Richard Lyons

Interpolated Narrowband Lowpass FIR Filters

This article describes a class of digital filters, called *interpolated finite impulse response (FIR) filters*, that can implement narrowband lowpass FIR filter designs with a significantly reduced computational workload relative to traditional FIR filters. Traditional N-tap direct-convolution FIR filters have been on the DSP scene since the early 1970s and have gained acceptance in a wide variety of applications. Their ability to exhibit guaranteed-stable linear-phase behavior, along with readily available filter design software [1], has made these filters the topic of more technical papers than any other digital filter type. As a result of all this study and analysis, several schemes have been developed to reduce the computational complexity of these FIR filters. In this article, we explore one of those schemes, the interpolated FIR filter, and show how the computational workload of traditional narrowband lowpass FIR filters can be reduced by more than 80% [2], [3]. The interpolated FIR filter is introduced through an example, parameter selection is discussed, and filter performance curves will be presented and used in a lowpass filter design example showing computational savings results.

Interpolated FIR (IFIR) filters are based upon the behavior of an N-tap nonrecursive linear-phase FIR filter when each of its unit delays are replaced with M-unit delays, with the expansion factor M being an integer, as shown in Figure 1(a). If the $h_{\mathrm{pr}}(k)$ impulse response of a nine-tap FIR filter is that shown in Figure 1(b), the impulse response of an expanded FIR filter, where, for example, $M = 3$, is the $h_{\mathrm{be}}(k)$ in Figure 1(c). The variable k is merely an integer time-domain index where $0 \le k \le N - 1$. To establish our terminology, we'll call the original FIR filter the prototype filter and introduce the filter with expanded delays as the band-edge shaping subfilter. Shortly we'll see why this nomenclature is appropriate.

We can express a prototype FIR filter's z-domain transfer function as

$$H_{\mathrm{pr}}(z) = \sum_{k=0}^{N-1} h_{\mathrm{pr}}(k) z^{-k} \tag{1}$$

where N is the length of h_{pr}. The transfer function of a general band edge shaping FIR filter, with z in (1) replaced with z^{M}, is

$$H_{\mathrm{be}}(z) = \sum_{k=0}^{N-1} h_{\mathrm{pr}}(k) z^{-kM}. \tag{2}$$

If the unit impulse response length (number of taps) of the prototype filter is N_{pr}, the band-edge shaping filter has N_{pr} nonzero taps and an expanded impulse response length of

$$L_{\mathrm{be}} = M(N_{\mathrm{pr}} - 1) + 1. \tag{3}$$

Later we'll see how L_{be} has an important effect on the implementation of IFIR filters.

The frequency-domain effect of those M-unit delays is shown in Figure 2 where the frequency axis is measured in hertz. As we should expect, an M-fold expansion of the time-domain filter impulse response causes an M-fold compression (and repetition) of the frequency-domain $|H_{\mathrm{pr}}(f)|$ magnitude response as in Figure 2(b). The frequency axis of these curves is normalized to f_s with f_s being the signal sample rate in samples/second. For example, the normalized frequency f_{pass} is equivalent to a cyclic frequency of $f_{\mathrm{pass}} \cdot f_s$ Hz. Those repetitive passbands in $|H_{\mathrm{be}}(f)|$ centered about integer multiples of $1/M$ (f_s/M Hz) are called images, and we now focus our attention on them.

If we follow the band-edge shaping subfilter with a lowpass *masking* subfilter [Figure 2(c)] whose task is to attenuate the image passbands, we can realize a multistage filter whose

This new column, titled "DSP Tips and Tricks," is an attempt by the new editorial team to introduce practical tips and tricks of design and implementation of signal processing algorithms so that you may be able to incorporate them into your designs. We welcome readers who enjoy reading this column to submit their contributions. Please contact Associate Editor Rick Lyons at ricklyon@onemain.com.

1053-5888/03/$17.00©2003IEEE

frequency response is shown in Figure 2(d). The resultant $|H_{\text{ifir}}(f)|$ frequency magnitude response is

$$|H_{\text{ifir}}(f)| = |H_{\text{be}}(f)| \cdot |H_{\text{ma}}(f)|. \qquad (4)$$

The structure of the cascaded subfilters is the so-called IFIR filter shown in Figure 3.

If a desired lowpass filter's passband width is f_{pass}, its stopband begins a f_{stop}, and the transition region width is $f_{\text{trans}} = f_{\text{stop}} - f_{\text{pass}}$, then the prototype subfilter's normalized frequency parameters are defined as

