A Systematic Algorithm for Designing Multiplierless Computationally Efficient Recursive Decimators and Interpolators

Juha Yli-Kaakinen and Tapio Saramäki

Institute of Signal Processing
Tampere University of Technology
P.O. Box 553, FI-33101 Tampere, Finland
email: {ylikaaki, ts}@cs.tut.fi

Abstract

It has been shown by Renfors and Saramäki that if the phase linearity is not required, then the single-stage and multistage decimators and interpolators based on the use of the so-called recursive Nth-band filters provide the lowest computational complexities among the existing decimators and interpolators. This paper describes an efficient algorithm for designing these decimators and interpolators for both the single-stage or multistage implementations in such a way that the resulting filters become multiplierless with short coefficient wordlength. For single-stage filters, the coefficient optimization is performed in two steps. First, a nonlinear optimization algorithm is used for determining a parameter space of the infinite-precision coefficients including the feasible space, where the filter meets the given criteria. The second step involves finding the filter parameters in this space so that the resulting filter meets the given criteria with the simplest coefficient representation forms. For multistage decimators and interpolators, these two steps are performed independently for each filter stage by properly sharing their attenuation responsibilities. This considerably reduces the overall optimization time. An example is included in order to illustrate the benefits of the proposed synthesis scheme.

1 Introduction

WHEN using a custom or a semi-custom integrated circuit or a programmable logic device for practically implementing a digital filter, the silicon area, the computational complexity, the power consumption, and the maximal achievable sampling rate is highly dependent on the coefficient wordlength. Therefore, the wordlength should be as short as possible but still sufficient to satisfy the given filter specifications. In addition, in highly customized very large-scale integration (VLSI) implementations, the general multiplier element is very costly. Therefore, it is beneficial to carry out the multiplication of a data sample by each filter coefficient value using a sequence of shifts and adds and/or subtracts [1]–[4]. The shifts are often hardwired and, therefore, essentially free. Thus, only a few adders and/or subtracters are required for implementing each coefficient. Such an implementation is usually called "multiplierless".

In order to generate multiplierless filter implementations, it is very essential that a digital filter is realized using a low-sensitivity structure being very insensitive to variations in the filter coefficients.

This work was supported by the Academy of Finland, project No. 44876 (Finnish Centre of Excellence Program (2000–2005)). Juha Yli-Kaakinen was also financed by a postdoctoral research grants from the Academy of Finland, project Nos. 75492 and 105823.

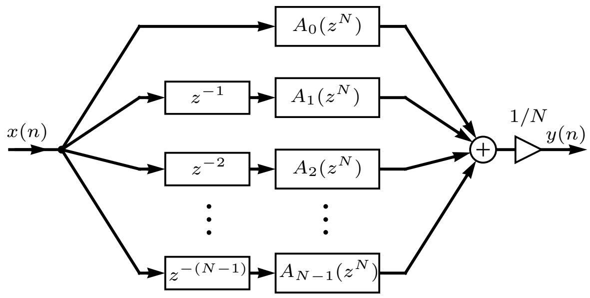

Figure 1. The polyphase structure.

The importance of such a structure is that if the effect of the coefficient value deviation from the ideal value is small, then short coefficient wordlengths can be used with only slightly violating the infinite-precision filter specifications, thereby resulting in a faster, smaller, and less expensive hardware [5].

The best structures for implementing decimation and interpolation filters in the case where the phase linearity is not important, are the so-called recursive Nth-band filters [6]–[8].[1] This structure is based on the polyphase decomposition of the overall transfer function into a set of N (N is the sampling rate alteration ratio) subfilters $A_n(z^N)$ such that the transfer function $H(z)$ can be restated as

$$H(z) = \frac{1}{N}\sum_{n=0}^{N-1} z^{-n} A_n(z^N), \tag{1a}$$

where the transfer functions $A_n(z)$ for $n = 0, 1, \ldots, N-1$ are the following cascades of first-order stable all-pass transfer functions:

$$A_n(z) = \prod_{l=1}^{K_n} \frac{-r_l^{(n)} + z^{-1}}{1 - r_l^{(n)} z^{-1}}. \tag{1b}$$

The transfer function of Eq. (1a) corresponds to the structure shown in Fig. 1. When this structure is used for decimation or interpolation purposes, the branch filters $A_n(z^N)$ can be realized at the lower sampling rate, thus reducing the arithmetic workload by a factor of N, as described in Figs. 2(a) and 2(b), respectively. In these figures, the highly efficient commutative structures [10] are used. The all-pass sections can also be implemented as wave digital all-pass structures, that have an inherently good dynamic range due to their

[1] It is also possible to design recursive Nth-band filters to have an approximately linear phase response in the passband [6], [9]. These filters require significantly higher computational complexities than the corresponding nonlinear-phase Nth-band filters, but they compare favorably with conventional finite-impulse response filters.

