# A Survey About IFIR Filters and Their More Recent Improvements

Felipe Espic, *Graduate Student, Music Engineering Technology*

***Abstract*—The overall characteristics and drawbacks of Finite Impulse Response (FIR) design are described. The Interpolated Finite Impulse Response (FIR) filter and its design steps are introduced and studied by using the basic two stages structure. A survey of the more recent improvements in IFIR filters is realized using an example to illustrate and compare results between methods. The methods studied are *Optimal Stretching Factor Design, IFIR Filter with Two Optimal Stretching Factors, Multirate Multistage IFIR Design, Decimation Techniques, Multiplier-Free.****

***Index Terms*— IFIR, FIR, multirate, narrowband, optimization.**

## I. Introduction

In certain applications, it is preferable to use Finite Impulse Response filters (FIR) instead of their counterpart, the Infinite Impulse Response (IIR), because of its advantaged impulse response basically. Specifically, using FIR filters allows us to get control over the phase and magnitude response independently and it also ensures the filter stability. On the contrary, the main disadvantage of this type of filtering is its high computational complexity, which primarily means a large number of multipliers [12].

Because of these reasons, the research about FIR filters and specifically how to reduce the computational complexity is quite studied, even more than any other type of filter [7]. Within the approaches to address this matter, there is one called Interpolated Finite Impulse Response (IFIR), which was introduced in [13].

Generally, the basic analysis of filter design is based on the typical low pass filter and its proper specifications, since this then can be applied on several filter types like band pass, high pass, notch filter, multiband only by using and modifying a complementary filter.

We could define certain filter specification to meet. These are the maximum ripple in the pass band $\delta_p$, the maximum ripple in the stop band $\delta_s$ and their respective edge frequencies $\omega_p$ y $\omega_s$.

Let us suppose that there is a FIR filter $H_M(z)$ and its impulse response $h_M(n)$. Then, we could insert *L-1* zeros between each consecutive sample in order to upsample it by a stretching factor *L* (see Fig. 1).

$$h'_M(n) = \begin{cases} h_M\left({}^{n}/_{L}\right), & {}^{n}/_{L} = 0, \pm 1, \pm 2, \ldots \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Besides, the value of *L* is limited by the following expression:

$$L_{MAX} = \left\lfloor \frac{\pi}{\omega_s} \right\rfloor \quad (2)$$

This ensures that the value of $\omega_s$ of $H_M(z)$ is lower than π, therefore realizable. Then, we could construct a new filter $H_{IFIR}(z)$ as a cascade structure formed by two filters:

$$H_{IFIR}(z) = H_M(z^L)G(z) \quad (3)$$

Where $H_M(z^L)$= $H'_M(z)$ and $G(z)$ is an interpolator filter.

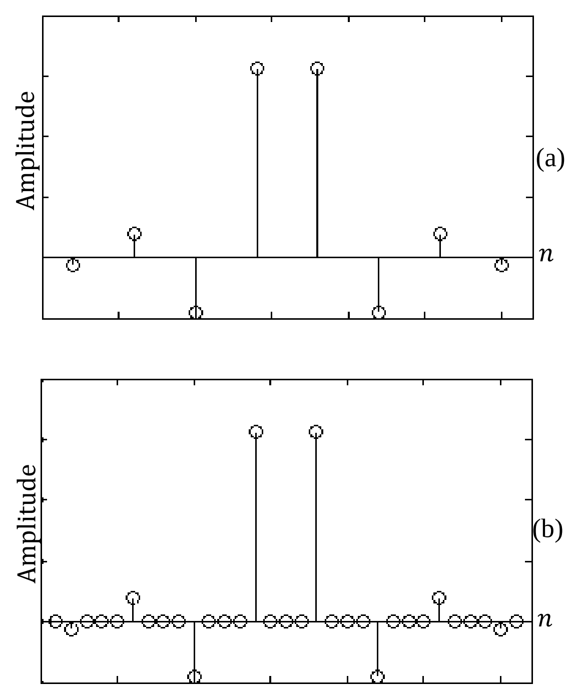


Fig. 1. Schematic comparison of the original impulse response $h_M(n)$ (a) and its upsampled version $h'_M(n)$ (b) stretched by a factor L = 4.

