Optimal Sharpening of CIC Filters and An Efficient Implementation Through Saramäki-Ritoniemi Decimation Filter Structure (Extended Version)

Çağatay Candan
Department of Electrical Engineering, METU, Ankara, Turkey.
email: ccandan@metu.edu.tr

***Abstract*—Conventional sharpened cascaded-integrator-comb (CIC) filters use generic sharpening polynomials to improve the frequency response. In contrast to the existing literature, an optimization framework is described for the selection of CIC sharpening polynomial and an efficient implementation through Saramäki-Ritoniemi decimation structure is suggested for its realization. The optimized sharpening polynomials are application specific and designed to meet given passband ripple and stopband attenuation specifications. Numerical results show that the optimized structure can be used without a secondary droop compensation filter, which is typically required for the conventional systems.**

***Index Terms*—Cascaded-Integrator-Comb (CIC) filters, sampling rate conversion, decimation, linear programming.**

I. Introduction

Cascaded-integrator-comb (CIC) filters are utilized in many applications that require efficient sampling rate conversion. An important application area for CIC filters is the software defined radio where the receiver can tune into a number of different bands with possibly different bandwidths, [1]. The conventional CIC filters, shown in the top part of Figure 1, do not have any multipliers making this structure particularly attractive for the FPGA implementations[1]. There are two major drawbacks of the conventional filters which are the large passband droop and limited stopband attenuation. These problems can be corrected to a certain degree by either modifying the conventional structure, [2], [3], [4], [5], [6] or implementing a secondary filter, after the conventional one, to compensate its undesired characteristics, [7], [8].

In certain applications, such as Σ/Δ converters, the filter input data can be significantly oversampled. In these applications, the CIC based decimators are utilized in the front stages of the processing chain to reduce the processing rate. For example, a CIC based decimator (say for 8-fold sampling rate reduction) is followed by a secondary decimator (say for 5-fold sampling rate reduction) is utilized to achieve large decimation ratios (which is 40-fold reduction). For such systems, the low pass filter of the secondary decimation block can also act as a compensation unit correcting the undesired characteristics of the front-end CIC stage, [4], [8], [9].

In this paper, we present an optimization framework for CIC filter sharpening and suggest the Saramäki-Ritoniemi structure for its efficient implementation. The Saramäki-Ritoniemi structure has been publicized in 1997, [2], [3]. In this paper, different from the original work of Saramäki-Ritoniemi, we approach the problem from the direction of filter sharpening. It should be noted that the application of the sharpening filters to the CIC decimation structure has been proposed by Kwentus *et al.* also in 1997, [4]. The current paper has been initiated with the goal of optimizing the ad-hoc filters suggested by Kwentus *et al.* and later it has been understood that the optimized structure is identical to the one suggested by Saramäki-Ritoniemi. Hence the current paper also establishes a connection between two lines of research for the CIC filter design.

The Saramäki-Ritoniemi structure shown in Figure 1 has a set of free parameters denoted with $\{\alpha_k, \beta_k, \gamma_k\}$, $k = \{0, \ldots, L\}$. Here L is the number of cascaded CIC blocks, as in the conventional scheme. The β_k and γ_k parameters indicate the delays appearing before and after the decimation-by-M unit and α_k parameters are the linear combination coefficients of the delayed sections. It can be noted that when $\alpha_k = 0$ for all k values, the Saramäki-Ritoniemi structure reduces to the conventional one given in the same figure. Furthermore by setting all α_k values to zero, except $\alpha_2 = 3$, $\alpha_3 = -2$ and adjusting the delays; the resultant filter is identical to the sharpened CIC filters proposed by Kwentus *et al.* [4].

In this paper, we present a framework for the optimization of the free parameters appearing in the Saramäki-Ritoniemi structure. The optimization process, different from [2], is not generic but specially designed for the optimization of decimation filters. Some optional optimization features that can be useful for high rate applications is suggested and ready-to-use MATLAB code is provided. The numerical results show that the frequency response of the optimized structure meets the specifications well enough that the compensation filter following the decimator can be eliminated with the optimized structure.

II. Saramäki-Ritoniemi Structure

The single stage non-recursive CIC filter calculates the average of M consecutive samples:

$$H_{\text{CIC}}(z) = \frac{1}{M}\left(1 + z^{-1} + \ldots + z^{-(M-1)}\right) = \frac{1}{M}\frac{1 - z^{-M}}{1 - z^{-1}} \quad (1)$$

[1] It is possible to move $(1/M)^L$ multiplication to the decimator output and combine with the subsequent processing stages.

