Studying different Interpolated FIR Digital Filters and how its implementation affects computational power

Digital Signal Processing

Rafael Marques Braga
School of Electrical and Computer Engineering
University of Miami
Miami, FL
Rxm403@miami.edu

***Abstract*—This paper studies interpolated finite impulse response filters (IFIR). It looks how these filters are done, its advantages and its effects on both the impulse response and magnitude response. It also shows on other techniques on how IFIR filters are done today, and discusses its advantages for each type of filters.**

***Index Terms*—Interpolation, upsampling, Finite Impulse Response filters, impulse response, magnitude response.** *(key words)*

I. Introduction

Finite impulse response (FIR) filters is one of the two primary types of digital types of filters, with the other type being Infinite impulse response filters (IRR). Having a FIR filter simply means that the impulse response of the filter is finite. In many of the cases, this means that the filter has no feedback. FIR filters can also contain feedback, however, these feedbacks need to be built in a certain method, in order to keep the impulse response of the filter finite.

FIR filters have its advantages and disadvantages. Usually its advantages outweigh its disadvantages, making it one of the most used types of filters. Some advantages of FIR filters are that it can easily be designed to be a linear phase filter, so it will not affect the phase of the filter, while still having a delay to it. Another advantage of the filter is that it is easy to implement on microprocessors and are also suited for interpolation and decimation making the filter easier to work with when using a higher order. However, some disadvantages of the FIR filter include that it is not every response that makes it practical to implement FIR filters and it can also use more memory space in computers.

There has been continuous studies on how to make FIR filters work better and faster. When using a very high sampling time in different signals, it makes it hard for a computer to follow with all the calculations in real time. On other times, the sampling rate is simply not fast enough, this brings a need for a increase of the sampling rate. Therefore there is a need for filter interpolation or decimation. This is when the sampling rate is increased or decreased respectively.

This paper will focus on filter interpolation. It will study various innovative methods for interpolating a filter, analyzing each method and concluding a final method to use when using interpolation.

II. The Problem

In this paper, the study will be made by creating and analyzing different methods of interpolation on a narrow band, FIR filter. This filter will have the following specifications.

Passband maximum ripple = 0.001;
Stopband maximum ripple = 0.001;
Passband edge frequency = 0.015π;
Stopband edge frequency = 0.020π.

III. Using an Optimum FIR Filter

To set a basic filter to compare the methods studied, a first filter will be implemented using the "*firpm*" function from MATLAB. This filter will be used to compare all the other filters from the other interpolated designs.

To start *figure 1* shows the impulse response of the filter.

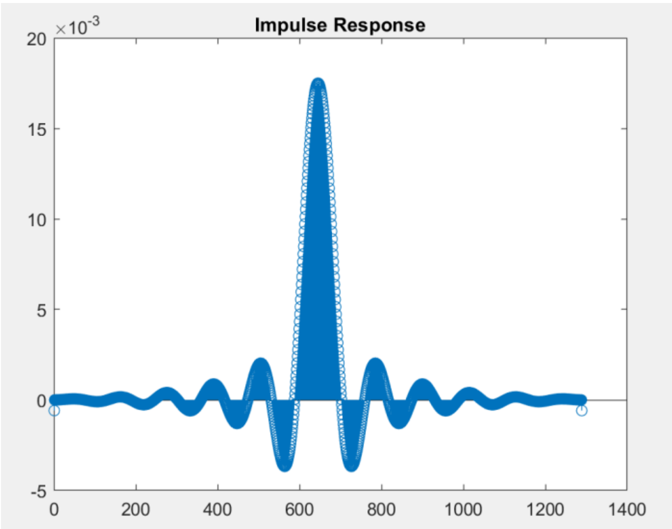


Figure 1

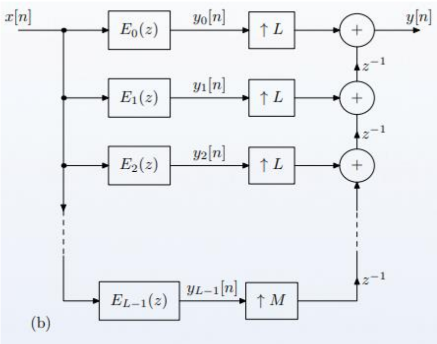


Figure 2

As it can be seen it has a finite impulse response with 1288 different impulses. This is because the calculated order of the filter was 1288. This was calculated using the Kaiser's formula shown below.

$$M = \frac{-10\log_{10}[\delta_p\delta_s]+13}{14.6(\omega_s - \omega_p)/2\pi} \; [1]$$

In *figure 3,* the magnitude response of the filter is shown.

