Digital Signal Processing Case Study: Interpolated FIR Filters

Ross L. Penniman, *Graduate Student*

Abstract—**The Interpolated FIR (IFIR) design technique provides a considerable reduction in computation cost for finite impulse response (FIR) filters with narrow bandwidth. The design process of an IFIR filter is explained and evaluated for computational savings. Different variations of the IFIR technique are examined, and their strengths and weaknesses are compared.**

Index Terms—**IFIR, interpolation, filter, narrowband**

I. Introduction

FINITE impulse response (FIR) filters are a very useful tool for signal processing as they are guaranteed to be stable and can be designed to have linear phase. The primary drawback of FIR filters, however, is that they typically require a large number of delays and a large number of multiplications. The interpolated finite impulse response (IFIR) filter is able to make significant improvements in terms of the number of multiplications required. It is not able to reduce the number of delays (it actually increases the number of delays by small percentage), but the net result is still considered a significant improvement in performance. The IFIR method is best used for high order filters that have a narrow bandwidth. The savings in computational cost increases as the passband of the filter narrows.

II. Original IFIR Design Method

The IFIR design method was first introduced by Neuvo, Dong, and Mitra [1]. The main idea is that the FIR filter is divided into two cascaded filters:

$$H(z) = F(z^L)G(z). \tag{1}$$

In this form $F(z^L)$ is the “original” form of the filter and defines the passband, transition band, and stop band. The z^L term means that the frequency response is periodically replicated with period $2\pi/L$. The $G(z)$ term is an additional FIR filter section designed to remove the additional passbands created by the periodicity of the first term. Since $F(z^L)$ determines the shape of the passband, transition band, and stopband, it will be referred to as the “shaping” filter. Since $G(z)$ in effect interpolates the impulse response of the previous filter, it will be referred to as the “interpolation” filter. An important parameter to consider is the “stretching” factor L. It determines how much the impulse response of $F(z^L)$ is stretched out, and correspondingly how much the frequency response of $F(z)$ is compressed on the frequency axis. Picking a small value of L will have little benefit in terms of computational savings, but will also only require an interpolation filter of relatively low order. Picking a large value of L, on the other hand, will result in a great benefit of computational savings for the shaping filter, but may in turn require a very high order interpolation filter, resulting in little net benefit. The maximum L possible while still preventing overlap between the periodic passbands is

$$L_{MAX} = \left\lfloor \frac{\pi}{\omega_s} \right\rfloor \tag{2}$$

where ω_s is the stopband edge. In most cases we will want to pick values significantly smaller than L_{MAX} in order to have the optimum computational savings.

A. IFIR Design Process

The process of designing an IFIR filter can be broken into several steps. As an example, we will use a filter with passband and stopband edge frequencies of 0.05π and 0.1π respectively. The allowable passband and stopband ripple are both 0.05.

1) Select a stretching factor L

Using (2) we compute L_{MAX} to be 10. Therefore the value of L must be in the range:

$$2 \le L \le L_{MAX} \tag{3}$$

If we arbitrarily take a value half-way between, we have $L = 6$.

2) Design the model filter $F(z)$

We need to carefully choose new values for the passband and stopband edge frequencies, so that as $F(z)$ is compressed on the frequency axis, it will result in having the desired parameters. For a desired passband edge frequency ω_p and stopband edge frequency ω_s the new values are given by $L\omega_p$ and $L\omega_s$ respectively. For the desired passband and stopband ripple amplitude of δ_p and δ_s respectively, the estimated filter

Manuscript received November 7, 2012. R. L. Penniman is a Graduate Student in the Music Engineering Department, Frost School of Music at the University of Miami, Coral Gables, FL 33124 USA (e-mail: rosspenn@gmail.com).

order can be computed via a formula developed by Kaiser [4]. The filter order N is given by the formula:

$$N = \frac{-20\log_{10}\sqrt{\delta_p\delta_s}-13}{2.324(\omega_s-\omega_p)} \tag{4}$$

Evaluating (4) for our example we have an estimated filter order of 6. The estimated filter order can then be used to run the McClellan-Parks algorithm [5] to find the optimum filter coefficients.

