Frequency Sampling Filters: The Lost Art

Richard Lyons

Besser Associates
201 San Antonio Circle, Bldg B, Suite 115
Mountain View, CA 94040
+1 650 949 3300
E-mail: r.lyons@ieee.org

1. INTRODUCTION

This paper describes a class of digital filters, called *frequency sampling filters*, used to implement linear-phase FIR filter designs. Although frequency sampling filters were developed over 35 years ago, the advent of the powerful Parks-McClellan nonrecursive FIR filter design method has driven them to near obscurity. Thus in the 1970s frequency sampling filter implementations lost favor to the point where their coverage in today's DSP classrooms and textbooks ranges from very brief to nonexistent. However frequency sampling filters remain *more computationally efficient* than Parks-McClellan-designed filters for certain applications where the desired passband width is less than roughly one fifth the sample rate. The purpose of this material is to introduce the DSP practitioner to the structure, performance, and design of frequency sampling filters.

Frequency sampling filters were founded upon the fact that a traditional N-tap nonrecursive (direct convolution) lowpass FIR filter in Figure 1(a) can be implemented as a comb filter in cascade with a bank of resonators as shown in Figure 1(b) with the resonator details given in Figure 1(c). the structure in Figure 1(b) is our frequency sampling filter (FSF). The equivalence of the FSF to the nonrecursive FIR filter is provided in Ref. [1]. That equivalency is such that the discrete Fourier transform of the nonrecursive FIR filter's $h(k)$ time-domain coefficients, where index $k = 0,1,...,N\text{-}1$, is represented by $|H(k)|e^{j\phi(k)}$, where the $|H(k)|$ and $\phi(k)$ values are coefficients actually used in the FSF implementation. Using the notation of Reference [1], we call the filter structure in Figure 1(b) a *Type-I* FSF because we'll explore another FSF structure later in this paper.

The basis of FSF design is for the designer (you) to define a desired FIR filter frequency response in the form of $H(k)$ frequency-domain samples, whose magnitudes are depicted as black dots in Figure 2(a) when, for

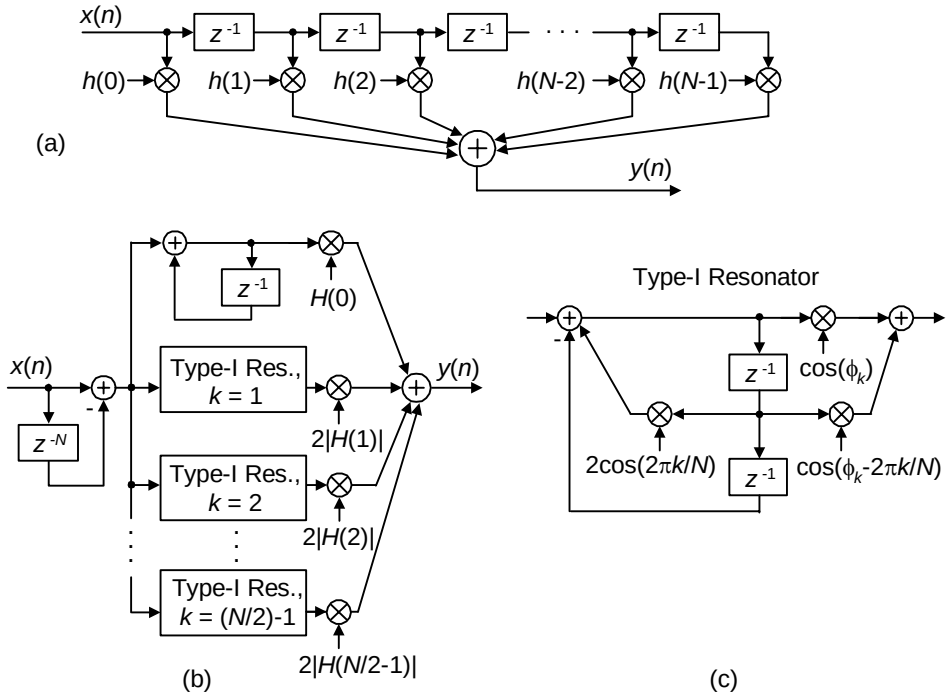

Figure 1. FIR filters: (a) N-tap nonrecursive; (b) equivalent N-section Type-I frequency sampling filter.