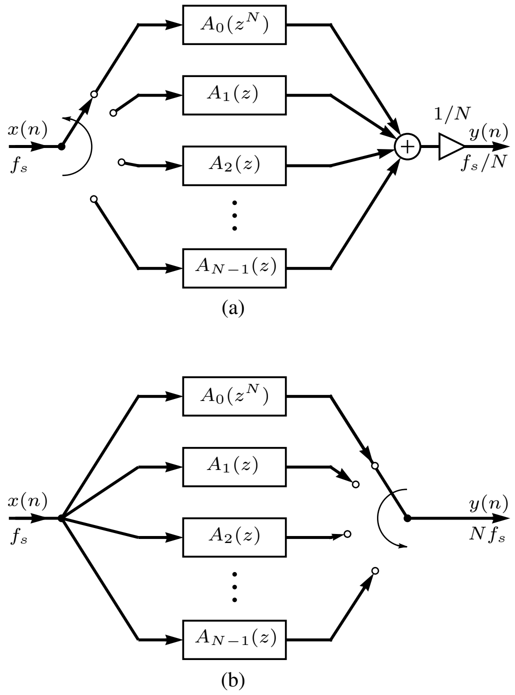

Figure 2. Commutative structures for polyphase filters. (a) N-to-1 decimator. (b) 1-to-N interpolator.

close relation to the continuous-time doubly terminated lossless networks [11]. The parallel connection of all-pass filters realized using such wave digital all-pass sections have also an extremely low passband sensitivity due to the structural passivity property [11]. Fettweis and others [5], [11], [12] have also shown that the parallel all-pass filters have other desirable properties, such as immunity from limit cycle oscillations due to the overflow or granularity oscillations.

Recursive Nth-band filters suffer from the drawback that, in the decimation case, their stopband cutoff frequency must be selected higher than half the output sampling rate, which causes aliasing into the transition band. In the interpolation case, this causes the corresponding imaging effects. If these effects can be tolerated and a linear-phase performance is not required, then these recursive polyphase filters require the lowest computational complexities among the known decimators and interpolators. From a computational point of view, it is also desirable to use a multistage interpolators and decimators, instead of using a single-stage realization. The design of recursive Nth-band filters and their use for decimation and interpolation has been discussed in detail in [6].

This paper describes an efficient algorithm for designing both single-stage and multistage interpolators and decimators based on the use of recursive Nth-band filters with short coefficient wordlength. This algorithm is based on the following observation: Finding the smallest and largest values for the radius of each real pole for all the branch filters so that the given criteria are still met by reoptimizing the remaining poles of the overall filter enables one to find a parameter space including the feasible space where the filter specifications are satisfied. After determining this larger space, all what is needed is to check whether in this space there exist the desired discrete values for the coefficient representations. For single-stage filters, the above steps have to be carried out only once,

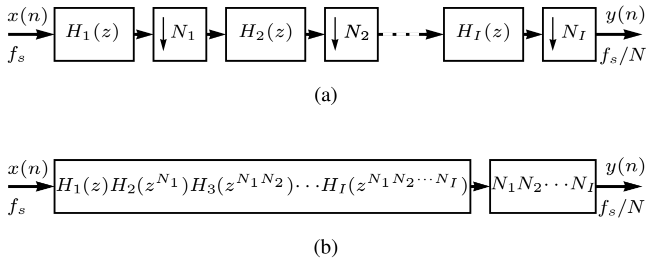

Figure 3. (a) A general implementation form for an N-to-1 decimator. (b) Its single-stage equivalent.

whereas for multistage decimators and interpolators, these two steps are performed independently for each filter stage by properly sharing their attenuation responsibilities. An example is included for illustrating the efficiency of the proposed synthesis scheme.

2 Single-Stage Decimators and Interpolators

The overall frequency response for the recursive Nth-band filter, as given by Eqs. (1a) and (1b), can be written as

$$H(e^{j\omega}) = \frac{1}{N}\sum_{n=0}^{N-1} e^{j\theta_n(\omega)}, \tag{2a}$$

where

$$\theta_n(\omega) = -n\omega + \arg[A_n(e^{jN\omega})] \tag{2b}$$

is the phase response of the nth branch all-pass filter. Consequently, the magnitude is limited by $|H(e^{j\omega})| \leq 1$.

