Available online at www.sciencedirect.com
SCIENCE DIRECT®
Digital Signal Processing 15 (2005) 522–535
Digital Signal Processing
www.elsevier.com/locate/dsp

Telescoping rounding for suboptimal finite wordlength FIR digital filter design

Dušan M. Kodek *, Marjan Krisper

Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia

Available online 16 March 2005

Abstract

Rounding of the so-called infinite precision coefficients to their nearest finite wordlength representation is often used for reasons of simplicity. This rounding, however, is typically far from optimal. A significantly better rounding method that uses a technique called telescoping is presented in this paper. Its practical effectiveness in the design of suboptimal finite wordlength filters is demonstrated. It can also be used to speed up the algorithms for optimal finite wordlength FIR filter design.

Keywords: FIR digital filters; Finite wordlength effects; Rounding; Quantization; Minimax approximation

1. Introduction

There are many practical situations in which the coefficients of an FIR digital filter must be represented with a finite number of bits. This means that the "infinite precision" coefficients have to be somehow replaced by the finite wordlength ones. The so-called infinite precision coefficients are typically 32-bit floating point numbers. Though the 32-bit wordlength is hardly infinite, it is much longer than practical finite wordlengths that we are interested in. Replacing the infinite precision coefficients with the finite wordlength coefficients degrades the filter's frequency response. The lowest degradation is obtained if an optimal finite wordlength design algorithm is used. Such algorithms, however, are

* Corresponding author. Fax: +386 1 426 4647.
E-mail addresses: duke@fri.uni-lj.si (D.M. Kodek), marjan.krisper@fri.uni-lj.si (M. Krisper).
URL: http://www.laps.fri.uni-lj.si.

1051-2004/$ – see front matter
doi:10.1016/j.dsp.2005.01.009

slow and it is not clear if filters with length 125, for example, can be designed this way. Simple rounding of the infinite precision coefficients to their nearest finite wordlength representation is therefore still often used.

Rounding of the infinite precision FIR filter coefficients to the nearest finite wordlength representation is well understood and was analyzed extensively in Refs. [1,2]. Its effect on the frequency response degradation is basically random and does not take into account any of the properties of FIR filter design problem. This paper presents a better rounding method that uses the properties of the minimax approximation problem. Its application is straightforward and usually produces finite wordlength filters that are significantly better than those obtained by simple rounding. The method is also useful in the algorithms for optimal finite wordlength FIR filter design where it can be used to reduce the amount of computation needed to produce the optimal solution.

2. Statement of the problem

To describe the method let us start with the infinite precision design problem. The frequency response $H^*(\omega)$ of a length N optimal infinite precision (i.e., filter coefficients can be any real number) linear phase FIR digital filter is equal to

$$H^*(\omega) = \sum_{k=0}^{N-1} h^*(k) e^{-j\omega k} = e^{j(L\pi/2-\omega(N-1)/2)} Q(\omega) \sum_{k=0}^{n} a_k^* \cos k\omega, \tag{1}$$

where $L = 0$ or 1. Depending on N (odd or even) and filter symmetry (positive or negative) there are exactly four types of FIR filters and four real functions $Q(\omega)$. The degree n of the cosine polynomial

$$P^*(\omega) = \sum_{k=0}^{n} a_k^* \cos k\omega, \tag{2}$$

is related to the filter length N and there are formulas which relate the optimal coefficients $h^*(k)$ and a_k^*. It is easy to see that $Q(\omega)$ is irrelevant from the point of view of the approximation problem and we will therefore assume $Q(\omega) = 1$. To find $P^*(\omega)$ one must solve the following minimax approximation problem:

$$\min_{P(\omega)} \max_{\omega\in\Omega} \left| W(\omega)\big(D(\omega) - P(\omega)\big)\right|. \tag{3}$$

The real function $D(\omega)$ is the desired frequency response, the weighting function $W(\omega)$ is by definition real and positive, and the set Ω is a subset of the interval $[0, \pi]$. The optimal infinite precision minimax approximation error or deviation E^* is equal to

$$E^* = \max_{\omega\in\Omega} \left| W(\omega)\left(D(\omega) - \sum_{k=0}^{n} a_k^* \cos k\omega\right)\right|. \tag{4}$$

Algorithms like linear programming and various versions of the exchange algorithm make solving (3) quite simple. The standard approach is to use the Remez algorithm in a way that was first described by Parks and McClellan [3]. The problem's complexity changes

dramatically when the finite wordlength constraint is introduced. Constrained minimax approximation problem is NP-complete and is much harder to solve than the infinite precision one. This is the reason for the interest in suboptimal methods like the rounding of the infinite precision coefficients.