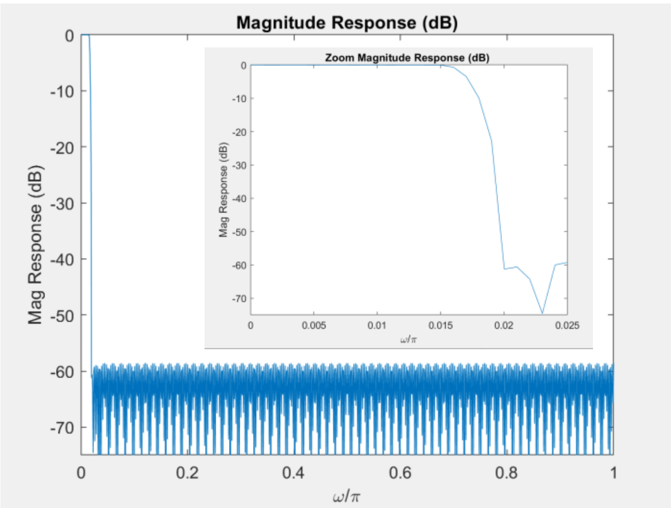


Figure 3

As seen, the filter created has an extremely narrow pass band, with a very steep cut off frequency. The filter also has a -60db ripple.

IV. Upsampling & Interpolation

Upsampling is when zeros are included in between the samples. An example of this can be seen on the picture bellow.

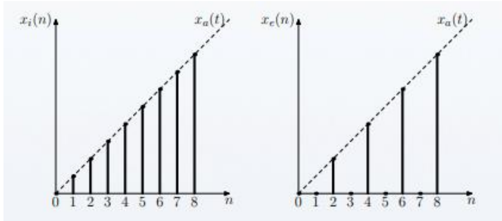


Figure 4

Interpolation is when you have an upsampler followed by a low pass filter. The low pass filter is there in order rebuild the signal after the interpolation. In *figure 4* a two fold upsampling can be seen as a zero is inserted in every 2nd sample. This kind of interpolation becomes useful because through different interpolations, it can then be implemented in "parallel" as shown below, and it can work in a way to make the each of the filters to work at a slower sampling rate.

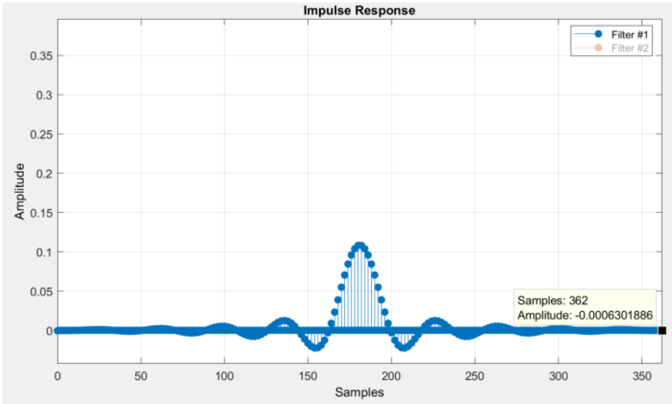


Figure 5

Sometimes in a signal, as mentioned in the introduction, the sampling time of a signal can be too fast for a filter to work properly. This is one of the uses for interpolation as it significantly reduces the computational workload when compared to the normal filters [2].

In Lyons paper [2], it is studied on how interpolated FIR filters are implemented. In the paper it is discussed on how the IFIR filter function can be expressed as

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

[2]

In the equation above N is the length of $h_{pr.}$ *Lyons* discusses on how a IFIR can be implemented similar to *figure 3.* This is by adding gradually adding delays, as for each delay, interpolate the signal, according to its delay. This is seen on *figure 4.*

So IFIR becomes useful becomes useful because it decreases the order of the filter that is needed to use the same filters. A perfect example to see the use of IFIR is in the example being studied in this paper since it is a narrow passband and a steep cut off frequency.