$$f_{\text{pr-pass}} = Mf_{\text{pass}} \qquad (5a)$$

$$f_{\text{pr-stop}} = Mf_{\text{stop}} \qquad (5b)$$

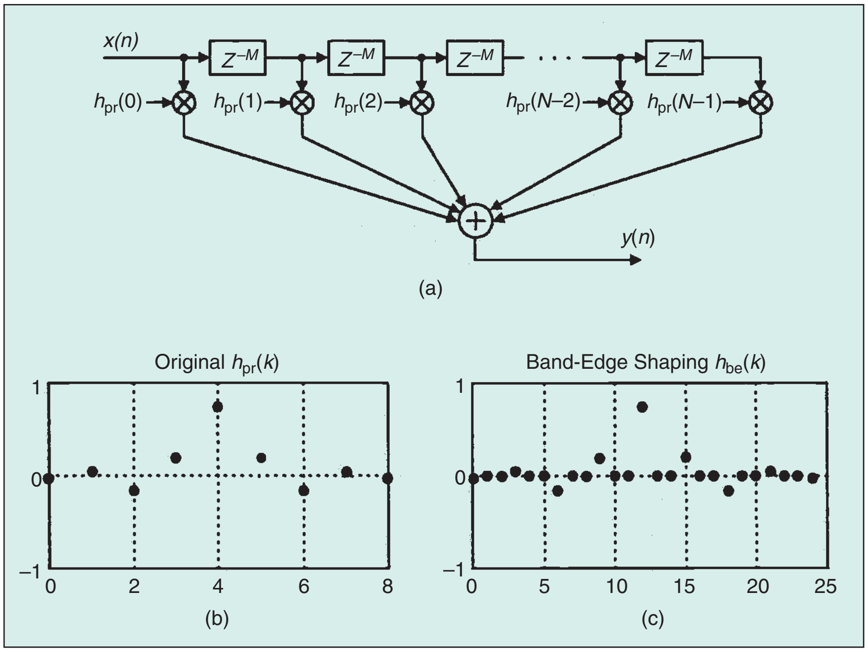
$$f_{\text{pr-trans}} = Mf_{\text{trans}} = M\left(f_{\text{stop}} - f_{\text{pass}}\right). \qquad (5c)$$

The masking subfilter's frequency parameters are

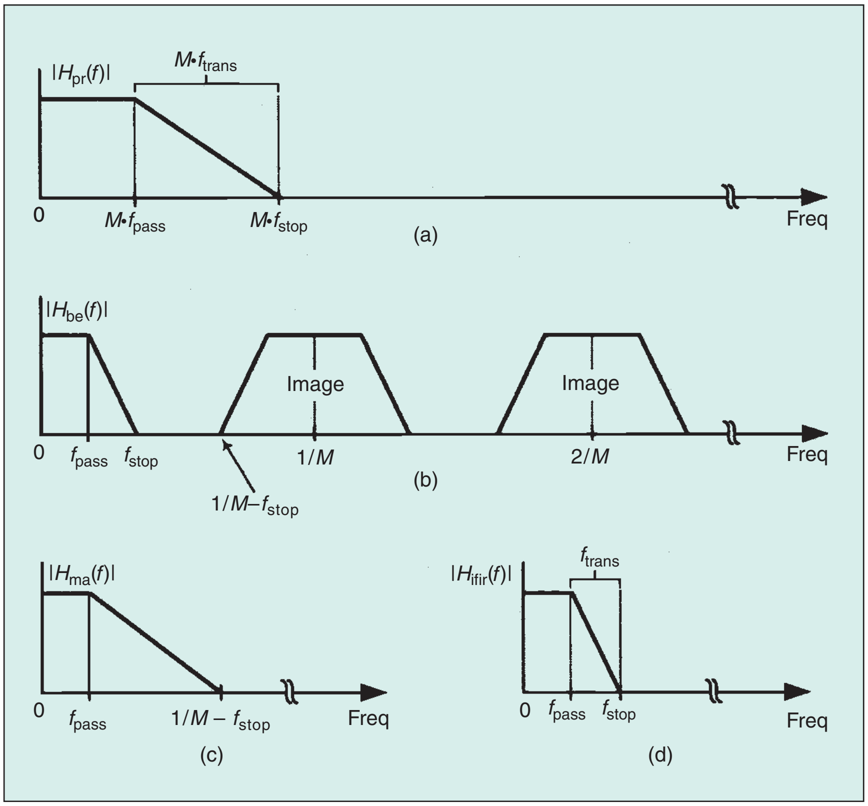
$$f_{\text{ma-pass}} = f_{\text{pass}} \qquad (6a)$$

$$f_{\text{ma-stop}} = \frac{1}{M} - f_{\text{stop}}. \qquad (6b)$$

The stopband attenuations of the prototype filter and masking subfilter are identical and set equal to the desired IFIR filter stopband attenuation. The word "interpolated" in the acronym IFIR is used because the masking subfilter interpolates the zero-valued samples in the band-edge shaping subfilter's $h_{\text{be}}(k)$ impulse response making the overall IFIR filter's impulse response nearly equivalent to that of a traditional L_{be}-length direct-convolution FIR filter. Some authors emphasize this attribute by referring to the masking subfilter as an interpolator. The sample rate remains unchanged within an IFIR filter, so no signal interpolation takes place.