Hence, the magnitude response of the new filter IFIR $H_M(z)$ must meet the following conditions:

$$1 - \delta_p \leq |H_M(\omega L)G(\omega)| \leq 1 + \delta_p \;, \qquad 0 \leq \omega \leq \omega_p \quad (4)$$

$$|H_M(\omega L)G(\omega)| \leq \delta_s \;, \qquad \omega_s \leq \omega \leq \pi \quad (5)$$

It is noticed that the magnitude response of $H_M(\omega L)$ is periodic, whose period is $2\pi/L$.

To satisfy the specifications, we could use several techniques to generate the interpolator $G(z)$. However, one of the most efficient techniques is the Parks-McClellan algorithm [4], since it generates lower filter-length and therefore, less multipliers.

### *A. Design Steps*

We can summarize the method to obtain an IFIR filter in three steps:

1) Choose a value of $L$ that meets (2).
2) Compute $G(z)$ to attenuate the undesired pass bands proper of the periodic filter $H_M(z^L)$.
3) Compute $H_M(z)$ multiplying the required edge frequencies $\omega_p$ and $\omega_s$ by $L$.

To illustrate the IFIR implementation, we have considered the following example:

$$\omega_p = 0.015\pi, \quad \omega_s = 0.020\pi\,, \quad \delta_p = 0.001, \quad \delta_s = 0.001 \qquad (6)$$

One conventional FIR filter realization using the efficient Parks-McClellan algorithm [4] to meet the specifications (6), consists in 652 multipliers and 1303 adders. It means a high computational complexity, which could be reduced by using the basic IFIR filter technique as follows.

Firstly, we choose $L$=4, since $L_{MAX}$=50, although we could choose any value between 2 and 50. Thus, we obtain the filter shown in Fig. 2.

The overall complexity of this IFIR realization involves 181 multipliers and 360 adders, which means a reduction at about 72% comparing with the conventional Parks-McClellan realization. No doubt, this new technique improves the performance considerably. The reasons of this improvement are briefly explained in the next section.

### *B. Computation Savings*

According to [13], the amount of multipliers that form the filter $H_M(z)$ and the new filter $H_{IFIR}(z)$ are almost the same, since $H_M(z)$ and $H_M(z^L)$ have the same number of non-zero coefficients and $G(z)$ provides more multipliers to the new filter, although this amount should not be considerable because of $G(z)$ should be a low order filter. However, this comparison does not yield any useful result, since the frequency response of $H_M(z)$ and $H_{IFIR}(z)$ are completely different. In fact, the pass and transition band widths of the new filter $H_{IFIR}(z)$ are $L$ times narrower than $H_M(z)$. It means that we should compare the number of multipliers of the new filter and an equivalent FIR filter.

