An Analysis of Interpolated Finite Impulse Response Filters and Their Improvements

Richard S. Juszkiewicz, *Member, IEEE*

Abstract— **This paper offers a brief overview of Interpolated Finite Impulse Response (IFIR) filters followed by a comprehensive and analytical literary survey of the improvements to the original design that have been made to reduce their computational complexity. Enhancements to the original design that are theoretically examined and compared are: stretching factor (L) optimization, arithmetic operation reduction, mipizing, alternate interpolator designs and coefficient re-quantization. Design examples are presented to accompany explanation and offer comparisons between various cases when applicable.**

***Index Terms*—IFIR, Optimization, Improvements**

I. Introduction

It is well known that Finite Impulse Response (FIR) filters have been long favored over Infinite Impulse Response (IIR) filters under certain circumstances due to their guaranteed stability and linear-phase behavior. However, the aforementioned desirable characteristics are valued at the expense of increased computational complexity. Standard FIR filters generally require twice as many multipliers and delays as their IIR counterparts. The width of the filter's transition band is inversely proportional to the number of calculations required so, logically, as the design specifications become more stringent, the effects of quantization noise also inevitably increase. In [1] an elegant method was introduced to drastically reduce these undesirable effects while maintaining filter stability. This technique, known as Interpolated Finite Impulse Response (IFIR) filters, along with the history of its advancements, is the focus of this paper.

II. The IFIR Filter

Two relatively simple concepts were combined in the invention of the IFIR filter: upsampling and interpolation. As shown in Fig. 1, the original IFIR filter design [1] consists of a model filter $H_M(z^L)$ upsampled by a factor of L that is cascaded with an interpolation filter labeled $G(z)$.

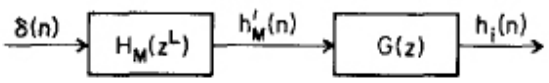


Fig. 1. The Interpolated Finite Impulse Response Filter [1]

The upsampling process inserts L-1 zero samples between each impulse response sample of the model filter in the time domain. This narrows the model filter's passband and creates undesired replicas of the passband in the frequency domain. The function of the interpolation filter is to fill in all of the zero samples and supress all unwanted images from the model filter. This implementation, when compared to an FIR filter with common design criteria, results in a reduction of the number of multipliers by a factor of approximately 1/L [1]. In order to avoid aliasing the max value of L is given in [1] to be

$$L_{MAX} = \left\lfloor \frac{\pi}{\omega_s} \right\rfloor \tag{1}$$

where ω_s is the desired stopband edge frequency. No further criteria for the choice of L is given in (1) so it can be arbitrarily chosen as long as it is less than L_{MAX}. Further research in [2] suggests that cascading multiple interpolator stages with lesser L values yields even greater computational benefits.

III. Design Specifications & Initial Design

Throughout the remainder of this paper the following low pass filter specifications will be used to compare various design methods

$$\begin{aligned} \delta_p &= \delta_s = .001 \\ \omega_p &= .015\pi \\ \omega_s &= .020\pi \end{aligned} \tag{2}$$

For reference purposes standard FIR and IFIR filters are designed using (2) and the methods outlined in [3] and [1] respectively. Using Kaiser's formula [3], and the kaiserord function in Matlab for confirmation, the approximate order of a conventional FIR filter with the design specifications given in (2) is calculated to be 1451. A plot of this filter is shown in Fig. 2

The amount of adders in this filter is equal to the order: 1451 while the number of multipliers in this design is found to be 726 by using (3) where M is the number of multipliers and N is the order of the filter.

$$M = \left\lfloor \frac{N+1}{2} \right\rfloor \tag{3}$$

Manuscript received November 14, 2005.

R. S. Juszkiewicz is a Graduate Student in the Music Engineering Technology Department at the University of Miami in Coral Gables, FL 33142 USA, (e-mail: r.juszkiewicz@umiami.edu).