▲ *1. (a) Band-edge shaping FIR filter with M-unit delays between the taps; (b) the impulse response of a prototype FIR filter; (c) the impulse response of a expanded delay band-edge shaping FIR filter with M = 3.*



▲ *2. IFIR filter magnitude responses: (a) the prototype filter; (b) band-edge shaping subfilter; (c) masking subfilter; (d) final IFIR filter.*

The following example illustrates the computational advantage of using IFIR filters. Consider the design of a desired linear-phase FIR filter whose normalized passband width is $f_{\text{pass}} = 0.1$, its passband ripple is 0.1 dB, the transition region width is $f_{\text{trans}} = 0.02$, and the stopband attenuation is 60 dB. (In this article, passband ripple is a peak-peak specification measured in decibels.) With an expansion factor of $M = 3$, the $|H_{\text{pr}}(f)|$ frequency magnitude response of the prototype filter is shown in Figure 4(a). The normalized frequency axis for these curves is such that a value of 0.5 on the abscissa represents the cyclic frequency $f_s/2$ Hz, half the sample rate. The frequency response of the band-edge shaping subfilter, for $M = 3$, is provided in Figure 4(b) with an image passband centered about $(1/M)$ Hz. The response of the masking subfilter is the solid curve in Figure 4(c) with the response of the overall IFIR filter provided in Figure 4(d).

Satisfying the original desired filter specifications in Figure 4(d) would require a traditional FIR filter with $N_{\text{tfir}} = 137$ taps, where the tfir subscript means *traditional FIR*. In our IFIR filter, the band-edge shaping and the masking subfilters require $N_{\text{pr}} = 45$ and $N_{\text{ma}} = 25$ taps, respectively, for a total of $N_{\text{ifir}} = 70$ taps. We can define the percent reduction in computational workload of an IFIR filter, over a traditional FIR filter, as

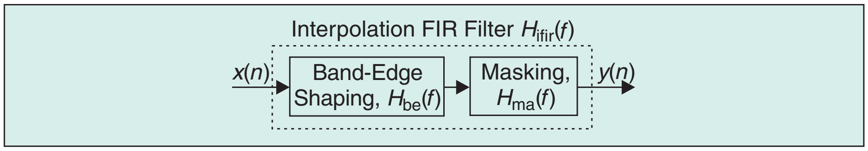
% computation reduction

$$= 100 \cdot \frac{N_{\text{tfir}} - N_{\text{pr}} - N_{\text{ma}}}{N_{\text{tfir}}}. \tag{7a}$$

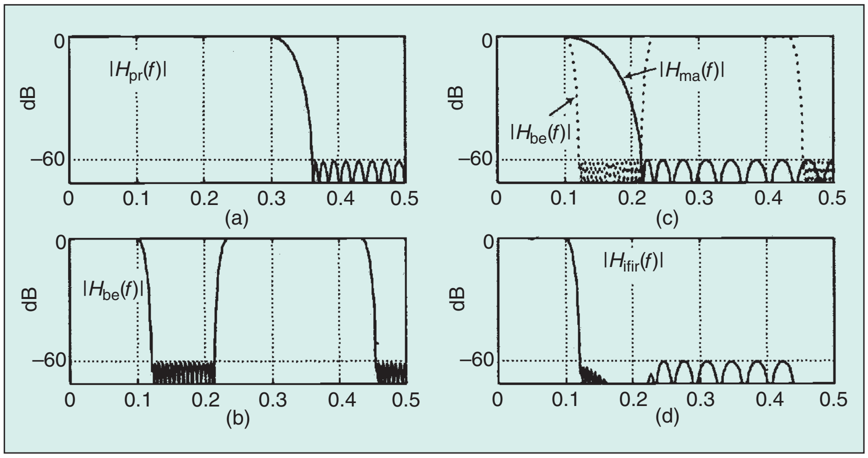
As such, the above example IFIR filter has achieved a computational workload reduction, over a traditional FIR filter, of

% computational reduction

$$= 100 \cdot \frac{137 - 70}{137} = 49\%. \tag{7b}$$


Interpolation FIR Filter $H_{\text{ifir}}(f)$
$x(n)$
Band-Edge Shaping, $H_{\text{be}}(f)$
Masking, $H_{\text{ma}}(f)$
$y(n)$

▲ *3. The structure of an interpolated FIR filter.*



▲ *4. Example lowpass IFIR filter magnitude responses: (a) the prototype filter; (b) band-edge shaping subfilter; (c) masking subfilter; (d) final IFIR filter.*

Figure 4 shows how the transition region width (the band-edge shape) of $|H_{\text{ifir}}(f)|$ is determined by the transition region width of $|H_{\text{be}}(f)|$, and this justifies the decision to call $h_{\text{be}}(k)$ the "band-edge shaping" subfilter.

Choosing the Optimum Expansion Factor *M*

The expansion factor M deserves our attention because it can have a profound effect on the computational efficiency of IFIR filters. To show this, had we used $M = 2$ in our Figure 4 example we would have realized an IFIR filter described by the $M = 2$ column in Table 1. In that case the computation reduction over a conventional FIR filter is 43%. With $M = 2$, a reduced amount of frequency-domain compression occurred in $H_{\text{be}}(f)$, which mandated more taps in $h_{\text{be}}(k)$ than that needed in the $M = 3$ case.

