Understanding Interpolated FIR Filters

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 article describes a class of digital filters, called interpolated FIR filters, that can implement narrowband lowpass finite impulse response (FIR) filter designs with a significantly reduced computational workload relative to traditional FIR filters. First we'll cover the theory of interpolated FIR filters and then present a design example showing the 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 L-unit delays, with the *expansion factor L* being an integer, as shown in Figure 1(a). If the $h_p(k)$ impulse response of a 9-tap FIR filter is that shown in Figure 1(b), the impulse response of an *expanded* FIR filter, where for example $L = 3$, is the $h_{sh}(k)$ 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 we'll introduce the filter with expanded delays as the *shaping* subfilter. Shortly we'll see why this terminology is sensible.

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

$$H_p(z) = \sum_{k=0}^{N_p-1} h_p(k)z^{-k} \qquad (1)$$

where N_p is the length of h_p. The transfer function of a general shaping FIR filter, with z in (1) replaced with z^L, is

$$H_{sh}(z^L) = \sum_{k=0}^{N_p-1} h_p(k)z^{-kL}. \qquad (2)$$

The frequency-domain effect of those L-unit delays is shown in Figure 2 where the frequency axis is measured in Hz.

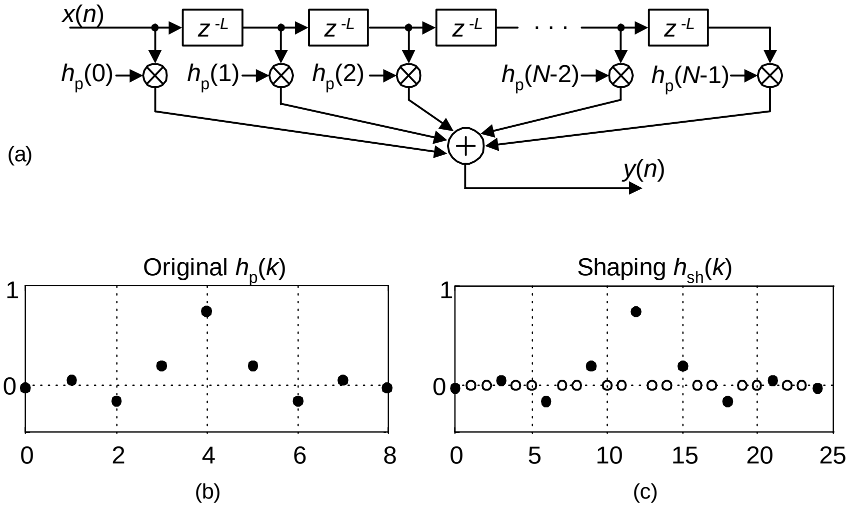

Figure 1. Shaping FIR filter with L-unit delays between the taps (a), the impulse response of a prototype FIR filter (b), and the impulse response of an expanded-delay shaping FIR filter with L = 3 (c).

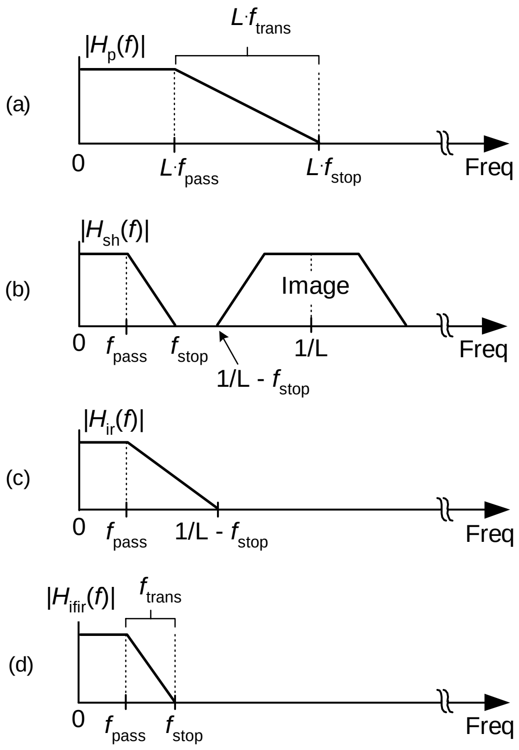

Figure 2. IFIR filter magnitudes responses: (a) the prototype filter, (b) shaping subfilter, (c) image-reject subfilter, (d) final IFIR filter.