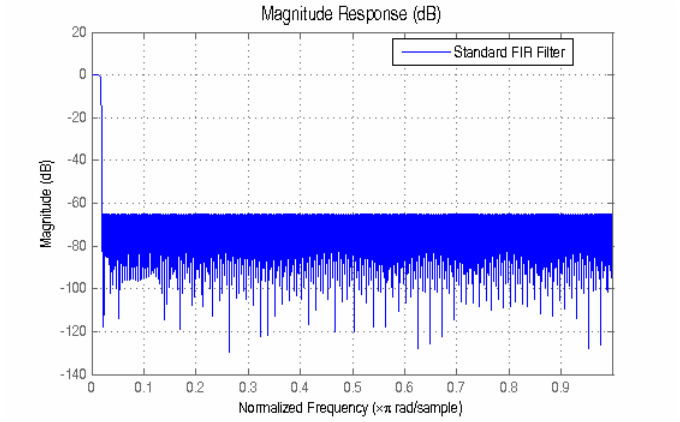
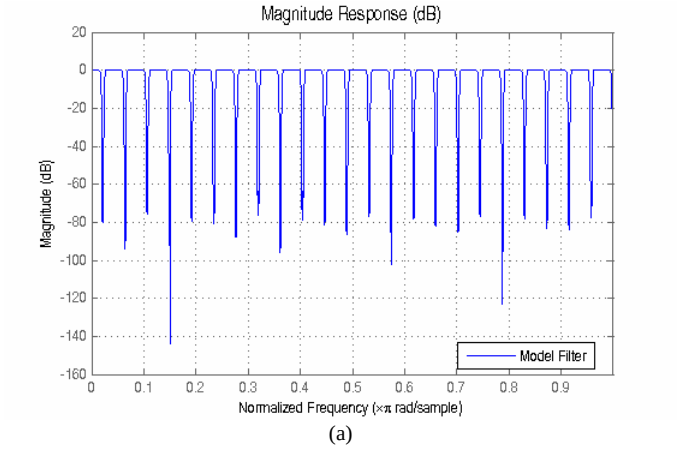


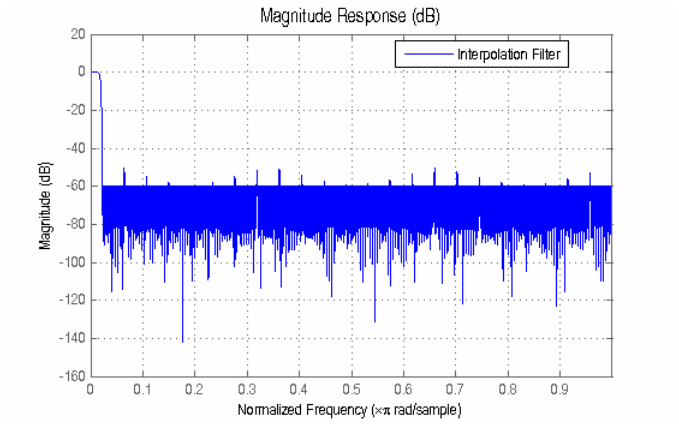
Fig. 2. The magnitude response of the FIR filter with order 1451 based on (2).

An IFIR filter meeting the same specifications is now designed using the procedure outlined in [1]. An L value of 47 is chosen to satisfy (1). The model filter is designed to have passband and stopband edge frequency values equal to L times the (normalized) desired values of .015 and .020, respectively, to account for the predictable shrinking of the transition band that occurs with the upsampling process. This model filter has an order of 30 and is shown in Fig. 3a. The model filter is then cascaded with an interpolation filter possessing a 3dB cutoff frequency of .0180 on the normalized abscissa to eliminate the unwanted images present in the model filter. Since a rather large value of L was chosen the order of the interpolation filter is 193. The image suppression filter is shown in Fig. 3b and the final cascaded filter (the IFIR filter) is shown in Fig. 3c.