In order to describe our method, the finite wordlength design problem needs to be defined more precisely. We can, without loss of generality, make the finite wordlength constraint equal to requesting the filter coefficients $h(k)$ to be b-bit integers from the set I_b, where $I_b = \{-2^{b-1}, \ldots, -1, 0, 1, \ldots, 2^{b-1}\}$. The integer set I_b is chosen for convenience only—any other finite set of numbers (sums of power-of-two, for example) can be used instead.

Constraining the coefficients $h(k)$ to the set I_b requires a redefinition or scaling of the original infinite precision approximation problem. This is necessary to bring the finite wordlength coefficients within the range of numbers in I_b and can be done with the help of a scaling factor s. Let us assume that s is known and denote $D_u(\omega)$, $W_u(\omega)$, and $P_u(\omega)$ as the original (unscaled) problem. The finite wordlength approximation problem can be rewritten as

$$\min_{P_u(\omega)} \max_{\omega\in\Omega} \bigl|W_u(\omega)/s\bigl(sD_u(\omega) - sP_u(\omega)\bigr)\bigr| = \min_{P(\omega)} \max_{\omega\in\Omega} \bigl|W(\omega)\bigl(D(\omega) - P(\omega)\bigr)\bigr|, \tag{5}$$

where

$$\begin{aligned} &D(\omega) = sD_u(\omega), \qquad W(\omega) = W_u(\omega)/s, \\ &P(\omega) = sP_u(\omega) = \sum_{k=0}^{n} a_k \cos k\omega, \quad a_k \in I_b. \end{aligned} \tag{6}$$

$P(\omega)$ is the finite wordlength polynomial and $D(\omega)$, $W(\omega)$ are the scaled input functions. Observe that $a_k \in I_b$ in (6). This requires an explanation since it is the filter coefficients $h(k)$ that must be from the set I_b. It is easy to see there is no problem here if the scaling factor is modified. The nature of modification follows from the formulas that relate the filter coefficients $h(k)$ to cosine polynomial coefficients a_k. For type 1 FIR filters (odd N, positive symmetry) there is

$$h(n) = a_0, \qquad h(n-k) = a_k/2, \quad k = 1, 2, \ldots, n. \tag{7}$$

This means that if $h(k) \in I_b$, a_k must be from the set of even numbers of twice the size of those in I_b for $k \geqslant 1$. Dividing the scaling factor s in (5) by 2 will also divide all a_k by 2 and the set I_b can now be used for both a_k and $h(k)$. Since all a_k were divided by 2 it is necessary to replace (7) by

$$h(n) = 2a_0, \qquad h(n-k) = a_k, \quad k = 1, 2, \ldots, n. \tag{8}$$

The coefficient a_0 is a special case — its values are constrained to the elements of I_b divided by 2. This property of a_0 must be taken into account in either rounding or optimal finite wordlength design.

Similar considerations apply to the type 2, 3, and 4 FIR filters. The difference is that s must be divided by 4 and not by 2 where a_0 is again a special case as above. The net effect of dividing s by 2 or 4 is a unification of all four cases from the point of view of the approximation problem.

It follows from (6) that scaling factor s can be interpreted as the filter gain. In most digital filtering applications one is allowed to use any gain s because its effect can usually be easily removed, if so desired. Scaling can also be used in the infinite precision case where it affects the size of coefficients, but does not affect the approximation error. Situation is different in the finite wordlength design where approximation error changes with s. The choice of s is therefore not trivial and it is important both in the optimal and in the suboptimal finite wordlength design.

Our rounding method can be used with any scaling factor s. We will start with an assumption that it is a known constant and explain later how a suitable s can be found.

