Chebyshev Stopbands for CIC Decimation Filters and CIC-Implemented Array Tapers in 1D and 2D

Jeffrey O. Coleman, *Senior Member, IEEE*

***Abstract*—The stopbands of a cascaded integrator-comb (CIC) decimation filter are ordinarily very narrow, as each results from a single multiple zero. Here response sharpening with a Chebyshev polynomial, using a previously reported CIC variant, separates each such multiple zero into an equiripple stopband. By trading unneeded depth at stopband center for improved depth at the stopband edge, the latter depth improves by some 6(N-1) dB in an Nth-order system. Increased computational complexity is modest: a few low-speed additions and multiplications by small integer coefficients that can often be chosen as powers of two. Alternatively, parameters can be configured to replace the many small stopbands with one large one, and this is demonstrated here with example spatial-processing CIC designs that create pencil beams for 1D and 2D receive antenna arrays.**

***Index Terms*—digital signal processing, digital filters.**

I. Introduction

HOGENAUER [1] introduced the cascaded integrator-comb (CIC) decimator of Fig. 1 in 1981, and its multiplierless elegance has made it a DSP favorite ever since, especially when the required stopband width is small. The input-referred frequency response $H_{RM}^N(f)$ of the Nth-order system shown is just the Nth power of this first-order response with impulse-response length parameter L set to RM:

$$H_L(f) = \frac{1-\mathrm{e}^{-j2\pi fL}}{1-\mathrm{e}^{-j2\pi f}} = L\,\mathrm{e}^{-j\pi f(L-1)}\,\mathrm{D}_L(f) \qquad (1)$$

where Lth-order Dirichlet kernel (a periodic sinc, more or less)

$$\mathrm{D}_L(f) \triangleq \begin{cases} 1 & \text{if } f \in \mathbb{Z}, \text{ the integers,} \\ \dfrac{\sin(\pi f L)}{L\sin(\pi f)} & \text{if } f \notin \mathbb{Z}. \end{cases}$$

The input-referred frequency response of the Fig. 1 system therefore comprises a DC gain of L^N, a delay of $N(L-1)/2$ samples (an integral number only if L is odd), and a zero-phase response shape $\mathrm{D}_L^N(f)$. An example first-order response shape $\mathrm{D}_L(f)$ appears in Fig. 2. One of the $R-1$ anti-aliasing stopbands is centered on every Mth of the $L-1$ nulls. In all applications known to this author $M=1$, but Hogenauer mentioned the occasional use of $M=2$. All numerical design examples in this paper will use $M=1$ so that $L=RM=R$.

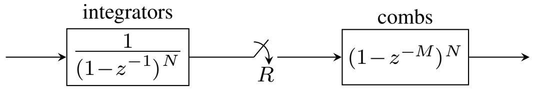

Fig. 1. Hogenauer's classic Nth-order CIC decimator typically has $M=1$. Implementation is in integer or fixed-point two's-complement arithmetic, so integrator overflow adds an integer multiple of the most-significant-bit (MSB) value to the output. Careful choice of MSB position makes this irrelevant.

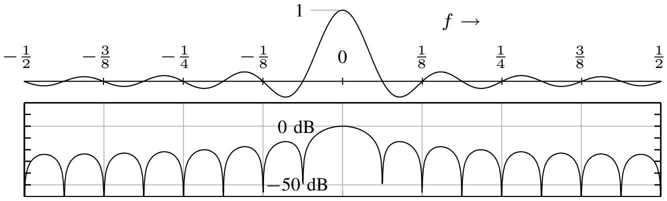

Fig. 2. Dirichlet kernel $\mathrm{D}_{RM}(f)$, top, and its decibel magnitude, bottom, for $R=8$ and $M=2$, with light vertical lines at integer multiples of $f=1/R$, where anti-aliasing stopbands are centered. This is the DC-normalized, zero-phase, input-referred frequency response of a first-order CIC decimator.

Obtaining adequate stopband rejection in challenging applications often requires some combination of (1) very narrow stopbands, (2) a high order N, and (3) varying the filter architecture in some way. In 1997 Saramäki and Ritoniemi [2] took the latter approach and introduced the modified CIC structure of Fig. 3 (more or less) to sharpen the CIC response using a polynomial. Conceptually their approach to sharpening the CIC filter generalizes on Kwentus et al. [3], which adapts to the CIC case a filter-sharpening approach of Kaiser and Hamming [4], itself a generalization of Tukey's "twicing" [5].

In [2] simple hand-tuned sharpening polynomials illustrated the sharpening concept, and choosing polynomials for best overall effect was not explored. Of course one can in principle optimize the internal coefficients for desired response characteristics, but this is very difficult because those coefficients must be integers in order to preserve the beautiful overflow-handling properties of the CIC structure. Integer programming is difficult and accessible to few designers. The alternate approach here uses sharpening polynomials scaled from Chebyshev prototypes with the aim of combining simple design and implementation with greatly improved control of stopband properties relative to what can be done with the classic CIC structure. No optimization tools are required.

A. Relationship to Recent CIC Literature

Though this paper builds on work published in and before 1997, recent literature on CIC decimators is not lacking. Here

Manuscript received August 18, 2011; revised December 22,2011 2011, April 17, 2012; accepted May 06, 2012. This work was supported by the base program of the Naval Research Laboratory. This paper was recommended by associate editor Chien-Cheng Tseng.

The author is with the Radar Division of the Naval Research Laboratory, Washington DC, USA (email: jeffc@alum.mit.edu).

Digital Object Identifier 10.1109/TCSI.2012.2206435

a sample of it is discussed.

1) Work Less Relevant to this Paper: CIC systems are often used simply as example systems on which to demonstrate interesting hardware techniques, for example the elegant optimization of FIR-filter adder systems in [6], [7] using integer linear programming. Other papers focus on CIC hardware issues, such as speed and (especially) power consumption [8]–[13]. Most notably, Dumonteix et al. [8]–[10] partition the CIC decimator into a polyphase filter-decimator followed by an efficient nonrecursive multirate structure and show that proper partitioning between these two known structures can be very power efficient when driven by short $\Delta\Sigma$-modulator output words. (Beware of one small math error: $H(z) = H_1(z)H_2(z)$ should be $H(z) = H_1(z)H_2(z^{M_1})$.) Laddomada's much later partial-polyphase implementation [14] of Lo Presti's rotated-zeros CIC response (discussed below) adapts this concept.

Many papers, like this one, aim to improve the frequency response of a CIC-based filter-decimation system. Replacing one filter-decimation step with several is common [15]–[23], as is cascading a CIC filter with compensator filters, typically but not always using simple multiplierless structures, to improve passband droop [18], [20]–[22], [24]–[30] or stopband suppression [19]–[21], [25], [27], [29], [31]. Occasionally the CIC no longer even dominates the cascade response [29]. Often portions of these cascades are realized [25], [29], [31] with the nonrecursive multirate structure mentioned above. Many of those approaches yield a sharpened filter cascade, but none actually modifies the structure of the CIC filter itself.

When actual CIC sharpening is found in recent literature, most often transition bands are sharpened only as a side effect of passband flattening, and stopbands are not broadened in any way when systems of equivalent overall order are compared. Typically Kaiser/Hamming/Kwentus sharpening with polynomial $x^2(3-2x)$ is simply adopted without modification [15], [32]–[35] in one or more CIC stages, sometimes with compensators cascaded outside the sharpened system [35]. Karnati et al. [36] modify such a "Kwentus 3-2" second stage slightly in order to permit polyphase realization for purposes of minimizing power consumption. Sharma et al. [34] report on a Xilinx FPGA realization. In [37] the impulse response of a Kwentus 3-2 sharpened CIC stage is realized not in Kwentus form but as conventional polyphase decimation realizing a frequency response designed using the Kwentus approach offline to limit coefficients to a few powers of two. Dolecek and Harris [23], [38], [39] go further conceptually to greatly increase passband flattening by setting x to a passband-compensated CIC response in sharpening polynomial $x(2-x)$. Occasional papers [16] consider many sharpening polynomials.

2) Surprises in That Body of Work: The impressive body of literature briefly discussed above has some surprising features. Certainly one cannot help but notice the impressive number of new ideas per paper averaged by some authors.

Independent reporting of the same systems by different researchers is also relatively common. For example, the authors of [15], [32] each propose two CIC stages with the second Kwentus 3-2 sharpened and the first implemented as one of the two integrator-free forms of Dumonteix (cited above). One author [32] chooses the nonrecursive form but mentions the polyphase alternative. The other [15] uses the polyphase form but mentions the nonrecursive alternative in a subsequent, more thorough examination of the architecture [16]. In another example, Laddomada and Mondin [40] and Dolecek and Mitra [17], in papers both submitted before either appeared, each modify second-stage Kwentus 3-2 sharpening by rotating two zeros in each of its stopbands (see next section). Journal paper [40] is of course far more thorough than conference paper [17], but the latter was actually submitted first.

And it is not only related work that gets overlooked. In [41], [42], the CIC definition is modified slightly so that its implementation can be structured, for efficiency, as an input polyphase decomposition feeding identical recursive, reduced-rate CIC decimators in parallel arms that drive an output sum. The authors then overlook the trivial possibility of moving those identical arm systems past the output sum, where they'd become one. The overall system would become a cascade of two CIC decimators with the first in polyphase form, a structure exploited earlier by others [18], [20], [39].

3) Work More Relevant to This Paper: The major alternative approach to broadening the CIC stopband by modifying the CIC filter structure itself is the elegant zero-rotation approach of Lo Presti and Laddomada. Lo Presti's original concept [43]–[45] rotated CIC stopband zeros slightly in the z plane to space them as desired, but the nominal recursive implementation was unstable because imperfect relationships between real coefficients caused pole-zero cancellation to fail. Laddomada then solved the stability problem, first with a nonrecursive implementation [46], later with a partially polyphased version of same [14], and finally using the recursive structure with mathematically ideal coefficient relationships for perfect pole-zero cancellation [26]. Both Laddomada and Mondin [40] and Dolecek and Mitra [17] modify Kwentus 3-2 sharpening by rotating two zeros in each stopband, a nice combination of ideas. Laddomada [47] generalizes to higher-order sharpening polynomials and tabulates required hardware.

This paper's approach broadens stopbands, so the proper comparison is with Lo Presti/Laddomada zero rotation (of which Dolecek/Mitra is a special case). However, zero rotation gives stopbands the same widths but different depths, whereas this paper's approach gives them the same depth and different widths, and while the present approach retains the overflow immunity of the original CIC system (see Section II-C), their approach does not, because of its noninteger coefficients. Further, the high-speed part of the present system is simpler.

B. Structure of the Paper

Section II analyzes the Saramäki-Ritoniemi CIC structure. Then Section III shows how sharpening with a Chebyshev-derived polynomial of degree N broadens each Nth-order zero of Hogenauer's original CIC response into a Chebyshev stopband, with stopband width and depth determined by the polynomial degree N and an internal scaling parameter γ. Section IV details the design process that maps the desired Chebyshev sharpening into the Fig. 3 structure.