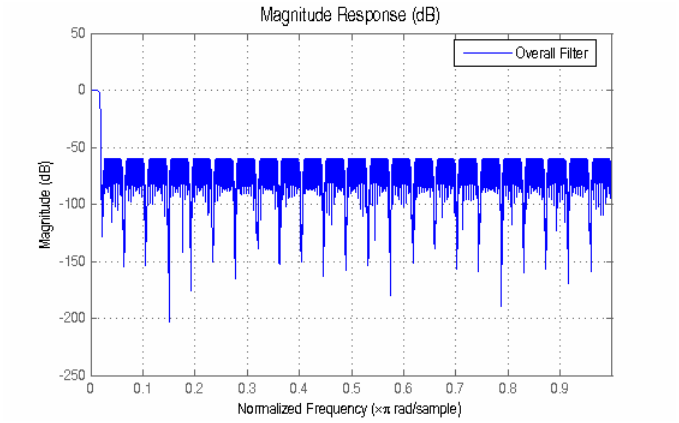
Notice that the stop band and attenuation of the IFIR filter and the FIR filter are identical. The total number of multipliers needed in the IFIR implementation is the sum of the number of multipliers required at each of the two stages. Using (3) the total number of multipliers in this design is calculated to be 121. This is drastically less than the 726 multipliers that are required for the standard FIR design shown in Fig. 2. The number of adders is equal to 223, also much less than that of the FIR filter. In the last twenty years copious new methods to further improve the efficiency of IFIR filters have been introduced and these techniques are the focus of the remainder of this paper.



(a)



(b)



(c)

Fig. 3. The output after each IFIR stage. (a) Model filter output. (b) Interpolator output. (c) Final IFIR output.

IV. Improvements

A. *Interpolation Using Cyclotomic Polynomials*

One of the most notable early methods of improving IFIR filters is introduced in [4] by proposing a new interpolator architecture utilizing cyclotomic polynomials. Prior (and more simplified) methods for interpolator design result in the amount of multiplications required to interpolate being inversely proportional to the stretching factor (L); this is not the case when the properties of cyclotomic polynomials are exploited for use in the interpolator. The first few cyclotomic polynomials [5] are:

$$\begin{aligned} C_1 &= x-1, \\ C_2 &= x+1 \\ C_3 &= x^2+x+1 \\ C_4 &= x^2+1 \end{aligned} \tag{4}$$

Since these polynomials never have any coefficients they are essentially multiplier-less. When converted into the transform domain none of the first 24 polynomials require more than ten additions or twelve delays. The plot in Fig. 4

illustrates the locations of zeros and the suppression bands for the first 24 cyclotomic polynomials.

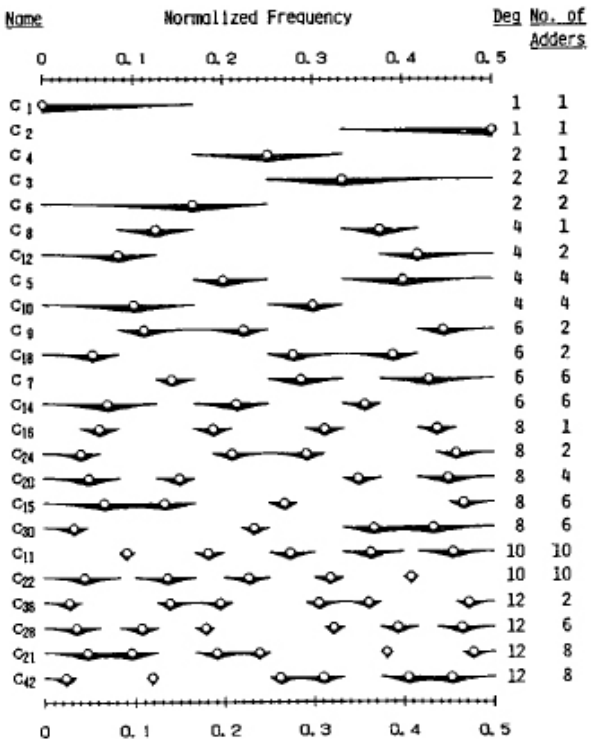


Fig. 4. The zeros of the first 24 cyclotomic polynomials and their suppression bands.

To successfully design an interpolator based upon Fig. 4 it is necessary to cascade multiple cyclotomic polynomials together. This cascading does increase the number of delays and adders needed but the number of multipliers necessary remains zero. One trade off of this design is that the model filter must account for the passband distortion that is created by such an interpolator such problems can be handled with the aid of computer software.