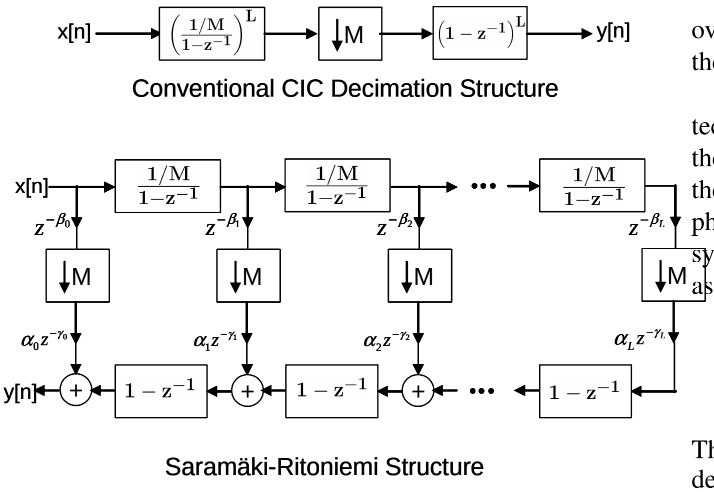


Fig. 1. Conventional CIC Structure and Saramäki-Ritoniemi Structure

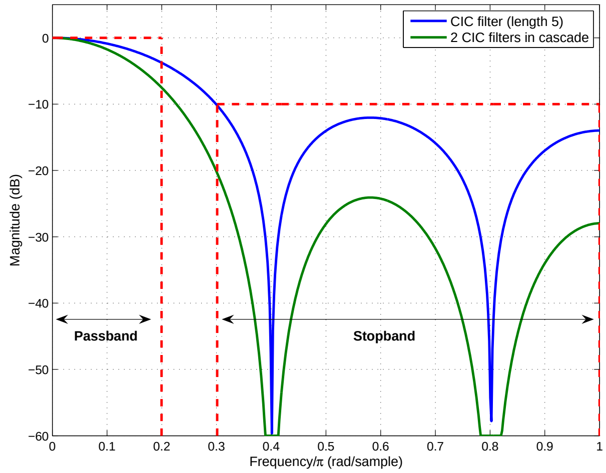


Fig. 2. CIC filters and passband/stopband definitions

Figure 2 shows the frequency response of the CIC decimation filter for the downsampling rate of 5, ($M = 5$). On the same figure, the desired pass/stop bands are also indicated. If the input signal is known to be oversampled by a factor of M, the rate after decimation becomes the Nyquist rate. For this case, the desired passband is $[-\pi/M, \pi/M]$, as shown in Figure 2.

In many applications, a CIC based decimator is followed by a secondary decimation stage. Hence the output of the front-end CIC decimator is not at the Nyquist rate. For such applications, the passband for the CIC filter extends to $\pi/(rM)$, where r is a scalar greater than 1 showing the residual oversampling rate at the front-end decimation output.

It can be noted from Figure 2 that the stopband attenuation of the CIC structure is only 12 dB. To increase the stopband attenuation, the CIC filters are used in cascade. Each cascade brings an additional 12 dB of attenuation. It should be remembered that while each cascade brings an additional 12 dB of attenuation to the stopband frequencies, the passband droop also increases with the number of cascades. (The passband drop increases by 3.5 dB per cascade for the given example.) If the desired passband is narrower, that is the residual oversampling rate at the output (r) is much larger than 1; the passband droop may not pose a significant problem.

Kwentus *et al.* have suggested to use the filter sharpening technique of Kaiser and Hamming [10], to partially alleviate these problems, [4]. Filter sharpening method improves both the passband and stopband characteristics of a prototype linear phase filter, [10]. In our case, the prototype filter is the even symmetric version of the CIC filter. This filter can be expressed as follows:

$$H_{\text{CIC}}(e^{j\omega}) = e^{-j\omega\frac{M-1}{2}} \underbrace{\frac{\sin(\omega M/2)}{M\sin(\omega/2)}}_{P(e^{j\omega})}. \qquad (2)$$

The first term on the right hand side of (2) is due to the group-delay of the filter. The second term, $P(e^{j\omega})$, is the prototype filter and it is a real valued function of ω that corresponds to the discrete-time Fourier transform (DTFT) of the symmetric version of the CIC filter.

Filter sharpening procedure constructs a new zero-phase filter from the given prototype. This procedure can be explained as follows: Let $g(x)$ be a polynomial in x defined from $[-1, 1]$ to $[-1, 1]$. The sharpened frequency response is simply $\widehat{P}(e^{jw}) = g(P(e^{j\omega}))$. In [10], a number of suitable $g(x)$ functions, for example $g(x) = 3x^2 - 2x^3$, have been suggested. These polynomials attain the value of 0 at $x = 0$ and the value of 1 at $x = 1$. Furthermore, a number of derivatives at $x = \{0, 1\}$ is equal to zero. The number of derivatives reducing to zero indicates the smoothness or the flatness of the function around $x = \{0, 1\}$. It is expected that a reasonably good prototype has an improved response both in passband ($P(e^{j\omega}) \approx 1$) and stopband ($P(e^{j\omega}) \approx 0$) after the application of sharpening.

It should be noted the sharpening polynomials in the literature are selected through the mentioned flatness considerations. Hence, these polynomials are not optimized for a particular problem. In this study, we suggest to optimize $g(x)$ polynomial to meet the passband and stopband specifications of the CIC based decimation systems.