example, $N = 22$ with a lowpass filter. Next, those $|H(k)|$ sample values as used as gain factors following the resonators in the Type-I FSF structure (block diagram). For a linear phase FSF, the resonators' $\phi(k)$ values would be set equal to $-k\pi(N-1)/N$. Because only three $H(k)$ values are nonzero, only three resonator sections need be implemented as shown in Figure 2(b).

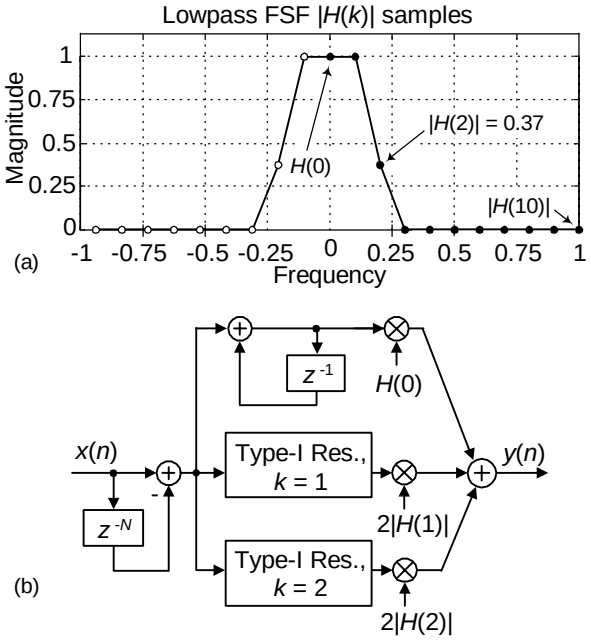

Figure 2. Desired lowpass filter mag. response for $N = 22$ (a), Type-I FSF implementation (b).

The *actual* time and frequency characteristics of the Figure 2(b) lowpass FSF are shown in Figure 3. There we see the comb filter's 22 zeros located on the *z*-plane's unit circle where the three resonators' poles cancel some of those zeros, and it this characteristic that provides the nonzero lowpass passband. The transfer function of this Type-I lowpass FSF is the messy

$$H_{\text{Type-I}}(z) = (1-z^{-N})\left[\frac{H(0)}{1-z^{-1}} + \sum_{k=1}^{N/2-1}\frac{2|H(k)|\,[\cos(\phi_k)-\cos(\phi_k-2\pi k/N)z^{-1}]}{1-[2\cos(2\pi k/N)]z^{-1}+z^{-2}}\right]$$

where ϕ_k is the desired phase angle of the kth section. The Type-I FSF requires four multiplies per resonator output sample.

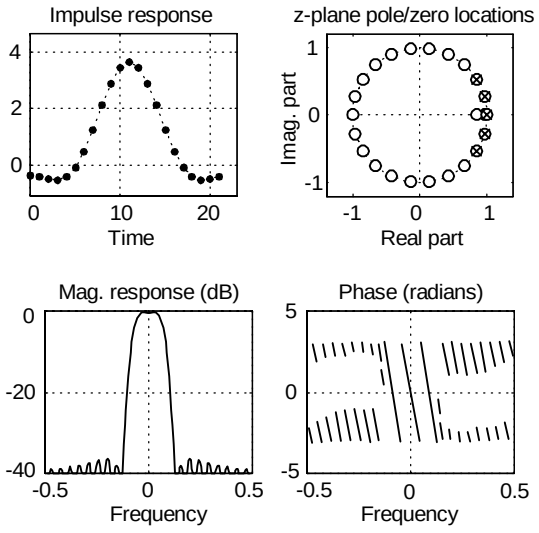

Figure 3. Type-I, lowpass, $N = 22$, FSF behavior.