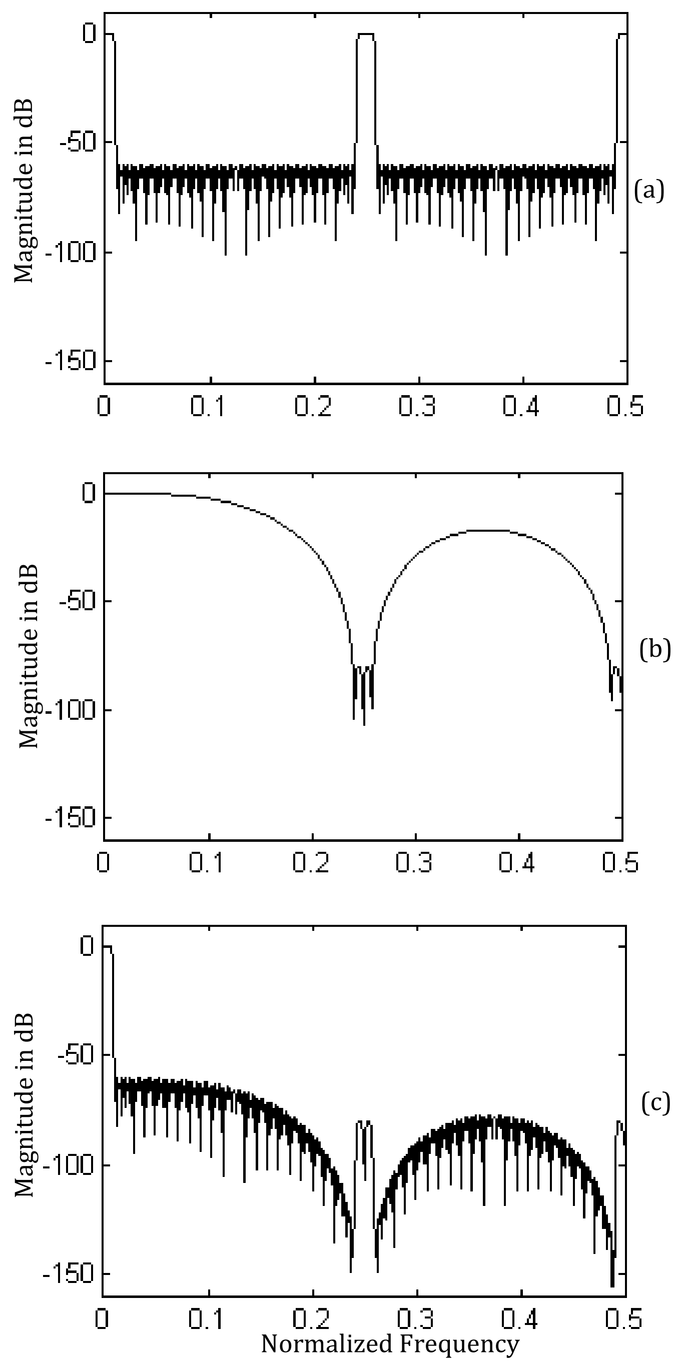


Fig. 2. Example of an IFIR LPF design with $L$=4 meeting the specifications $\omega_p$ = 0.015π, $\omega_s$ = 0.020π , $\delta_p$ = 0.001, $\delta_s$ = 0.001. Frequency responses of (a) $H_M(z^L)$, (b) $G(z)$, (c) the overall filter $H_{IFIR}(z)$.

The Kaiser's formula to estimate the order $N$ of a conventional FIR filter is [8]:

$$N \cong \frac{-20\log_{10}\left(\sqrt{\delta_p\delta_s}\right)-13}{14.6(\omega_s-\omega_p)/2\pi} \quad (7)$$

If the FIR filter were direct form and linear phase, the number of its multipliers $Mul_{FIR}$ is [1]:

$$Mul_{FIR} = \left\lceil\frac{N+1}{2}\right\rceil \quad (8)$$

Besides, the number of multipliers $Mul_{IFIR}$ of the new filter $H_{IFIR}(z)$ is given by:

$$Mul_{IFIR} = \left\lfloor\frac{N_L+1}{2}\right\rfloor + Mul_{INTER}, \qquad N_L = {}^{N}/_{L} \quad (9)$$

Where $Mul_{INTER}$ is the number of multipliers provided by the interpolator $G(z)$ and $N_L$ is the filter order estimator for $H_M(z^L)$ [12]. Then, we could relate the respective number of multipliers:

$$Mul_{IFIR} \approx \frac{Mul_{FIR}}{L} \quad (10)$$

It is noticeable the computation saving that is carried out by using IFIR filtering. Besides, as we can see in (10), it is supposed that the higher is $L$; the lower value of $Mul_{IFIR}$ should be. Nevertheless, with a higher $L$, the complexity of $G(z)$ increases, which means that the relation between $L$ and $Mul_{IFIR}$ is not linear, fact that is studied on the next section.

The number of adders is also an important factor to evaluate the computational complexity of filters. We can relate the respective number of adders as [13]:

$$A_{IFIR} = \frac{A_{FIR}}{L} + A_{INTER} \quad (11)$$

Therefore, this is a similar case as the multipliers. The number of adders should decrease at about $L$ times. Nevertheless, the price paid for this savings is that the number of delays is increased slightly [12].

On the other hand, according to (9) the filter order also decreases at about $L$ times, which is significant for multipliers and adders savings; but it is also relevant for the time domain response, decreasing the ringing effect and latency.

## II. Improvements

### A. Optimal Stretching Factor

As mentioned, the relation between the number of multipliers of the IFIR filter and the stretching factor $L$ is not linear. If we take the example addressed in Fig. 2, we can compute the number of multipliers for various values of $L$ (see Fig. 3).