3 Multistage Recursive Decimators and Interpolators

Due to the duality between interpolators and decimators, the discussion in this paper will be concentrated on designing decimators. If the sampling rate conversion ratio can be factored into the product

$$N = \prod_{i=1}^{I} N_i, \tag{3}$$

where $N_1, N_2, \ldots, N_I$ are integers, the decimation can be implemented using I stages as shown in Fig. 3(a). An equivalent single-stage implementation is shown in Fig. 3(b) [6]. Hence, the transfer function of this implementation can be written in the following notationally simplified form:

$$H(z) = \prod_{i=1}^{I} H_i(z^{\widetilde{N}_i}), \tag{4a}$$

where

$$\widetilde{N}_1 = 1, \quad \text{and} \quad \widetilde{N}_r = \prod_{i=1}^{r-1} N_i \quad \text{for } r = 2, 3, \ldots, I. \tag{4b}$$

The amplitude response of the overall filter is thus

$$|H(e^{j\omega})| = \prod_{i=1}^{I} H_i(e^{j\widetilde{N}_i\omega}). \tag{5}$$

4 Statement of the Problem

Before stating the optimization problem, the transfer function of the overall filter is denoted by $H(\Phi,\, z)$, where Φ is the following adjustable parameter vector:

$$\Phi = \left[r_1^{(0)}, \ldots, r_{K_0}^{(0)}, r_1^{(1)}, \ldots, r_{K_1}^{(1)}, \ldots, r_1^{(N-1)}, \ldots, r_{K_{N-1}}^{(N-1)}\right] \quad (6)$$

for the single-stage transfer function, as given by Eqs. (1a) and (1b). For multistage implementation, the transfer functions of the substages are denoted by $H_i(\Phi_i,\, z)$'s, where Φ_i for $i = 1, 2, \ldots, I$ are the corresponding parameter vectors of the substages, whereas

$$\Phi = [\Phi_1,\, \Phi_2, \ldots,\, \Phi_I] \quad (7)$$

is the overall adjustable parameter vectors.

If the desired sampling rate alteration factor is N, then the requirements for the decimation filter can be stated as

$$1 - \delta_p \leq |H(\Phi,\, e^{j\omega})| \leq 1 \quad \text{for } \omega \in [0,\, \omega_p] \quad (8a)$$

$$|H(\Phi,\, e^{j\omega})| \leq \delta_s \quad \text{for } \omega \in \Omega_s, \quad (8b)$$

where $\omega_p < \pi/N$ and the selection of the stopband region Ω_s depends on whether or not aliasing is allowed into the transition band of the filter. Due to the properties of recursive Nth-band filters for $N > 2$, it is not possible to make the stopband region continuous near the frequencies $3\pi/N$, $5\pi/N$, $7\pi/N$, etc [6]. Therefore, in addition to the transition band of width $2(\pi/N - \omega_p)$ and centered at $\omega = \pi/N$, these filters have don't care bands of the same width and centered at $\omega = 3\pi/N, 5\pi/N, \cdots$ (see Fig. 6). In this case, the stopband covers only those frequencies which are aliased into the passband in the decimation case. If necessary, the continuous stopband region starting at π/N can be achieved by cascading the decimator with an additional filter stage working at the output sampling rate [6].

This contribution considers only the case when aliasing is allowed into the transition band $[\omega_p,\, \pi/N]$. In this case, the stopband region is given as

$$\Omega_s = \bigcup_{k=1}^{\lfloor N/2 \rfloor} \left[k\frac{2\pi}{N} - \omega_p,\, \min\Big(k\frac{2\pi}{N} + \omega_p,\, \pi\Big)\right]. \quad (9)$$

The detailed description on how to determine the stopband regions for each filter stage in the multistage implementation is given in [6].

Alternatively, the criteria for the recursive Nth-band filters can be expressed as

$$E(\Phi,\, \omega) = |H(\Phi,\, e^{j\omega})| - \delta_s \leq 0 \quad \text{for } \omega \in \Omega_s. \quad (10)$$

These specifications are typical of recursive Nth-band filters built using all-pass filters as building blocks. In these cases, the filter structure constrains the minimum value of the squared-magnitude response in the passband region to be at least $1 - (N-1)\epsilon$, where ϵ is the maximum value of the squared-magnitude response in the stopband region [6]. Consequently, the passband ripple is very small and the design of Nth-band filters can concentrate on the stopband region.

The stability of the resulting filter is guaranteed if the poles of the all-pass sections $A_n(z)$'s, as given by Eq. (1b), lie inside the unit circle, that is, it is required that for $l = 1, 2, \ldots, K_n$ and for $n = 0, 1, \ldots, N-1$

$$|r_l^{(n)}| < 1. \quad (11)$$

This contribution concentrates on coefficient quantization in fixed-point arithmetic. In many hardware or VLSI implementations, it is attractive to carry out the multiplication of a data sample by a filter coefficient value using a sequence of shifts and adds and/or subtracts. For such a purpose, it is desired to express the coefficient values in the form

$$\sum_{r=1}^{R} a_r 2^{-P_r}, \quad (12)$$

where each a_r is either 1 or -1 and the P_r's are nonnegative integers in the increasing order. The goal is to find all the coefficient values so that, first, R, the number of powers-of-two terms, is made as small as possible and, second, P_R, the maximum number of shifts, is made as small as possible. For this purpose, it is attractive to use the canonic-signed-digit (CSD) representation. This representation is characterized by the fact that no two consecutive digits a_r are both nonzero, that is, for the minimal R, $a_r a_{r+1} = 0$ for $r = 1, 2, \ldots, R-1$. In the sequel, $\mathbb{CSD}_{(R,\, P_R)}$ denotes the space of the CSD numbers with the maximum number of power-of-two terms and the maximum number of fractional bits being R and P_R, respectively.