The phase response in Figure 3 looks linear, but it turns out to be only *moderately* linear because there are zeros inside the unit circle which have no reciprocal zeros outside the unit circle. (That's a strict requirement for phase linearity!) Thus Type-I FSFs can have a group delay peak-peak fluctuation of up to two sample periods ($2/f_s$) when multiple resonators are used.

While the Type-I FSF is one of the most common types described in the literature of DSP, its inability to yield exact linear phase has not been emphasized. No matter though, because there's a better FSF structure on which we now focus our attention.

Before we jump to the topic of an *improved* FSF, did you notice what the FSF in Figure 2 would be if we set $H(1)$ and $H(2)$ equal to zero? That's right, such a single-stage FSF would be a *cascaded integrator-comb* (CIC) filter that's found wide appeal in hardware lowpass filtering implementations in association with time-domain decimation and interpolation.

2. AN IMPROVED FSF

There are many resonators that can be used in FSFs, but of particular interest to us is the so-called *Type-IV* resonator presented in Figure 4(a). This resonator deserves attention because it provides highly linear phase, is computationally efficient, and yields stopband attenuation performance superior to the Type-I FSF.

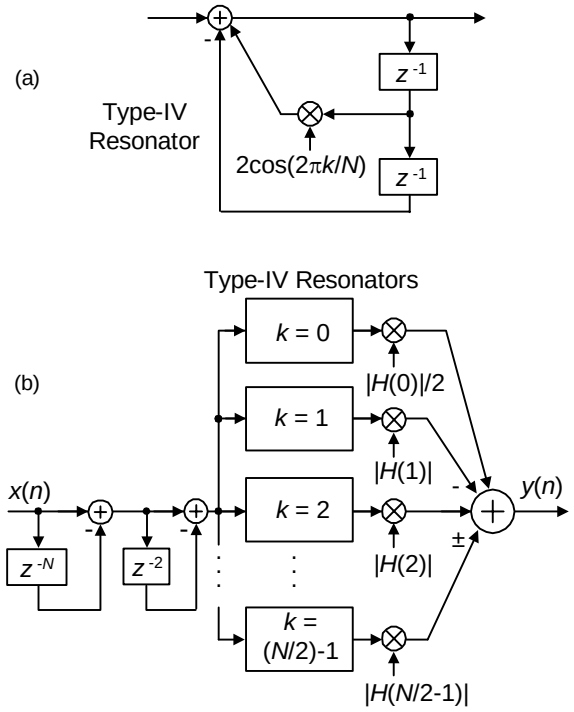

Figure 4. Type-IV resonator structure (a), Type-IV lowpass FSF implementation (b).

Cascading Type-IV resonators with a comb filter provides a Type-IV lowpass FSF, Figure 4(b), with a transfer function of

$$H_{\text{Type-IV}}(z) = (1 - z^{-N}) \sum_{k=0}^{N/2} \frac{(-1)^k |H(k)| \, (1 - z^{-2})}{1 - 2\cos(2\pi k/N) z^{-1} + z^{-2}}$$

where N is even. A surprising fact (at least to me) is that using the alternating signs at the final summation, expressed as the $(-1)^k$ factor in $H_{\text{Type-IV}}(z)$, guarantees linear phase! (The proof of this behavior can be found in Reference [1].) This means we no longer need be concerned with the phase factor ϕ_k as in $H_{\text{Type-I}}(z)$.

The time and frequency characteristics of a Type-IV lowpass FSF, implementing the desired frequency response of Figure 2(a), are shown in Figure 5. The Type-IV FSF requires only two multiplies per resonator output sample.