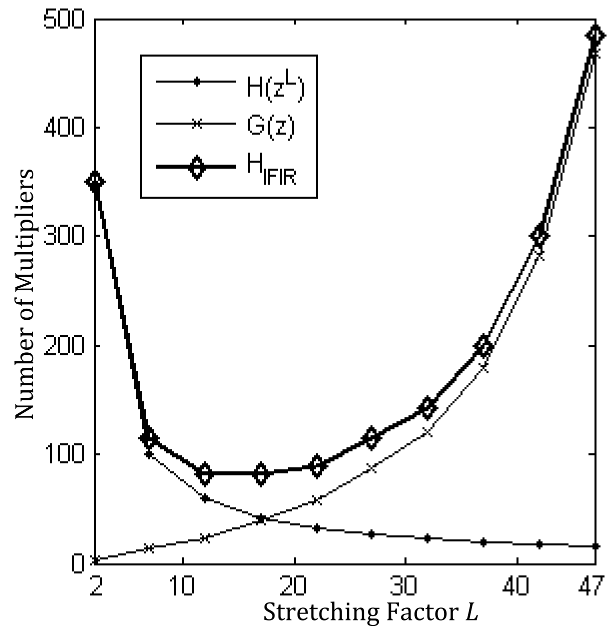


Fig. 3. Number of multipliers needed by $H_M(z^L)$, $G(z)$, and $H_{IFIR}(z)$ versus the stretching factor $2<L<47$ for the example shown in Fig. 2.

It is noticed that there is a minimum at about $L$=17. The relation between $L$ and the total number of multipliers behaves in the same way generally. In other words, it behaves as an unimodal function that is possible to minimize. Hence, a function to estimate the optimal stretching factor $L_{OPT}$ is the following [1]:

$$L_{OPT} = \frac{2\pi}{\omega_p+\omega_s+\sqrt{2\pi(\omega_s-\omega_p)}} \quad (12)$$

Thus, if the above expression is used on the previous studied example, we obtain an optimal stretching factor $L_{OPT}$ = 15, which is very close to what we see in Fig. 2.

The IFIR realization for the specifications (6) with the optimal stretching factor $L_{OPT}$ = 15, involves 75 multipliers and 147 adders yielding a computational complexity reduction at about 88%.

### B. Modified Basic IFIR Filter

There is an approach [12] which is focused in modifying the basic conditions defined for IFIR filters (4) and (5) as follows:

$$1-\delta_p \le |H_M(\omega)G(\omega L)| \le 1+\delta_p, \quad 0 \le \omega \le L\omega_p \quad (13)$$

$$|H_M(\omega)G(\omega L)| \le \delta_s, \quad L\omega_s \le \omega \le \pi \quad (14)$$

$$G(0) = 1 \quad (15)$$

$$|H_M(\omega L)G(\omega)| \leq \delta_s\,, \qquad \omega \in R_\omega \quad (16)$$

Where $R_\omega$ is the union of all the portions of the spectrum where the undesired pass bands are located. One of the most relevant factors that increases the number of multipliers is the large order of the shaping filter $H_M(z)$. On the contrary, the order of the interpolator filter is always very small, due to the simplicity of the task that has to carry out. Thus, this approach focuses on sharing the complexity of the overall filter between its two stages in a better way, allowing the shaping filter to get a flatter response.

Hence, these conditions make possible to decrease the number of multipliers needed by the filter. A good example to illustrate its performance is the one of the Fig. 2, the result is shown in Fig. 4.

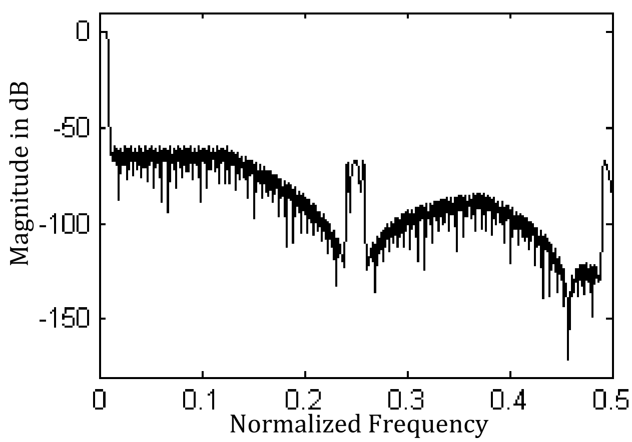


Fig. 4. The overall filter $H_{IFIR}(z)$ using the modified basic IFIR approach with a stretching factor $L$=4. It is noticed the difference in shape with Fig. 2 (c).