A reasonable estimate for the cost for the hardware and VLSI implementations of the structure of Fig. 1 is the number of adders and/or subtracters required to implement all the filter coefficients.

The optimization problem under consideration is the following:

Optimization Problem: Given ω_p, N, I, and δ_s, find K_n's for all the filter stages and the parameter vector Φ in such a manner that, first, the criteria of Eq. (8) [or Eq. (10)] and Eq. (11) are met after quantizing the filter coefficient values corresponding to the parameters included in Φ to achieve the above-mentioned form for their representations and, then, the above implementation cost is minimized.

5 Filter Optimization

For the single-stage decimators, the solution to the stated optimization problem can be found in the following two steps. In the first step, the smallest and largest values are determined for each adjustable parameter by reoptimizing the remaining unknowns in the parameter vector in such a manner that the given specifications are met. This enables one to find the parameter space of the infinite-precision coefficients including the feasible space where the filter meets the specifications. The second step involves finding the filter parameters in this space so that the resulting filter meets the given criteria with the simplest coefficient representation forms. For multistage decimators, it is beneficial to perform the above-mentioned steps independently for each filter stage in order to reduce the computational burden.[2]

5.1 Optimization of Infinite-Precision Filters

It has turned out that a very straightforward quantization scheme for the filter coefficients is obtained as follows: For each real pole, the smallest and largest values for the radius are determined so that by

[2] It should be pointed out that the stopband attenuation of the multistage decimators can be only slightly improved by simultaneously optimizing all the filter stages.

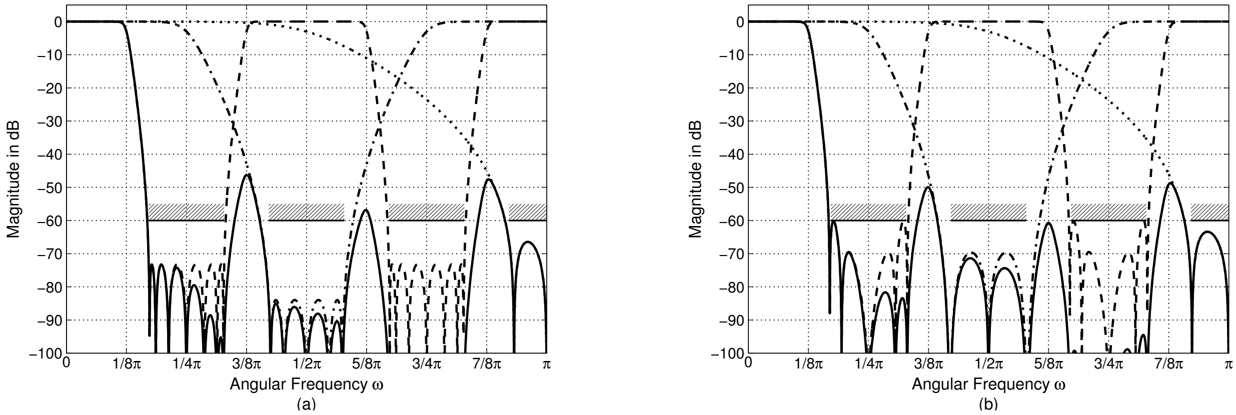

Figure 4. Magnitude responses for (a) the initial infinite-precision and (b) the optimized finite-precision three-stage eighth-band decimator. The solid line gives the magnitude response for the single-stage equivalent $H_1(z)H_2(z^2)H_3(z^4)$, whereas the dotted, dot-dashed, and dashed lines give the responses for $H_1(z)$, $H_2(z^2)$, and $H_3(z^4)$, respectively.

reoptimizing the locations of the remaining poles the given overall criteria, as given by Eq. (8) [or Eq. (10)] and Eq. (11) can still be met.

The above procedure gives for each real pole $r_l^{(n)}$ for $l = 1, 2, \ldots, K_n$ and for $n = 0, 1, \ldots, N-1$, the region $r_l^{(n)(\min)} \leq r_l^{(n)} \leq r_l^{(n)(\max)}$ that is the feasible region, where the pole can be located such that by relocating the remaining poles the given overall criteria are still met by using infinite-precision arithmetic. In order to simplify the overall synthesis scheme, this procedure is carried on independently for all the filter stages.

In order to find the above-mentioned regions for all the poles, there are

$$2 \prod_{n=0}^{N-1} K_n \tag{13}$$

problems of the following form: Find the adjustable parameter vector Φ to minimize ψ subject to the conditions of Eq. (8) [or Eq. (10)] and Eq. (11). For these problems, ψ is $r_l^{(n)}$ and $-r_l^{(n)}$ for $l = 1, 2, \ldots, K_n$ and for $n = 0, 1, \ldots, N-1$.