Now had $M = 4$ been used, the computation reduction would only be 8% as shown in the rightmost column of Table 1. This is because the $H_{\text{be}}(f)$ passband images would be so close together that a high performance (increased number of taps) masking subfilter would be required. As so often happens in signal processing designs, there is a tradeoff to be made. We would like to use a large value for M to compress the $H_{\text{be}}(f)$'s transition region width as much as possible, but a large M reduces the transition region width of the masking subfilter which increases the number of taps in $h_{\text{ma}}(k)$ and its computational workload. In the Figure 4 IFIR filter example an expansion factor of $M = 3$ is optimum because it yields the greatest computation reduction over a traditional FIR filter.

It follows from Figure 2(b) that the maximum M is the largest integer satisfying $1/M - f_{\text{stop}} \geq f_{\text{stop}}$, ensuring no passband image overlap. This yields an upper bound on M of

$$M_{\text{max}}\left\lfloor \frac{1}{2f_{\text{stop}}} \right\rfloor \tag{8a}$$

where $\lfloor x \rfloor$ indicates truncation of x to an integer. Thus the acceptable expansion factors are integers in the range $2 \le M \le M_{\text{max}}$. Evaluating (8a) for the Figure 4 IFIR filter example yields

$$M_{\text{max}}\left\lfloor \frac{1}{2(0.1+0.02)} \right\rfloor = 4 \tag{8b}$$

justifying the range of M used in Table 1.

Estimating the Number of FIR Filter Taps

To estimate the computation reduction of IFIR filters, an algorithm is needed to compute the number of taps, N, in an arbitrary traditional FIR filter. Several authors have proposed empirical relationships for estimating N for traditional direct-convolution FIR filters (also known as optimal FIR, Parks-McClellan, remez exchange, Chebyshev approximation, or equiripple filters) based on passband ripple, stopband attenuation, and transition region width [4]-[7]. A particularly simple expression for N, giving results consistent with other estimates for passband ripple values near 0.1 dB, is

$$N_{\text{tfir}} \approx \frac{\text{Atten}}{22\cdot\left(f_{\text{stop}} - f_{\text{pass}}\right)} \tag{9a}$$

where Atten is the stopband attenuation measured in decibels, and f_{pass} and f_{stop} are the normalized frequencies in Figure 2(d) [7]. Likewise, the number of taps in the prototype and masking subfilters can be estimated using

$$N_{\text{pr}} \approx \frac{\text{Atten}}{22\cdot M\left(f_{\text{stop}} - f_{\text{pass}}\right)} \tag{9b}$$

$$N_{\text{ma}} \approx \frac{\text{Atten}}{22\cdot\left(1/M - f_{\text{stop}} - f_{\text{pass}}\right)}. \tag{9c}$$

Modeling IFIR Filter Performance

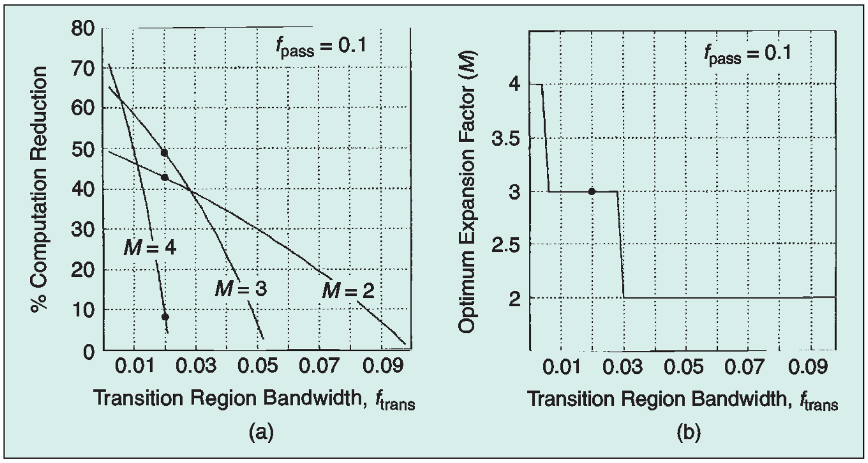
As it turns out, IFIR filter computational workload reduction depends on the expansion factor M, the passband width, and transition region width of the desired IFIR filter. To show this, we substitute the expressions in (9) into (7a) and write

$$\%\ \text{computation reduction} = 100 \cdot\left[\frac{M-1}{M} - \frac{Mf_{\text{trans}}}{1 - Mf_{\text{trans}} - 2Mf_{\text{pass}}}\right]. \tag{10}$$

Equation (10) is plotted in Figure 5(a), for a passband width is 10% of the sample rate ($f_{\text{pass}} = 0.1$) showing the percent computation reduction afforded by an IFIR filter versus transition region width for expansion factors of 2, 3, and 4. Focusing on Figure 5(a), we see when the transition region width is

Table 1. IFIR filter computation reduction versus *M*.