3. Telescoping polynomials

Notation $P(\omega)$ will from here on denote a polynomial with b-bit coefficients $a_k,\ k = 0, 1, \ldots, n$, from I_b, while $P^*(\omega)$ is the optimal infinite precision polynomial with unconstrained coefficients $a_k^*,\ k = 0, 1, \ldots, n$. $D(\omega)$ and $W(\omega)$ are the scaled input functions in both cases. This means that the effect of the scaling factor s is already included in coefficients a_k^*.

The well-known Chebyshev equioscillation theorem (also known as the alternation theorem) [4] provides the conditions for the optimal infinite precision minimax approximation of degree n: there are at least $n+2$ so-called extremal points in Ω at which the approximation error achieves its maximum. Let $\omega_i,\ \omega_0 < \omega_1 < \cdots < \omega_{n+1}$, be these extremal points. The following equations hold:

$$W(\omega_i)\left(D(\omega_i) - \sum_{k=0}^{n} a_k^* \cos k\omega_i\right) = (-1)^i d, \quad i = 0, 1, \ldots, n+1, \tag{9}$$

where $E^* = |d|$ is the optimal approximation error.

Let us now take the highest order coefficient a_n^* and replace it with its nearest finite wordlength neighbor $a_n \in I_b$,

$$a_n = a_n^* + \Delta a_n, \tag{10}$$

where $|\Delta a_n|$ is defined as the lowest possible distance from a_n^* to a number in I_b. Obviously, if $\Delta a_n < 0$ the finite wordlength coefficient a_n is the nearest lower neighbor of a_n^*—it is the nearest upper neighbor otherwise. For the integer set I_b the value of Δa_n always lies between -0.5 and 0.5.

Using a_n instead of a_n^* gives the approximation error that is greater than E^* for nonzero Δa_n. The increase in approximation error can be made smaller if the remaining coefficients $a_{n-1}^*, a_{n-2}^*, \ldots, a_0^*$ are suitably modified. Derivation of a simple modification that achieves this goal is the main purpose of this paper. We will show that it can be done with the help of a so-called telescoping cosine polynomial $C_n(\omega)$ that was first described in Ref. [5],

$$C_n(\omega) = \cos n\omega + \sum_{k=0}^{n-1} c_{nk} \cos k\omega. \tag{11}$$

Telescoping polynomial $C_n(\omega)$ is defined as a solution of a minimax problem

$$E_{c_n} = \min_{c_{nk}} \max_{\omega\in\Omega} \big|W(\omega)C_n(\omega)\big|. \tag{12}$$

This means that for a given set Ω and a given weighting function $W(\omega)$ no other cosine polynomial of degree n with leading coefficient 1 can have smaller extreme value than $C_n(\omega)$. In this respect $C_n(\omega)$ is similar to a Chebyshev polynomial of degree n divided by 2^{n-1} for which the same is true [6] when $\Omega = [-1, 1]$ and $W(\omega) = 1$. Telescoping polynomial $C_n(\omega)$ can therefore be viewed as a Chebyshev polynomial that is altered to conform to a particular FIR design problem.

Computing the coefficients c_{nk}, $k = 0, 1, \ldots, n-1$, is quite easy. We simply make $D(\omega) = -\cos n\omega$ and the Remez algorithm can be used to solve the minimax approximation problem as in (3). Having $C_n(\omega)$ let us modify the remaining coefficients $a^*_{n-1}, a^*_{n-2}, \ldots, a^*_0$ in the following manner:

$$a^{(1)}_k = a^*_k + \Delta a_n c_{nk}, \quad k = 0, 1, \ldots, n-1. \tag{13}$$

These coefficients define a new polynomial $P^{(1)}(\omega)$ of degree n

$$P^{(1)}(\omega) = a_n \cos n\omega + \sum_{k=0}^{n-1} a^{(1)}_k \cos k\omega = P^*(\omega) + \Delta a_n C_n(\omega). \tag{14}$$

The coefficients of $P^{(1)}(\omega)$ are different from the optimal coefficients a^*_k, which means that they give the approximation error

$$E^{(1)} = \max_{\omega\in\Omega} \big|W(\omega)\big(D(\omega) - P^{(1)}(\omega)\big)\big|, \tag{15}$$

that is greater than E^*. The increase in the approximation error, however, is upper bounded and the upper bound is the lowest when telescoping polynomial $C_n(\omega)$ is used as in (13). This is proven formally in the following theorem.