As we should expect, an L-fold expansion of the time-domain filter impulse response causes an L-fold compression (and repetition) of the frequency-domain $|H_p(f)|$ magnitude response as in Figure 2(b). While $H_p(f)$ has a single passband, $H_{sh}(f)$ was L passbands. (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 $f_{pass}{\cdot}f_s$ Hz.) Those repetitive passbands in $|H_{sh}(f)|$ centered about integer multiples of $1/L$ (f_s/L Hz) are called *images*, and on them we now focus our attention.

If we follow the shaping subfilter with a lowpass *image-reject* subfilter, Figure 2(c), whose task is to attenuate the image passbands, we can realize a multistage filter whose frequency response is shown in Figure 2(d). The resultant $|H_{ifir}(f)|$ frequency magnitude response is, of course,

$$|H_{ifir}(f)| = |H_{sh}(f)|{\cdot}|H_{ir}(f)|. \qquad (3)$$

The structure of the cascaded subfilters is the so-called IFIR filter shown in Figure 3(a), with its interpolated impulse response given in Figure 3(b).

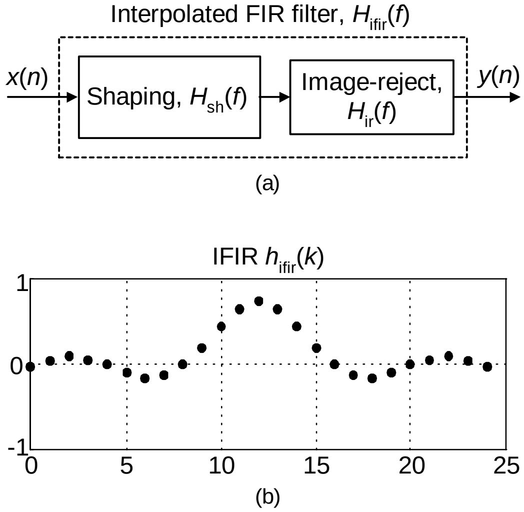

Figure 3. Lowpass interpolated FIR filter: (a) the structure, (b) impulse response.

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

$$f_{p\text{-}pass} = Lf_{pass} \qquad (4a)$$

$$f_{p\text{-}stop} = Lf_{stop} \qquad (4b)$$

$$f_{p\text{-}trans} = Lf_{trans} = L(f_{stop} - f_{pass}). \qquad (4c)$$

The image-reject subfilter's frequency parameters are

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

$$f_{ir\text{-}stop} = \frac{1}{L} - f_{stop}. \qquad (5b)$$

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

A scheme for estimating the optimum L expansion factor for a particular IFIR filter application has been developed and resulted in the design curves in Figure 4 [See Ref. 1]. By optimum we mean selecting the expansion factor L resulting in a minimum IFIR filter computational workload. The various curves in Figure 4 apply to various desired IFIR filter passband widths f_{pass}. The best expansion factor depends both on the desired filter's transition region width and passband width. Let's see, now, how those curves are used in an IFIR filter design example.