Expansion factor, M	2	3	4
Number of $h_{\text{be}}(k)$ taps	69	45	35
Number of $h_{\text{ma}}(k)$ taps	8	25	95
Total number of IFIR filter taps	77	70	130
Number of traditional FIR taps	137	137	137
Computation reduction	43%	49%	8%



▲ *5. IFIR filter performance versus transition region width for* $f_{\text{pass}} = 0.1$*: (a) percent computation reduction; (b) optimum expansion factors.*

large (say, $f_{\text{trans}} = 0.07$), the IFIR filter's passband plus transition region width is so wide rhat only an expansion factor of $M = 2$ will avoid passband image overlap.

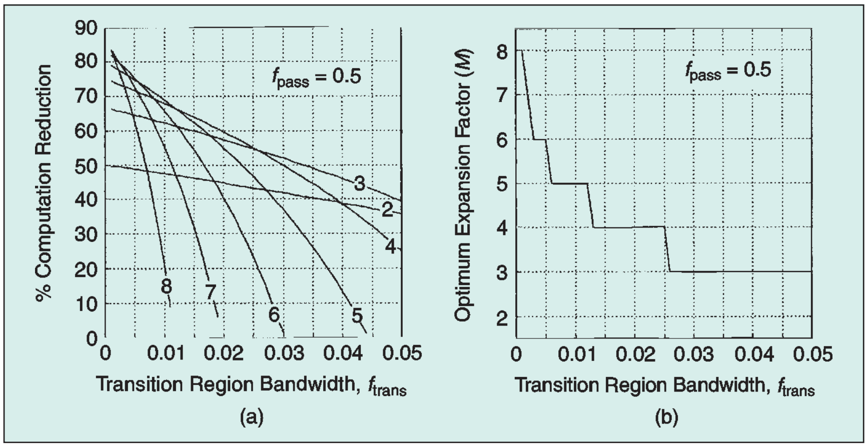
At smaller transition region widths expansion factors of 3 and 4 are possible. For example, over the transition region width range of roughly 0.005 to 0.028 an expansion factor of $M = 3$ provides greater computation reduction than using $M = 2$. The optimum (greatest percent computation reduction) expansion factor, as a function of transition region width, is shown in Figure 5(b). The black dots in Figure 5 represent the Figure 4 IFIR filter example with a transition region width of $f_{\text{trans}} = 0.02$.

To see how the percent computation reduction of IFIR filters varies with the desired passband width, Figure 6 shows IFIR filter performance when the desired passband width is 5% of the sample rate ($f_{\text{pass}} = 0.05$). The numbers on the curves in Figure 6(a) are the expansion factors.

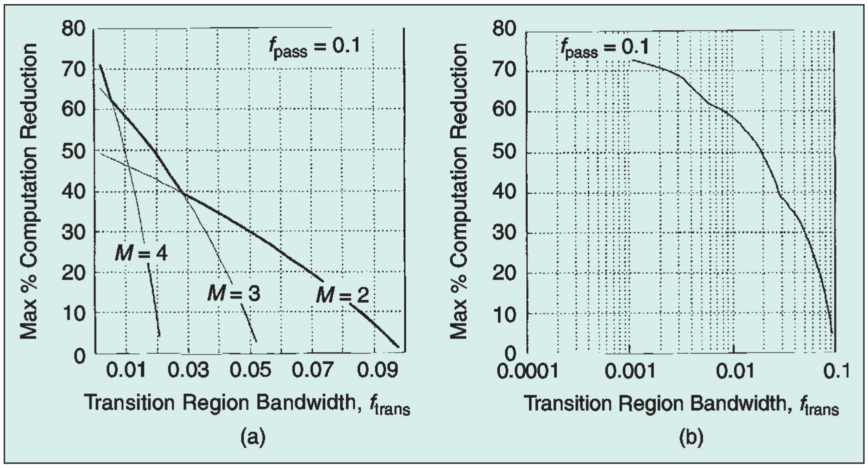
The optimum M values versus transition region width are presented in Figure 6(b). The curves in Figure 6(a) illustrate, as the first ratio within the brackets of (10) indicates, that when the transition region width approaches zero the percent computation reduction approaches $100(M-1)/M$.

We continue describing the efficiency of IFIR filters by considering the bold curve in Figure 7(a), which is the maximum percent computation reduction as a function of transition region width for the $f_{\text{pass}} = 0.1$ IFIR filter described for Figure 5(a). That bold curve is plotted on a logarithmic frequency axis, in Figure 7(b), to show *maximum* percent computation reduction over a wider transition region width range.

Next, we duplicate the curve in Figure 7(b) in Figure 8(a) and include the maximum percent computation reduction versus transition region width curves for five other IFIR filters having different passband widths, showing how significant computation reduction can be achieved by using lowpass IFIR filters. The optimum expansion factors, used to compute the curves in Figure 8(a), are shown in Figure 8(b). To keep the lower portion of Figure 8(b) from being too cluttered, curve fitting was performed to convert the *stairstep* curves to smooth curves. Shortly we'll see how the curves in Figure 8(b) are used in an IFIR filter design example.