Theorem 1. *Let $P^{(1)}(\omega)$ be a cosine polynomial of degree n defined by* (14)*. Then its approximation error is bounded by*

$$E^{(1)} \leqslant E^* + \Delta a_n E_{c_n}, \tag{16}$$

where E_{c_n} is given by (12)*.*

Proof. The approximation error $E^{(1)}$ can be rewritten as

$$\begin{aligned}
E^{(1)} &= \max_{\omega\in\Omega} \left| W(\omega) \left(D(\omega) - \big(a^*_n + \Delta a_n\big)\cos n\omega - \sum_{k=0}^{n-1} \big(a^*_k + \Delta a_n c_{nk}\big)\cos k\omega \right) \right| \\
&= \max_{\omega\in\Omega} \left| W(\omega) \left(D(\omega) - \sum_{k=0}^{n} a^*_k \cos k\omega - \Delta a_n \left(\cos n\omega + c_{nk} \sum_{k=0}^{n-1} \cos k\omega \right) \right) \right| \\
&= \max_{\omega\in\Omega} \big| W(\omega)\big(D(\omega) - P^*(\omega) - \Delta a_n C_n(\omega)\big) \big| \\
&\leqslant E^* + \Delta a_n E_{c_n},
\end{aligned} \tag{17}$$

where the triangle inequality was used in the last line. Since E_{c_n} is by definition the lowest possible value for any cosine polynomial of degree n with leading coefficient 1, the upper bound (16) cannot be lower for any polynomial that is different from $C_n(\omega)$. This completes the proof. □

The theorem does not put any restrictions on the nature of the discrete set I_b. It is also easy to show that it holds for any set of functions and not only for cosine polynomials. This level of generality is not needed in this paper although it may be useful in other cases.

The upper bound (16) is important because it demonstrates the special role of telescoping polynomial $C_n(\omega)$. It is also pessimistic since it represents the worst case in which the extremal points ω_i from (9) and (12) coincide. The actual increase in the approximation error is usually lower than $\Delta a_n E_{c_n}$. Nevertheless, the telescoping polynomial promises lower increase than any other easily computed polynomial.

4. Telescoping rounding

Let us examine the polynomial $P^{(1)}(\omega)$. Its coefficient a_n is finite wordlength while the remaining $a_{n-1}^{(1)}, a_{n-2}^{(1)}, \ldots, a_0^{(1)}$ are not. They can, however, be made finite wordlength if the procedure described by (10)–(13) is applied repeatedly. The coefficient $a_{n-1}^{(1)}$ is replaced by the nearest finite wordlength $a_{n-1} \in I_b$,

$$a_{n-1} = a_{n-1}^{(1)} + \Delta a_{n-1}, \tag{18}$$

where $|\Delta a_{n-1}|$ is again the lowest distance from $a_{n-1}^{(1)}$ to a number in I_b. A new telescoping polynomial of order $n-1$ is defined as before

$$C_{n-1}(\omega) = \cos(n-1)\omega + \sum_{k=0}^{n-2} c_{n-1,\,k} \cos k\omega. \tag{19}$$

Telescoping polynomial $C_{n-1}(\omega)$ is a solution of a minimax problem

$$E_{c_{n-1}} = \min_{c_{n-1,\,k}} \max_{\omega\in\Omega} \left| W(\omega) C_{n-1}(\omega) \right|, \tag{20}$$

and the remaining coefficients are modified giving

$$a_k^{(2)} = a_k^{(1)} + \Delta a_{n-1} c_{n-1,\,k}, \quad k = 0, 1, \ldots, n-2. \tag{21}$$

These coefficients define a new polynomial $P^{(2)}(\omega)$ of degree n,

$$P^{(2)}(\omega) = a_n \cos n\omega + a_{n-1} \cos(n-1)\omega + \sum_{k=0}^{n-2} a_k^{(2)} \cos k\omega. \tag{22}$$

This polynomial has finite wordlength coefficients a_n and a_{n-1} while the remaining $n-1$ coefficients are not finite wordlength. The above procedure is repeated for polynomials $P^{(3)}(\omega)$, $P^{(4)}(\omega), \ldots, P^{(n+1)}(\omega)$ and it is clear that all $n+1$ coefficients of $P^{(n+1)}(\omega)$ are finite wordlength. As noted in (8), the coefficient $a_0^{(n)}$ is a special case—the nearest finite

wordlength coefficient a_0 must be selected from elements of I_b that are divided by 2. Note that this does not mean that elements of I_b must be divisible by 2.