Section V explores parameter choices that merge a CIC filter's multiple stopbands into a single, large stopband. Two

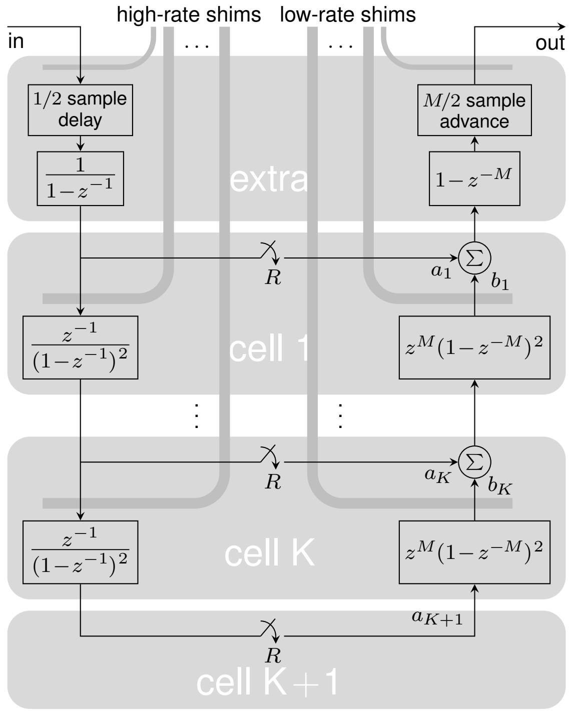

Fig. 3. Zero-phase form of a polynomial-sharpened variant of the CIC decimator of even or odd order $2K$ or $2K+1$ according as the "extra" section is omitted or included. Specific choices for integer weights $\{a_i\}$ and $\{b_i\}$ yield Chebyshev stopbands.

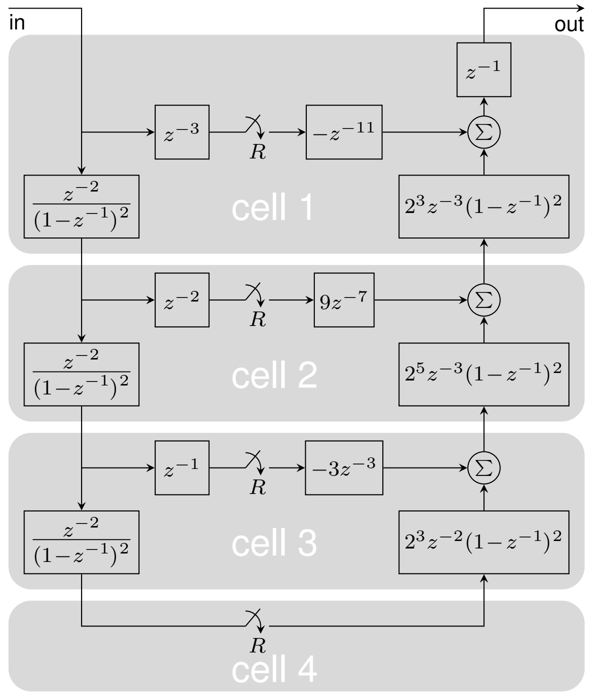

Fig. 4. Causal and fully pipelined example of an even-order Fig. 3 system realizes frequency response $G_{L,N,\gamma}(f)$ of (8) to within a scale factor and delay, with specific coefficients $a_1,\ldots,a_4$ and $b_1,\ldots,b_3$ corresponding to $M=1$ or $L=R$, to $N=6$, and to $\gamma=2$. Its frequency response for decimation ratio $R=5$ appears on the lower right in Fig. 7.

examples each spatially realize a taper to equip an array antenna with a pencil beam. One example is for a uniform line array. The other realizes a nonseparable 2D taper for a square array on the square lattice. However, parameters may or may not exist that will create practical tapers for real systems, as the CIC structure inherently favors sidelobes lower than necessary (or achievable in real hardware) and receiver SNR losses higher than are typically tolerable. However, in spite of appearances, arrays are not the central point of the section. Instead, the point is that a CIC structure can be considered for narrow-beam, single-stopband DSP applications. Arrays here are primarily a vehicle for exploring the topic.

II. The Saramäki-Ritoniemi CIC Structure

A. System Description

Fig. 3 shows the proposed realization system in zero-phase form. Cells $1,\ldots,K$ differ only in the integer weights $\{a_i\}$ and $\{b_i\}$ of the "Σ" linear combiners, and cell $K{+}1$ comprises only decimation and scaling by a_{K+1}. As shown the low-rate right side of the system is not causal, but causality can be restored by inserting shimming delays where the "low-rate shims" lines cross signal paths. Those shims are of M or $M/2$ samples respectively according as the shim lines shown are wide or narrow. Similarly, the integrator additions in the high-rate subsystem on the left can be pipelined by inserting shims of a whole or half sample where indicated by wide or narrow "high-rate shims" lines. Once such high-rate shimming is added, $z^{-1}/(1-z^{-1})$ delayed integrators can be used throughout. Similarly, inserting delays of three samples and one sample along wide and narrow "low-rate shims" lines respectively puts each comb in pipelined form $z^{-1}(1-z^{-M})$ and gives each explicit summer pipeline delays on both inputs.

The structure that Saramäki and Ritoniemi presented in [2] is precisely the Fig. 3 system in causal form with high-rate pipelining, with input scaling, with $b_1,\ldots,b_K=1$, and with any number of "extra" sections. Here we need at most one such extra section, and the other differences are less than they seem. The overall scaling that Saramäki and Ritoniemi provided explicitly at the system input—an arbitrary but reasonable location—is actually assumed here as well, though it is not shown in Fig. 3. Further, the $\{b_i\}$ used here could certainly be pushed upstream and folded into the $\{a_i\}$ with no loss of generality. Keeping the $\{b_i\}$ in place here instead aids in multiplierless realization by permitting one b_i to replace what would otherwise be additional factors in many of the $\{a_i\}$. For example, setting $b_1=3$ is equivalent to leaving it at unity but including a factor of 3 in each of $a_2,\ldots,a_{K+1}$. Of course further low-rate shims may be required if linear combining with coefficients a_k and b_k is realized in pipelined form.

Fig. 4 is a causal and fully pipelined example version of Fig. 3 with $M=1$, four cells, no extra section, and coefficients derived below in Section IV of $a_1=-1$, $b_1=2^3$, $a_2=9$, $b_2=2^5$, $a_3=-3$, $b_3=2^3$, and $a_4=1$. Single pipeline delays on each summer input, placed using the procedure described above, have been replaced in Fig. 4 by single pipeline delays on summer outputs. The summers in cells 1 and 3 would be realized as differencers, since a_1 and a_3 are negative. All but two coefficients are powers of two and so yield zero-

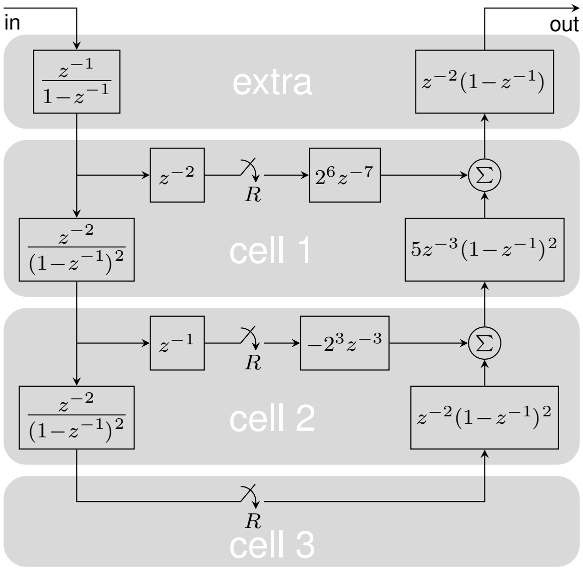

Fig. 5. Causal example of an odd-order Fig. 3 system with integrators, combs, and summers pipelined and realizing frequency response $G_{L,N,\gamma}(f)$ of (8) to within a scale factor and delay. Here a_1, a_2, a_3, b_1, and b_2 correspond to $N = 5$ and $\gamma^2 = 2^{-5}5$ with (as usual) $M = 1$ or $L = R$. See Fig. 9 for the input-referred magnitude response when $R = 16$.

cost multiplication in a parallel implementation. The remaining coefficients have absolute values $9 = 8 + 1$ and $3 = 2 + 1$, so those multiplies can each be realized with a single adder and pipelined using registers taken from shims already present. Fig. 5 is similar but has an "extra cell" because there N is odd.

B. Frequency Response

The Fig. 3 system's frequency response is simple to derive. Noble-identity replacement of the decimators with a single output decimator first replaces each M-sample time interval on the right with an L-sample interval (remembering $L = RM$). We can then write the transfer function of everything preceding that output decimator in either of two straightforward forms. If we ignore the so-called extra section for now, this input-referred equivalent filter has transfer function

$$\sum_{k=1}^{K+1} b_1 \cdots b_{k-1}\, a_k \left(z^{L-1} \Big(\frac{1 - z^{-L}}{1 - z^{-1}} \Big)^2 \right)^{k-1}, \tag{2}$$

where $b_1 \cdots b_{k-1}$ means $\prod_{i=1}^{k-1} b_i$ and so degenerates to unity—the multiplicative identity—when $k = 1$ gives it no factors. The summand is the input-referred transfer function of the signal path through the horizontal arm of cell k in Fig. 3.

But it is both more straightforward and ultimately more useful to let C_k denote the input-referred transfer function of cell k, from its upper input to its upper output, and then to write recursion relation

$$C_k = \begin{cases} a_k + b_k\, C_{k+1}\, z^{L-1} \Big(\dfrac{1 - z^{-L}}{1 - z^{-1}} \Big)^2 & \text{for } k = 1, \ldots, K, \\ a_{K+1} & \text{for } k = K+1. \end{cases}$$

so that C_1 effectively realizes polynomial (2) in the variable $z^{L-1}(1-z^{-L})^2/(1-z^{-1})^2$ in classic Horner-algorithm form.

Rather than show the dependence of C_k on z explicitly, let us substitute $\mathrm{e}^{-j2\pi f}$ for unit delay z^{-1} and write C_k as a function of f in this form, used exclusively henceforth:

$$C_k(f) = \begin{cases} a_k + b_k\, X^2(f)\, C_{k+1}(f) & \text{for } k = 1, \ldots, K, \\ a_{K+1} & \text{for } k = K+1, \end{cases} \tag{3}$$

$$X(f) \triangleq L\, \mathrm{D}_L(f). \tag{4}$$

Here intermediate function $X(f)$ is defined for convenience.

Optionally, the extra section of Fig. 3 can be used to compute C_0 as C_1 times $(1 - z^{-L})/(1 - z^{-1})$ and advanced in time by $(L - 1)/2$ samples or, in the frequency domain,

$$C_0(f) = X(f)\, C_1(f).$$

C. Irrelevance of Overflow

In Fig. 3 there is one explicit addition in each of cells $1, \ldots, K$. Further, each integrator and each comb requires one addition (or subtraction, a special case of addition). Each addition has the potential of overflowing, and in the case of integrators, overflow is certain except in the case of very special inputs.

Let us briefly review overflow irrelevance in two's-complement arithmetic. In two's-complement an integer y is in effect represented in ρ-bit unsigned binary as $y + m2^\rho$, where m is the one integer for which $0 \le y + m2^\rho < 2^\rho$. A two's-complement adder simply adds the unsigned integers and discards the leftmost carry bit, a value in $\{0, 2^\rho\}$, to obtain yet another ρ-bit unsigned binary quantity $y + m2^\rho$ with y the desired integer and with integer m unknown.