This design allows for large values of L to reduce the order of the model filter without increasing the complexity of the interpolation filter. In the design example introduced in the previous section C_{15}, C_3 and C_4 could be used as part of the interpolator to suppress the unwanted images in the model filter. More cyclotomic polynomials may be needed but the number of adders necessary for a successful interpolation filter will surely be less than 193 and since the number of multipliers in the interpolator will be zero this method does significantly reduce the amount of arithmetic required to realize the given IFIR filter.

B. Interpolator Design Using B-Spline Functions

In a method similar to that of cyclotomic polynomials [6] uses the uniform B-Spline function to form the interpolation filter and create a multiplier-less solution. Once again, the maximum value of L is used when designing the model filter since the order of the interpolator is not of concern. Based on the value of L a design sub-interval is established. Only using this established subinterval for the remaining calculations facilitates the rest of the design procedure by eliminating irrelevant arithmetic. From this subinterval the Chebyshev approximation is used to find the solution of the upsampled model filter thus establishing the criteria for the interpolation filter.

The improvement in computation efficiency with this method is similar to that of cyclotomic polynomials in that there are no multipliers required in the interpolator.

C. Sigma-Delta Coefficient Truncation

All digital filters, whether they are implemented in hardware or software, are a series of unique coefficients that are stored in a finite amount of memory. The coefficient's binary wordlength determines the complexity of the filter's arithmetic operations and memory demands. Reference [7] proposes using Sigma-Delta coefficient truncation in conjunction with powers-of-two filter design algorithms to reduce the amount of memory needed to store IFIR filter coefficients without increasing quantization noise in the process.

Through the use of feedback loops and a canonic signed-digit (CSD) code with three possible values for each bit (-1,0,1) the effects of re-quantizing to a lower level are shaped to not affect the low frequency data present in the signal. Since the low frequency data in the model filter is the only data of interest in IFIR filters this method proves to be suitably effective.

It is shown in [7] that 10-bit representations of numbers using CSD can be truncated down to two bits and still be able to represent 148 different levels of quantization between -512 and 512. This decreases the number of multipliers needed in conventional IFIR implementations by a considerable amount in each of the examples given in [7]. Perhaps if this method of coefficient quantization was used in conjunction with one of the previously mentioned multiplier-less interpolators there would for an even greater decrease in the overall number of multipliers.

D. Simplifying the Design of Minimum Phase IFIR Filters

The method presented in [8] is an improvement to IFIR filters that differs from every improvement that has been addressed thus far. Instead of a reduction in an IFIR's arithmetic operations being proposed a process to simplify the calculations involved in minimizing the phase of the final IFIR filter is presented. Prior to IFIR filters if one desired to minimize the phase of a very high order FIR filter it was necessary to arduously calculate all of the zeros of the filter's lengthy transfer function and then bring inside all of the zeros that were outside of the unit circle in the imaginary plane. This process of "bringing in the zeros" is referred to as mipizing.

It is presented in [8] that if the model filter and interpolation filters are both mipized prior to upsampling and

cascading then the resultant filter will also be minimum phase. Mipizing the two substantially lower order filters prior to cascading them is radically easier than mipizing the final filter. Since IFIR filters are not inherently minimum phase this procedure allows for a simpler mipizing process if minimum phase is a desired criterion of the filter.

E. Selecting an Optimal L Factor

One of the initial calculations in the design of IFIR filters is of L. Using (1) a maximum value of L is attained; however, the optimal value of L is somewhere in between two and L_{MAX}. Since L is proportional to the number of multipliers needed in the interpolation filter optimizing its value is critical. In [8] a process for calculating an optimal L value is addressed. The first step in optimizing L is to calculate L_{MAX}. In this paper the equation is presented in a slightly different way from (1) but both equations yield a solution of 50. Once the maximum value of L is computed the reduction of computation can be calculated for every value of L between 1 and L_{MAX} using (5).

$$\left\lfloor \frac{L-1}{L} - \frac{L(f_s - f_p)}{1 - L(f_s - f_p) - 2Lf_p} \right\rfloor \quad (5)$$

In (5) f_s and f_p are the values of ω_s and ω_p in Hz. The results of this equation are shown in Fig. 5 where computational reduction (as a percent) is plotted against L.