As seen on the impulse response of the optimum FIR filter without interpolation there was a need of 1288 different samples. However, when upsampling the same filter by 2, it already changes the order of the filter to 362 samples as it can be seen in the figure below.

The magnitude response can also be seen in *figure 6*. In *figure 6*, it can also be seen, in red, the low pass filter from interpolation. The low pass filter is needed so the signal does not get repeated, since the signal is being compressed in the frequency domain. As the upsampling number increases, it can also be seen (*figure 6*) that the number of samples in the impulse response decreases.

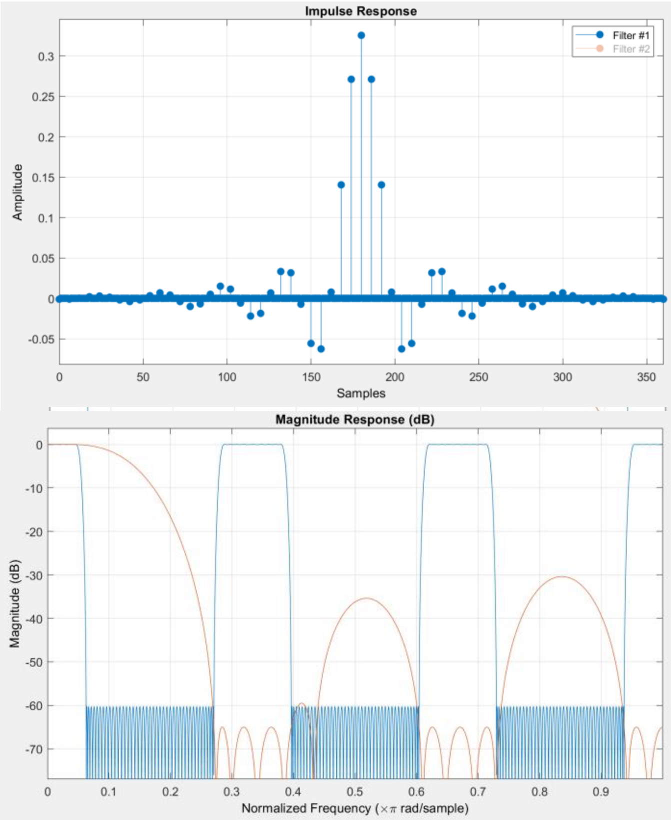


Figure 6

In *figure 6*, the filter, with the same specifications as listed in the beginning of the paper were done, however it was upsampled by 6. Clearly, in the impulse response for the figure abore, the samples are spread apart, a lot more than in the FIR filter. As Lyons mentions it in his paper, this gives the computer less effort to do the same computations as it would do using the normal FIR filter, making the computer do the computations a lot faster as well as being more efficient. The red line in *figure 5*, again, shows the low pass filter that is used so the signal is not repeated. This is because when a signal is upsampled, the signal is compressed, and it will then get repeated in the frequency domain, creating many periods. This also generates the need to use a low pass filter, so there is no aliasing happening.

V. Frequency Response Masking Filters

Frequency response masking (FRM) filters are another type of filters that can be used for narrow transitions bands, like the one studied in this paper. This type of filters usually produces sharper transition bands. The transfer function of a FRM filter is the following.

$$H_{FRM}(z) = F(z^L)M(z) + \left[z^{-\frac{L(n-1)}{2}} - F(z^L)\right]M_c(z)$$

Where N is an odd integer and it means the length of F(z). A FRM filter with a single stage is connected in parallel with the periodic filter with upsampling ($F(z^L)$) and its delay periodic filer (the second part of the equation). A representation of a single stage filter can be seen in *figure 7 [3]*.