Similarly, two's complement scaling by a coefficient c operates on unsigned quantity $y + m2^\rho$ to form product $cy + cm2^\rho$ and discards the leftmost carry bit. *If coefficient c is an integer*, this result remains an unsigned integer of form $y + m2^\rho$, with y the desired integer result and with integer m unknown.

It follows (technically by induction on the sample number) that any system comprising only addition and integer-coefficient scaling of this sort, in whatever combination, must yield a system output $y_{\text{raw}} = y_{\text{desired}} + m2^\rho$.

To determine $y_{\text{desired}} = y_{\text{raw}} - m2^\rho$, we must know *a priori*, perhaps by system dynamic-range analysis, that y_{desired} falls in some range $n_{\text{low}}, \ldots, n_{\text{high}}$ for which $n_{\text{high}} - n_{\text{low}} < 2^\rho$, because such a restriction is only compatible with one choice of m. Custom takes the range to be $-2^{\rho-1}, \ldots, 2^{\rho-1} - 1$, in which case the most significant bit (bit value $2^{\rho-1}$) of y_{raw} determines the correct m unambiguously:

$$m = \begin{cases} 0 & \text{if } \mathrm{MSB}(y_{\text{raw}}) = 0, \\ -1 & \text{if } \mathrm{MSB}(y_{\text{raw}}) = 1. \end{cases}$$

This is the origin of the usual two's-complement sign convention, that in four-bit two's complement for example, unsigned four-bit values $0, \ldots, 7$ represent themselves, while unsigned four-bit values $8, \ldots, 15$ have $2^4 = 16$ subtracted in interpretation so that they represent values $-8, \ldots, -1$.

By this reasoning, a two's-complement realization of Fig. 3 will be overflow-free if coefficients $a_1, \ldots, a_{K+1}$ and $b_1, \ldots, b_K$ are integers and wordwidth ρ is such that the ideal system output always falls into range $-2^{\rho-1}, \ldots, 2^{\rho-1} - 1$. We will take care below to meet the integer-coefficient requirement. The rest is up to the system designer.

TABLE I
CHEBYSHEV POLYNOMIALS OF THE FIRST KIND THROUGH DEGREE 10, IN TWO FORMS

$$\begin{aligned}
T_0(x) &= 1\\
T_1(x) &= x\\
T_2(x) &= -1 + 2x^2\\
T_3(x) &= -3x + 4x^3\\
T_4(x) &= 1 - 8x^2 + 8x^4\\
T_5(x) &= 5x - 20x^3 + 16x^5\\
T_6(x) &= -1 + 18x^2 - 48x^4 + 32x^6\\
T_7(x) &= -7x + 56x^3 - 112x^5 + 64x^7\\
T_8(x) &= 1 - 32x^2 + 160x^4 - 256x^6 + 128x^8\\
T_9(x) &= 9x - 120x^3 + 432x^5 - 576x^7 + 256x^9\\
T_{10}(x) &= -1 + 50x^2 - 400x^4 + 1120x^6 - 1280x^8 + 512x^{10}
\end{aligned}$$

$$\begin{aligned}
T_3(x) &= x(-3 + 4x^2)\\
T_4(x) &= 1 + 8x^2(-1 + x^2)\\
T_5(x) &= x(5 + 4x^2(-5 + 4x^2))\\
T_6(x) &= -1 + 2x^2(9 + 8x^2(-3 + 2x^2))\\
T_7(x) &= x(-7 + 8x^2(7 + 2x^2(-7 + 4x^2)))\\
T_8(x) &= 1 + 32x^2(-1 + x^2(5 + 4x^2(-2 + x^2)))\\
T_9(x) &= x(9 + 8x^2(-15 + 2x^2(27 + 4x^2(-9 + 4x^2))))\\
T_{10}(x) &= -1 + 2x^2(25 + 8x^2(-25 + 2x^2(35 + 8x^2(-5 + 2x^2))))
\end{aligned}$$

III. CHEBYSHEV STOPBANDS

A. How the Chebyshev Ripples are Created

The Chebyshev polynomial $T_N(x)$ of the first kind of degree N, the first few of which are listed in the top half of Table I, is generated by the well-known recursion

$$T_N(x) = \begin{cases} 1 & \text{for } N = 0,\\ x & \text{for } N = 1,\\ 2xT_{N-1}(x) - T_{N-2}(x) & \text{for } N > 1 \end{cases} \tag{5}$$

and has the useful properties illustrated in Fig. 6.

The central point of this paper is that the Fig. 3 structure can be used to realize decimation by R preceded by one of the frequency responses

$$C_0(f) = T_{2K+1}(\gamma L\,\mathrm{D}_L(f)) \tag{6}$$

$$C_1(f) = T_{2K}(\gamma L\,\mathrm{D}_L(f)), \tag{7}$$

according as the extra section is included or not, to within a gain factor. Let us defer to the next section consideration of why this is so and first examine why the expressions on the right amount to CIC responses sharpened with polynomials to create Chebyshev stopbands and how scaling constant γ affords some control over stopband width.

For convenience let us first put the right sides of (7) and (6) in a common form by writing them with $N = 2K$ and $N = 2K + 1$ respectively. If we then recognize $X(f)$ from (4) and DC normalize the result, we obtain a normalized frequency response parameterized by integers L and N and scaling parameter γ,

$$G_{L,N,\gamma}(f) \triangleq \frac{T_N(\gamma X(f))}{T_N(\gamma L)}. \tag{8}$$

The DC normalization is not shown explicitly in Fig. 3 but is assumed. A power-of-two component of the normalization factor is certainly trivial to include, and any residual could be located in many places in a typical application system.

The example in Fig. 7 shows why response $G_{L,N,\gamma}(f)$ has Chebyshev stopbands. On the upper right, scaled Dirichlet kernel $X(f) = L\,\mathrm{D}_L(f)$ has a passband (unmarked) centered at integer frequencies—only the one at $f = 0$ is shown—and single-null stopbands centered on other integral multiples of $f = 1/L$. Here there are four such one-null stopbands per period. The frequencies where $\gamma X(f) = \pm 1$ or, equivalently, where $\mathrm{D}_L(f) = \pm 1/(\gamma L)$ become band-edge frequencies on the lower left, where $T_N(\gamma X)$ is shown on a decibel scale versus X and normalized, by $T_N(\gamma L)$, to 0 dB at $X(0)$. On the lower right plotting versus f rather than $X(f)$ yields the desired DC-normalized magnitude response.

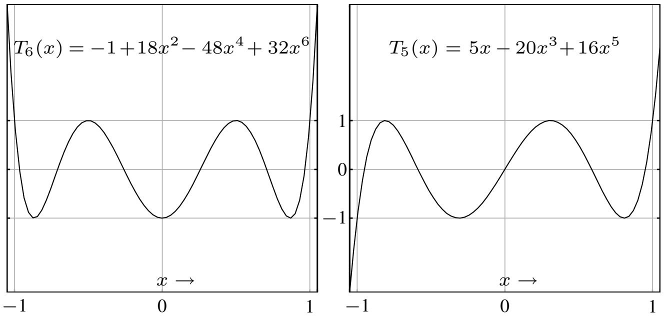

Fig. 6. The degree-N Chebyshev polynomial $T_N(x)$ of the first kind contains only even powers or only odd powers of x according as N is even or odd, oscillates between ±1 for x between ±1, and has $2^{N-1}x^N$ as its highest-degree term, which of course it approaches asymptotically for large $|x|$.

B. Performance Comparisons

Technically approximate but practically accurate performance analyis can be carried out using the fact that for $x \gg 1$,

$$T_N(x) \approx 2^{N-1}x^N. \tag{9}$$

In the passband $X(f) \approx L$ by definition (4), so in the usual $\gamma L \gg 1$ case, approximation (9) applies and (8) becomes

$$G_{L,N,\gamma}(f) \approx \frac{2^{N-1}(\gamma X(f))^N}{2^{N-1}(\gamma L)^N} = \mathrm{D}_L^N(f)$$

the normalized zero-phase frequency response of the original Hogenauer CIC filter. Chebyshev sharpening therefore leaves passbands unchanged. This can be seen in the Fig. 7 example.

The creation of those Chebyshev-sharpened stopbands relies on the unit-amplitude ripples of Chebyshev polynomial $T_N(x)$ for $|x| \le 1$, as these become ripples in $T_N(\gamma X(f))$ for f between bandedges. The first stopband, for example in Fig. 7, is the narrowest, because $X(f)$ is steepest in that band. Further, the lower edge of that first stopband is closer to the nominal $f = 1/L$ stopband center than is the upper stopband edge. If that first stopband's lower edge is f_e, Fig. 7 implies that $T_N(\gamma L D_L(f_e)) = 1$. But the largest argument at which a Chebyshev polynomial attains unit value is unity, so

$$\gamma L D_L(f_e) = 1. \tag{10}$$

It follows then from definition (8) that $G_{L,N,\gamma}(f_e) = 1/T_N(\gamma L) \approx 2^{N-1}(\gamma L)^N$ in the usual $\gamma L \gg 1$ case, so

$$\text{dB stopband depth} \approx N \times 20\log_{10}(2\gamma L) - 6.02 \text{ dB}. \tag{11}$$

The original Hogenauer CIC filter of order N has stopband depth at that bandedge f_e of $(LD_L(f_e))^N$, which by (10) is just $(1/\gamma)^N$, giving the present scheme an advantage in

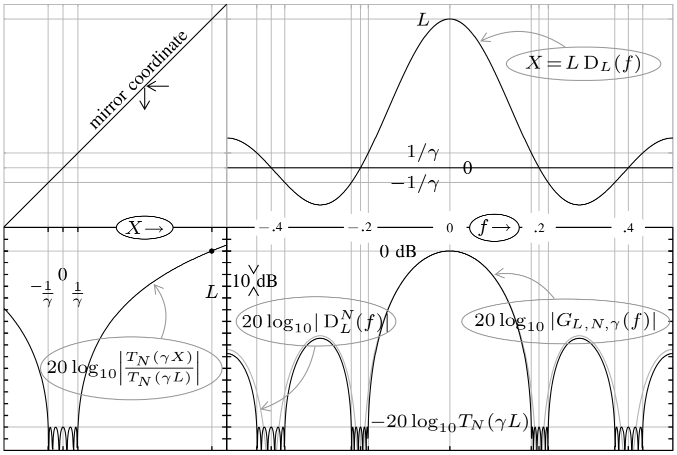

Fig. 7. The zero-phase, DC-normalized, input-referred dB magnitude response $G_{L,N,\gamma}(f)$ of the Fig. 3 structure equipped with Chebyshev stopbands is shown on the lower right for $L = 5$, $N = 6$, $\gamma = 2$, plotted over a gray reference, the dB magnitude of conventional CIC response $D_L^N(f)$. To see how the stopbands arise, proceed counterclockwise starting on the upper right.

stopband depth of a factor of approximately 2^{N-1} or about $6.02 \times (N-1)$ dB. Of course going on to compare the performance of the present system to a Kaiser/Hamming/Kwentus-sharpened CIC system would be unfair to the latter, which for a given total number of integrators trades away some of the standard stopband performance for improved passband flatness that the present system makes no attempt to obtain. (However, the passband-flattening approaches suggested in Section VI as further research should indeed be compared to the Kaiser/Hamming/Kwentus approach of [3].) Comparison with the polynomial-design approach of Saramäki and Ritoniemi [2] is deferred to Section IV-B.