### C. *Generalized IFIR Filter with Two Optimal Stretching Factors*

Mehrnia and Willson [1] have proposed a new approach, which consists in to expand the filter in three cascade stages, instead of the basic design with only one shaping filter and its respective interpolator. Specifically, this new type is focused on dividing the interpolator in two stages, each one tuned on its optimal stretching factor.

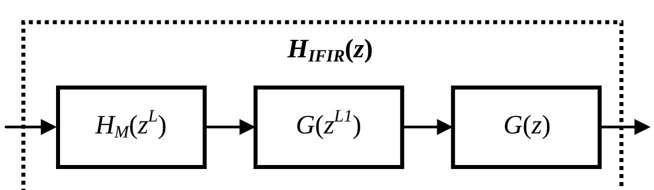


Fig. 5. Basic structure of an IFIR filter with two optimal stretching factors.

To find the optimal $L$ and $L_1$ they propose two expressions that minimize the number of multiplications, which may be computed numerically.

$$\frac{-1}{(\omega_s - \omega_p)} + \frac{2\pi}{L_1\left[\left(\frac{2\pi}{L} - \omega_s\right) - \omega_p\right]^2} - \frac{2\pi}{\left[\frac{2\pi}{L_1} - \left(\frac{2\pi}{L} - \omega_s\right) - \omega_p\right]^2} = 0 \quad (17)$$

$$\frac{-1}{\left[\left(\frac{2\pi}{L} - \omega_s\right) - \omega_p\right]} + \frac{2\pi}{\left[\frac{2\pi}{L_1} - \left(\frac{2\pi}{L} - \omega_s\right) - \omega_p\right]^2} = 0 \quad (18)$$

Using these formulas for the specifications (6), they yield $L_{OPT}$ = 27 and $L_{1OPT}$ = 6. However, with the intention of confirming and illustrating these results, a matrix that contains the total number of multiplications of every possible combination between $L_{OPT}$ and $L_{1\,OPT}$ , was generated and illustrated in the Fig. 6. Besides, in order to pick suitable *L's*, it was necessary to meet the condition (2) and for *$L_1$'s* the following condition was determined:

$$L_1 \leq \frac{\pi L}{2\pi - \omega_s L} \quad (19)$$

As a result, the minimum number of multipliers was 50, which was found in two consecutives sets of combinations; for $L_1$ = 7, 25 ≤ $L$ ≤30 and $L_1$ = 6, 24 ≤ $L$ ≤ 29. These results confirm the accuracy of the previous estimation based on (17) and (18). Finally, the computational complexity was reduced at a surprising 92%.

The minimum number of adders is localized at the same region, since these are proportional to the number of multipliers[1]. For the optimal stretching factors, the total number of adders was 93.

### A. *Multirate Multistage IFIR Design*

Considering the interpolator $G(z)$ , we could implement a multistage filter as a multirate cascade structure [5]:

$$G(z) = G_1(z)G_2(z^{L_1})G_3(z^{L_2}) \ldots G_K(z^{L_{K-1}}) \quad (20)$$

The value of $K$ depends on the chosen value of $L$, since they are proportional. This is because the higher $L$, the higher is the number of undesired passbands, and therefore more periodic filters $G_k(z^{L_{k-1}})$ are required.

[1] All of the analyses are based on linear-phase filers, unless otherwise indicated.

In order to obtain an optimal decomposition, the relation between $L$ and $L_k$ is the following:

$$L_{k-1} = \frac{L}{L_1 L_2 \dots L_{k-2}} \tag{21}$$

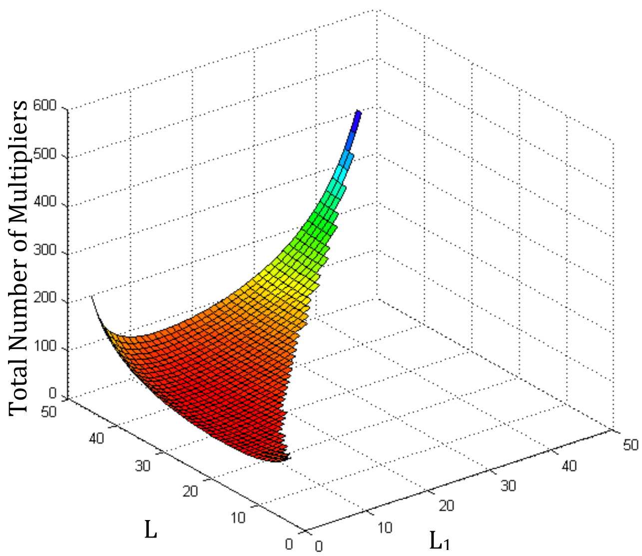


Fig. 6. Number of Multipliers of the overall IFIR filer for the Example1 using the two stretching factors approach.