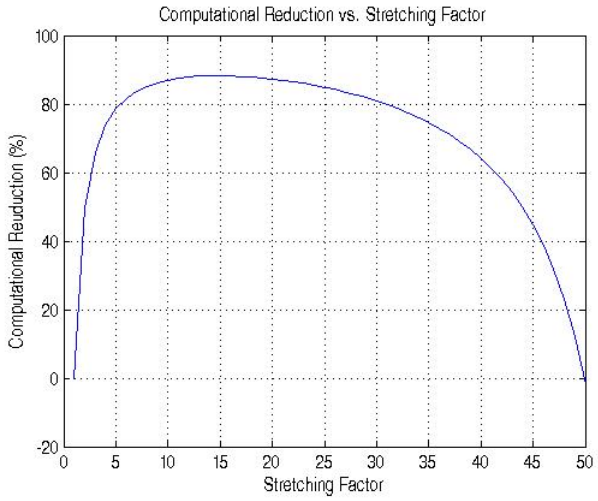


Fig. 5. Plot of computational Reduction vs. Stretching Factor (L).

The optimal value of L is the value on the x-axis that corresponds to the peak of the plot. It may not be completely obvious from Fig. 5 but the Matlab routine that was written to generate this plot outputted 15 as the optimal value of L before displaying the graph.

Using L=15 and the previously mentioned design specifications an IFIR filter was designed. Its final output is shown in Fig. 6.

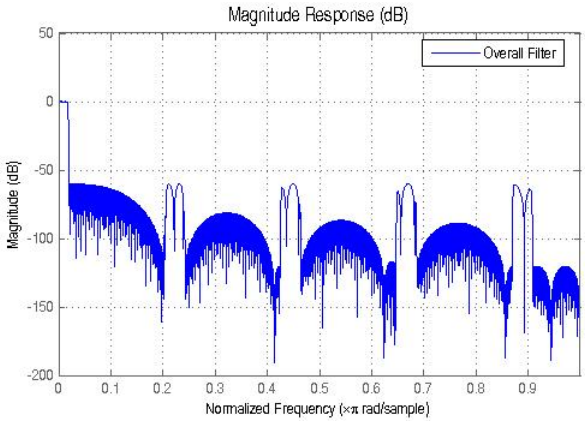


Fig. 6. Magnitude response of the IFIR with L = 15.

The magnitude response of this IFIR filter only differs from Fig. 3c in the shape of the passband ripples which is a trivial matter. Fig. 6 and Fig. 3c have identical transition bands is and attenuations which makes this design a success. The order of the model filter is 96 and the order of the interpolation filter is 56 which means 76 multipliers and 152 adders are needed to realize the entire filter. Both of these values offer significant improvements over the values that were calculated in the previous section however, the multiplier-less interpolator methods presented in *A* and *B* of this section can potentially offer less multipliers. The design in this section can be entirely automated and is less complicated than the multiplier-less solutions.

F. Upsampling the Interpolator

One of the most recent advancements in the field of IFIR digital filters is creating an IFIR structure out of the interpolation filter. This is introduced in [10] and shown in Fig. 7.

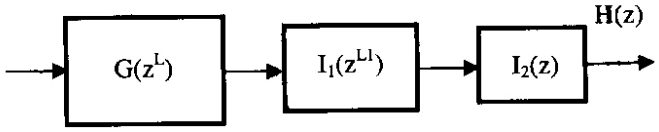


Fig 7. An IFIR filter using an IFIR implementation on the first interpolation filter (I_1).

As evident in Fig. 7 two values of L (L and L_1) are required for this design and they are dependent on each other. This dependence leads to a pair of non-linear simultaneous equations (6a) that must be solved to obtain optimal values for both L and L_1. The design and implementation of this method is rather complex. Once L and L_1 are obtained the model filter is designed based on L. I_1 is then designed to suppress the nearest undesired images created by the upsampling of G(z) by a factor of L. Since I_1 is upsampled by a factor of L_1 as part of its own IFIR structure it has unwanted images associated with it as well. These images are attenuated by $I_2(z)$, the final interpolator. The stopband and passband edges

for each interpolation filter are defined in (6b) and (6c).