C. Setting the Stopband Width and Depth

Fig. 8 illustrates selection of stopband width and depth parameters in the present approach. For the traditional anti-aliasing application, the first-stopband edge frequency f_e determines the maximum acceptable γ through (10) as

$$\gamma_{\text{max}} = \frac{1}{L\,\mathrm{D}_L(f_e)}. \tag{12}$$

Once γ is chosen, at least approximately, N can be set to the least integer that yields sufficient stopband depth according to (11). This is a trial-and-error process, since N determines which γ values are simple to implement. This will become clearer in the examples of the next section.

The special case that pairs $\gamma = 1$ with an odd N always combines the stopbands on either side of $f = 1/2$ into one equiripple stopband. This is because $X(0.5) = L\,\mathrm{D}_L(0.5) = 1$ irrespective of L.

IV. Realization

A. Choosing γ for Implementation Convenience

Suppose we wish to realize $G_{L,6,\gamma}(f)$, an $N = 6$ version of the numerator of definition (8), with the intent of eventually normalizing with a reasonable approximation of the denominator of (8) and perhaps setting $L = 5$ and $\gamma = 2$ to obtain the response shown in Fig. 7.

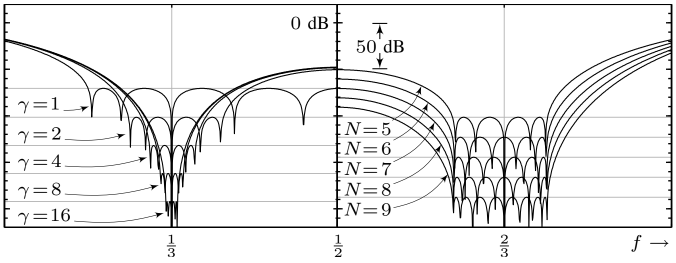

Fig. 8. The two Chebyshev stopbands of $G_{L,N,\gamma}(f)$ for impulse-response length parameter $L = 3$, on the left with Chebyshev degree $N = 5$ and several choices of scaling parameter γ, and on the right with $\gamma = 2$ and several choices of N.

The first step is to put Chebyshev polynomial $T_6(x)$ from the top half of Table I in form

$$T_6(x) = -1 + x^2(18 + x^2(-48 + 32x^2)).$$

using a straightforward Horner-algorithm factorization. The high-degree coefficient of a Chebyshev polynomial is always a power of two, and matters will be clearer if we push as many of its factors to the left to the extent possible, though this doesn't really affect implementation beyond the question of where binary points go. Here this yields

$$T_6(x) = -1 + 2x^2(9 + 8x^2(-3 + 2x^2)).$$

For convenience the bottom half of Table I lists these forms for Chebyshev polynomials through $T_{10}(x)$.

At this stage we replace x by $\gamma X(f)$ as per the numerator of (8). When γ^2 is an integer, a simple labeling of constants and parenthesized sums matches the result up with recursion (3) and the Fig. 3 implementation. Here trivial items $C_4 = a_4 = 1$ are not labeled:

$$T_6(\gamma X) = \overset{C_1}{(}-\underset{a_1}{1}+2\underset{b_1}{\gamma^2}X^2\,\overset{C_2}{(}\underset{a_2}{9}+8\underset{b_2}{\gamma^2}X^2\,\overset{C_3}{(}-\underset{a_3}{3}+2\underset{b_3}{\gamma^2}X^2))). \tag{13}$$

If this were an odd-degree polynomial, a factor of γ in front would then be folded into the overall scaling.

Where to go next very much depends on the choice of γ^2, and in fact this is the appropriate place to ask what γ^2 choices are reasonable. Certainly (13) has the integer coefficients required to give the Fig. 3 system the proper overflow behavior for any $\gamma^2 = \ell/2$ with $\ell \in \mathbb{Z}$. As a special case, when $\gamma^2 = 2^\ell$ with $\ell \geq -1$, direct substitution makes three of the coefficient multiplies realizable as simple shifts:

$$T_6(\gamma X) = -1 + 2^{1+\ell}X^2(9 + 2^{3+\ell}X^2(-3 + 2^{1+\ell}X^2)). \tag{14}$$

A CIC system incorporating this specific polynomial is realized in Fig. 4.

Perhaps surprising is that $1/2$ is not the only fractional value acceptable for γ^2. When $\gamma^2 = 2^\ell$ with $\ell < -1$, the fractional

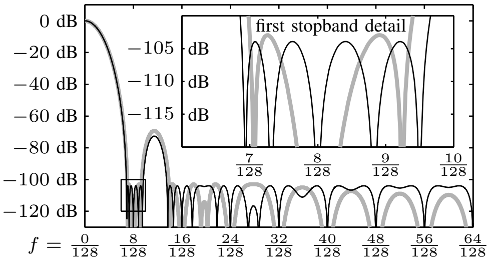

Fig. 9. Magnitude responses of the $N = 5$, $L = 16$, $\gamma = \sqrt{2^{-5}5}$ design of Fig. 5 (thin dark line) and a reference design (thick gray line) from [2]. Both are normalized here to 0 dB at $f = 0$.

coefficients in (14) can be factored out algebraically to obtain

$$\begin{aligned} &T_6(\gamma X) \\ &= 2^{5+3\ell}(-2^{-3\ell-5} + X^2(2^{-2\ell-4}9 + X^2(-2^{-\ell-1}3 + X^2))). \end{aligned} \tag{15}$$

Leading factor $2^{5+3\ell}$ can be absorbed into the overall scaling and so need not be realized directly. The other powers of two are nonnegative integers.

A simple generalization of (15) in principle permits γ^2 to be chosen to be arbitrarily close to any positive real number. Simply set $\gamma^2 = \eta 2^\ell$ with $\ell < -1$ as before but now with η any positive integer. Then

$$\begin{aligned} &T_6(\gamma X) = \\ &2^{5+3\ell}(-2^{-3\ell-5} + \eta X^2(2^{-2\ell-4}9 + \eta X^2(-2^{-\ell-1}3 + \eta X^2))), \end{aligned}$$

which has only integer coefficients internally.

The practical question then is not what values of γ^2 are allowable but what values lead to efficient implementations. Certainly $\gamma^2 = 2^\ell$ with $\ell \geq -1$ is efficient. The power-of-two multiplies in (14) are trivial shifts, and its linear combiners require exactly five adders using relationships $-3u = -4u + u$ and $9v = 8v + v$.

Another efficient choice results from making b_K in a formulation like (13) a power of two times one or more of the integer factors of $|a_K|$. For example, in (13) let $\gamma^2 = 2^\ell 3$, where the 3 is chosen specifically to be equal to integer $|a_3|$. Then when $\ell \geq -1$,

$$\begin{aligned} T_6(\gamma X) &= -1 + 2^{\ell+1}\ 3X^2(9 + 2^{\ell+3}3X^2(-3 + 2^{\ell+1}3X^2)) \\ &= -1 + 2^{\ell+1}27X^2(1 + 2^{\ell+3}\ X^2(-1 + 2^{\ell+1}\ X^2)). \end{aligned}$$

The one nontrivial coefficient multiply is $27u = 4v - v$ with $v = 8u + u$, so the linear combiners again require five adders. Similarly, when $\ell < -1$, adapting the idea in (15) yields

$$\begin{aligned} &T_6(\gamma X) \\ &= 2^{5+3\ell}(-2^{-3\ell-5} + 27X^2(2^{-2\ell-4} + X^2(-2^{-\ell-1} + X^2))) \end{aligned}$$

and so requires the same five adders. It is by happenstance that the number of adders—five—is unchanged by the inclusion of the $|a_3| = 3$ factor in γ^2. This trick in some cases increases or reduces adder count.

B. An Example Design from Saramäki and Ritoniemi

The example response in Fig. 7 and discussed above has $\gamma > 1$ and L distinct Chebyshev stopbands. Section V below considers the $\gamma \ll 1$ case that leads to a single stopband. The design examples of Saramäki and Ritoniemi [2] happen to fall midway between those two γ regimes, so let us compare with their results here, at a midway point in the discussion.

The reference response in Fig. 9 is the "$K = 16$, $A_{\text{min}} = 90$ dB, $M = 3$, $N = 1$" (different variable names than here) design from Fig. 6 of [2]. That design has our $f_e = \frac{7}{128} f_s$, from which (12) gives $\gamma^2_{\text{max}} \approx 0.1996$ so that stopband depth (11) exceeds $A_{\text{min}} = 90$ dB for $N > 4.2$ or so. With $N = 5$ and $\gamma^2 = 2^\ell 5$, the lower T_5 Table I entry with $x = \gamma X$ yields $T_5(\gamma X)/(5\sqrt{2^\ell 5}) = X(1 + 2^{\ell+2}5X^2(-1 + 2^{\ell+2}X^2))$, implying simple implementation. The largest ℓ with $\gamma \leq \gamma_{\text{max}}$ is $\ell = \lfloor \log_2(\gamma^2_{\text{max}}/5) \rfloor = -5$, for a stopband depth (11) of ≈ 104 dB. The above polynomial scaled by 2^6 is realized in Fig. 5 as $X(2^6 + 5X^2(-2^3 + X^2)))$ with causality and pipelining. The thin line in Fig. 9 is its magnitude response.

This design's performance is comparable to that of the reference design from [2], and the two approaches' performance is in fact similarly close for the other two designs of their Fig. 6. But the design here can be realized with integer coefficients and therefore be free of overflow, because γ was quantized by hand to an acceptable value. In [2] real polynomial coefficients—from 1 to 4 of them in their examples—are obtained and must be hand quantized. The effects of this are less predictable than when γ is quantized.

The present approach is also simpler, as only a few parameter calculations are needed. The design approach in [2], however, uses a McClellan transformation in an unusual 1D-to-1D manner (in the spirit of Section 2 of [48]), and it also requires a nonstandard optimization. For the latter they suggest a custom-weighted Parks-McClellan formulation, but this author found a linear program to be easier to set up [49].

V. Single Stopbands and Arrays

A value of γ below unity creates a stopband centered on $f = 0.5$, and values sufficiently below unity yield only that one stopband, which becomes large, and no other. Multiple stopbands are eliminated. The implied design choices are interesting in both 1D and 2D. A 2D application will be spatial by nature, so the discussion below makes the similar 1D system spatial as well for easy comparability.

Readers without array backgrounds can find a tutorial on the basics of receive arrays from a signal-processing point of view in roughly the first half of [50]. For an explanation of the usual "taper loss" measure of receive SNR and transmit power efficiency, see Section 3.2.3 of [48].