3. IMPROVING PERFORMANCE WITH TRANSITION BAND COEFFICIENTS

Did you notice that the $H(2)$ sample value in the desired lowpass frequency response example in Figure 2(a) was not unity? We set $H(2) = 0.39$ to reduce the passband ripple and improve the stopband attenuation of the Type_IV FSF.

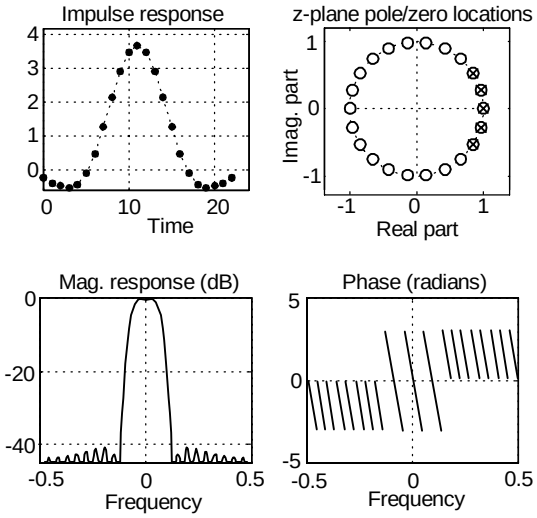

Figure 5. Type-IV, lowpass, N = 22, FSF behavior.

Had we set $H(2)$ to zero, the resultant FSF would have had a stopband attenuation of only, roughly, 17 dB. The assignment of a transition band magnitude sample, coefficient $H(2)$, to a value between 0 and 1 reduces the abruptness in the transition region between the passband and the stopband of our desired frequency magnitude response. The price we pay for the improved performance is the computational cost of an additional FSF section and increased width of the transition region.

Assigning a coefficient value of $H(2) = 0.39$ was not arbitrary nor magic. Measuring the maximum stopband sidelobe level for various values of $0 \le H(2) \le 1$, for our $N = 22$ lowpass FSF, reveals the existence of an optimum value for $H(2)$. Figure 6 shows the maximum stopband sidelobe level is minimized when $H(2) = 0.39$.

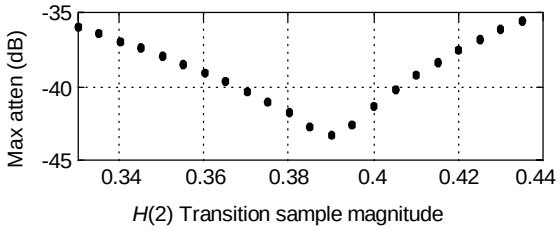

Figure 6. Maximum stopband attenuation versus $H(2)$ for a 3-section Type-IV FSF when N = 22.

This venerable and well-known method of employing a transition region coefficient to reduce passband ripple and minimize stopband sidelobe levels also applies to bandpass FSF design where the transition sample is used just before and just after the filter's passband unity-magnitude $|H(k)|$ samples.

Further stopband sidelobe level suppression is possible if two transition band coefficients, T_1 and T_2, are used such that $0 \le T_2 \le T_1 \le 1$. (Note: for lowpass filters, if T_1 is the $|H(k)|$ sample, then T_2 is the $|H(k+1)|$ sample.) Each additional transition coefficient used improves the stopband attenuation by approximately 25 dB. However, finding the optimum values for the T coefficients is a daunting task. Optimum transition region coefficient values depend on the number of unity-gain FSF sections, the value of N, and the number of coefficients used; and unfortunately there's no closed form equation available to calculate optimum transition coefficient values. We must search for them empirically. For Type-IV FSFs, tables of optimum transition coefficient values have been compiled by the author and are provided in Reference [1]. With this good news in mind, let's look at a real-world FSF design example to appreciate the benefits of FSFs.

4. A TYPE-IV FSF EXAMPLE

Think about designing of a linear-phase lowpass FIR filter with the cutoff frequency being 0.05 times the sample rate f_s, the stopband must begin at 0.095 times f_s, maximum passband ripple is 0.3 dB peak-to-peak, and the minimum stopband attenuation must be 65 dB. If we designed a six-section Type-IV FSF with $N = 62$, its frequency-domain performance satisfies our requirements and is the solid curve in Figure 7.