The polynomial $P^{(n+1)}(\omega)$ represents the rounded FIR filter that we were looking for. Its approximation error

$$E_{I_b} = E^{(n+1)} = \max_{\omega\in\Omega} \left| W(\omega)\left(D(\omega) - P^{(n+1)}(\omega)\right)\right|, \tag{23}$$

is almost always lower than the one obtained by the simple rounding of the infinite precision coefficients a_k^* to their nearest finite wordlength representations from I_b. It can also be improved considerably by introducing additional search into the telescopic rounding process.

The idea for this improvement comes from the observation that in cases where $|\Delta a_k|$ is close to 0.5 the choice of the lowest $|\Delta a_k|$ is not necessarily the best. The absolute distances from an infinite precision coefficient $a_k^{(i)}$ to its nearest lower and upper finite wordlength neighbors are similar and therefore both worth investigating. This differs significantly from the cases where $|\Delta a_k|$ is close to zero. Since it is not possible to know in advance which choice is better, both the nearest lower and the nearest upper neighbor are tried in such cases—the choice that gives lower E_{I_b} is selected.

The notion of "close to 0.5" must be defined more precisely. It was determined experimentally that $|\Delta a_k| = 0.3$ represents a suitable threshold for deciding whether to try both choices or not. Our telescoping rounding method can now be summarized in the following steps:

1. Compute and save the coefficients of telescoping polynomials $C_1, C_2, \ldots, C_n$. This, as mentioned before, can be done using the same Remez algorithm that is used to compute the infinite precision coefficients a_k^*.
2. Redefine the infinite precision coefficients as

   $$a_k^{(0)} = a_k^*, \quad k = 0, 1, \ldots, n. \tag{24}$$

   Set the index i of the coefficient that is to be rounded next to n.
3. For coefficient $a_i^{(n-i)}$ compute its distance Δa_i to its nearest finite wordlength representation from I_b. If $|\Delta a_i| > 0.3$ go to step 4. Otherwise compute the finite wordlength coefficient $a_i = a_i^{(n-i)} + \Delta a_i$ and use telescoping polynomial C_i to compute the coefficients of polynomial $P^{(n+1-i)}(\omega)$,

   $$a_k^{(n+1-i)} = a_k^{(n-i)} + \Delta a_i c_{ik}, \quad k = 0, 1, \ldots, i-1. \tag{25}$$

   Go to step 5.
4. Both lower and upper neighbors of $a_i^{(n-i)}$ must be tried. Use Δa_i first and compute telescoping polynomials $P^{(n+1-i)}(\omega), P^{(n+2-i)}(\omega), \ldots, P^{(n+1)}(\omega)$ as described by (21)–(23). All $n+1$ coefficients of $P^{(n+1)}(\omega)$ are finite wordlength and its approximation error E_{I_b} is computed and saved. If $\Delta a_i < 0$ change Δa_i to $\Delta a_i + 1$, otherwise change it to $\Delta a_i - 1$. Repeat the procedure with the changed Δa_i and compute the corresponding E_{I_b}. If it is lower than the previous one, compute the finite wordlength coefficient $a_i = a_i^{(n-i)} + \Delta a_i$ and the coefficients (25) using the changed Δa_i. Otherwise use the starting Δa_i.

5. Replace index i by

$$i \leftarrow i - 1. \tag{26}$$

Return to step 3 if $i \geqslant 1$. Otherwise the coefficients $a_k,\ k = 0, 1, \ldots, n$, are all finite wordlength and we have obtained the rounded polynomial.

