right subplot of figure 11 shows the pole positions of the 4-times bandwidth filter as well as the shifted positions due to operating in the 1-to-4 zero packed mode. Also shown on this plot is the nominal pole-zero positions prior to the quantization. The pole and zero shifts due to quantization are seen to be significantly smaller for this operation mode.

A comparison of the filter spectra for the prototype filter and for the zero packed filter is seen in figure 13. Here we see that the spectra in the neighborhood of the pass-band are essentially indistinguishable. The effect of the additional processing to suppress the spectral replicates is also seen here.

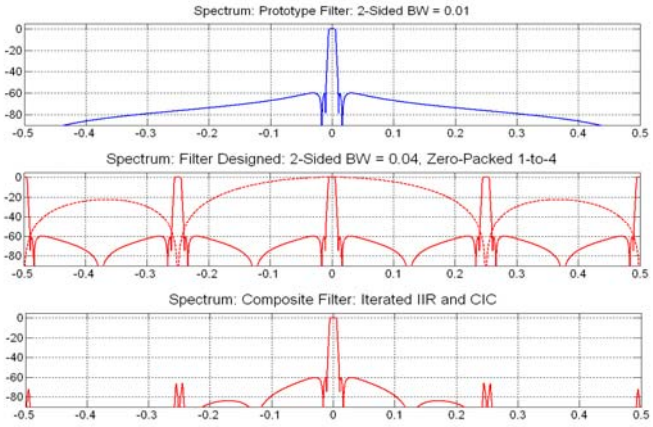

Figure 13. Spectrum of Prototype Filter, of the 4-Times BW Filter 1-to-4 Zero Packed, and the CIC-Filtered Composite Spectrum

4. REVIEW AND CONCLUSIONS

We have reviewed the source of difficulty we encounter when operating a recursive filter at very high ratios of sample rate to bandwidth. These are primarily related to the affects of finite arithmetic on sensitivity of pole positions to coefficient quantization. This sensitivity is compounded by the presence of multiple roots in the same filter structure. Good design practices have us decouple roots by placing them in different polynomials implemented in distinct subfilters. Due to similar considerations, the numerical gain of a filter is proportional to the ratio of sample rate to filter bandwidth. This gain had to be scaled out of the filter and good design strategies dictate that the scaling be distributed over multiple small filters to avoid having to deal with very wide words in the processing stream.

We then reviewed the finite arithmetic effects of the viable building blocks of recursive filters, first and second order sub-filters. We reminded the reader that the interaction between the two coefficients of a second order polynomial leads to an ill conditioned coupling when the recursive filter is designed for low bandwidth near zero frequency. The conditioning is worse for closely spaced poles, as required for low bandwidth filters near DC, and the conditioning is best for widely spaced poles, as encountered for filters centered at the quarter sample rate. The traditional method of accessing pole positions near DC is to raise the number of bits representing the filter coefficients. Alternatively we try to avoid the bad operating condition by good design practice and elect not to implement high Q digital filters. The modern approach to this avoidance involves reducing the sample rate by multirate signal processing techniques, conducting the desired filtering task at a reduced sample rate, and finally returning the sample rate to its original by a second multirate filter.

In the event the filter must be designed to operate with large ratios of sample rate to bandwidth we offer an option besides brute force to access pole positions in the normally vacant neighborhood about the Z=1 point. The process involves zero packing a recursive filter in a manor similar to the iterated designs used in FIR filter implementations. In the FIR filter case the intent of zero packing is to reduce the number of computations. It the IIR case, the intent of zero packing is to reduce the bit width required in the recursion. The M-th roots of the nominal pole positions accessible to second order polynomials enlarges the available region of accessible roots.

5. REFERENCES

[1] fred harris and Benjamin Egg, “Forming Narrowband Filters at a Fixed Sample Rate with Polyphase Down and UP Sampling Filters”, DSP-2007, Cardiff, Great Britain, 1-4 July 2007

[2] fred harris and Bruce Fette, “Algorithms for Arbitrary Resampling Filters”, 2003 Software Defined Radio Conference, SDR-2003, Orlando, FL, 17-19 November 2003

[3] W. Mills and Richard Roberts, “Low Roundoff Noise and Normal Realizations of Fixed Point IIR Digital Filters”, Acoust. Speech and Signal Processing, pp. 893-903, Vol 29, Issue 4, Aug. 1981.

[4] fred harris, “Novel Approach to the Design of Optimal Second Order Digital Filters, SOUTHCON/82; Orlando, FL., March 1982.

[5] fred harris, “Multirate Signal processing for Communication Systems”, Prentice-Hall 2004