A. An Example Uniform Line Array

Let us explore a design with $\gamma < 1$ using a simple $N = 3$ example. A small γ is appropriate, so let us apply the last section's logic to $T_3(x) = x(-3 + 4x^2)$ from Table I and set $4\gamma^2$ to $2^{-\ell}3$ for some integer $\ell \geq 0$. Definition (8) gives, if

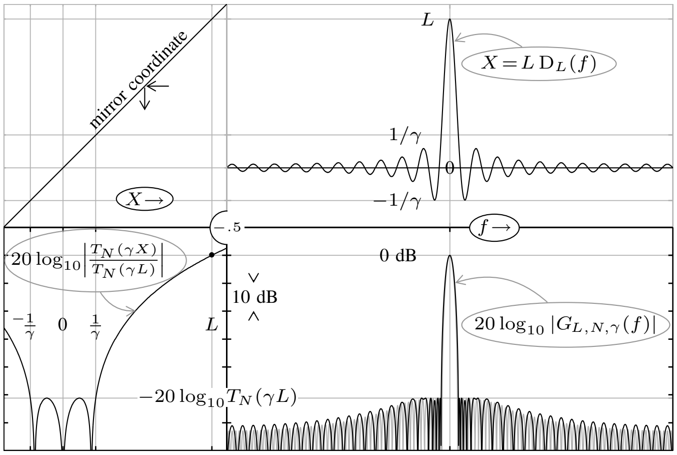

Fig. 10. Boresight-normalized dB magnitude of the array factor $G_{42,3,\gamma}(f)$ with $\gamma = 2^{-4}\sqrt{3}$ realized by the 124-element Fig. 12 array structure, plotted over a gray Taylor reference response with $\bar{n} = 18$ and sll $= -51$ dB.

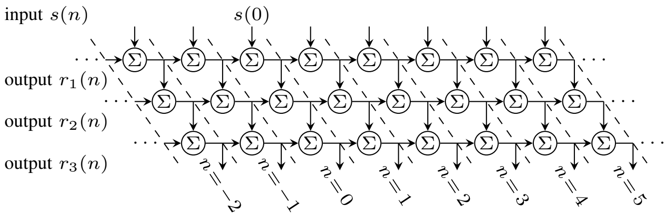

Fig. 11. When $s(0)$ is the only nonzero input to this "summer wall," outputs $r_1(n) = s(0)$, $r_2(n) = (n+1)s(0)$, and $r_3(n) = (n+1)(n+2)s(0)/2$ for $n \geq 0$, $n \geq -1$, and $n \geq -2$ respectively. First, second, and third differences of $r_1(n)$, $r_2(n)$, and $r_3(n)$ respectively are always zero when constructed from samples taken in those regions, where outputs are of degree 0, 1, and 2 in n. Input $s(0)$ and inputs to its left cannot contribute to such differences.

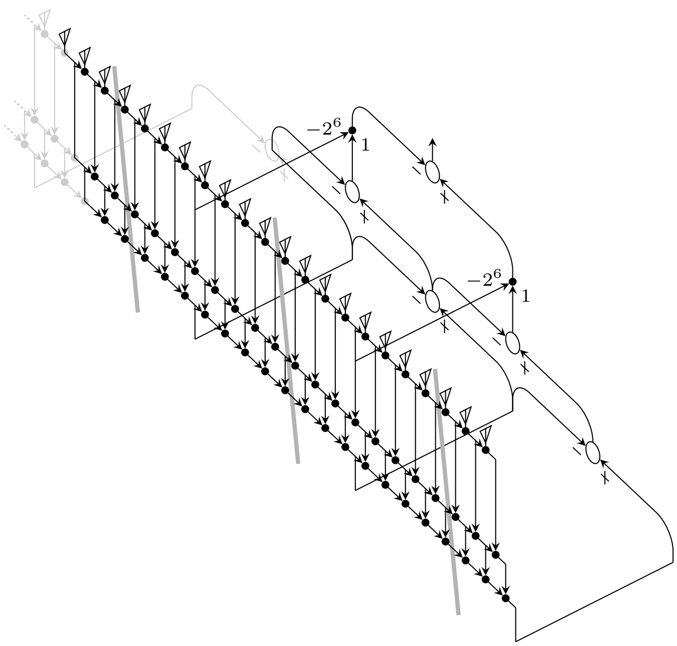

Fig. 12. A "$K = 1$ with extra" spatial version of Fig. 3 "unrolled" to create a single output sample realizes a tapered beamforming sum for a uniform line array of antenna elements. Summers and comb differencers are respectively dark balls and open circles. Faint material on the upper left was removed using the Fig. 11 result. Weights $a_1 = -2^{-6}$ and $b_1 = 1$ shown yield the Fig. 10 array factor, assuming that periodic extension of the element-summer structure transforms this $L = 8$ array with 22 elements into an array with $L = 8 + 34 = 42$ and 124 elements by adding 34 elements at each gray bar.

we suppress f dependence on the right,

$$G_{L,3,\gamma}(f) = \frac{T_3(\gamma X)}{T_3(\gamma L)} = \frac{\gamma X(-3+4\gamma^2X^2)}{\gamma L(-3+4\gamma^2L^2)} = \frac{X(X^2-2^\ell)}{L\,(L^2-2^\ell)}. \tag{16}$$

The form chosen for γ has made $a_1 = -2^\ell$ and $b_1 = 1$ trivial to realize. Choosing $\ell = 6$ and $L = 42$ then yields normalization constant $L(L^2 - 2^\ell) \approx 1.0895 \times 2^{16}$, a 16-bit shift and a residual of some 0.74 dB to be made up elsewhere.

With these parameters, the decibel magnitude of response (16) and the mappings that produce it are as shown in Fig. 10. The response shape suggests an antenna application: a uniform-line-array receive antenna. The filter frequency response becomes an array factor, and taking θ to be the angle between the propagation direction and the line normal, f becomes the normalized spatial frequency, along the line, given by $\sin(\theta)$ times element spacing in wavelengths. A classic Taylor array factor is included in Fig. 10 for reference. In practical terms, both array factors have very low sidelobe levels and concomitantly weak beamwidth and SNR-performance numbers. Assuming classic half-wavelength element spacing, the 3 dB points are separated by $1.41°$ and $1.27°$ on the CIC Chebyshev and reference Taylor beams respectively. The taper losses—SNR losses relative to the theoretical optimum for this number of elements—are respectively 2.04 dB and 1.57 dB.

What is most interesting about this example design is the spatial signal-processing structure that realizes the one required spatial output sample. The single-summer recursive integrator that is key to the elegance of the temporal CIC structure is lost, because what was time index n now becomes position index n, meaning the integrators must be "unrolled" in n to create the "wall" of summers depicted in Fig. 11. (Two's-complement overflow in these integrator sums remains both acceptable and, when large signals are present, to be expected.) There outputs $r_1(n)$, $r_2(n)$, and $r_3(n)$ represent respectively one, two, and three applications to input $s(n)$ of integration $1/(1 - z^{-1})$. When this summer wall and correspondingly explicit realizations of the comb steps, advances, and delays are substituted into a $K = 1$ version of the Fig. 3 system with the "extra" section included, the Fig. 12 spatial realization results. Coefficients shown are for the Fig. 10 design.

Just as the nominally infinite memory of an integrator is rendered finite by the comb's difference operation in an ordinary temporal CIC filter, the infinite extent of the left side of the Fig. 11 summer wall is in effect truncated by the action of the combs in Fig. 12. The details of where truncation can be applied are developed in the Fig. 11 caption, for easy reference while studying the diagram. The argument there holds even if differences are generalized from the usual form $u(n) = v(n) - v(n-1)$ to $u(n) = v(n-k_1) - v(n-k_2)$ with integers k_2 and k_1 fixed but arbitrary. Offsets k_1 and k_2 can even differ for each difference sequence computed, though that flexibility is not needed in this application.

A line array can be phase-shift steered with an element-output phase shift linear in position. Phase slope determines steering direction [50]. In Fig. 12 multiplication of analytic or complex-baseband summer-output signals along the summer wall's top row by $e^{j\phi}$ is equivalent to applying linear phase $e^{j\phi n}$ at element outputs, assuming element numbering $n = 0, 1, \ldots$ from the left in Fig. 12. Obtaining a linear phase slope by integrating a phase constant in this way makes an element's local processing independent of its position in the array, a minor implementation convenience.

Has losing the integrator's single-summer simplicity cost us all of the CIC's implementation advantage? The Fig. 12 system requires $9L - 2 = 376$ adders, four more than three for each of the $3L{-}2 = 124$ array elements. In comparison, 473 adds are required to realize the same weighted element-output sum in classic multiplierless form, in which an expansion of those weights in the canonical-signed-digit (CSD) number system [51] specifies a shift-add network. However, the 11 CSD digits per coefficient of that exact realization is really overkill, because discarding the two least-significant digits degrades array-factor sidelobes only at roughly the -55 dB level and below. Using only the 9 remaining CSD digits reduces the adder count to 369. And simple extensions of the CSD approach [52] do even better—the simple "base$(4, 1)$" system of [53] reduces adder count to 310 even without the sidelobe compromises—and modern algorithms along the lines of H_{cub} [54] generally do better yet.

Superficially then, it appears that the Fig. 12 system is only weakly competitive within the range of multiplierless implementation choices available to realize the same weights, give or take modest sidelobe roughening. This ignores a key factor however: The Fig. 12 system is very regular in structure, and this is apt to be a great help in FPGA or VLSI realization. The straight CSD approach is also quite regular, as all terms are summed into one long bus, but realizing the base$(4, 1)$ approach requires summing most terms into one of six busses, which larger number begins to complicate area management in a VLSI or FPGA realization. And shift-add networks created by graph-manipulation algorithms like H_{cub} have no regularity of structure at all. So it's a question of having signal paths running haphazardly around the chip like Boston streets, which are said to derive from old cow paths, or running largely in neat, parallel rows like streets in cities on the plains like, say, Denver. This difference may well make Fig. 12 the most realizable of the approaches in practical terms.

B. An Example Planar Array

Suppose we again wish to implement a single spatial output sample of the Fig. 3 system, a beam output for a receive array, but now in 2D. It turns out that the concepts work in 2D as easily as in 1D. As before, the system is realizable in zero-phase form. Each Fig. 3 integrator $1/(1 - z^{-1})$ becomes integration in each of two directions, with sequential realization the obvious choice: integrate in one direction, and then integrate the result in the other. Delays, advances, and comb differences $(1 - z^{-M})$ are likewise required in each direction. The resulting separability of the transfer functions from the input to the decimators does not, of course, imply separability of linear-combiner outputs, so the system's 2D impulse and frequency responses are generally nonseparable. In 1D an integrator's one-sided, infinite impulse response is truncated to L samples when the corresponding comb is applied. Similarly, the 2D integrator impulse response extends across an infinite quarter plane, but the combined integrator-comb response has support only on an $L \times L$ square.

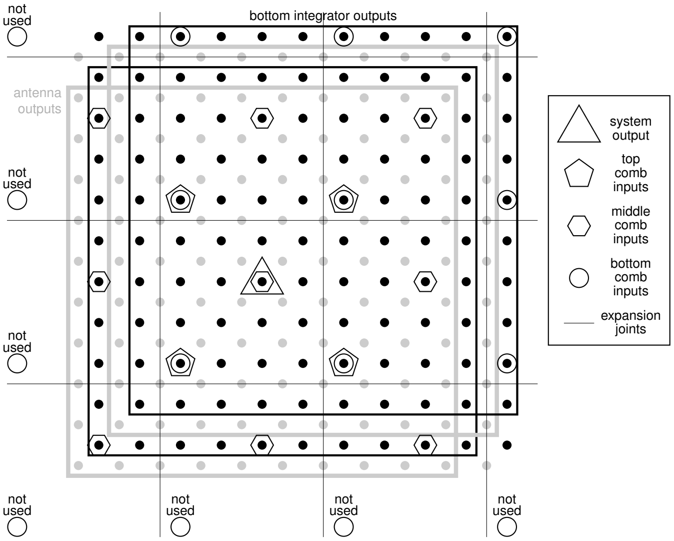

Fig. 13. Sample locations for signals internal to a 2D spatial realization of the Fig. 3 system in "K=1 plus extra" form to sharpen the 2D CIC response by a Chebyshev polynomial $T_3(x)$.