For illustration purposes, let's assume that the sharpening polynomial $g(x)$ is a Lth order polynomial:

$$g(x) = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \ldots + \alpha_L x^L. \qquad (3)$$

Then the sharpened filter has the frequency response of

$$\widehat{P}(e^{jw}) = \sum_{k=0}^{L} \alpha_k \left[P(e^{j\omega})\right]^k = \sum_{k=0}^{L} \alpha_k \left[\frac{\sin(\omega M/2)}{M\sin(\omega/2)}\right]^k. \qquad (4)$$

It should be noted that the sharpened filter, $\widehat{P}(e^{jw})$, is also a zero-phase filter and its frequency response is a linear combination of the prototype filter frequency response and its powers.

We would like to present a concrete example for the impulse response construction of the sharpened filter. For the decimation rate of $M = 5$, the inverse DTFT of the prototype response, i.e. $\mathcal{F}^{-1}\{P(e^{j\omega})\}$, is a 5-point sequence whose symmetry center is the 3rd sample. The second power of the prototype response, i.e. $\mathcal{F}^{-1}\{P^2(e^{j\omega})\}$, is a 9 point sequence whose symmetry center is the 5th sample. Similarly,

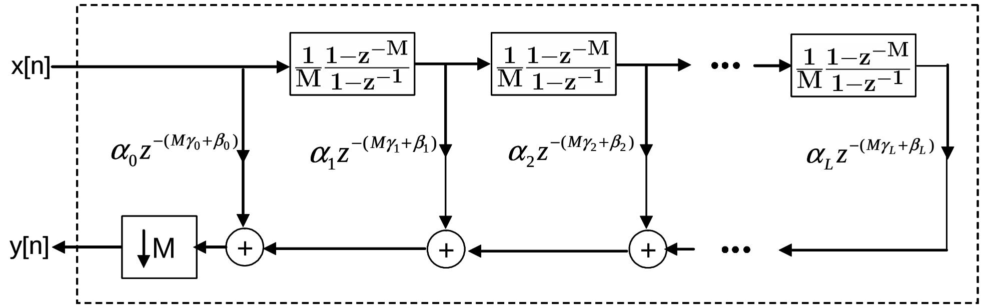

x[n]
$\frac{1}{M}\frac{1-z^{-M}}{1-z^{-1}}$
$\frac{1}{M}\frac{1-z^{-M}}{1-z^{-1}}$
$\frac{1}{M}\frac{1-z^{-M}}{1-z^{-1}}$
$\alpha_0 z^{-(M\gamma_0+\beta_0)}$
$\alpha_1 z^{-(M\gamma_1+\beta_1)}$
$\alpha_2 z^{-(M\gamma_2+\beta_2)}$
$\alpha_L z^{-(M\gamma_L+\beta_L)}$
y[n]
↓M
+
+
+

Fig. 3. A direct implementation for the sharpened CIC filters

$\mathcal{F}^{-1}\{P^3(e^{j\omega})\}$ is of length 13 and has the symmetry center at the 7th sample. The sharpened filter is a linear combination of these sequences. It is important to note that before the linear combination of these sequences, a number of zeros should be appended to the front of each sequence so that all sequences have a common symmetry center. For the presented example, the longest sequence in the combination is of length 13, then 4 zeros should be appended to the front of 5-point sequence to align their symmetry centers. The delays appearing in the vertical branches of proposed system shown in Figure 1 is to align the symmetry centers of each section.

Figure 3 shows a direct implementation of the described structure. This implementation is not efficient, but it is conceptually straightforward. The direct implementation can be transformed to the efficient structure, which is the Saramäki-Ritoniemi structure, shown in Figure 1 in a few steps: First, move the M-fold decimation block into the summations and relocate it on the vertical branches. Then interchange the delay operators and the decimation operators. Move the factor of $1-z^{-M}$ (and its powers) to the vertically oriented branches, interchange this block with decimator. (After the interchange, $1-z^{-M}$ is converted to $1-z^{-1}$.) Finally, collect the common $1-z^{-1}$ term (and its powers) lying on the summation branches together and move the common term to the output of the summation. Once these steps are completed, we get the efficient implementation shown in Figure 1.

III. Optimal Sharpening Polynomial

In this section, we present a linear programming based optimization framework for the selection of the sharpening polynomial. The goal is to minimize the worst case passband and stopband ripples. In the original work of Kaiser and Hamming, the sharpening polynomials are designed to improve the response of generic filters, [10]. Here, we would like to present an optimization framework specific for the CIC filters.

The linear program can be written as follows:

$$\begin{aligned} &\underset{\mathbf{x}}{\text{minimize}} \quad \mathbf{f}^{\mathbf{T}}\mathbf{x} \\ &\text{subject to} \quad \mathbf{A}\mathbf{x} \le \mathbf{b} \quad \text{and} \quad \mathbf{A}_{\mathbf{eq}}\mathbf{x} = \mathbf{b}_{\mathbf{eq}}. \end{aligned} \tag{5}$$

Below, we present the inequality and equality constraints appearing in the problem and also the vector $\mathbf{f}$ producing the cost.