To solve these problems, the stopband region is discretized into the frequency points $\omega_i \in \Omega_s$, $i = 1, 2, \ldots, L_s$. The resulting discrete minimization problem is to find Φ to minimize ψ subject to

$$E(\Phi, \omega_i) \leq 0 \quad \text{for} \ \ i = 1, 2, \ldots, L_s \tag{14}$$

and the constraints of Eq. (11).

The above-mentioned problems can be conveniently solved by using the second algorithm of Dutta and Vidyasagar [13] or the function `fmincon` from the optimization toolbox provided by MathWorks, Inc. [14]. For more details, see [15].

5.2 Optimization of Finite-Precision Filters

It has been experimentally proved that the parameter space defined above forms a space including the feasible space where the filter specifications are satisfied. After finding this larger space, all what is needed is to check whether in this space there exist combinations of the discrete pole positions with which the given overall criteria are met.

This search can be done in a straightforward manner by first finding the sets of CSD numbers $\Gamma_l^{(n)}$ for $l = 1, 2, \ldots, K_n$ and for $n = 0, 1, \ldots, N-1$ between the smallest and largest values of each filter coefficient, that is, for $l = 1, 2, \ldots, K_n$ and for $n = 0, 1, \ldots, N-1$

$$\left\{ \Gamma_l^{(n)} \in \mathbb{CSD}_{(R,P_R)} \mid r_l^{(n)(\min)} \leq \Gamma_l^{(n)} \leq r_l^{(n)(\max)} \right\}, \tag{15}$$

where $\mathbb{CSD}_{(R,P_R)}$ denotes the space of CSD numbers [cf. Eq. (12)]. The magnitude response is then evaluated for each combination of $\Gamma_l^{(n)}$'s in order to check whether the filter meets the given specifications.

6 Illustrative Example

This example is included to illustrate the performance of the proposed overall synthesis scheme. It is desired to design a eighth-band filter with the passband edge at $\omega_p = 0.0785\pi = 0.628\pi/8$. The minimum required stopband attenuation is 60 dB ($\delta_s = 10^{-3}$). In this case, the stopband region, as given by Eq. (9), is $\Omega_s = [0.1715\pi,\ 0.3285\pi] \cup [0.4215\pi,\ 0.5785\pi] \cup [0.6715\pi,\ 0.8285\pi] \cup [0.9215\pi,\ \pi]$, that is, the aliasing into to the transition band $[0.0785\pi, 0.125\pi]$ is allowed from the bands $[0.3285\pi,\ 0.4215\pi]$, $[0.5785\pi,\ 0.6715\pi]$, and $[0.8285\pi,\ 0.9215\pi]$.

For the three-stage design, the only alternative to factor the sampling rate conversion ratio is $N_1 = N_2 = N_3 = 2$ ($\widetilde{N}_1 = 1$, $\widetilde{N}_2 = 2$, and $\widetilde{N}_3 = 4$), that is, the overall filter is a cascade of three half-band filters. In this case, the last stage $H_3(z^4)$ provides the attenuation on region $\Omega_s = [0.1715\pi,\ 0.3285\pi] \cup [0.6715\pi,\ 0.8285\pi]$, the second stage $H_2(z^2)$ on $\Omega_s = [0.4215\pi,\ 0.5785\pi]$, and the first stage $H_1(z)$ on $\Omega_s = [0.9215\pi,\ \pi]$. The minimum number of multipliers for $H_i(z^{\widetilde{N}_i})$ for $i = 1, 2, 3$ to meet the given criteria are $K^{(1)} = 1$, $K^{(2)} = 2$, and $K^{(3)} = 3$, respectively.

The initial infinite-precision Nth-band subfilters can be determined by utilizing the design algorithm described in [6]. The initial coefficient values for $H_3(z^4)$ are $r_1^{(0)} = -0.085\,523$, $r_2^{(0)} = -0.718\,273$, and $r_1^{(1)} = -0.326\,452$, for $H_2(z^2)$ $r_1^{(0)} = -0.116\,797$ and $r_1^{(1)} = -0.548\,630$, whereas for $H_1(z)$ $r_1^{(0)} = -0.338\,473$. The stopband attenuations for the initial substages $H_3(z^4)$, $H_2(z^2)$, and $H_1(z)$ are 73.21 dB, 83.97 dB, and 66.45 dB, respectively. The magnitude responses for the initial infinite-precision substages as well as for the single-stage equivalent are

Table 1: The smallest and largest infinite-precision coefficient values for the subfilters $H_3(z^4)$, $H_2(z^2)$, and $H_1(z)$ in Example.