The most time consuming part of the telescoping rounding method are two computations of E_{I_b} in step 4. Computing the coefficients of telescoping polynomials C_k in step 1 may seem substantial since it requires n applications of the Remez algorithm, but is in fact modest. It is easy to see that the number of operations grows polynomially with n and not exponentially as is the case with optimal finite wordlength algorithms. The total time is almost negligible when compared to the time needed for a typical optimal finite wordlength solution.

The idea of searching for the best Δa_i can be extended to two or more coefficients. A two coefficient version of the method that searches simultaneously along Δa_i and Δa_{i-1} was implemented and tested. A direct extension of the one coefficient method would be to compute

$$a_{i-1}^{(n+1-i)} = a_{i-1}^{(n-i)} + \Delta a_i c_{ii-1}, \tag{27}$$

if $|\Delta a_i| > 0.3$ and then compute Δa_{i-1}. If there is also $|\Delta a_{i-1}| > 0.3$ approximation error E_{I_b} is computed for all four combinations of lower/upper neighbors of $a_i^{(n-i)}$, $a_{i-1}^{(n+1-i)}$ and the Δa_i that gives the lowest E_{I_b} is selected. Otherwise the procedure would remain the same as described in the one coefficient method. Although this works, the experiments have shown that it is better to use the criterion $|\Delta a_i + \Delta a_{i-1}| \leqslant 1.2$. Approximation error E_{I_b} is computed only for those combinations of lower/upper neighbors that satisfy this criterion and the Δa_i that gives the lowest E_{I_b} is selected. The reason for better performance of this criterion lies in the fact that it takes into account the property that opposite signs of Δa_i and Δa_{i-1} tend to produce approximation errors which, to a certain extent, cancel each other. Other criteria were tried and none performed better.

The two coefficient version is approximately two times slower than the one coefficient version. The corresponding rounded polynomial is usually, but not always, better than the polynomial obtained by the one coefficient method. It follows from (24)–(27) that it is possible to construct a set of infinite precision coefficients a_k^* that give a one coefficient rounded polynomial that is better than a two coefficient rounded polynomial. We therefore combined both methods and used the two coefficient polynomial only if it is better.

The amount of computation for a three coefficient version again increases by a factor of two and the same exponential increase follows for four or more coefficient search. The three coefficient version was tested and abandoned because the results show that it is only rarely better than the two coefficient version.

5. Results

Fifteen filters with five different sets of frequency-domain specifications, denoted A through E, were used for testing. The frequency specifications are identical to those that

Table 1
The five sets of filter specifications. The frequency edges are divided by 2π

Filter	Band 1	Band 2	Band 3
A			
Edges	0–0.2	0.25–0.5	
$D(\omega)$	1	0	
$W(\omega)$	1	1	
B			
Edges	0–0.2	0.25–0.5	
$D(\omega)$	1	0	
$W(\omega)$	1	10	
C			
Edges	0–0.12	0.2–0.34	0.42–0.5
$D(\omega)$	1	0	1
$W(\omega)$	1	1	1
D			
Edges	0–0.12	0.2–0.34	0.42–0.5
$D(\omega)$	1	0	1
$W(\omega)$	1	10	1
E			
Edges	0.01–0.21	0.26–0.49	
$D(\omega)$	1	0	
$W(\omega)$	1	1	

were used in Ref. [7] and are given in Table 1. A is a low-pass filter with unit weighting in both bands. B is the same, except that the stopband has a weighting of 10. C is a bandstop filter with unit weighting in all bands, while D has a weighting of 10 in stopband. E is a low-pass filter whose passband and stopbands do not include $\omega = 0$ or π.

We denote by A35/8 the filter design problem for specification A, length $N = 35$ ($n = 18$ independent coefficients), and $b = 8$ bits (sign included); similarly for A45/8, B35/9, and so on. Table 2 shows a summary of the results, comparing the infinite precision approximation error E^* and the finite wordlength approximation errors E_{I_b} obtained by different methods. The following methods are included: simple rounding to the nearest, one coefficient telescoping rounding, two coefficient telescoping rounding, and the optimal finite wordlength design.

The last column gives the relative quality of the two coefficient telescoping rounding. The results show that it is within 90% of the optimal solution in 12 out of 15 examples, but there are also two examples (C35/8 and D125/22) in which they are equal. Integer set I_b and a constant scaling factor $s = 2^{b-1}$ were used in all examples. This scaling was chosen for simplicity as well as to allow easier comparison with the older results.