3) Up-sample the model filter to obtain $F(z^L)$

If the impulse response of the model filter is given by h_M we obtain the impulse response of the shaping filter h_F by inserting $L-1$ zeros between each sample. This can be represented as:

$$h_F = \begin{cases} h_M\left(\frac{n}{L}\right) & n = iL,\ i = 0, \pm1, \pm2, \dots \\ 0 & otherwise \end{cases} \tag{5}$$

The original impulse response and the upsampled version of the impulse response are shown in Fig. 1. The change in the frequency response of the filter as the result of the upsampling is shown in Fig. 2.

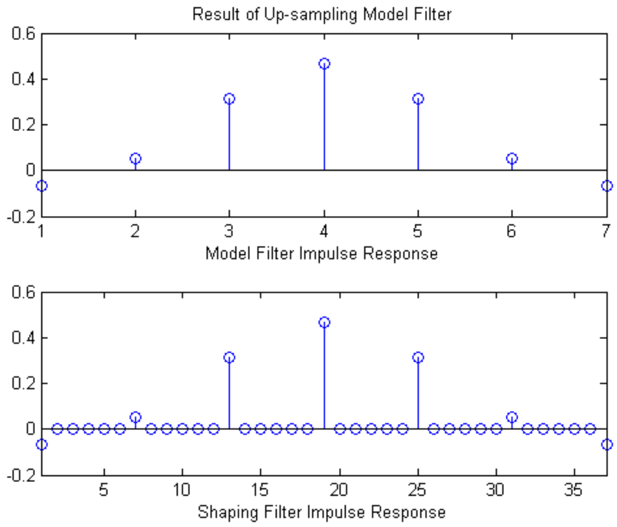

Fig. 1: Comparison of impulse responses of model and shaping filters.

4) Design the interpolation filter

The role of the interpolation filter is to remove all of the periodic passbands from the response of the shaping filter while leaving the primary passband unchanged. Accordingly, the ω_p value will be the same for both. In order to keep the order of the interpolation filter as low as possible, the ω_s value should be chosen to be as high as possible. Since the shaping filter is periodic with period $2\pi/L$, this value represents the center frequency of the first “image” of the filter. If we come down ω_s below this center frequency, we have the point at which the first “image” rises about the stop band level. Thus the stopband edge frequency for the interpolation filter ω_{si} is given by:

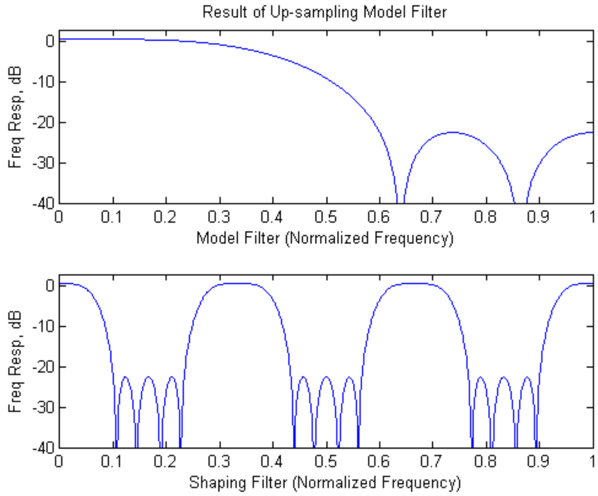

Fig. 2: Comparison of frequency responses of model and shaping filters.

$$\omega_{si} = \frac{2\pi}{L} - \omega_s \tag{6}$$

where ω_s is taken from the shaping filter. Evaluating (6) and (4) for the interpolation filter gives us a value $N = 10$ for the filter order. The McClellan-Parks algorithm can be used to design the interpolation filter as well.

5) Total result

The combination of the shaping filter in cascade with the interpolating filter can be shown to meet or exceed the requirements of the design. Fig. 3 shows the individual and combined frequency responses for the example filter.

B. Computation cost

Since the motivating factor in using the IFIR technique is its savings in computational cost, it is important to be able to evaluate what that savings is. Fortunately, linear phase FIR filters are very easy to evaluate given that there is typically only one structure used to implement them [6]. The number of additions is equal to the order of the filter and the number of multiplications is given by the equation:

$$Multiplications = \left\lceil\frac{N+1}{2}\right\rceil \tag{7}$$

Evaluating (4) for a traditional FIR filter with the specifications given in the example would produce a filter with order $N = 36$. Given that the component filters of the IFIR have orders of 6 and 10 respectively, it is clear that there is a significant reduction in number of multiplications and additions which must be performed. The computation cost of the two options are compared in Table 1.