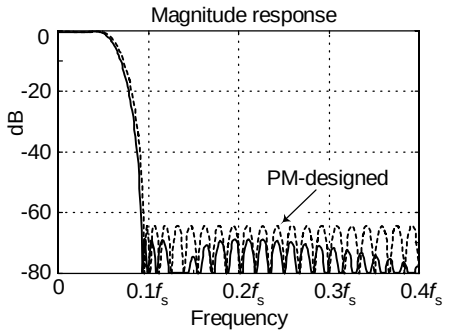

Figure 7. Magnitude response of an N = 62, six-section Type-IV real FSF [solid] versus a 60-tap PM-designed filter [dashed].

For this FSF, two transition region coefficients of $|H(4)| = T_1 = 0.5899$, and $|H(5)| = T_2 = 0.1049$ were used. Included in Figure 7, the dashed curve, is the magnitude response of a PM-designed 60-tap nonrecursive FIR filter. Both filters meet the performance requirements and have linear phase. The structure of the Type-IV FSF is provided in Figure 8. A Parks-McClellan-designed (PM-designed) nonrecursive filter implemented using a *folded* nonrecursive FIR structure, exploiting its impulse response symmetry to halve the number of multipliers, requires 30 multiplies and 59 adds per output sample. We see the computational savings of the Type-IV FSF which requires only 17 multiplies and 19 adds per output sample. (Note, the FSF's $H(k)$ gain factors are all zero for $6 \le k \le 31$.)

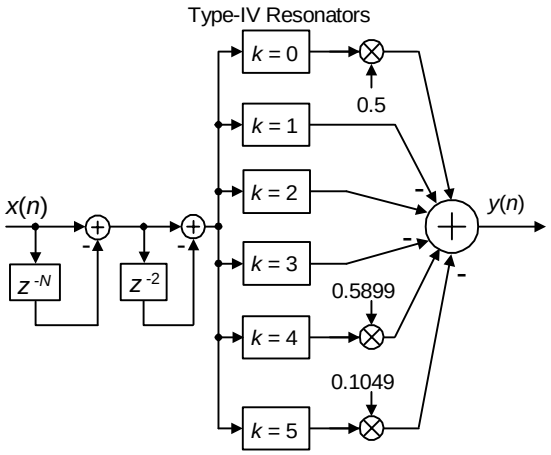

Figure 8 Six-section, $N = 62$, lowpass Type-IV FSF with two transition region coefficients.

5. WHEN TO USE AN FSF

Here we attempt to answer the burning question; when is it smarter (more computationally efficient) to use a Type-IV FSF instead of a PM-designed nonrecursive FIR filter? First, if your desired linear phase filter requires a stopband attenuation within or below one of the shaded bands in Figure 9, then an FSF *may* solve your filter design problem. For example, if your desired filter requires a stopband attenuation of 60 dB, from Figure 9 then, a one-transition coefficient FSF won't solve your problem, but a two-transition band coefficient FSF may be appropriate. Maybe, maybe not. You must now consider the passband width and transition bandwidth of your desired filter to see if an FSF is appropriate. Here's how.

The computational workload of a PM-designed FIR filter is directly related to the length of its time-domain impulse response, which in turn is inversely proportional to the filter transition bandwidth. The more narrow the transition bandwidth, the greater the nonrecursive FIR filter's computational workload measured in arithmetic operations per output sample. On the other hand, the computational workload of an FSF is roughly proportional to its passband width. The more FSF sections needed for wider bandwidths and the more transition region coefficients used, the higher the FSF's computational workload.