▲ 6. *IFIR filter performance versus transition region width for* $f_{\text{pass}} = 0.05$*: (a) percent computation reduction; (b) optimum expansion factors.*



▲ 7. *Maximum percent computation reduction versus transition region width for* $f_{\text{pass}} = 0.1$ *plotted on (a) a linear axis and (b) a logarithmic axis.*

Passband Ripple Considerations

The passband ripple of an IFIR filter is a function of the individual band-edge shaping and masking subfilters' passband ripple. If we represent an arbitrary FIR filter's peak passband ripple as shown in Figure 9, we can estimate IFIR filter passband ripple.

From (4), the upper bound of an IFIR filter's passband response is the product of the band-edge shaping and masking subfilters' peak passband responses as

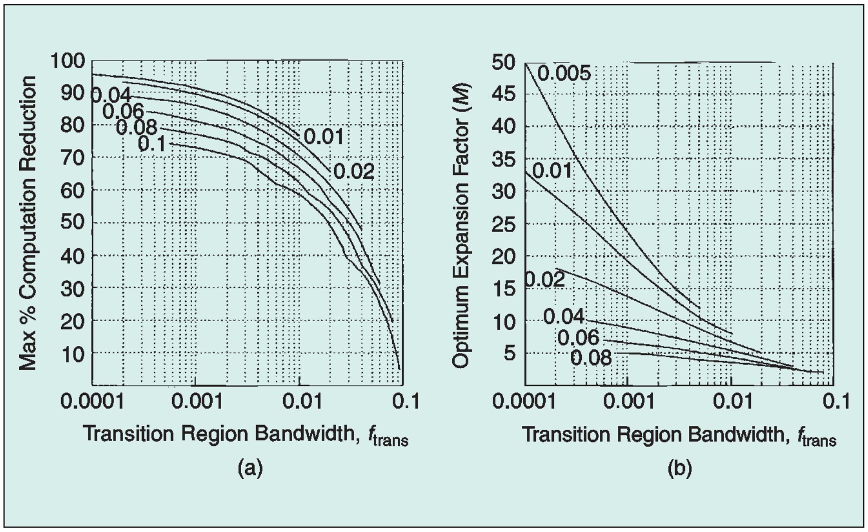
$$\begin{aligned} 1+\delta_{\text{ifir}} &= (1+\delta_{\text{be}})\cdot(1+\delta_{\text{ma}}) \\ &= 1+\delta_{\text{be}}+\delta_{\text{ma}}+\delta_{\text{be}}\delta_{\text{ma}}. \end{aligned} \quad (11)$$

For small values of δ_{be} and δ_{ma}, the $\delta_{be}\delta_{ma}$ term becomes negligible, and we state

$$\delta_{ifir} = \delta_{be} + \delta_{ma}. \tag{12}$$

Thus, in the design of the band-edge shaping and masking subfilters, it's prudent to specify their passband ripple values to be roughly half the desired ripple specification for the final IFIR filter as

$$\delta_{be} = \delta_{ma} \approx \delta_{ifir} / 2. \tag{13}$$



▲ *8. IFIR filter performance versus transition region width for various passband widths: (a) maximum percent computation reduction; (b) optimum expansion factors.*

IFIR Filter Implementation Issues

The computation reduction of IFIR filters is based on the assumption that they are implemented as two separate subfilters as in Figure 3. We have resisted the temptation to combine the two subfilters into a single filter whose coefficients are the convolution of the subfilters' impulse responses. Such a maneuver would eliminate the zero-valued coefficients of the band-edge shaping subfilter and eliminate any computation reduction.

The curves in Figure 8(b) indicate an important implementation issue when using IFIR filters. With decreasing IFIR filter passband width, larger expansion factors, M, can be used. When using programmable DSP chips, larger values of M require a larger block of hardware data memory, in the form of a circular buffer, be allocated to hold a sufficient number of input $x(n)$ samples for the band-edge shaping subfilter. The size of this data memory must be equal to L_{be} as indicated in (3). Some authors refer to this data memory allocation requirement, to accommodate all the stuffed zeros in the $h_{be}(k)$ impulse response, as a disadvantage of IFIR filters. This is a misleading viewpoint because, as it turns out, the L_{be} length of $h_{be}(k)$ is only few percent larger than the length of the impulse response of a traditional FIR filter having the same performance as an IFIR filter. So from a data storage standpoint the price we pay to use IFIR filters is a slight increase in the memory of size to accommodate L_{be}, plus the data memory of size L_{ma} needed for the masking subfilter. In practice, for narrowband lowpass IFIR filters L_{ma} is typically less than 10% of L_{be}. The L_{be}-sized data memory allocation, for the band-edge shaping subfilter, is not necessary in field programmable gate array (FPGA) IFIR filter implementations because the FPGA area is not a strong function of the expansion factor M [8].

When implementing an IFIR filter with a programmable DSP chip, the filter's computation reduction gain can only be realized if the chip's architecture enables zero-overhead looping through the circular data memory using an increment equal to the expansion factor M. That looping capability ensures that only the nonzero-valued coefficients of $h_{be}(k)$ are used in the band-edge shaping subfilter computations.