The example given is relatively low order filter and as a result the computational savings are fairly modest. It has been demonstrated by [3] that savings of 80% or more are attainable by high order filters.

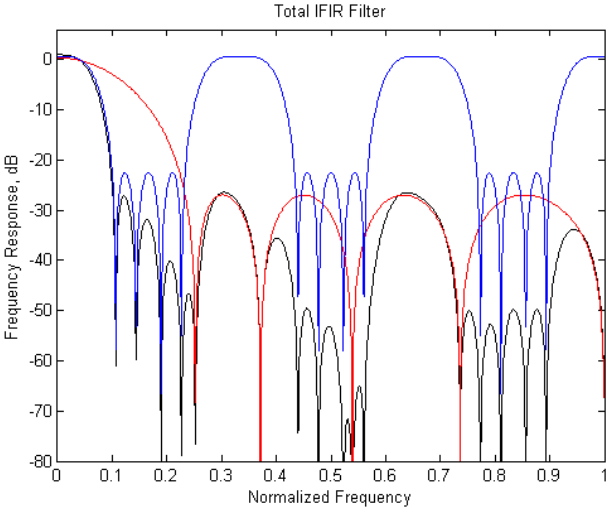

Fig. 3: Frequency responses of shaping filter (blue), interpolation filter (red), and combination (black).

TABLE 1
COMPARISON OF COMPUTATIONAL COST

Filter	*Order*	*Mult.*	*Add.*
Shaping	6	4	6
Interpolating	10	6	10
Combined	**16**	**10**	**16**
FIR	**36**	**19**	**36**

C. *Memory requirement*

It is important to compare the memory requirements of the two filters. While the IFIR filter does have fewer calculations, it will require more memory than the traditional FIR filter. This is because the impulse response of the stretched shaping filter will typically be the same length as the impulse response of the filter it is replacing. The IFIR also has the additional memory requirement of the interpolating filter, which can vary depending on the design parameters chosen.

The memory requirement of the shaping filter must take into account the zero values between each filter coefficient. Thus the total number of delays for an IFIR filter will be:

$$N_s * L + N_i \tag{8}$$

where N_s is the order of the shaping filter, L is the stretching factor, and N_i is the order of the interpolating filter. Using these figures for our example, we can see that the IFIR filter will require 46 delays as opposed to 36 delays for the traditional FIR filter.

D. *Design example*

The differences between using an IFIR filter as opposed to a traditional FIR filter are more pronounced when working with a more real-world example. A filter suggested in [7] has the following values:

$$\delta_p = 0.001$$
$$\delta_s = 0.001$$
$$\omega_p = 0.015\pi$$
$$\omega_s = 0.02\pi$$

The stretching factor has been found to yield good results for a value of $L = 15$ (see discussion on optimization). Plots of the frequency response of the model and shaping filters are shown in Fig. 4. The combined effect of shaping and interpolation filters is shown in Fig. 5. A summary of the computational cost of the IFIR and traditional FIR filters are shown in Table 2.

In this example the difference in performance is quite dramatic. The savings is almost 90% for the IFIR when compared to the traditional FIR. Comparing memory values calculated from data in Table 2, we can see that the memory requirement for the IFIR filter is $87 * 15 + 66 = 1371$. This is only a 5% increase (approximately) over the requirement of 1303 for the traditional FIR filter.

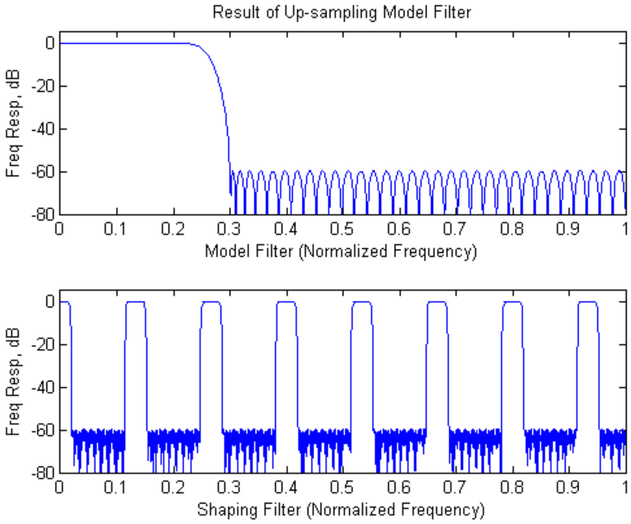

Fig. 4: Comparison of frequency responses of model and shaping filters.