Careful tracking of the locations of spatial samples is key to realizing the 2D system. Fig. 13 maps those locations for a system with Chebyshev degree $N = 3$, a $K = 1$ system with the "extra" section. The map is for $L = 4$ as shown, but the integrators can be extended periodically at the "expansion joints" to any $L > 4$, in exactly the way the 1D Fig. 12 system can be extended at the vertical gray bars.

The four large squares in Fig. 13, two light and two dark, are the locations of the element outputs and of the outputs of each of the three 2D "delayed" integrators. Each square contains $(3L{-}2)^2$ samples, which for the $L = 4$ configuration shown means $10 \times 10 = 100$. Consider the squares in order from lower left to upper right. Antenna-element outputs are located at the light dots in the light, lower-leftmost square. The "extra" Fig. 3 integration comes first and is combined with a half-sample delay that moves that integrator's outputs upward and rightward by half a sample in each direction, to the dark dots in the next, dark-outlined square. Samples of that signal at the four dark dots marked with pentagons comprise the output of the $L \times L$ decimator in cell 1. The first integration in that cell, when combined with half the one-sample delay, then produces outputs located at the light dots in the next light-outline square box, the third of the four boxes. The second integration in Fig. 3's cell 1, again with a half-sample delay, yields outputs located in Fig. 13 at the dark dots in the last and upper-rightmost square box, which has a dark outline. The $L \times L$ decimator below cell 1 outputs samples taken from the sixteen circled output locations. Eleven are dark dots, but five

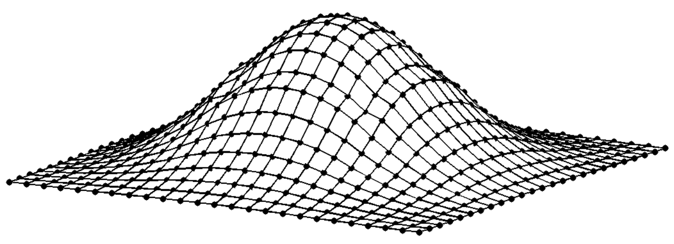

Fig. 14. Integer-valued, symmetric, but nonseparable 25×25 array taper (dots) in 2D designed by sharpening a 2D CIC response of size $L \times L = 9 \times 9$ using Chebyshev polynomial $T_3(\gamma X)$ with design constant $\gamma^2 = 2^{-10}3$. Corner, mid-edge, and center samples have values 1, 61, and 3,465 respectively. Its Fourier transform, the array factor, appears in Fig. 15.

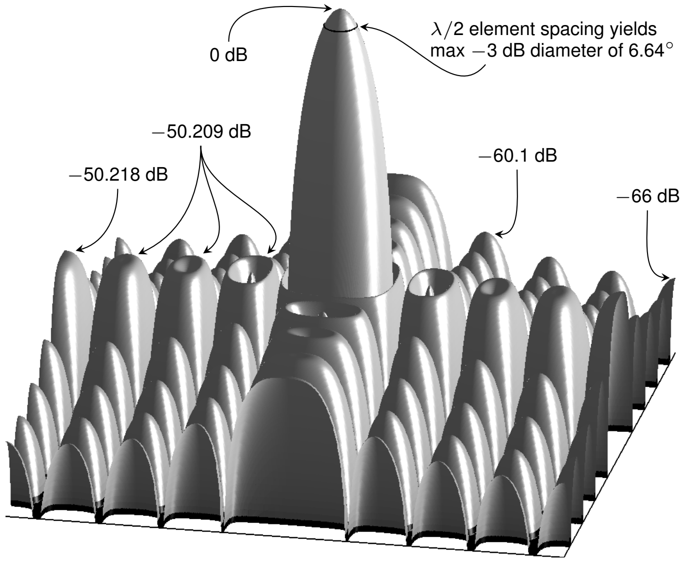

Fig. 15. One period of the boresight-relative dB magnitude of the array factor corresponding to the Fig. 14 taper. Roughly speaking, the array factor is array sensitivity versus direction specified as the array-plane projection of the wavenumber vector. At the classic $\lambda/2$ element spacing, the radius $1/\lambda$ solid circle of possible projections, the "visible region," is inscribed in an array-factor period. More precisely, the dB directive gain of an N element receive array is the dB directive gain of an embedded element (varies slowly), plus the $10\log_{10} N$ ideal array gain (a large constant), minus the dB taper loss (a small constant), plus the dB magnitude of this normalized array factor.

on the left and bottom are marked "not used" for the reasons outlined in the "summer-wall" discussion of Fig. 11. Those five samples correspond to the grayed-out material at the upper left in Fig. 12 and likewise are not actually implemented here. The corresponding comb inputs are permanently fixed at zero.

Proceeding up from the bottom right of cell 1 in Fig. 3, the first 2D comb, when paired with its share of the "advance," takes the sixteen decimator-output samples and yields nine output samples, at the dark-dot locations marked with hexagons. The second advanced comb in cell 1 reduces the set of sample locations to the four dark dots marked with pentagons. The one linear combiner then combines the four comb-output samples with the four samples from the upper decimator using weights b_1 and a_1 respectively. The final advanced comb combines the four linear-combiner output samples to produce the one output sample, at the triangle-marked dark dot in the Fig. 13 map.

Planar-array steering is similar to line-array steering. Phase shifts can always be applied at element outputs [50], but here the realized phase shifts can be made position independent using the existing 2D spatial integration to provide the phase slope: in the first Fig. 13 planar integration—lower-left, light square—impose phase shifts $\mathrm{e}^{j\phi_x}$ and $\mathrm{e}^{j\phi_y}$ at summer nodes' outputs that feed signals in the x and y directions respectively.

The process by which coefficients a_1 and b_1 are chosen is analogous to that used above in the 1D Fig. 12 design, which was based on the frequency-response map of Fig. 10. That same map serves as a rough guide here as well, as actually drawing the 2D version is somewhat impractical. The key difference is that now $X(f_1, f_2) = L^2\,\mathrm{D}_L(f_1)\,\mathrm{D}_L(f_2)$, so the 2D version of the drawing on the upper right in Fig. 10 peaks at L^2 rather than L. This necessitates adjustments to the procedure for putting the $1/\gamma$ line at or just above the inner-sidelobe peak. Again we can set $4\gamma^2 = 2^{-\ell}3$, exactly as for the uniform line array, but now the peak value of the inner sidelobe is $|L^2\,\mathrm{D}_L\big(\frac{3}{2L}\big)|$. Let us call that value p. To have the $1/\gamma$ ripple threshold exceed p by the narrowest margin requires just satisfying $\frac{1}{4\gamma^2} = 2^{-\ell}3 > p^2/4$, achieved by setting $\ell = \lceil \log_2(3p^2/4) \rceil$. Choosing $L = 9$ for an array of size 25×25 then leads to $p = 18$ so that $\frac{3}{4}p^2 = 3^5 = 243$ exactly, resulting in $\ell = 8$. Ripple threshold $1/\gamma$ exceeds the sidelobe peak by $20\log_{10}(256/243) \approx 0.45$ dB.

Implementation constants are $a_1 = -2^{\ell} = -256$ and $b_1 = 1$ as per (16), which applies here exactly except that normalization is a bit different: each L in the ratios on the right in (16) must now be replaced with L^2. This design yields the 2D array taper of Fig. 14 and the 2D array factor of Fig. 15. Each is invariant to rotation by 90° in the plane and mirror symmetric both about each of the two axes and the two diagonals between them. The array factor has Chebyshev ripple level $-20\log_{10}(T_3(\gamma L^2)) = -50.209$ dB.

If the Fig. 15 array factor were separable, the sidelobes in the corner would be twice as deep, in decibels, as at the axis extremes. Clearly they are not, and indeed the array factor and taper are nonseparable. The deviation from array-factor separability mostly happens far down from the central peak of the Fig. 14 taper, however, so it is no surprise that its nonseparability is far from obvious visually. However, separability would imply unit rank for the taper taken as a matrix, and its two nonzero singular values actually differ only by some 37 dB. The SNR taper loss of a separable 2D taper is the decibel sum of the taper losses of its 1D factors, so we expect the taper loss of a "nearly separable" taper like this one to be perhaps a touch better than twice what we'd look for in a 1D version. Here the taper loss is about 3.57 dB.

One might hope, based on this example, that 2D Chebyshev-shaped CIC array tapers are easily tailored to application requirements. It does not appear to be generally so, however. A great many applications require better SNR performance than this author's experimentation suggests this approach can provide. The extreme tapering at the edges displayed in Fig. 14 is not good for receive-array SNR, and that extreme tapering appears to be inherent in the CIC structure.

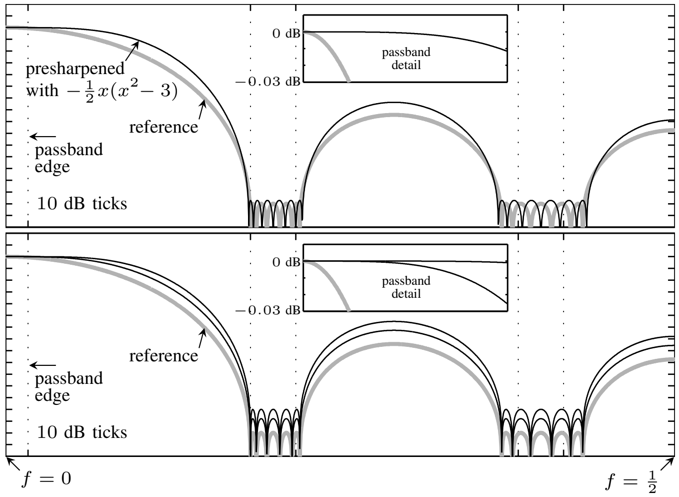

Fig. 16. Preliminary experiments in passband flattening. Here a reference case (gray) without flattening duplicates the Fig. 7 example and has passband droop off the detail at around -0.59 dB. Flattening with presharpening (top plot) is probably impractical, but derivative cancellation (bottom plot) may work. Here cancellation zeros one (middle curve) and two (top curve) derivatives at $f = 0$ by increasing polynomial order from 6 to 8 and 10 respectively.

VI. Invitation to Further Research

There are many related topics that are impractical to cover here given space limitations. And indeed, this was intended from the outset as a preliminary study, one less intended to close the door on all related worthy topics than to open as many promising doors as possible. In the interest of the latter, this section sketches some specific topics of possible interest.

A. Flattening the Passband

On hearing about CIC filters with Chebychev stopbands, the typical first question is about passbands. Can they be flattened while keeping the Chebyshev stopbands? Certainly it is possible in principle. A quick look at two possible approaches will show that the challenge may lie in working out the details of making such methods practical.