Constraint on Maximum Passband Ripple: Let ω_{p_k} represent a frequency value in the desired passband. The magnitude deviation of the sharpened filter from the desired response can be written as $|g(P(e^{j\omega_{p_k}}))-1|$. Our goal is to minimize the deviation through a proper selection of α_k coefficients, which are given in (3).

We assume that $|g(P(e^{j\omega_{p_k}}))-1| \le \epsilon_p$ or $-\epsilon_p \le g(P(e^{j\omega_{p_k}}))-1 \le \epsilon_p$ for some unknown ϵ_p. Here ϵ_p is the passband ripple value that can be attained. The goal is to reduce ϵ_p for a set of dense ω_{p_k} values in the passband, i.e. to minimize the worst case ripple.

The inequalities can be summarized as follows:

$$\begin{aligned} &\begin{bmatrix} -1 & P(e^{j\omega_{p_k}}) & P(e^{j\omega_{p_k}})^2 & \dots & P(e^{j\omega_{p_k}})^L \end{bmatrix} \mathbf{x} \le 1 \\ &\begin{bmatrix} -1 & -P(e^{j\omega_{p_k}}) & -P(e^{j\omega_{p_k}})^2 & \dots & -P(e^{j\omega_{p_k}})^L \end{bmatrix} \mathbf{x} \le -1 \end{aligned} \tag{6}$$

Here $\mathbf{x}$ is the vector of unknowns:

$$\mathbf{x} = \begin{bmatrix} \epsilon_p & \alpha_0 & \alpha_1 & \dots & \alpha_L \end{bmatrix}^{\mathbf{T}} \tag{7}$$

This concludes the derivation of the inequality constraints for ω_{p_k}, a single sample of passband frequencies. Similar inequalities should be reproduced for a dense set of frequencies covering the passband.

Constraint on Maximum Stopband Ripple: Let ω_{s_k} represent a frequency value lying in the desired stopband. The stopband ripple for the frequency of ω_{s_k} can be bounded as $|g(P(e^{j\omega_{s_k}}))| \le \epsilon_s$. The goal is to reduce the worst case ϵ_s for the stopband frequencies. To that aim, we introduce a weight W which is defined as the ratio of maximum passband ripple to the maximum stop band ripple, $\epsilon_s = \epsilon_p/W$.

The filter designer sets W to trade-off between the amount of passband and stopband ripples. It can be noted that a higher W value decreases the stopband ripple at the expense of increased passband ripple.

The inequalities can be summarized as follows:

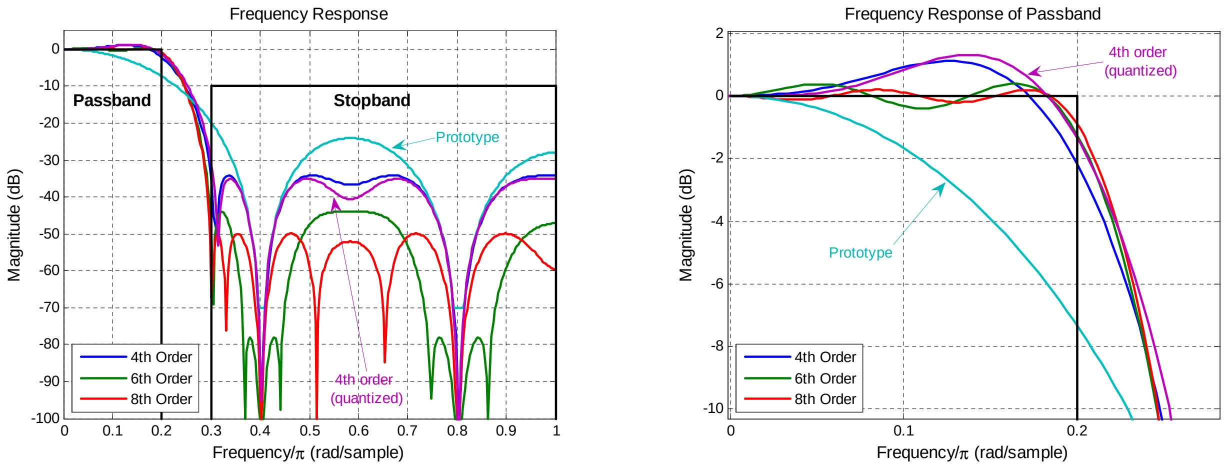
$$\begin{aligned} &\begin{bmatrix} -1/W & P(e^{j\omega_{s_k}}) & P(e^{j\omega_{s_k}})^2 & \dots & P(e^{j\omega_{s_k}})^L \end{bmatrix} \mathbf{x} \le 0 \\ &\begin{bmatrix} -1/W & -P(e^{j\omega_{s_k}}) & -P(e^{j\omega_{s_k}})^2 & \dots & -P(e^{j\omega_{s_k}})^L \end{bmatrix} \mathbf{x} \le 0 \end{aligned} \tag{8}$$

The inequalities should be reproduced for a dense set of frequencies covering the desired stopband.