TABLE 2
COMPARISON OF COMPUTATIONAL COST

Filter	*Order*	*Mult.*	*Add.*
Shaping	87	44	87
Interpolating	66	34	66
Combined	**153**	**78**	**153**
FIR	**1303**	**652**	**1303**

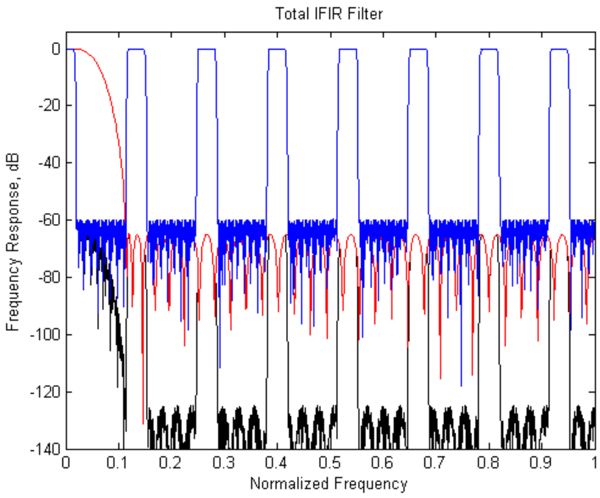

Fig. 5: Frequency responses of shaping filter (blue), interpolation filter (red), and combination (black).

III. Refinements of IFIR Design

A. Selecting an optimum stretch factor

By calculating and plotting the number of calculations required for a given stretch factor L, it is easy to determine the optimum value for a given filter specification. This has been done in Fig. 6 for the design example given above. From this plot it is easy to see that $L = 15$ is the optimum value for this filter.

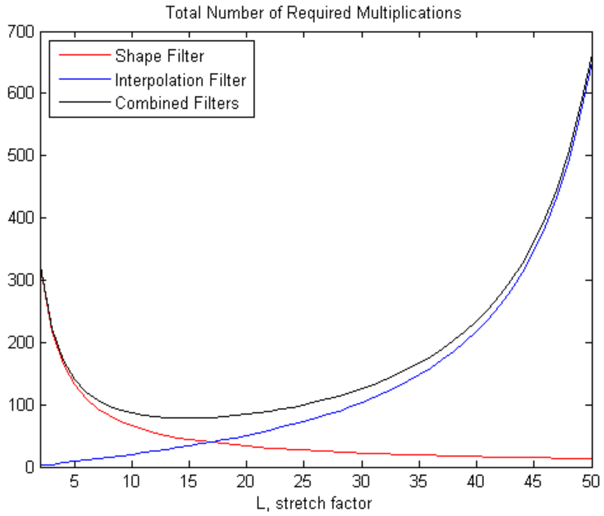

Fig. 6: Optimization plot for determining stretch factor L for lowest computation requirement.

It is also possible to determine the optimum value of L directly using a formula given in [8], which is then rounded to the nearest integer. This formula also yields a value of 15.

$$L_{opt} = \frac{2\pi}{\omega_p + \omega_s + \sqrt{2\pi(\omega_s - \omega_p)}} \tag{9}$$

B. Use of cascaded interpolation filters

It has been suggested in [2] and [8] that for particularly demanding designs, additional improvements can be made by implementing the interpolation filter as an IFIR filter itself. This results in having two (or more) cascaded filters following the shaping filter. Each of the interpolating filters in this case is designed to remove a subset of all of the images of the original passband. This is illustrated in Fig. 7.