In (6) ω_s and ω_p are the original specifications given in the previous section, ω_{s1} and ω_{p1} are the stopband and passband edge frequencies respectively for $I_1(z)$, and ω_{s2} and ω_{p2} are the respective edges for $I_2(z)$. L_1 and L_2 are the values obtained from solving the two equations in (6a) simultaneously.

$$\frac{-1}{\omega_s-\omega_p}+\frac{2\pi}{L_1\left[\left(\frac{2\pi}{L}-\omega_s\right)-\omega_p\right]^2}-\frac{2\pi}{\left[\frac{2\pi}{L_1}-\left(\frac{2\pi}{L}-\omega_s\right)-\omega_p\right]^2}=0$$

$$\frac{-1}{\left[\left(\frac{2\pi}{L}-\omega_s\right)-\omega_p\right]}+\frac{2\pi}{\left[\frac{2\pi}{L_1}-\left(\frac{2\pi}{L}-\omega_s\right)-\omega_p\right]^2}=0$$

(a)

$$\omega_{p1}=L_1\omega_p$$

$$\omega_{s1}=.L_1\left(\frac{2\pi}{L-\omega_s}\right) \qquad (6)$$

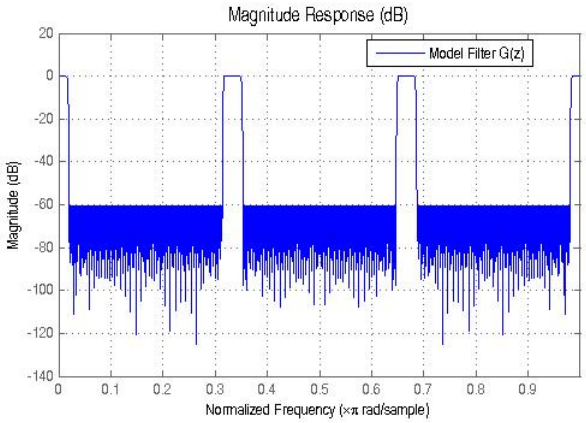
(b)

$$\omega_{p2}=\omega_p$$

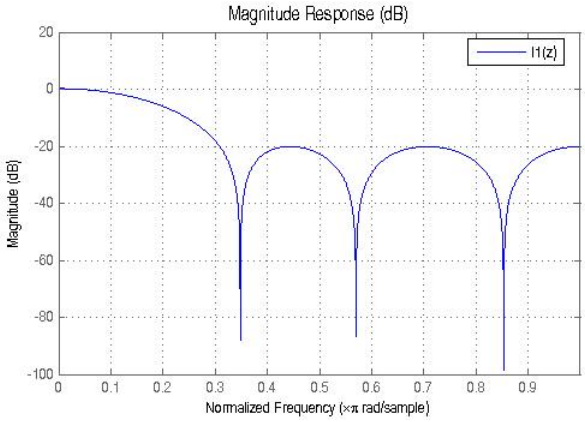
$$\omega_{s2}=.\frac{2\pi}{L_1}-\left(\frac{2\pi}{L-\omega_s}\right)$$

(c)

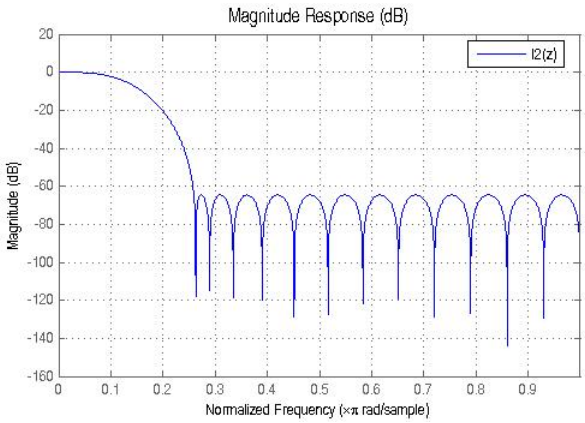
For the specifications given in the previous section of this paper L_1 was calculated to be 6 and L to be 27. These L values are rounded values so actual optimal values may vary slightly. Using L_1 and L, the three filters in Fig. 7 were designed and are shown in Fig. 8 along with the final IFIR filter.