1) Polynomial Presharpening: The cubic polynomial $p(x) \triangleq -\frac{1}{2}x(x^2-3) = -\frac{1}{2}x(x+\sqrt{3})(x-\sqrt{3})$ is an odd function with derivative $p'(x) = -\frac{3}{2}(x-1)(x+1)$ equal to zero for $x \in \{-1, 1\}$ and with $p(x)$ having a minimum and maximum respectively at $x = -1$ and $x = 1$. It is strictly monotonically increasing on interval $(0, 1)$. Rather than apply the realization procedure of Section IV to $T_N(\gamma X(f))$, one might instead apply it to $T_N(\gamma L\, p(X(f)/L)\,)$. In the top plot of Fig. 16 the normalized magnitude of $T_N(\gamma L\, p(X(f)/L)\,)$ with $L = 5$, $N = 7$, and $\gamma = 1.25$ is shown—the dark, thin line—over the (thick gray) normalized magnitude $T_N(\gamma X(f))$ from Fig. 7, which realizes (8) with $L = 5$, $N = 6$, and $\gamma = 2$.

Many such passband-sharpening polynomials are also possible, though for realizability $p(x)$ should always be even or odd. The quartic polynomial $p(x) \triangleq 2x^2 - x^4 = x^2(2+x)(2-x)$ is an even function with derivative $p'(x) = 4x(1-x)(1+x)$ equal to zero for $x \in \{-1, 0, 1\}$ and with $p(x)$ having a minimum and maxima respectively at $x = 0$ and at $x = \pm 1$. It is strictly monotonic ascending on interval $(0, 1)$. When the realization procedure of Section IV is applied to $T_N(\gamma L p(X(f)/L))$ with $L = 5$, $N = 4$, and $\gamma = 8$, the normalized magnitude response has a stopband depth and width similar to the responses in the top plot of Fig. 16, though with fewer lobes. Its passband droop is about $2/3$ that shown in the Fig. 16 inset for the cubic sharpening polynomial. (When this curve is added the plot becomes unreasonably cluttered, as the curves track closely.)

With either polynomial $p(x)$, the obvious implementation begins with an expansion of $T_N(\gamma L p(X(f)/L))$. For the odd- and even-degree $p(x)$ discussed, this yields polynomials of degrees 21 and 16 respectively. These are very high degrees indeed, so the next strategy is probably preferable in practice.

2) Cancel Derivatives at Passband Center: Define

$$
\begin{aligned}
V_N(x) \triangleq \quad & T_N(\ x\ )T'_{N+2}(\gamma L) \\
& - T'_N(\gamma L)T_{N+2}(\ x\ )
\end{aligned} \tag{17}
$$

$$
\begin{aligned}
W_N(x) \triangleq \quad & V_N(\ x\ )V''_{N+2}(\gamma L) \\
& - V''_N(\gamma L)V_{N+2}(\ x\ )
\end{aligned} \tag{18}
$$

so that

$$
\begin{aligned}
\left.\frac{dV_N(\gamma X)}{dX}\right|_{X=L} &= \gamma V'_N(\gamma L) = && \gamma\ T'_N(\gamma L)T'_{N+2}(\gamma L) \\
& && - \gamma\ T'_N(\gamma L)T'_{N+2}(\gamma L) = 0 \\
\left.\frac{dW_N(\gamma X)}{dX}\right|_{X=L} &= \gamma W'_N(\gamma L) = && \gamma\ V'_N(\gamma L)V''_{N+2}(\gamma L) \\
& && - \gamma\ V''_N(\gamma L)V'_{N+2}(\gamma L) = 0 \\
\left.\frac{d^2W_N(\gamma X)}{dX^2}\right|_{X=L} &= \gamma^2 W''_N(\gamma L) = && \gamma^2 V''_N(\gamma L)V''_{N+2}(\gamma L) \\
& && - \gamma^2 V''_N(\gamma L)V''_{N+2}(\gamma L) = 0
\end{aligned}
$$

where the second of these is zero because the first is.

Given these relationships, if $T_N(\gamma X(f))$ is replaced in the realization procedure of Section IV with $V_N(\gamma X(f))$, a system results that has increased frequency-response flatness in the passband region. Further, the stopband region would be generally like that of $T_N(\gamma X(f))$, because setting $x = \gamma X(f)$ in definition (17) leaves the first term of $V_N(\gamma X)$ strongly dominant in the stopband region. The operative word is "generally," because the re-engineering of the lower-left Fig. 7 curve to show $20\log_{10}|V_N(\gamma X)/V_N(\gamma L)|$ rather than $20\log_{10}|T_N(\gamma X)/T_N(\gamma L)|$ bends the curve downward to the horizontal as $X = L$ is approached, and this reduces the stopband depth modestly. As an example, in the bottom plot of Fig. 16 the design of Fig. 7 (bottom curve, gray) is modified using this procedure to zero one polynomial derivative at the center of the passband (middle curve). This reduces the decibel passband droop by a factor of about 23, to around 0.026 dB.

Similarly, if the realization procedure of Section IV is applied to $W_N(\gamma X(f))$, the two zero polynomial derivitives yield an even flatter passband. This is illustrated in the top curve of the bottom plot of Fig. 16, where decibel passband droop has been reduced by another factor of almost 27, to less than 0.001 dB. In principle the approach could be extended to flatten the passband even further.

In the exact form presented, this approach is practical only when N is small, as otherwise the integer scaling constants (sampled derivatives) in the definitions of $V_N(x)$ and $W_N(x)$ in (17) and (18) become unworkably large. To avoid that issue as N increases, those constants can be replaced with small integers chosen to keep key ratios "essentially" unchanged.

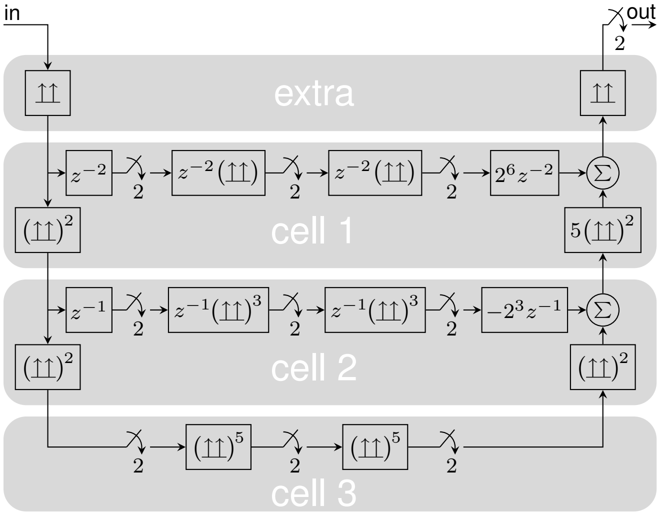

Fig. 17. Nonrecursive equivalent of Figs. 5 with $R = L = 16$ as in Fig. 9. If impulse responses are assumed causal so that (↥↥)^3 means $(1 + z^{-1})^3$, the system is also causal. (Pipelining registers are not shown.) Set $z = 1$ and center impulse responses about $t = 0$ to obtain the zero-phase version.

B. Nonrecursive Implementation

Many authors have noted that factorizations like

$$\sum_{i=0}^{15} z^{-i} = (1 + z^{-1})(1 + z^{-2})(1 + z^{-4})(1 + z^{-8}) \qquad (19)$$

yield efficient computational structures for CIC systems [8]–[10], [25], [29], [31]. This is also true here: the Fig. 17 example uses factorization Fig. 19. Multiple internal decimations (more than two factors in (19)) mean inner blocks on all levels except the top must be repeated, as shown in Fig. 17 with exponents. As usual, filter-decimator combinations can be realized using efficient polyphase approaches.

VII. Conclusions

Sharpening with a Chebyshev polynomial here equips a CIC decimation filter of any nonunit order with equiripple stopbands using (given modest parameter restrictions) only integer coefficients, a key strength of the CIC approach. Roughly $6(N - 1)$ dB of stopband depth is added to an Nth-order system, potentially permitting a lower order. The CIC-sharpening computational structure was previously reported. What is new here is a simple way to choose parameters to specialize this structure to Chebyshev sharpening.

Ordinary CIC decimation filters have multiple stopbands, but Chebyshev sharpening can be parameterized to merge those many stopbands into one (possible with other sharpening polynomials also). Passband shape remains a narrow peak.

The single-stopband version of the Chebyshev-sharpened CIC filter can be adapted for spatial implementation to give pencil beams with low sidelobes to sensor arrays. The approach is easily scaled to large array sizes at an implementation cost competitive with other approaches, particularly in 1D. This is perhaps surprising, since spatial integration is not realizable with one adder the way integration in time is, The CIC filter, both in ordinary and Chebyshev-sharpened versions, can be realized in 2D as well. The Chebyshev-sharpened version can be used, for example to equip a large square receive array with a pencil beam as demonstrated here by example. Per-element implementation cost is roughly twice that of a 1D array, so this approach is less competitive than the 1D approach, though its computational cost is certainly not high enough to automatically rule it out in applications.

Both 1D and 2D array applications of Chebyshev-sharpened CIC responses suffer from so-so SNR taper losses, so their use in receivers is reasonable only where SNR is of little concern. In a transmitter application the unimpressive receive taper loss becomes unimpressive transmit power efficiency: the ratio of pointing-direction power density to total power supplied to the array from transmit drivers is further below the theoretical ideal than is tolerable in many systems.

The CIC approach appears to have only limited usefulness for arrays, so the space devoted to them here is largely to illustrate that a CIC structure can yield a frequency response that has one wide stopband instead of many narrow ones, for which there are of course many potential DSP applications.

References

[1] E. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 29, no. 2, pp. 155–162, Apr. 1981.

[2] T. Saramaki and T. Ritoniemi, "A modified comb filter structure for decimation," in *IEEE Int'l Symp. on Circuits and Systems*, vol. 4, June 1997, pp. 2353–2356 (vol. 4), see also US patent 5,689,449 (est. expiration: Sept. 27, 2015).

[3] A. Kwentus, Z. Jiang, and J. Willson, A.N., "Application of filter sharpening to cascaded integrator-comb decimation filters," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 457–467, Feb. 1997.

[4] J. Kaiser and R. Hamming, "Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 25, no. 5, pp. 415–422, Oct. 1977.

[5] J. W. Tukey, *Exploratory data analysis*. Reading, MA: Addison-Wesley, 1977.

[6] A. Blad and O. Gustafsson, "Bit-level optimized FIR filter architectures for high-speed decimation applications," in *IEEE Int'l Symp. Circuits and Syst.*, May 2008, pp. 1914–1917.

[7] ——, "Redundancy reduction for high-speed FIR filter architectures based on carry-save adder trees," in *IEEE Int'l Symp. Circuits and Syst.*, May 30–June 2, 2010, pp. 181–184.

[8] H. Aboushady, Y. Dumonteix, M. Louërat, and H. Mehrez, "Efficient polyphase decomposition of comb decimation filters in $\Sigma\Delta$ analog-to-digital converters," in *IEEE Int'l Midwest Symp. Circuits and Syst.*, vol. 1, 8–11 Aug., 2000, pp. 432–435 (vol. 1).