As expected, both telescopic rounding methods are consistently better than the simple rounding to the nearest. The two coefficient telescoping method is better than the one coefficient method in 10 out of 15 filters, although the difference is often small. It is probably worth using since it is still quite fast. The computing time for the two coefficient telescoping rounding for all 15 filters was less than 5.1 s. Compare this with 2462 s that were

Table 2
Comparison of approximation errors using a constant scaling factor $s = 2^{b-1}$

Filter	E^*	Rounding to nearest E_{I_b}	One coefficient telescopic E_{I_b}	Two coefficient telescopic E_{I_b}	Optimal $E^*_{I_b}$	Relative quality $E^*_{I_b}/E_{I_b}$
A35/8	0.01594584	0.03266230	0.03266230	0.03266230	0.02983816	0.91
A45/8	0.00712762	0.03701502	0.03186462	0.03186462	0.02962304	0.93
A125/21	0.00000797	0.00001532	0.00001248	0.00001179	0.00001077†	0.91†
B35/9	0.05271937	0.15891206	0.09709565	0.07851429	0.07709547	0.98
B45/9	0.02104800	0.11718750	0.06640625	0.06640625	0.05679037	0.86
B125/22	0.00002489	0.00004675	0.00003322	0.00003293	0.00002959†	0.90†
C35/8	0.00262898	0.04687500	0.01787084	0.01787084	0.01787084	1.00
C45/8	0.00066997	0.03045225	0.02287919	0.02103044	0.01609009	0.77
C125/21	0.00000001	0.00000878	0.00000220	0.00000210	0.00000206†	0.98†
D35/9	0.01043321	0.12197080	0.03368525	0.03368525	0.03252775	0.97
D45/9	0.00223461	0.10904023	0.03224953	0.02859819	0.02612254	0.91
D125/22	0.00000004	0.00004034	0.00000380	0.00000216	0.00000216†	1.00†
E35/8	0.01760590	0.04692227	0.04221047	0.03399270	0.03299053	0.97
E45/8	0.00653752	0.03577490	0.03549788	0.03403126	0.02887703	0.85
E125/21	0.00000787	0.00001429	0.00001158	0.00001127	0.00001034†	0.92†

needed to compute all 15 optimal finite wordlength filters (time for each of the $N = 125$ filters was limited to 600 s). A 2.4 GHz Pentium 4 PC was used as a platform for all experiments.

Knowing the optimal finite wordlength approximation error $E^*_{I_b}$ is of course necessary for evaluation of any rounding method. This creates a problem when long filters are used because it is impossible to find the optimal solution in a reasonable time. Such is the case of length $N = 125$ filters which were included to demonstrate that the telescoping method also works for long filters. The values $E^*_{I_b}$ for $N = 125$ are marked by † to indicate that they are estimates that were obtained after 600 s of computation and were not proved to be optimal.

The long finite wordlength filters deserve an additional comment. It has been shown in Ref. [8] that for a given number of bits b there exists a nonzero lower bound on the approximation error, below which it is not possible to go, no matter how large the length N. Furthermore, it is possible to demonstrate that for all optimal finite wordlength filters there exists an index l beyond which the optimal finite wordlength coefficients a_k are all zero. Or formally

$$a_k = 0, \quad k \geqslant l + 1, \tag{28}$$

where l is a function of b and of desired frequency response. No method for computation of l is known at this time, although it can be determined experimentally. For example, the optimal finite wordlength filter D45/9 in Table 2 is in fact of length 39—the remaining coefficients are zero. Increasing its length to, say, $N = 301$ would only give additional zero coefficients. This means that designing long finite wordlength filters is appropriate only if a correspondingly large number of bits b is used. Such is the case of $N = 125$ filters in our examples where 21 and 22 bits were used.