In practice the band-edge shaping and masking subfilters should be implemented with a folded nonrecursive FIR structure, exploiting their impulse response symmetry, to reduce the number of necessary multiplications by a factor of two. Using a folded structure does not alter the performance curves provided in this article. Regarding an IFIR filter's implementation in fixed-point hardware, its sensitivity to coefficient quantization errors is no greater than that exhibited by traditional FIR filters [2].

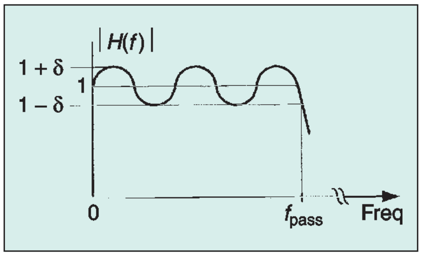
IFIR Filter Design Example

The design of practical lowpass IFIR filters is straightforward and comprises four steps:

▲ define the desired lowpass filter performance requirements

▲ determine a candidate value for the expansion factor M

▲ design and evaluate the band-edge shaping and masking subfilters



▲ *9. Definition of FIR filter peak passband ripple δ.*

▲ investigate IFIR performance for alternate expansion factors near the initial M value.

As a design example, refer to Figure 2(d) and assume that we desire a lowpass IFIR filter with $f_{\text{pass}} = 0.02$, a passband ripple of 0.1 dB, a transition region bandwidth of $f_{\text{trans}} = 0.005$ (thus $f_{\text{stop}} = 0.025$), and 60 dB of stopband attenuation. First, we find the $f_{\text{trans}} = 0.005$ point on the abscissa of Figure 8(b) and follow it up to the point where it intersects the $f_{\text{pass}} = 0.02$ curve. This intersection indicates we should start our design with an expansion factor of $M = 8$. (The same intersection point in Figure 8(a) suggests we can achieve a computational workload reduction of roughly 80%.)

With $M = 8$ and applying (5), we use our favorite traditional FIR filter design software to design a linear-phase prototype FIR filter with the following parameters:

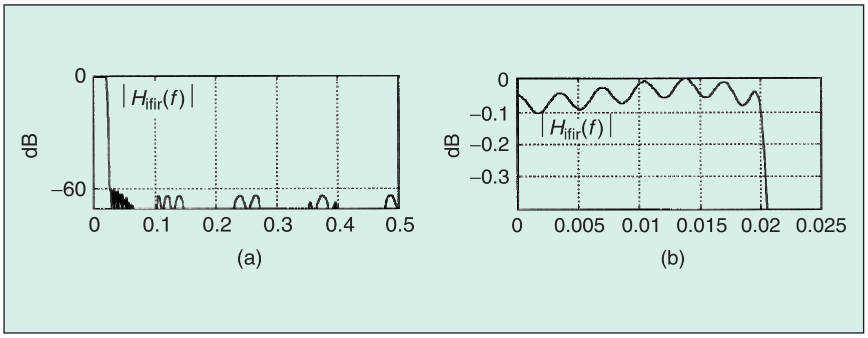
$f_{\text{pr-pass}} = M \cdot (0.02) = 0.16$,

passband ripple = (0.1)/2 dB = 0.05 dB,

$f_{\text{pr-stop}} = M \cdot (0.025) = 0.2$, and

stopband attenuation = 60 dB.

Such a prototype FIR filter will have $N_{\text{pr}} = 76$ taps and, from (3), when expanded by $M = 8$ the band-edge shaping subfilter will have an impulse response length of $L_{\text{be}} = 601$ samples.



▲ 10. *IFIR filter design example magnitude responses: (a) full response; (b) passband ripple response.*

Next, using (6) we design a masking subfilter having the following parameters:

$f_{\text{ma-pass}} = f_{\text{pass}} = 0.02$,

passband ripple = (0.1)/2 dB = 0.05 dB,

$f_{\text{ma-stop}} = 1/M - f_{\text{stop}}$

$= 1/8 - 0.025 = 0.1$,

and stopband attenuation = 60 dB.

This masking subfilter will have $N_{\text{ma}} = 40$ taps and when cascaded with the band-edge shaping subfilter will yield an IFIR filter requiring 116 multiplications per filter output sample. The frequency response of the IFIF filter is shown in Figure 10(a), with passband response detail provided in Figure 10(b).

A traditional FIR filter satisfying our design example specifications would require approximately $N_{\text{tfir}} = 588$ taps. Because the IFIR filter requires only 116 multiplications per output sample, using (7a), we have realized a computational workload reduction of 80%.

Further modeling for alternate expansion factors yields the IFIR filter performance results in Table 2. There we see how the M expansion factors

Table 2. IFIR design example computation reduction versus *M*.