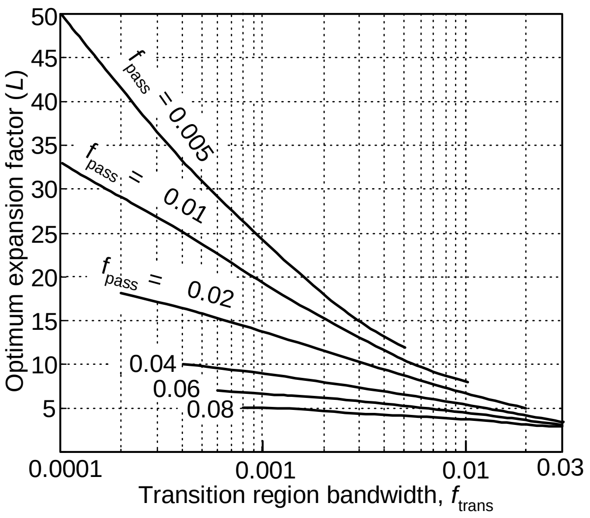

Figure 4. IFIR filter optimum expansion factors.

2. IFIR FILTER DESIGN EXAMPLE

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

1. Define the desired lowpass filter performance requirements,
2. Determine a candidate value for the expansion factor L from Figure 4,
3. Design and evaluate the shaping and image-reject subfilters using Eqs. (4) and (5), and
4. Investigate IFIR performance for alternate expansion factors near the initial L value.

As a design example, refer to Figure 2(d) and assume we want to build a lowpass IFIR filter with $f_{\text{pass}} = 0.02$, a peak-peak passband ripple of 0.5 dB, a transition region bandwidth of $f_{\text{trans}} = 0.01$ (thus $f_{\text{stop}} = 0.03$), and 50 dB of stopband attenuation. First, we find the $f_{\text{trans}} = 0.01$ point on the abscissa of Figure 4 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 $L = 7$.

With $L = 7$, and applying Eq. (4), we use our favorite traditional FIR filter design software to design a linear-phase prototype FIR filter with the following parameters:

$f_{\text{p-pass}} = L \cdot 0.02 = 0.14$,
passband ripple = (0.5)/2 dB = 0.25 dB,
$f_{\text{p-stop}} = L \cdot (0.03) = 0.21$, and
stopband attenuation = 50 dB.

(Notice how we specified the prototype filter's passband ripple to be half our final desired ripple, and we'll do the same for the image-reject subfilter. The justification for this can be found in Chapter 6 of [1].) Such a prototype FIR filter will have $N_{\text{p}} = 33$ taps and when expanded by $L = 7$ the shaping subfilter will have an impulse response length of 225 samples. Its frequency magnitude response is shown in Figure 5(a). The shaping subfilter's frequency magnitude response is shown as the solid curve in Figure 5(b).

Next, using Eq. (5) we design a image-reject subfilter having the following parameters:

$\text{f}_{\text{ir-pass}} = f_{\text{pass}} = 0.02$,
passband ripple = (0.5)/2 dB = 0.25 dB,
$\text{f}_{\text{ir-stop}} = 1/L - f_{\text{stop}} = 1/7 - 0.03 = 0.113$, and
stopband attenuation = 50 dB.

This image-reject subfilter will have $N_{\text{ir}} = 27$ taps, and when cascaded with the shaping subfilter will yield an IFIR filter requiring 60 multiplications per filter output sample. The frequency response of the image-reject subfilter is shown in as the dashed curve in Figure 5(b). The result of this filter design is the frequency response of the IFIR filter as shown in Figure 5(c), with passband response detail provided in Figure 5(d).

A traditional FIR filter satisfying our design example specifications would require approximately $N_{\text{tfir}} = 240$ taps. Because the IFIR filter requires only 60 multiplications per output sample, we have realized a computational workload reduction of 75%. The final IFIR filter design step is to sit back and enjoy a job well done.

3. IFIR FILTER IMPLEMENTATION ISSUES

The computation reduction of IFIR filters is based on the assumption they are implemented as two separate subfilters as in Figure 3(a). We have resisted the temptation to combine the two subfilters into a single filter whose coefficients are the convolution of the subfilters' impulse responses.