To illustrate the functionality of this method, we addressed the specifications (6) by using this approach, whose process and result are shown on Fig. 7.

As a result, this realization yields 52 multipliers and 93 adders as a total. Hence, it is noticed that it turns out a high improvement comparing to the basic IFIR filter introduced in this paper, and it practically equals the performance achieved by the *Optimal Two Stretching Factors* technique.

### *B. Decimation Approach*

The decimation process can be used along with interpolation to obtain better performance. The main idea is about to use certain algebraic properties of decimators and transfer functions called *Noble Identities* [5]. Specifically, the identity used is the following:

$$x(n) \rightarrow [\downarrow M] \rightarrow H(z) \rightarrow y_1(n)$$
$$\Updownarrow \tag{22}$$
$$x(n) \rightarrow H(z^M) \rightarrow [\downarrow M] \rightarrow y_2(n)$$

Where $M$ is the decimator factor and $y_1(n) = y_2(n)$. With this equivalence, it is possible to derive a new structure from the desired filter as shown on Fig. 8.

### *C. Multiplier-Free*

The basic idea of this method is to exploit some numerical advantages that allow avoiding using multipliers under computational systems. We can define a rounded filter $g(n)$, such that [2]:

$$g(n) = r \cdot g_I(n) = r \cdot round\left({}^{h(n)}/_{r}\right) \tag{23}$$

Where $h(n)$ is the original FIR filter, $r$ is a scaling factor, $g_I(n)$ is the rounded version of $h(n)$, and the function $round(x)$ rounds $x$ to the closest integer. Thus, the main objective of this expression is rounding every coefficient of the original filter to the nearest integer in order to take the advantage of the fact that multiplications between integers can be replaced by adding and shifting operations, so that the multipliers can be computed without multiplying. Besides, the constant $r$ conditions the approximation accuracy. As expected, it yields a deformation in the frequency response; although there are several sharpening techniques that improve its performance [3].

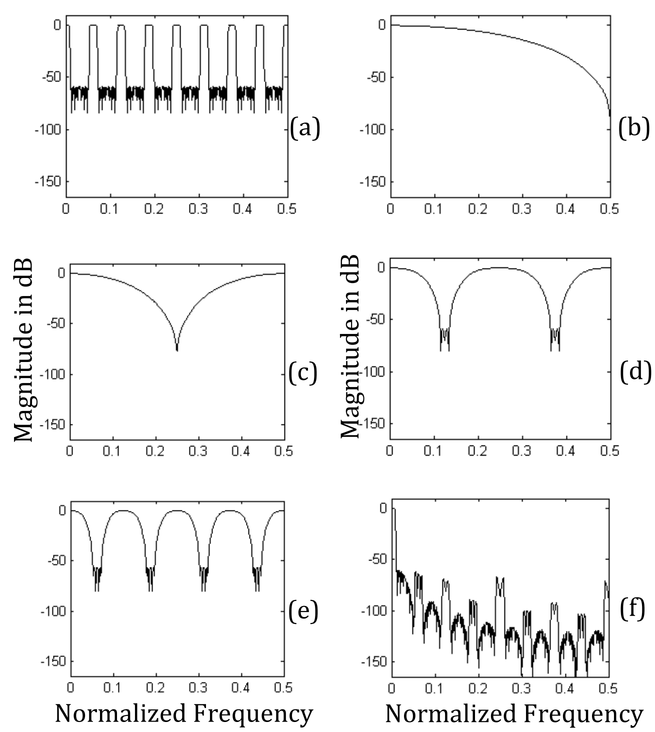


Fig. 7. Process and result of the design of a multirate IFIR filter for Example1 using four interpolators. (a) The shaping filter $H_M(z^{16})$. (b) $G_1(z)$. (c) $G_2(z^2)$. (d) $G_3(z^4)$. (e) $G_4(z^8)$. (f) The overall filter $H_{IFIR}(z)$.

Hence, this technique is perfectly suitable for low pass interpolator filters, since their task is basically to maintain the gain reduction in the stop band. Any error in this range due to rounding could be solved by increasing the order of the original filter.