Equality Constraint for DC frequency: It is desirable to attain the frequency response of 1 at the DC frequency. This condition is satisfied if $g(P(e^{j\omega}))\downarrow_{\omega=0}= 1$. Since $P(e^{jw})\downarrow_{\omega=0}= 1$ for the prototype filter, the constraint reduces to $g(1)=1$ and can be expressed as follows:

$$\begin{bmatrix} 0 & 1 & 1 & \dots & 1 \end{bmatrix} \mathbf{x} = 1 \tag{9}$$

Equality Constraint for Image Nulling: In some sampling rate conversion systems, the input contains the images of the



(a) Frequency Response

(b) Frequency Response of Passband

Fig. 4. Frequency Response of 5-fold Decimation Filter whose target output rate is the Nyquist Rate.

desired spectrum centered around the multiples of $2\pi/M$, where M is the upsampling ratio. For such systems, it is desirable to have nulls centered at the integer multiples of $2\pi/M$. This can be achieved with $g(P(e^{j\omega}))\downarrow_{\omega=2\pi k/M}=0$ for $k\neq 0$. Since that $P(e^{jw})\downarrow_{\omega=2\pi k/M}=0$ $(k\neq 0)$ for the prototype filter, the constraint reduces to $g(0)=0$ (or $\alpha_0=0$) and can be expressed as follows:

$$[0 \quad 1 \quad 0 \quad \dots \quad 0\,]\,\mathbf{x}=0 \tag{10}$$

The Cost Function: The goal is to minimize ϵ_p via a proper selection of the sharpening polynomial coefficients. The cost function, $\mathbf{f^T x}$, can be written as $\mathbf{f^T}=[1\ 0\ 0\ \dots\ 0]$ where $\mathbf{x}$ is defined in (7).

The inequality constraints of the linear program can be written by concatenating the set of inequalities given in (6) and (8) for dense sets of passband and stopband frequencies. The equality constraints are optional for the lowpass filter design problem. If desired, they can be easily accommodated by concatenating the equations given in (9) and (10).

Once the problem is expressed in the standard form of linear programming, the solution can be found efficiently through a general purpose solver. Readers can retrieve a ready-to-use MATLAB function from [11].

IV. Numerical Results

To illustrate the described structure, we present two examples. In both examples, the cascade of two CIC filters is used as the prototype filter, i.e. $P(e^{j\omega}) = [\sin(\omega M/2)/(M\sin(\omega/2))]^2$. This choice is due to insufficient stopband attenuation of the single stage CIC structure.

Example 1: 5-fold Decimation to the Nyquist Rate Figure 5 shows the suggested CIC based low-pass filtering structure for 5-fold decimation. The sharpening polynomial specific for this problem is $g(x)=10x^4-23x^3+15x^2-x$. It should be noted that the coefficients of the sharpening polynomial are all integers making the system especially attractive for the FPGA implementations.

The pink line with the label '4th order (quantized)' in Figure 4 shows the frequency response of the suggested system. The other curves show the response of the prototype system and the response of the filters having 4th, 6th, 8th order optimal sharpening polynomials similarly found through the described linear programming procedure. The sharpening polynomial with integer coefficients (quantized coefficients) is formed by rounding the coefficients of the optimal 4th order polynomial to the nearest integers.

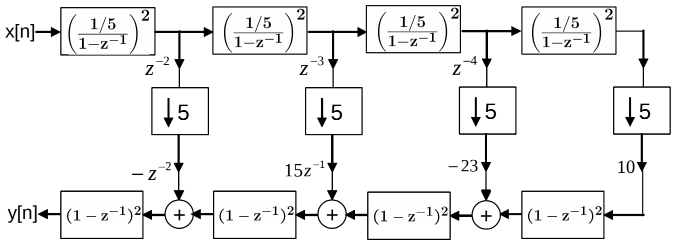
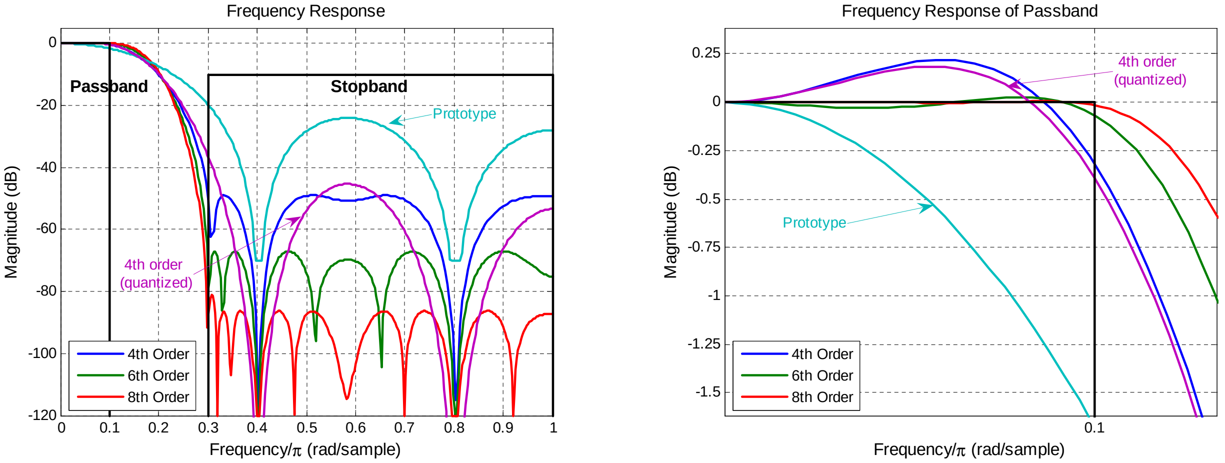