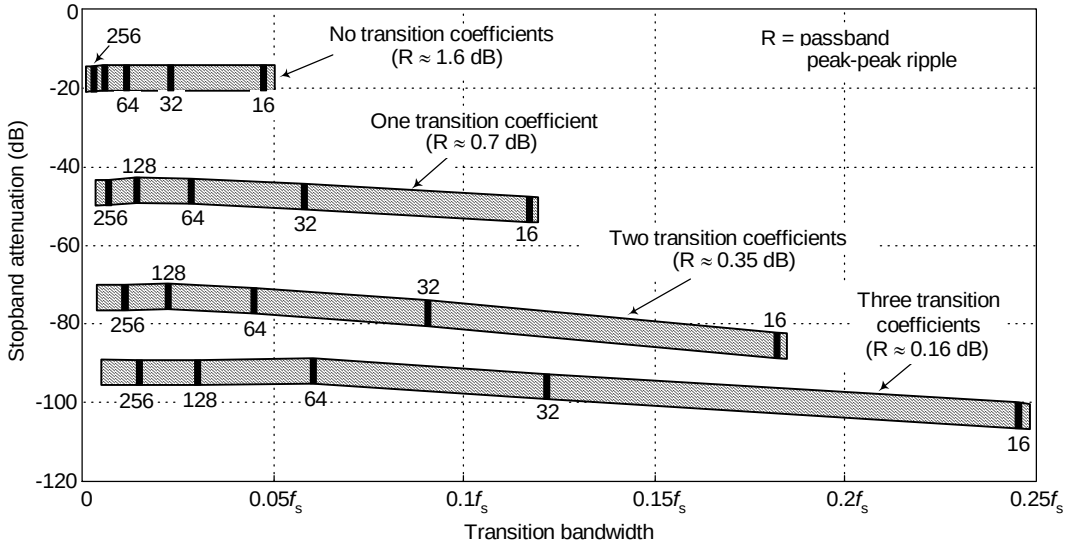

Figure 9. Typical Type-IV lowpass FSF stopband attenuation performance as a function of transition bandwidth.

So in comparing the computational workload of FSFs to PM-designed FIR filters, both transition bandwidth and passband width must be considered.

Figure 10 provides a computational comparison between even-*N* Type-IV FSFs and PM-designed nonrecursive lowpass FIR filters. The curves represent desired filter performance parameters where a Type-IV FSF and a PM designed filter have approximately equal computational workloads per output sample. (The bandwidth values in the Figure 10 axis are normalized to the sample rate, so for example a frequency value of 0.1 equals $f_s/10$.) If the desired FIR filter transition region width and passband width combination (a point) lies beneath the curves in Figure 10, then a Type-IV FSF is computationally more efficient than a PM-designed filter. Figure 10(a) is a comparison accounting only for multiply operations. For filter implementations where multiplies and adds require equal processor clock cycles, Figure 10(b) provides the appropriate comparison. The solitary black dot in the figure indicates the comparison-curves location for the above Figure 7 Type-IV FSF filter example.

The PM filter implementation used in the Figure 10 comparison assumes a symmetrical impulse response, an *M*-tap folded structure requiring *M*/2 multiplies, and *M*-1 additions. The performance criterion levied on the PM filter are typical the Type-IV FSF properties (when floating-point coefficients are used), for various numbers of transition band coefficients, given in Table 1.

Table 1 Typical even-*N* Type-IV FSF properties.

Parameter	One coefficient	Two coefficients	Three coefficients
Passband peak-peak ripple (dB)	0.7	0.35	0.16
Minimum stopband atten. (dB)	-45	-68	-95

As of this writing, programmable-DSP chips typically cannot take advantage of the folded FIR filter structure assumed in the creation of Figure 10. In this case, an *M*-tap direct convolution FIR filter has the disadvantage that it must perform the full *M* multiplies per output sample. However, DSP chips have the advantage of *zero-overhead looping* and single-clock-cycle *multiply and accumulate* (MAC) capabilities making them more efficient for direct convolution FIR filtering than for filtering using the recursive FSF structures. Thus, a DSP chip implementation's disadvantage of more necessary computations and its advantage of faster execution of those computations roughly cancel each other. With those thoughts in mind, we're safe to use Figure 10 as a guide in deciding whether to use a Type-IV FSF or a PM-designed FIR filter in a programmable DSP chip filtering application.