$H_3(z^4)$	$r_1^{(0)(\min)} = -0.111\,647$	$r_1^{(0)(\max)} = -0.057\,811$
	$r_2^{(0)(\min)} = -0.771\,093$	$r_2^{(0)(\max)} = -0.681\,117$
	$r_1^{(1)(\min)} = -0.395\,188$	$r_1^{(1)(\max)} = -0.268\,425$
$H_2(z^2)$	$r_1^{(0)(\min)} = -0.156\,770$	$r_1^{(0)(\max)} = -0.082\,365$
	$r_1^{(1)(\min)} = -0.618\,978$	$r_1^{(1)(\max)} = -0.489\,915$
$H_1(z)$	$r_1^{(0)(\min)} = -0.341\,785$	$r_1^{(0)(\max)} = -0.336\,582$

shown Fig. 4(a), whereas the pole-zero plot for the single-stage equivalent of the initial filter is shown in Fig. 5(a). In Fig. 4(a), the hatched bars indicate both the stopband region and the required attenuation for the overall decimator.

The smallest and largest values for the coefficients of the substages $H_3(z^4)$, $H_2(z^2)$, and $H_1(z)$ after the infinite-precision optimization of Subsection 5.1 are shown in Table 1. The overall CPU-time required for solving all these infinite-precision optimization problems is approximately ten seconds when using a MATLAB code running on a 1.4 GHz Pentium-M. The number of discrete frequency points were $L_s = 200$, $L_s = 100$, and $L_s = 50$ for $H_3(z^4)$, $H_2(z^2)$, and $H_1(z^2)$, respectively.

For this overall filter, the maximum number of power-of-two terms required to implement all the filter coefficients is four ($R = 4$), whereas eight fractional bits ($P_R = 8$) are required to meet the magnitude specifications. For this coefficient representation form, the number of discrete coefficient values between the smallest and largest values for the coefficients of $H_3(z^3)$ is 14, 21, and 33, that is, the number of coefficient combinations for the first stage is $14 \cdot 21 \cdot 33 = 9702$. The number of discrete coefficient values between the smallest and largest values for the coefficients of $H_2(z^2)$ are 19 and 33, that is, the number of coefficient combinations for the second stage is 627. For the first stage, there exists only one discrete coefficient value between the smallest and largest values of the coefficient. The CPU time required when using a Fortran 95 program on a 500 MHz AlphaServer DS20 to evaluate all these combinations with $L_s = 100$ was less than one second.

The number of adders and/or subtracters required to implement all the coefficients is nine. The optimized finite-precision coefficients values are shown in Table 2, whereas the magnitude responses for the substages as well as for the single-stage equivalent are depicted in Fig. 4(b). The passband and stopband ripples for the optimized finite-precision overall filter are $\delta_p = 4.9184 \cdot 10^{-7}$ and $\delta_s = 0.9766 \cdot 10^{-3}$ (60.21 dB), respectively.

The best two-stage design is obtained with the decimation ratios $N_1 = 4$ and $N_2 = 2$ ($\widetilde{N}_1 = 1$ and $\widetilde{N}_2 = 4$). The minimum number of multipliers for $H_i(z^{\widetilde{N}_i})$ for $i = 1, 2$ to meet the given criteria are $K^{(1)} = 6$ and $K^{(2)} = 3$, respectively. The stopband attenuation for the initial filter is $0.9792 \cdot 10^{-3}$ (60.18 dB). The overall specifications are satisfied by $R = 4$ and $P_R = 7$. The optimized finite-precision coefficients values for the first stage are shown in Table 3. The optimized coefficient values for the last stage are the same as for the last stage for the three-stage design. In this case, the number of adders and subtracters required to implement all the coefficients is eight. The passband and stopband ripples for the optimized filter are $\delta_p = 1.0724 \cdot 10^{-6}$ and $\delta_s = 0.9762 \cdot 10^{-3}$ (60.21 dB), respectively.

Table 2: Optimized finite-precision coefficient values for the three-stage eighth-band filter.

$H_3(z^4)$	$r_1^{(0)} = -0.078\,125\,00 = -2^{-4} - 2^{-6}$
	$r_2^{(0)} = -0.710\,937\,50 = -1 + 2^{-2} + 2^{-5} + 2^{-7}$
	$r_1^{(1)} = -0.312\,500\,00 = -2^{-2} - 2^{-4}$
$H_2(z^2)$	$r_1^{(0)} = -0.125\,000\,00 = -2^{-3}$
	$r_1^{(1)} = -0.562\,500\,00 = -2^{-1} - 2^{-4}$
$H_1(z)$	$r_1^{(0)} = -0.339\,843\,75 = -2^{-1} + 2^{-3} + 2^{-5} + 2^{-8}$

Table 3: Optimized finite-precision adaptor coefficients for the first stage of the two-stage eighth-band decimator. The optimized coefficient values for the last stage are the same as for the last stage of the three-stage eighth-band decimator.

$H_1(z)$	$r_1^{(0)} = -2^{-6}$	$r_2^{(0)} = -2^{-1} - 2^{-5}$
	$r_1^{(1)} = -2^{-4}$	$r_2^{(1)} = -1 + 2^{-2}$
	$r_1^{(2)} = -2^{-3}$	
	$r_1^{(3)} = -2^{-2} - 2^{-4}$	

Table 4: Optimized finite-precision adaptor coefficients for the single-stage eighth-band decimator.