Due to this method avoids any multiplier in the interpolator section; theoretically, this should present the best performance for the case of the specifications (6). Hence, the *Multilier-Free* method can be applied for the two interpolator stages $G(z)$ and $G(z^{L1})$, involving 13 and 12 multipliers respectively. It would yield in an overall number of multipliers equals to 25, which means a computational complexity reduction at about 96%.

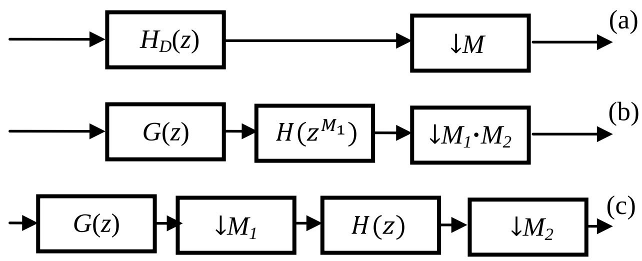


Fig. 8. (a) The decimation by $M$ of the desired filter $H_D(z)$. (b) The desired filter is expressed in terms of a periodic filter $H(z^{M_1})$ stretched by $M_1$ and the interpolator $G(z)$, where $M= M_1 \cdot M_2$. (c) Final structure equivalent to

## III. Conclusions

The IFIR filter design approach is a very useful way to reduce the computational complexity of FIR filters showing excellent results. Moreover, the simplicity on designing this type of structure makes it more appealing.

Regarding the example provided in this paper; the basic model consisting in only two stages turned out a complexity reduction of 72%. However, the best performance is shown by the *Two Optimal Stretching Factors* and the *Multirate Multistage* techniques which present a reduction of complexity equals to 92%. Although a *Multiplier-Free* realization for the specifications (6) was not developed; theoretically, it would reach 96% of computational cost. However, the number of delays would increase considerably. These results support what was stated in the first paragraph, showing the excellent performances yielded by all of the realizations analyzed on this paper, even considering the complexity of the narrowband filter for the specifications (6).

## References

[1] 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.

[2] Bartolo. B. D. Clviner, R. C. Burges. and J. P. Turnbull. “An efficient method of FIR filtering based on impulse response rounding.” *IEEE Trans. on Signal Processing.* vol. 46, No. 8. pp.2243-2248. August 1998.

[3] G. Jovanovic-Dolecek, S. K. Mitra, “Multiplier-free FIR filter design based on IFIR structure and rounding,” *IEEE 48th Midwest Symposium on Circuits and Systems*., vol. 1, pp. 559-562, 2005.

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

[5] M. C. Lin: “ Design Techniques for Low complexity and High speed Multirate Multistage FIR Digital Filters, Department of Electrical Engineering National Taiwan University Taipei, Taiwan, ROC 2007.

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

[7] R. Lyons, “Interpolated narrowband lowpass FIR filters,” IEEE Signal Processing Magazine, vol. 20, no. 1, pp. 50–57, Jan.2003.

[8] S. K. Mitra. “Digital Signal Processing - A computer-Based Approach. Third Edition.” McGraw-Hill Inc.. New York. 2006.

[9] T. Saramäki, ‘‘Computationally efficient narrowband linear phase FIR filters,” *IEEE Proc. G, Electron Circuits and S-yst.,* vol. 130,1975, pp. 23-126. pp. 20-24, Feb. 1983.

[10] T. Saramäki, “A class of linear-phase FIR filters for decimation, interpolation, and narrow-band filtering.” *IEEE Trans. Acoust., Speech. Signal Processing,* vol. ASSP-32, pp. 1023-1036, Oct. 1984.

[11] T. Saramäki, “Design of optimal multistage IIR and FIR filters for multirate filtering-Part I: Design of decimators and interpolators: Part 11: Design of narrowband filters, *Proc.* 1986 *IEEE Int. Symp. Circuits Syst.,* (San Jose, CA), pp. 227-230, May 1986.

[12] T. Saramäki, T. Neuvo, and S. K. Mitra, “Design of computationally efficient interpolated FIR filters,” IEEE Transactions on Circuits and Systems, vol. 35, no. 1, pp. 70–88, Jan. 1988.

[13] Y. Neuvo, C.-Y. Dong, and S. K. Mitra, “Interpolated finite impulse response filters,” IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 32, no. 3, pp. 563–570, June 1984.