Finally we conclude, from the last figure, that Type-IV FSFs are more computationally efficient than PM-designed FIR filters in lowpass applications where the passband is less than approximately $f_s/5$ and the transition bandwidth is less than roughly $f_s/8$.

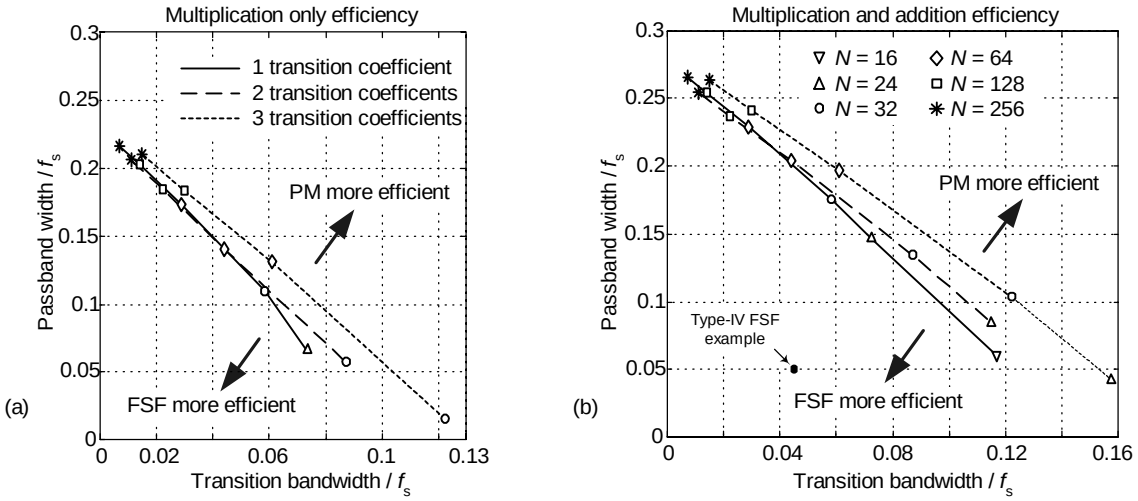

Figure 10. Computational workload comparison between even-*N* lowpass Type-IV FSFs and nonrecursive PM FIR filters: (a) only multiplications considered; (b) multiplications and additions considered.

6. CONCLUDING REMARKS

We've introduced the structure, performance, and a few design details regarding frequency sampling FIR filters. Performance curves were presented to aid the designer in choosing between a Type-IV FSF and a Parks-McClellan-designed FIR filter for a given narrowband linear-phase filtering application. We can state:

- Type-IV FSFs are more computationally efficient, for certain stopband attenuation levels, than Parks-McClellan-designed nonrecursive FIR filters in lowpass applications where the passband is less than $f_s/5$ and the transition bandwidth is less than $f_s/8$,
- FSFs are modular; their components (sections) are computationally identical and well understood,
- tables of optimum transition region coefficients, used to improve Type-IV FSF performance, are available; and
- although FSFs use recursive structures, they can be designed to be guaranteed stable, and have linear phase.

Due to space constraints, this discussion of FSFs was necessarily *very* brief. Reference [1] provides far more details concerning mathematical derivations describing various complex and real FSF structures, phase linearity performance curves, frequency response equations, methods for ensuring FSF stability (while this paper described FSFs having poles on the *z*-plane's unit, there are ways to guarantee filter stability), FSF gain scaling, tips on modeling FSFs, details for designing and implementing frequency sampling filters, and references for further reading.

7. REFERENCES

[1] R. Lyons, "*Understanding Digital Signal Processing*", 2nd Ed., Prentice Hall, Upper Saddle River, New Jersey, 2004, Chapter 7.