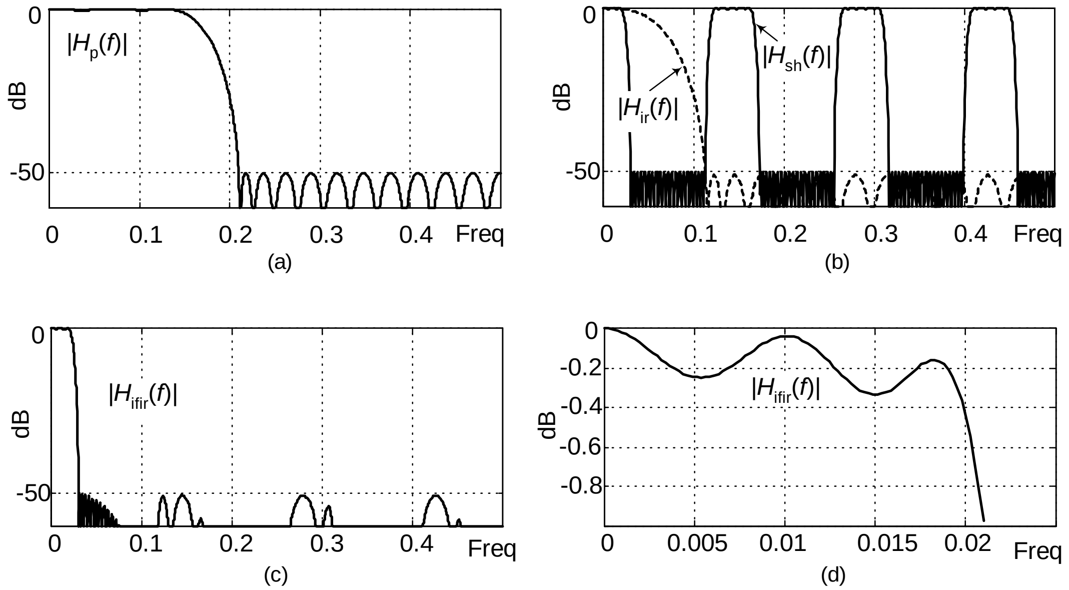

Figure 5. IFIR filter design example magnitudes responses: (a) prototype filter, (b) shaping and image-reject subfilters, (c) IFIR filter, (d) IFIR filter passband response detail.

Such a maneuver would eliminate the zero-valued coefficients of the shaping subfilter and we'd lose all computation reduction.

The curves in Figure 4 indicate an important implementation issue when using IFIR filters. With decreasing IFIR filter passband width, larger expansion factors, L, can be used. When using programmable DSP chips larger values of L 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 shaping subfilter. The size of this data memory must be equal to

$$K_{sh} = L(N_p-1) + 1. \qquad (6)$$

Some authors refer to this data memory allocation requirement, to accommodate all the stuffed zeros in the $h_{sh}(k)$ impulse response, as a disadvantage of IFIR filters. This is a misleading viewpoint because, as it turns out, the K_{sh} length of $h_{sh}(k)$ 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 K_{sh}, plus the data memory of size K_{ir} needed for the image-reject subfilter. In practice, for narrowband lowpass IFIR filters, K_{ir} is typically less than 10% of K_{sh}. The K_{sh}-sized data memory allocation, for the shaping subfilter, is not necessary in field programmable gate array (FPGA) IFIR filter implementations because FPGA area is not a strong function of the expansion factor L.

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 L. That looping capability ensures only the nonzero-valued coefficients of $h_{sh}(k)$ are used in the shaping subfilter computations.

In practice the shaping and image-reject 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 Figure 4. Regarding an IFIR filter's implementation in fixed-point hardware, its sensitivity to coefficient quantization errors is no greater than the errors exhibited by traditional FIR filters.