Fig. 5. Proposed 5-fold Decimation Filter. (Decimation output is at Nyquist rate)

As shown in Figure 4, the desired passband is the interval of $[-\pi/5, \pi/5]$. For this system, the target rate after the decimation is the Nyquist rate. For the desired bandwidth, the passband droop of the prototype filter is around 8 dB. The described 4th order implementation with integer valued linear combination coefficients has a maximum passband ripple of 1 dB and has a stopband attenuation of at least 34 dB. These values can be acceptable in many applications.

It should be noted that the sharpening filters of higher orders have further improved droop and stopband characteristics. For the 6th and 8th order sharpening polynomials, the maximum ripple reduces to 0.5 dB and 0.2 dB respectively and the worst case stopband attenuation increases to 44 dB and 50 dB respectively.

As a last note, we would like to remind that the designs shown in Figure 4 are specific for the given passband and stopband pair. Furthermore, the weighting factor W, trading the passband ripple with the stopband attenuation, is chosen as 7 in this example. By changing W, sharpening polynomials having reduced droop at the expense of worse stopband atten-



(a) Frequency Response

(b) Frequency Response of Passband

Fig. 6. Frequency Response of 5-fold Decimation Filter whose target output rate is two times the Nyquist Rate.

uation (or vice versa) can be found. We believe that the weight W can be instrumental to achieve difficult specifications.

Example 2: 5-fold Decimation to Double Nyquist Rate In many applications, there is a sequence of decimation blocks progressively reducing the sampling rate. The CIC based decimators appear in the front end of the chain due to their low implementation complexity.

In this example, we examine a system with a target decimation rate of 10. 10-fold decimation is achieved through a cascade of 5-fold and 2-fold decimations. We assume that the first stage of the system is a CIC based structure, that is the target output rate of the CIC structure is the double Nyquist rate. It should be noted that in many practical systems, the decimations at the subsequent stages can be much higher, [4].

Figure 6 shows the frequency response of 5-fold decimation system whose target rate is the double Nyquist rate. Different from the earlier example, the passband of this system is $[-\pi/10, \pi/10]$.

As in the first example, the results for the sharpening polynomials having the orders of 4, 6 and 8 (designed for the given passband and stopband pair and $W = 7$) and the quantized version of 4th order polynomial are presented. The quantized 4th order polynomial for this case is $g(x) = -2.5x^4 + 2.25x^3 + 1.25x^2$. Figure 6 shows that the quantized design has the passband ripple of 0.25 dB and the stopband attenuation of 45 dB. These values are very much welcomed in many applications.

V. Conclusion

The main goal of this paper is to underline the utilization of the application specific sharpening filters in CIC decimation filter design in contrast to generic sharpening polynomials. It has been observed that the optimally sharpened filters can produce high performance decimators virtually eliminating the need of a secondary compensation filter in certain cases, [12], [9].

The suggested optimally sharpened CIC filters can be efficiently implemented through the Saramäki-Ritoniemi structure, [2], [3]. As noted before, the present paper has been initiated to provide an optimization framework for the sharpening of the CIC filters. The connection between the Saramäki-Ritoniemi structure has been understood during the initial review cycle of this paper. Hence the current paper can also serve as a link between two respectable lines of research for the CIC filter design.

References

[1] T. Hentschel and G. Fettweis, "Sample rate conversion for software radio," *IEEE Communications Magazine*, vol. 38, pp. 142–150, Aug. 2000.

[2] T. Saramaki and T. Ritoniemi, "A modified comb filter structure for decimation," in *Proc. IEEE Int. Symp. on Circuits and Systems*, vol. 4, pp. 2353–2356, June 1997.

[3] T. Saramäki, T. Ritoniemi, V. Eerola, T. Husu, E. Pajarre, and S. Ingalsuo, "Decimation filter," US Patent #5689449, Nov. 1997.

[4] A. Y. Kwentus, J. Z. Jiang, and A. N. W. Jr., "Application of Filter Sharpening to Cascaded Integrator-comb Decimation Filters," *IEEE Trans. Signal Process.*, vol. 45, pp. 457–467, Feb. 1997.

[5] W. A. Abu-Al-Saud and G. L. Stüber, "Modified CIC Filter for Sample Rate Conversion in Software Radio Systems," *IEEE Signal Process. Letters*, vol. 10, pp. 152–154, May 2003.

[6] G. J. Dolecek and S. K. Mitra, "A New Two-Stage Sharpened Comb Decimator," *IEEE Trans. Circuits and Syst. I*, vol. 52, pp. 1414 – 1420, July 2005.