$r_1^{(0)} = -2^{-6} - 2^{-8}$	$r_2^{(0)} = -2^{-1} - 2^{-5}$
$r_1^{(1)} = -2^{-4} + 2^{-6}$	$r_2^{(1)} = -2^{-1} - 2^{-3}$
$r_1^{(2)} = -2^{-4} - 2^{-6}$	$r_2^{(2)} = -1 + 2^{-2} + 2^{-5}$
$r_1^{(3)} = -2^{-3} + 2^{-8}$	$r_2^{(3)} = -1 + 2^{-2} - 2^{-4} + 2^{-8}$
$r_1^{(4)} = -2^{-2} + 2^{-4} + 2^{-7}$	$r_2^{(4)} = -1 + 2^{-3} - 2^{-8}$
$r_1^{(5)} = -2^{-2} + 2^{-7}$	$r_2^{(5)} = -1 + 2^{-4} - 2^{-6} + 2^{-8}$
$r_1^{(6)} = -2^{-2} - 2^{-4} - 2^{-7}$	
$r_1^{(7)} = -2^{-1} + 2^{-4} + 2^{-8}$	

Table 5: Summary of filter designs.

	$N_{\rm M}$	$N_{\rm A}$	$N_{\rm O}$	δ_p	δ_s
Three-stage	6	9	45	$4.918 \cdot 10^{-7}$	$0.977 \cdot 10^{-3}$
Two-stage	9	8	61	$1.072 \cdot 10^{-6}$	$0.976 \cdot 10^{-3}$
Single-stage	14	23	140	$1.703 \cdot 10^{-6}$	$0.979 \cdot 10^{-3}$

For the single-stage design, that is, for an eighth-band filter, the minimum number of multipliers, K, required to meet the specifications is 14. For this design, $K_i = 2$ for $i = 0, 1, \ldots, 5$ and $K_6 = K_7 = 1$. The stopband attenuation of the initial filter is 60.84 dB. Again, the specifications are met by $R = 4$ and $P_R = 8$. The optimized finite-precision coefficients values are shown in Table 4. In this case, the number of adders and subtracters required to implement the overall filter is 23. The passband and stopband ripples for the optimized filter are $\delta_p = 1.7033 \cdot 10^{-6}$ and $\delta_s = 0.9792 \cdot 10^{-3}$ (60.18 dB), respectively. The magnitude responses for the finite-precision single-stage decimator is depicted in Fig. 6.

The summary of the decimator designs is shown in Table 5. In this table, $N_{\rm M}$, $N_{\rm A}$, and $N_{\rm O}$ give the number of multipliers, number

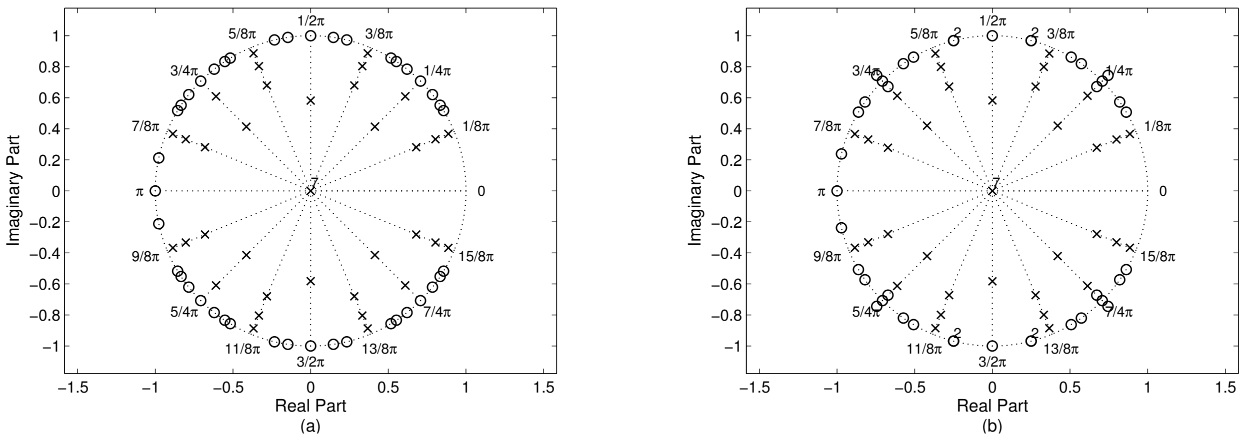

Figure 5. Pole-zero plot for (a) the initial infinite-precision and (b) optimized finite-precision three-stage eighth-band decimator.