4. IFIR FILTERS WITH SAMPLE RATE CHANGE

IFIR filters particularly useful for signal sample rate change, decimation or interpolation, applications. To see why this is so, we refer to Figure 6(a). There we show the standard IFIR filter structure followed by downsampling by an integer L process. The downsampling (decimation) operation '$\downarrow L$' means discard all but every Lth sample. Because the $H_{sh}(z^L)$ shaping subfilter and the $H_{ir}(z)$ image-reject subfilter are linear, we can swap their order in the IFIR filter as depicted in Figure 6(b). Now here comes the good part.

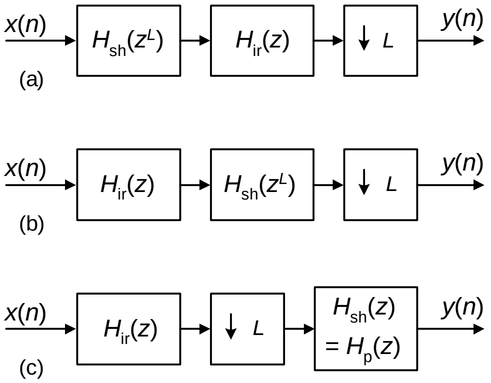

Figure 6. IFIR filtering with decimation by *L*.

Because a filter whose *z*-domain transfer function is defined in terms of a polynomial in z^L (Eq. (2)) followed by *L* downsampling is equivalent to an *L* downsampler followed by the filter defined in terms of a polynomial in *z*, we swap the order of filter $H_{sh}(z^L)$ with the downsampler and arrive at the structure shown in Figure 6(c). Every *L*-unit delay in $H_{sh}(z^L)$ is then replaced by a unit delay, which takes use back to using our original low-order prototype filter $H_p(z)$.

This happy scenario reduces the signal data storage requirements of our traditional IFIR filter. In addition, the $H_{ir}(z)$ and *L* downsampler combination can be implemented using polyphase filtering techniques to reduce its computational workload [1].

In a similar manner, IFIR filters can be used for interpolation (upsampling) as shown in Figure 7(a). There we show an upsampling process followed by a standard IFIR filter structure. The upsampling (interpolation) operation '↑*L*' means insert *L*-1 zero-valued samples between each $x(n)$ sample.

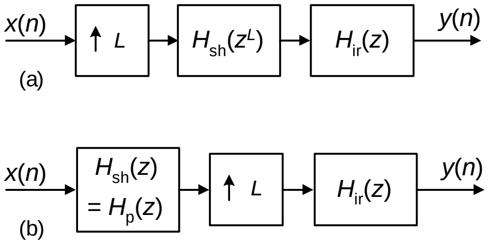

Figure 7. IFIR filtering with interpolation by *L*.

Because an *L* upsampler followed by a filter defined in terms of a polynomial in z^L is equivalent to the filter defined in terms of a polynomial in *z* followed by *L* upsampler, we swap the order of filter $H_{sh}(z^L)$ with the upsampler and arrive at the structure shown in Figure 7(b). Every *L*-unit delay in $H_{sh}(z^L)$ is then replaced by a unit delay which, again, takes use back to using our original prototype filter $H_p(z)$ with its reduced data storage requirements. As with decimation, the *L* upsampler and $H_{ir}(z)$ combination can use polyphase filtering to reduce its computational workload.

5. 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 and can also be employed in narrowband two-dimensional filtering applications.

To conclude our linear-phase IFIR filter coverage, we reiterate the good news that they can achieve significant computational workload reduction relative to traditional single-stage direct-convolution FIR filters. IFIR filters are lean, mean, filtering machines. Happily, their implementation is a straightforward cascade of nonrecursive (tapped-delay line) FIR filters designed using readily-available traditional FIR filter design software.

Due to space constraints, this discussion of IFIR filters was necessarily brief. Reference [1] provides more details concerning mathematical derivations describing the performance curves, computational workload reduction estimates, and further implementation issues regarding IFIR filters.

6. REFERENCES

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