[7] G. J. Dolecek and F. Harris, "Design of wideband CIC compensator filter for a digital IF receiver," *Elsevier Digital Signal Processing*, vol. 19, pp. 827–837, April 2009.

[8] G. J. Dolecek and M. Laddomada, "An Economical Class of Droop-Compensated Generalized Comb Filters: Analysis and Design," *IEEE Trans. Circuits and Syst. II*, vol. 57, pp. 275–279, April 2010.

[9] R. G. Lyons, *Understanding Digital Signal Processing*. Prentice Hall, 2010.

[10] J. Kaiser and R. Hamming, "Sharpening the Response of a Symmetric Nonrecursive Filter by Multiple Use of the Same Filter," *IEEE Trans. Signal Process.*, vol. 25, pp. 415–422, Oct. 1977.

[11] C. Candan, "Optimal Sharpening of CIC Filters and An Efficient Implementation Through Saramäki-Ritoniemi Decimation Filter Structure (Extended Version)." [ONLINE] http://www.eee.metu.edu.tr/∼ccandan/pub.htm.

[12] G. Molnar and M. Vucic, "Closed-Form Design of CIC Compensators Based on Maximally Flat Error Criterion," *IEEE Trans. Circuits and Syst. II*, vol. 58, pp. 926–930, Dec. 2011.

MATLAB CODES:

The following is the MATLAB function prepared for the optimal sharpening of CIC filters. The usage of this function is illustrated in the script provided. The script generates Figure 4 of this document.