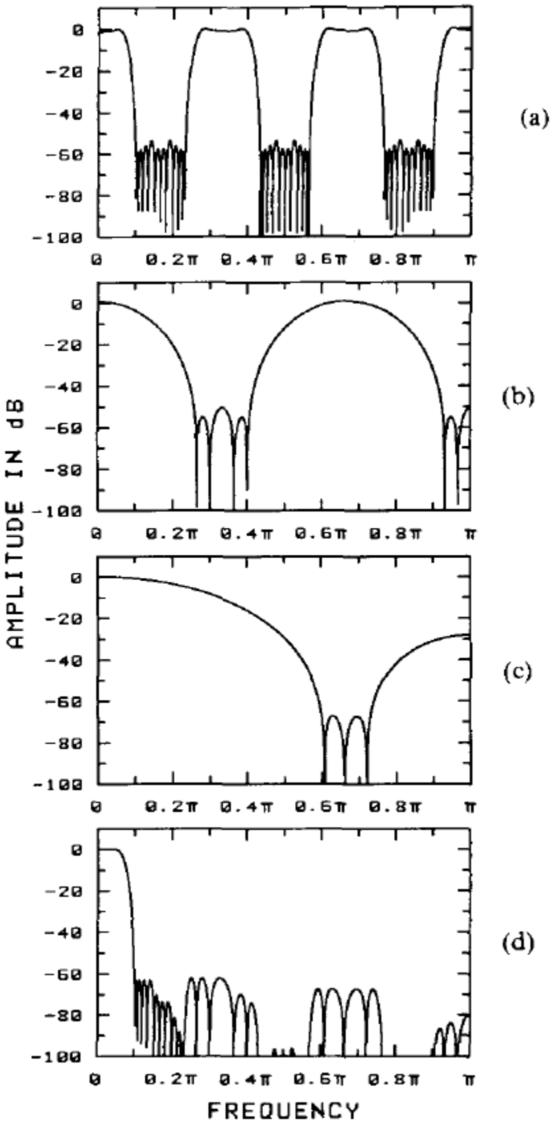

Fig 7: Example of cascaded interpolating filters. (a) Shaping filter, (b) First interpolating filter, (c) Second interpolating filter, (d) Frequency response of entire system. (image taken from [2]).

C. Periodicity of interpolation filter zeros

The observation has been made in [9] that since the required stopbands of the interpolation filter are periodic, it is possible to utilize the periodic properties of transfer functions in order to reduce the number of computations. A transfer function of the form:

$$H(z) = 1 - 2\cos(K\alpha)z^{-K} + z^{-2K} \tag{10}$$

has K zeros between 0 and π with pairs of zeros at $\frac{2\pi i}{K} \pm \alpha$ and only requires one multiplication. If zeros are occurring periodically, or nearly periodically in the target transfer function, (10) can be used to replace those terms with a transfer function that is more computationally efficient. The steps for using this method are:

1) *Design the filter using the McClellan-Parks algorithm or similar[5].*
2) *Examine where transfer function zeros occur nearly periodically on the unit circle.*
3) *Out of the selected zeros, find those that mostly affect the passband and transition band response.*
4) *Replace the zeros with a periodic series implemented via (10).*
5) *Implement any remaining zeros in a separate transfer function.*

IV. Conclusion

The IFIR method of designing linear-phase FIR filters can provide dramatic reductions in the number of computations required for filters that have a high order and narrow bandwidth. The design process was reviewed and two examples were shown, including an analysis of differences in computations and memory between the IFIR and the traditional FIR design. Additional improvements over the method described in [1] were also reviewed.

References

[1] 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.

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

[3] R. Lyons, “Interpolated narrowband lowpass FIR filters,” *IEEE Signal Process. Mag.*, vol. 20, no. 1, pp. 50-57, Jan. 2003.

[4] J.F. Kaiser, “Nonrecursive digital filter design using I_0-sinh window function,” in *Proc. 1974 IEEE Int. Symp. Circuits Syst.*, Apr. 1974, pp. 20-23.

[5] J. H. McClellan, T. W. Parks, and L. R. Rabiner, “A computer program for designing optimum FIR linear phase digital filters,” *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 506-526, Dec. 1973.

[6] D. G. Manolakis, V. K. Ingle, *Applied Digital Signal Processing*, New York, NY: Cambridge University Press, 2011, pp. 503-507.

[7] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*, ser. Prentice-Hall Signal Processing Series, A. V. Oppenhein, Ed. Englewood Cliffs, NJ: Prentice Hall, 1993.

[8] A. Mehrnia, J. A. Willson, “On optimal IFIR filter design,” *IEEE Proceedings of the 2004 International Symposium on Circuits and Systems*, vol. 3, pp. 133-136, May 2004.

[9] J.L. Webb, D. Munson “A new approach to designing computationally efficient interpolated FIR filters” *IEEE Trans. Signal Processing*, vol. 44, pp. 1923-1931, Aug. 1996.