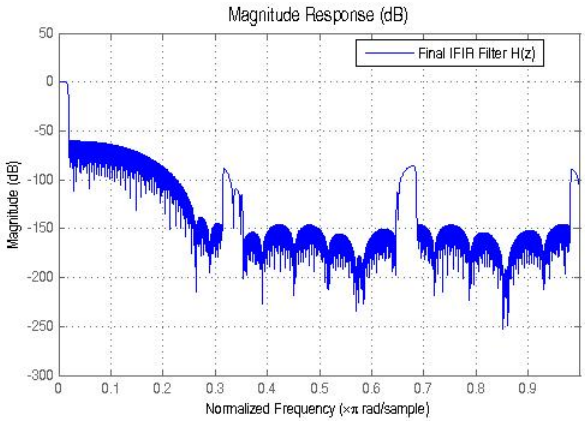
(a)



(b)



(c)



(d)

Fig. 8. The output after every stage of the IFIR filter design using an IFIR for the interpolator. (a) The model filter G(z). (b) $I_1(z)$. (c) $I_2(z)$ (d) The final output of the three cascaded stages H(z).

The order of the model filter is 242, the order of $I_1(z)$ is 40 and the order of $I_2(z)$ is 65 which gives the total number of multipliers to be 174 and the total number of adders to be 347. This method, while it is a novel idea, uses more multipliers than most of the methods previously discussed in this paper. The author of [10] mentions a three-variable implantation may be even more beneficial but does not offer any evidence. Such a procedure would involve the solving of three non-linear simultaneous equations thus further complicating the design process. A plot is also supplied in [10] to illustrate when this method is most beneficial and the design

specifications in (4) fall into the range of the plot that is described to benefit most from this method; however in a two variable design it does not offer any innovative improvements.

V. Conclusion

Since IFIR filters were first introduced to the digital signal processing community in 1984 many optimization methods have been proposed. Some of these methods were analyzed in this paper either theoretically or with design examples. The combination of one or more of these methods may provide the most reduction in the complexity and amount of mathematics required to realize these filters thus leaving the doors open for the continuing of research in the area of IFIR filters.

Acknowledgment

The author would like to thank the creators of the Matlab software package for providing many beneficial filter design and digital signal processing tools.

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. Circuis Syst.*, vol, CAS-35, pp. 70-88, Jan. 1998.

[3] S. K. Mitra, *Digital Signal Processing: A Computer Based Approach*. New York, NY: McGraw Hill, 1998, pp427-589.

[4] H. Kikuchi, H. Watanabe and T. Yanagisawa, "Interpolated FIR filters using cyclotomic polynomials," Trans ISACS, pp 2009-2012 August 1988.

[5] E. W. Weisstein. "Cyclotomic polynomial." From *MathWorld*—A Wolfram Web Resource http://mathworld.wolfram.com/CyclotomicPolynomial.html

[6] D. Pang, L. A. Ferrari and P. V. Sankar, "A unified approach to IFIR filter design using B-Spline functions," *IEEE Trans. On Signal Processing*, vol. 39. no. 9, pp. 2115-2217, September 1991.

[7] S. R. Powell and P. M. Chau, "Efficient narrowband FIR and IFIR Filters based on powers-of-two sigma-delta coefficient truncation," *IEEE Trans. on Circuits and Systems—Analog and Digital Signal Processing*, vol. 41, no. 8, pp.497-505, August 1994.

[8] G. Jovanovic-Dolecek and J. J.D. Carmona, "Lowpass minimum phase filter design using IFIR filters," *Electronics Letters*, vol. 33 no. 23, pp. 1933-1935, November 1997.

[9] R. L. Lyons, "Interpolated narrowband lowpass FIR filters," *IEEE Signal Processing Magazine*, vol 1053-5888, pp. 50-57, January 2003.

[10] A. Mehrnia and A. N. Wilson, Jr., "On optimal IFIR filter design." *IEEE ISCAS 2004*, pp. III-133-III136, 2004.