[9] Y. Dumonteix, H. Aboushady, H. Mehrez, and M. M. Louërat, "Low power Comb Decimation Filter Using Polyphase Decomposition For Mono-Bit $\Sigma\Delta$ Analog-to-Digital Converters," in *Int'l Conf. on Signal Processing Applications & Technology*, Oct. 2000.

[10] H. Aboushady, Y. Dumonteix, M.-M. Louërat, and H. Mehrez, "Efficient polyphase decomposition of comb decimation filters in $\Sigma\Delta$ analog-to-digital converters," *IEEE Trans. Circuits Syst. II: Analog and Digit. Signal Process.*, vol. 48, no. 10, pp. 898–903, Oct. 2001.

[11] T. Shahana, R. James, B. Jose, K. Poulose Jacob, and S. Sasi, "Polyphase Implementation of Non-recursive Comb Decimators for Sigma-Delta A/D Converters," in *IEEE Conf. on Electron Devices and Solid-State Circuits*, Dec. 2007, pp. 825–828.

[12] M. Abbas, O. Gustafsson, and L. Wanhammar, "Power estimation of recursive and non-recursive CIC filters implemented in deep-submicron technology," in *Int'l Conf. on Green Circuits and Systems*, June 2010, pp. 221–225.

[13] M. Abbas and O. Gustafsson, "Switching activity estimation of CIC filter integrators," in *Asia Pacific Conf. on Postgraduate Research in Microelectronics and Electronics*, Sept. 2010, pp. 21–24.

[14] M. Laddomada, “On the Polyphase Decomposition for Design of Generalized Comb Decimation Filters,” *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 55, no. 8, pp. 2287–2299, Sept. 2008.

[15] G. Jovanovic-Dolecek and S. Mitra, “Efficient sharpening of CIC decimation filter,” in *IEEE Int’l Conf. on Acoustics, Speech, and Signal Processing*, Apr. 2003, pp. 385–8 (vol. 6).

[16] ——, “Sharpened comb decimator with improved magnitude response,” in *IEEE Int’l Conf. on Acoustics, Speech, and Signal Processing*, May 2004, pp. 929–932 (vol. 2).

[17] G. Dolecek and S. Mitra, “Efficient comb-rotated sinc (RS) decimator with sharpened magnitude response,” in *IEEE Int’l Midwest Symp. Circuits and Syst.*, July 2004, pp. 117–120 (vol. 2).

[18] ——, “Stepped Triangular CIC-Cosine Decimation Filter,” in *7th Nordic Signal Process Symp.*, June 2006, pp. 26–29.

[19] ——, “Stepped Triangular CIC Filter for Rational Sample Rate Conversion,” in *IEEE Asia Pacific Conf. on Circuits and Systems*, Dec. 2006, pp. 916–919.

[20] ——, “A New Two-Stage CIC-Based Decimation Filter,” in *5th Int’l Symp. on Image and Signal Process. and Analysis*, Sept. 2007, pp. 218–223.

[21] G. Dolecek, “Modified CIC filter for rational sample rate conversion,” in *Int’l Symp. on Communications and Information Technologies*, Oct. 2007, pp. 252–255.

[22] G. Dolecek and S. Mitra, “Two-stage CIC-based decimator with improved characteristics,” *IET Signal Processing*, vol. 4, no. 1, pp. 22–29, Feb. 2010.

[23] G. Dolecek, “Compensated sharpened comb decimation filter,” in *7th Int’l Symp. on Image and Signal Processing and Analysis*, Sept. 2011, pp. 15–19.

[24] G. Jovanovic Dolecek, “Simple wideband CIC compensator,” *Electronics Letters*, vol. 45, no. 24, pp. 1270–1272, Nov. 19, 2009.

[25] G. Dolecek and M. Laddomada, “Comb-Cosine prefilter based decimation filter,” in *IEEE Int’l Conf. on Industrial Technology*, Mar. 2010, pp. 205–210.

[26] ——, “An Economical Class of Droop-Compensated Generalized Comb Filters: Analysis and Design,” *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 4, pp. 275–279, Apr. 2010.

[27] P. Hong and Z. Yuhua, “The Efficient Design and Modification of Cascaded Integrator Comb Filter,” in *Int’l Conf. on Information Engineering*, Aug. 14–15, 2010, pp. 127–130 (vol. 1).

[28] A. Fernandez-Vazquez and G. Jovanovic Dolecek, “Passband and stopband CIC improvement based on efficient IIR filter structure,” in *IEEE Int’l Midwest Symp. Circuits and Syst.*, Aug. 2010, pp. 765–768.

[29] G. Dolecek and J. Carmona, “Generalized CIC-cosine decimation filter,” in *IEEE Symp. on Industrial Electronics Applications*, Oct. 2010, pp. 640–645.

[30] B. Li, Y. Qiu, and B. Ma, “The Design of Compensation Filter Based on Digital Receiver,” in *Int’l Conf. on Intelligent Computation Technology and Automation*, vol. 2, Mar. 2011, pp. 753–756.

[31] G. Dolecek and J. Carmona, “A new cascaded modified CIC-cosine decimation filter,” in *IEEE Int’l Symp. Circuits and Syst.*, May 2005, pp. 3733–3736 (vol. 4).

[32] G. Stephen and R. Stewart, “High-speed sharpening of decimating CIC filter,” *Electronics Letters*, vol. 40, no. 21, pp. 1383–1384, Oct. 2004.

[33] G. Jovanovic-Dolecek and S. Mitra, “A new two-stage sharpened comb decimator,” *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 52, no. 7, pp. 1414–1420, July 2005.

[34] S. Sharma, S. Kulkarni, M. Vanitha, and P. Lakshminarsimhan, “Hardware Realization of Modified CIC Filter for Satellite Communication,” in *Int’l Conf. on Computational Intelligence and Communication Networks*, Nov. 2010, pp. 41–44.

[35] Z. He, Y. Hu, K. Wang, J. Wu, J. Hou, and L. Ma, “A Novel CIC Decimation Filter for GNSS Receiver Based on Software Defined Radio,” in *7th Int’l Conf. on Wireless Communications, Networking and Mobile Computing*, Sept. 2011, pp. 1–4.

[36] N. Karnati, K.-S. Lee, J. Carletta, and R. Veillette, “A power-efficient polyphase sharpened CIC filter for sigma-delta ADCs,” in *IEEE Int’l Midwest Symp. on Circuits and Systems*, Aug. 2011, pp. 1–4.

[37] X. Liu and A. Willson, “A 1Gsample/Sec non-recursive sharpened cascaded integrator-comb filter with 70 dB alias rejection and 0.003 dB droop in 0.18-μm CMOS,” in *IEEE 8th Int’l Conf. on ASIC*, Oct. 2009, pp. 867–870.

[38] G. Dolecek and F. Harris, “Design of CIC Compensator Filter in a Digital IF Receiver,” in *Int’l Symp. on Communications and Information Technologies*, Oct. 2008, pp. 638–643.

[39] ——, “On design of two-stage CIC compensation filter,” in *IEEE Int’l Symp. on Industrial Electronics*, July 2009, pp. 903–908.

[40] M. Laddomada and M. Mondin, “Decimation schemes for $\Sigma\Delta$ A/D converters based on Kaiser and Hamming sharpened filters,” *IEE Proc., Vis. Image Signal Process.*, vol. 151, no. 4, pp. 287–296, Aug. 2004.

[41] X. Liu, “A High Speed Digital Decimation Filter with Parallel Cascaded Integrator-Comb Pre-Filters,” in *2nd Int’l Congress on Image and Signal Processing*, Oct. 2009, pp. 1–4.

[42] X. Liu and A. Willson, “A 1.2 Gb/s recursive polyphase cascaded integrator-comb prefilter for high speed digital decimation filters in 0.18-μm CMOS,” in *IEEE Int’l Symp. Circuits and Syst.*, May 30–June 2 2010, pp. 2115–2118.

[43] L. Lo Presti and A. Akhdar, “Efficient antialiasing decimation filter for $\Delta\Sigma$ converters,” in *IEEE Int’l Conf. on Electronics, Circuits and Systems*, 1998, pp. 367–370 (vol. 1).

[44] L. Lo Presti, “Efficient modified-sinc filters for sigma-delta A/D converters,” *IEEE Trans. Circuits Syst. II: Analog and Digit. Signal Process.*, vol. 47, no. 11, pp. 1204–1213, Nov. 2000.

[45] M. Laddomada, L. Lo Presti, M. Mondin, and C. Ricchiuto, “An efficient decimation sinc-filter design for software radio applications,” in *IEEE Third Workshop on Signal Processing Advances in Wireless Communications*, 2001, pp. 337–339.

[46] M. Laddomada, “Generalized Comb Decimation Filters for $\Sigma\Delta$ A/D Converters: Analysis and Design,” *IEEE Trans. Circuits Syst. I: Reg. Papers*, vol. 54, no. 5, pp. 994–1005, May 2007.

[47] ——, “Comb-Based Decimation Filters for $\Sigma\Delta$ A/D Converters: Novel Schemes and Comparisons,” *IEEE Trans. Signal Process.*, vol. 55, no. 5, pp. 1769–1779, May 2007.

[48] J. O. Coleman, “Amplitude tapers for planar arrays using the McClellan transformation: concepts and preliminary design experiments,” Naval Research Laboratory, NRL Report NRL/MR/5320--10-9231, April 29, 2010. [Online]. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/a523137.pdf

[49] J. Coleman, D. Scholnik, and J. Brandriss, “A specification language for the optimal design of exotic FIR filters with second-order cone programs,” in *Thirty-Sixth Asilomar Conf. on Signals, Systems and Computers*, Nov. 2002, pp. 341–345 (vol. 1), the file in the IEEE database is unreadable, but a clean preprint is not hard to find.

[50] J. O. Coleman, “Planar arrays on lattices and their FFT steering, a primer,” Naval Research Laboratory, Washington DC, USA, Formal Report NRL/FR/5320–11-10,207, Apr. 29, 2011. [Online]. Available: http://www.dtic.mil/docs/citations/ADA544059

[51] A. Avizienis, “Signed-Digit Number Representation for Fast Parallel Arithmetic,” *IRE Trans. Electron. Comp.*, vol. EC-10, pp. 389–400, 1961.

[52] J. O. Coleman, “Express Coefficients in 13-ary, Radix-4 CSD to Create Computationally Efficient Multiplierless FIR Filters,” in *European Conf. on Circuit Theory and Design*, Aug. 28–31, 2001.

[53] ——, “Cascaded coefficient number systems lead to FIR filters of striking computational efficiency,” in *2001 Int’l IEEE Conf. on Electronics, Circuits, and Systems*, Sep. 2–5, 2001.

[54] Y. Voronenko and M. Püschel, “Multiplierless multiple constant multiplication,” *ACM Trans. Algorithms*, vol. 3, May 2007. [Online]. Available: http://doi.acm.org/10.1145/1240233.1240234

Jeffrey O. Coleman (S’75-M’79-SM’99) received the SBEE degree from the Massachusetts Institute of Technology in 1975, the MSEE degree from Johns Hopkins University in 1979, and the Ph.D. degree from the University of Washington (Seattle) in 1991. He was with the Radar Division of the Naval Research Laboratory (NRL) in Washington, DC from 1978 to 1985, with The Boeing Company through 1992, and then on the electrical engineering faculty at Michigan Technological University until 1997, when he returned to NRL. His research is on theory and design methods in digital signal processing.