Expansion factor, M	5	6	7	8	9	10	11
Number of $h_{\text{be}}(k)$ taps	123	100	86	76	69	62	56
Number of $h_{\text{ma}}(k)$ taps	21	26	33	40	50	59	71
Total IFIR filter taps	144	126	119	116	119	121	127
Traditional FIR taps	588	588	588	588	588	588	588
L_{be} data storage rqmt.	611	595	596	601	613	611	606
Computation reduction	76%	79%	80%	80%	80%	79%	78%

of 7, 8, and 9 provide equivalent computational reductions; however we select $M = 7$ because it requires the smallest of L_{be}-sized data storage for the band-edge shaping subfilter. The final IFIR filter design step is to sit back and enjoy a job well done.

Concluding Remarks

IFIR filters are suitable whenever narrowband lowpass linear-phase filtering is required; for example, the filtering prior to decimation for narrowband channel selection within wireless communications receivers or in digital television. IFIR filters are essential components in sharp-transition wideband frequency-response masking (FRM) FIR filters [9], [10]. In addition, IFIR filters can also be employed in narrowband two-dimensional filtering applications.

While this article focused on lowpass IFIR filters, highpass IFIR filters can be designed using the same prototype filter expansion and masking principles. Upon expansion by M of a highpass prototype filter's impulse response, the resulting highpass band-edge shaping subfilter will exhibit passband images centered about odd integer multiples of $1/(2M)$. There are two restrictions associated with the design of a highpass IFIR filter. First, the prototype highpass FIR filter must have an odd impulse response length. (Even-length nonrecursive FIR filters have a zero magnitude response at $f_s/2$, preventing their use as highpass filters.) The second restriction is the band-edge shaping subfilter's expansion factor must be an odd integer to ensure an $|H_{be}(f)|$ passband image residing at $f_s/2$.

Additional, and more complicated, IFIR design methods have been described in the literature. Improved computational workload reduction, on the order of 30-40% beyond that presented here, has been reported using an intricate design scheme when the Figure 3 masking subfilter is replaced with multiple stages of filtering [11].

We've introduced the theory of linear-phase narrowband lowpass IFIR filters and shown how they achieve significant computational workload reduction (as large as 90%) relative to traditional direct-convolution FIR filters, at the cost of less than a 10% increase in hardware data memory requirements. IFIR filters were seen to have a simple cascade structure and straightforward implementation. Performance curves were presented to aid the designer in choosing the appropriate band-edge shaping subfilter expansion factor to maximize IFIR filter efficiency. Finally, a straightforward IFIR filter design procedure was presented based on frequency-domain analysis and the use of readily-available traditional FIR filter design software.

Richard Lyons (ricklyon@onemain.com) is a consulting systems engineer and lecturer with Besser Associates in Mt. View, California. He has been the lead hardware engineer for numerous multimillion dollar signal processing systems for both the National Security Agency (NSA) and TRW Inc. and has taught at the University of California Santa Cruz Extension. He is an associate editor for *IEEE Signal Processing Magazine* and author of *Understanding Digital Signal Processing* (Prentice-Hall, 1997). He is a member of the IEEE and the Eta Kappa Nu honor society and rides a 1981 Harley Davidson.

References

[1] T. Parks and J. McClellan, "A program for the design of linear phase finite impulse response digital filters," *IEEE Trans. Audio Electroacoust.*, vol. AU-20, pp. 195-199, Aug. 1972.

[2] Y. Neuvo, C.Y. Dong, and S.K. Mitra, "Interpolated finite impulse response filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, pp. 563-570, June 1984.

[3] P. Vaidyanathan, *Multirate Systems and Filter Banks.* Englewood Cliffs, NJ: Prentice Hall, 1993.

[4] C. Rorabaugh, *DSP Primer*. New York: McGraw-Hill, 1999, pp. 279.

[5] R. Crochiere and L. Rabiner, "Interpolation and decimation of digital signals—A tutorial review," *Proc. IEEE*, vol. 69, pp. 300-331, Mar. 1981

[6] J. Kaiser, "Nonrecursive digital filter design using I_o-sinh window function," in *Proc. 1974 IEEE Int. Symp. Circuits Systems*, Apr. 1974, pp. 20-23.

[7] F. Harris, "Digital signal processing for digital modems," in *DSP World Spring Design Conf., Lecture Notes, Non-Recursive Filters*, Santa Clara, CA, Apr. 1999, pp. 5-6.

[8] C. Dick, "Implementing area optimized narrow-band FIR filters using Xilinx FPGAs," in *Proc. SPIE Int. Symp. Voice, Video and Data Communications*, Boston, MA, Nov. 1998, pp. 227-238 [Online]. Available: http://www.xilinx.com/products/logicore/dsp/ifir.pdf

[9] Y. Lim, "Frequency-response masking approach for the synthesis of sharp linear phase digital filters," *IEEE Trans. Circuits Syst*., vol. CS-33, pp. 357-364, Apr. 1986.

[10] R. Yang, B. Liu, and Y.C. Lim, "A new structure of sharp transition FIR filters using frequency-response masking," *IEEE Trans. Circuits Syst.*, vol. 35, pp. 955-966, Aug. 1988.

[11] T. Saramaki, Y. Neuvo, and S. Mitra, "Design of computationally efficient interpolated FIR filters," *IEEE Trans. Circuits Syst*., vol. 35, pp. 70-88, Jan. 1988.