MATLAB codes can also be downloaded from the author's webpage, http://www.eee.metu.edu.tr/~ccandan/pub.htm

```
function [coef,delays,hdes] = CICsharpen(downby,order,weight,passb,stopb)
%function [coef,delays,hdes] = CICsharpen(downby,order,weight,passb,stopb)
%
% Generates optimal sharpening coefficients for CIC decimation filters
% (See explanations)
%
%
% INPUTS:
% -------
% downby : decimation ratio
% order  : order of the sharpening polynomial (order = #coef + 1)
% weight : stopband / passband ripple weighting factor
% passb  : passband definition, [0 1/4] -> DC to pi/4
% stopb  : stopband definition, [1/2 1] -> pi/2 to pi
%
% if passb or stopb is not provided
%        passb=[0 1/downby]; stopb=[1/downby*1.5 1]
%
% OUTPUTS:
% --------
% coef   : coefficients of sharpening polynomial
% delays : delays required for the implementation (see explanations)
% hdes   : filter
%
% Note : You can run fvtool(hdes) at the workplace to examine the design.
%
% Cagatay Candan
% METU, Ankara, TURKEY
% Oct. 2011,
%

if exist('passb')==0, passb=[0 1/downby]; end;
if exist('stopb')==0, stopb=[1/downby*1.5 1]; end;

NFFT=128;
hp=conv(ones(1,downby),ones(1,downby))/downby^2;
h=[hp(downby:end) zeros(1,NFFT-2*downby+1) hp(1:downby-1)];
hf=real(fft(h)); hf=hf(:);

if passb(2)>1/downby, disp('Caution: Passband exceeds 1/downby'); end;
passband=(floor(passb(1)*NFFT/2):floor(passb(2)*NFFT/2)) + 1;
stopband=(floor(stopb(1)*NFFT/2):floor(stopb(2)*NFFT/2)) + 1;

% CONSTRUCT PASSBAND CONSTRAINTS
thisone=hf(passband);
dum1=[ones(size(thisone)) repmat(thisone,[1 order])];
dum2=ones(size(thisone))*[0:order];
Amat=dum1.^dum2;
Amatp1=-[ones(size(thisone)) Amat];
bp1=-1*ones(size(thisone));

Amatp2=[-ones(size(thisone)) Amat];
bp2=ones(size(thisone));

Amatp=[Amatp1; Amatp2];
bp=[bp1; bp2];

%CONSTRUCT STOPBAND CONSTRAINTS
thisone=hf(stopband);
dum1=[ones(size(thisone)) repmat(thisone,[1 order])];
dum2=ones(size(thisone))*[0:order];
Amat=dum1.^dum2;
Amats1=-[ones(size(thisone)) Amat];
bs1=zeros(size(thisone));

Amats2=[-ones(size(thisone)) Amat];
bs2=zeros(size(thisone));

Amats=[Amats1; Amats2];
Amats(:,1)=Amats(:,1)/weight;
```

```
bs=[bs1; bs2];

%CONSTRUCT EQUALITY CONSTRAINT AT DC FREQ.
Aeq=[0 ones(1,order+1)];beq=1;

%CONSTRUCT EQUALITY CONSTRAINT FOR IMAGE NULLING
AeqImg=[0 1 zeros(1,order)];beqImg=0;  %OPTIONAL
Aeq=[Aeq; AeqImg]; beq=[beq; beqImg];  %OPTIONAL

%SOLVE FOR OPTIMAL COEF.
f=[1 zeros(1,order+1)];
dum=linprog(f,[Amatp; Amats],[bp;bs],Aeq,beq);
coef=dum(2:end);

%GENERATE THE FILTER
fillength=(order-1)*(length(hp)-1)+length(hp);
center=(fillength-1)/2+1;
hdes=zeros(1,fillength);
hdes(center) = hdes(center) + coef(1);

hdum=1;
delays=zeros(order+1,1);delays(1)=center;
for dum=1:order,
    hdum = conv(hdum,hp);
    q=length(hdum); qhalf=(q-1)/2;
    index = center-qhalf:center+qhalf;
    hdes(index)=hdes(index) + coef(dum+1)*hdum;
    delays(dum+1)=center-qhalf-1;
end;

%fvtool(hdes)
if nargout==0,
figure,
plotfft(hdes,'db',12); grid on; hold on;
plot(passb,[0 0],'-r');
plot(stopb,[-10 -10],'-r');


disp('-Power-/-Coef-/-Delay-/')
disp([(0:order)' coef delays])

hdesf=fft(hdes,NFFT);
hfnew=polyval(flipud(coef),hf);dumind=linspace(0,2,NFFT);
figure,
plot(dumind,max(-100,20*log10(abs([hf hfnew hdesf(:)]))));
end;

%fvtool(hdes)
```

```
%
%THIS SCRIPT GENERATES FIGURE 4
%
downby=5;

NFFT=128*4;
hp=conv(ones(1,downby),ones(1,downby))/downby^2;
h=[hp(downby:end) zeros(1,NFFT-2*downby+1) hp(1:downby-1)];
hf=real(fft(h)); hf=hf(:);

hclas=polyval([-2 3 0 0],hf);
[a,b,hdes2]=CICsharpen(downby,4,7);hdes2f=fft(hdes2(:),NFFT);
[a,b,hdes3]=CICsharpen(downby,6,7);hdes3f=fft(hdes3(:),NFFT);
[a,b,hdes4]=CICsharpen(downby,8,7);hdes4f=fft(hdes4(:),NFFT);
close all;

plotfft(hdes2,'dB',12,'linewidth',2); grid on; hold all;
plotfft(hdes3,'dB',12,'linewidth',2);
plotfft(hdes4,'dB',12,'linewidth',2);

%prototype
hpf=polyval([1 0],hf);
plot(linspace(-1,1,NFFT),max(-70,fftshift(20*log10(abs(hpf)))),'--','linewidth',2);

%quantized 4th order
hq4f=polyval([10 -23 15 -1 0],hf);
plot(linspace(-1,1,NFFT),fftshift(20*log10(abs(hq4f))),'--','linewidth',2);


dum = axis;
axis([0 1 dum(3) 5])
xtickstr=[0:0.1:1];
set(gca,'xtick',xtickstr);set(gca,'XtickLabelMode','auto');

plot([0 1/downby],[0 0],'--k','linewidth',2);
plot([1/downby 1/downby],[dum(3) 0],'--k','linewidth',2);
plot([1/downby*1.5 1/downby*1.5],[dum(3) -10],'--k','linewidth',2);
plot([1/downby*1.5 1],[-10 -10],'--k','linewidth',2);
plot([1 1],[dum(3) -10],'--k','linewidth',2);

text(0.55,  -14, 'Stopband','fontsize',11,'fontweight','bold');
text(0.02,  -14, 'Passband','fontsize',11,'fontweight','bold');
legend('4th Order','6th Order','8th Order','Location','SouthWest');

h=xlabel('Frequency/\pi (rad/sample)');set(h,'fontsize',11);
h=ylabel('Magnitude (dB)');set(h,'fontsize',11);
```

```
function out=plotfft(input,format,times_p,varargin)
%plotfft(input,format,times_p,plotstring)
%
%Plots magnitude spectrum of input
%input: vector, output=fft(input,N);
%times: scalar, (default value = 1)
%
%Number of fft points taken: length(input)*times
%
%Oct.2005
%CC
%

warning off;

if exist('times_p')==0, times_p=1; end;
if length(times_p)==0, times_p=1; end;

N = length(input)*times_p;

out = fft(input,N);
if nargin≥2 & length(format)==2 & all(lower(format)=='db')==1
    out = max(-120,db(abs(out),'voltage'));
    plot(linspace(-pi,pi,N)/pi,fftshift(out),varargin{:});
    ylabel('Magnitude Spectrum (dB)');
else
   plot(linspace(-pi,pi,N)/pi,fftshift(abs(out)),varargin{:});
   ylabel('Magnitude Spectrum (linear)');
end;

xlabel('Discrete frequency (rad/sample)');

q=axis; q(1:2)=[-pi pi]/pi;
axis(q);
set(gca,'Xtick',[-pi -pi*3/4 -pi/2 -pi/4 0 pi/4 pi/2 3/4*pi pi]/pi)
set(gca,'Xticklabel',{'-pi';'-3/4pi';'-pi/2';'-pi/4'; '0'; 'pi/4'; 'pi/2'; '3/4pi'; 'pi'})

if nargout==0, out=[]; end;
```