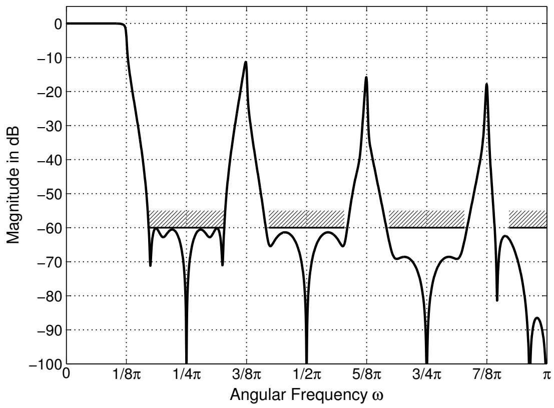

Figure 6. Magnitude response for the optimized finite-precision single-stage eighth-band decimator.

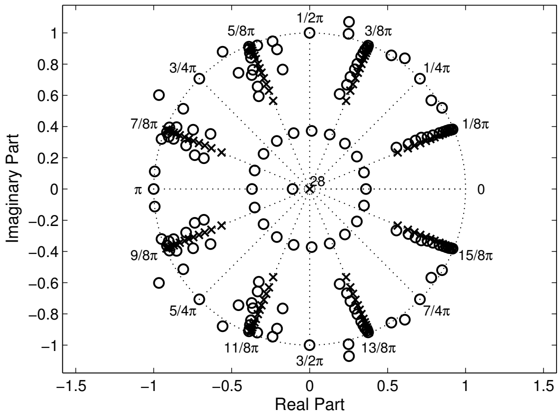

Figure 7. Pole-zero plot for the optimized finite-precision single-stage eighth-band decimator.

of adders and subtracters, and the overall order of the single-stage equivalent, respectively.

References

[1] J. Yli-Kaakinen and T. Saramäki, "Design of low-sensitivity and low-noise recursive digital filters using a cascade of low-order lattice wave digital filters," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 906–914, July 1999. [Online]. Available: http://alpha.cc.tut.fi/~ylikaaki/CAS99.pdf

[2] J. Yli-Kaakinen and T. Saramäki, "A systematic algorithm for the design of multiplierless FIR filters," in *Proc. IEEE Int. Symp. Circuits Syst.*, vol. II, Sydney, Australia, May 6–9 2001, pp. 185–188. [Online]. Available: http://alpha.cc.tut.fi/~ylikaaki/iscas2001.pdf

[3] D. Brückmann, "Flexible digital receiver architecture with optimized components," *AEÜ–Int. J. Electron. Commun.*, vol. 55, no. 6, pp. 408–416, Dec. 2001.

[4] J. Yli-Kaakinen and T. Saramäki, "A systematic algorithm for the design of multiplierless lattice wave digital filters," in *Proc. 2004 IEEE Int. Symp. Control., Commun., Signal Process.*, Hammamet, Tunisia, Mar. 21–24 2004, pp. 415–418.

[5] L. Wanhammar, *DSP Integrated Circuits*. New York: Academic, 1998.

[6] M. Renfors and T. Saramäki, "Recursive Nth-band digital filters — Part I: Design and properties, Part II: Design of multistage decimators and interpolators," *IEEE Trans. Circuits Syst.*, vol. CAS-34, no. 1, pp. 24–51, Jan. 1987.

[7] T. Saramäki and M. Renfors, "Nth-band filter design," in *Proc. IX European Signal Processing Conf.*, Island of Rhodes, Greece, Sept. 1998, pp. 1943–1947.

[8] J. Yli-Kaakinen, T. Kupiainen, M. Hu, R. Uusikartano, and M. Renfors, "Multirate digital filter design for a PAL TV modulator," *IEEE Trans. Consumer Electron.*, vol. 45, no. 3, pp. 970–974, Aug. 1999. [Online]. Available: http://alpha.cc.tut.fi/~ylikaaki/ce.ps.gz

[9] R. Ansari and B. Liu, "Efficient sampling rate alteration using recursive IIR digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 1366–1373, Dec. 1983.

[10] R. E. Crochiere and L. R. Rabiner, *Multirate Digital Signal Processing*. Englewood Cliffs: NJ: Prentice-Hall, 1983.

[11] A. Fettweis, J. A. Nossek, and K. Meerkötter, "Reconstruction of signals after filtering and sampling rate reduction," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 893–902, Aug. 1985.

[12] P. P. Vaidyanathan, S. K. Mitra, and Y. Neuvo, "A new approach to the realization of low sensitivity IIR digital filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 2, pp. 350–361, Apr. 1986.

[13] S. R. K. Dutta and M. Vidyasagar, "New algorithms for constrained minimax optimization," *Math. Program.*, vol. 13, pp. 140–155, 1977.

[14] T. Coleman, M. A. Branch, and A. Grace, *Optimization Toolbox User's Guide*, The MathWorks, Inc., Jan. 1999, Version 2.

[15] T. Saramäki and J. Yli-Kaakinen, "Design of digital filters and filter banks by optimization: Applications," Tampere International Center for Signal Processing, Tech. Rep. No. 15, Apr. 2002. [Online]. Available: http://alpha.cc.tut.fi/~ylikaaki/TICSP02.pdf