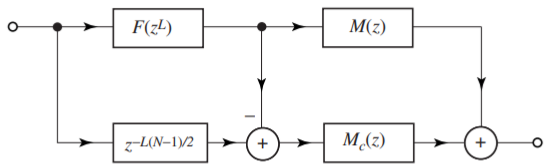
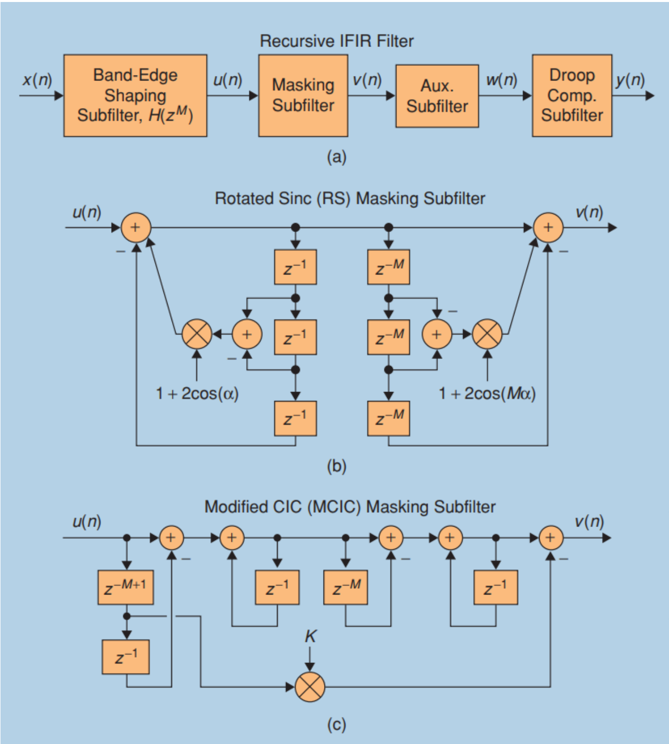


Figure 7

VI. RECURSIVE IFIR FILTERS

Recursive filters are filters which have feedback in the system. It can have one or more of its outputs used as its inputs. This can usually end up as being a IIR filter, however, if implemented correctly, it can be used is a IFIR filter. A recursive IFIR filter can also increase computational efficiency as mentioned by Lyons [4]. A diagram of a recursive IFIR filter can be seen in *figure 8.*



[FIG1] Recursive IFIR filter: (a) subfilters, (b) RS-masking subfilter structure, (c) MCIC-masking subfilter.

Figure 8

The recursive part of this filter can be seen in the first part of the Rotated Sinc (RS) masking subfilter and also in the modified CIC (MCIC) Masking subfilter. In the RS subfilter, there is feedback when the signal has a z^{-3} delay but then it is later decimated to make the filter linear. [4]

VII. CONCLUSION

In this paper various methods of implementing Interpolated Finite Impulse Response were studied including recursive IFIR filters, using frequency response masking filters with IFIR, only using IFIR and using a FIR to build a filter.

It was seen that for a narrow band, low pass filter, when using a optimum FIR, the order of the filter can be too high. This will require high amounts of computer power, making the filter inefficient and also a lot harder to implement. When using an IFIR, it could be seen that even though is a little bit harder to implement, it can make the filter a lot more efficient when doing the computer calculations. This is because it will change the impulse response of the system, giving it less samples and not requiring as much computer power to make all the calculations for the filter.

Other possible ways of implementing interpolation into filters were also seen, however these were not implemented to see how they affect the impulse response of the filter and how much less computer power these filters would require. However papers listed in the references have shown and proved that recursive IFIR filters and using IFIR filters with different masking filters can require even less computer power to make calculations.

It was also seen that the more the filter is interpolated the wider the impulse response will be, however, this will not always have a positive impact on the filter, since data can be lost in the interpolation since it is increasing the sampling time. This should be done carefully in order to figure out what is the best number for the filter to be upsampled by.

VIII. REFERENCES

[1] [Manolakis and Ingle, 2011] Manolakis, D. G. and Ingle, V. K. (2011). Applied Digital Signal Processing: Theory and Practice. Cambridge University Press, New York, NY.

[2] R. Lyons "Interpolated narrowband lowpass FIR filters" IEEE Signal Processing Mag. vol. 20 no. 1 pp. 50-57 Jan. 2003.

[3] W. Lu and T. Hinamoto, "A unified approach to the design of interpolated and frequency-response-masking FIR filters," 2016 IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, QC, 2016, pp. 2174-2177.

[4] R. Lyons, "Turbocharging Interpolated FIR Filters [DSP Tips & Tricks]," in IEEE Signal Processing Magazine, vol. 24, no. 5, pp. 140-143, Sept. 2007.