To further demonstrate the effectiveness of telescoping rounding method, we also tested it on a more complicated case of variable scaling factor s. The optimal scaling factor s_{opt}

can be obtained together with the optimal finite wordlength coefficients if s is included as a variable in the minimax approximation problem (5)

$$\min_{s,\,P(\omega)} \max_{\omega\in\Omega} \big| W(\omega)\big(D(\omega) - sP(\omega)\big)\big|, \tag{29}$$

where $P(\omega)$ is the finite wordlength polynomial. This problem is significantly more difficult to solve than the one in which s is a constant. A much simpler method that gives a good suboptimal s is obviously needed when rounding is used. We use a heuristic that is similar to the one used in Ref. [9] and can be used with any rounding method. Its details depend somewhat on the nature of discrete set I_b. A version for integer set I_b that was used in our experiments is described in the following steps:

1. Starting with $s = 2$ use (6) to compute the scaled $D(\omega)$, $W(\omega)$ and use the Remez algorithm to compute the infinite precision coefficients a_k^*. Round a_k^* to the finite wordlength $a_k \in I_b$ using telescoping or some other rounding method. The corresponding approximation error E_{I_b} is computed and saved. The scaling factor s is multiplied by 2, the coefficients a_k^* are again computed, rounded, and E_{I_b} is computed. This process is repeated until the maximum $|a_k^*|$ exceeds the maximum integer in I_b by $n/2$. As noted before, coefficient a_0^* is a special case and must be multiplied by 2 during the search for maximum $|a_k^*|$. The scaling factor s that gave the lowest E_{I_b} is saved as s_2.
2. The power of 2 factor $s_0 = s_2$ is used as a starting value for additional search upwards and downwards from s_2. The integer search step Δ_s is defined as

$$\Delta_s = \max(1, s_2/128). \tag{30}$$

   Starting with $s = s_2 + \Delta_s$ the upward search with s increasing by Δ_s continues until the maximum $|a_k^*|$ exceeds the maximum integer in I_b by $n/2$ as in step 1. If a lower E_{I_b} is found, the corresponding s is used as the new best scaling integer s_0. The highest s that was used is saved as $s_{\max}$ and the search is repeated in the downward direction starting with $s = s_2 - \Delta_s$. The downward search stops when s falls below $s_{\max}/2$. It follows from (6) that it is extremely unlikely for such s to improve E_{I_b} because they give the coefficients a_k^* which are $1/2$ of those that were already tried.

The basic idea is to get a rough estimate for s in step 1 and then improve it in step 2. Because of (30) the number of E_{I_b} computations is typically less than 128. The criterion "maximum integer in I_b plus $n/2$" that is used to stop the upward search is steps 1 and 2 is based on the observation that E_{I_b} starts to grow when infinite precision coefficients a_k^* begin to exceed the maximum element of I_b. The $n/2$ part is used to ensure that this also holds for the telescoped coefficients $a_k^{(n-i)}$ in (25). As is true for any heuristic, we do not claim that the integer s_0 is the best possible integer scaling factor.

Instead of using s_0 we include an additional improvement that gives a better noninteger scaling factor s^*. This improvement follows from the observation that a lower value of E_{I_b} is available with some additional computation. Assume that for a given scaling factor s all coefficients a_k are integers from I_b. The approximation error E_{I_b} that is given by (23) can be reduced if the following minimax approximation problem is solved:

$$\min_{t} \max_{\omega\in\Omega} \big| W(\omega)\big(D(\omega) - tP^{(n+1)}(\omega)\big)\big|, \tag{31}$$

Table 3
Comparison of approximation errors using variable scaling factors s

Filter	E^*	Rounding to nearest E_{I_b}	One coefficient telescopic E_{I_b}	Two coefficient telescopic E_{I_b}	Optimal $E^*_{I_b}$	Relative quality $E^*_{I_b}/E_{I_b}$
A35/8	0.01594584	0.02235840	0.02092188	0.02075809	0.01979400	0.95
A45/8	0.00712762	0.01628142	0.01332987	0.01332987	0.01332987	1.00
A125/21	0.00000797	0.00001025	0.00000959	0.00000939	0.00000903†	0.91†
B35/9	0.05271937	0.08009994	0.06161027	0.06008090	0.05858363	0.98
B45/9	0.02104800	0.06010068	0.03189451	0.03189451	0.03176571	0.99
B125/22	0.00002489	0.00003587	0.00002788	0.00002755	0.00002682†	0.97†
C35/8	0.00262898	0.01371235	0.01024694	0.01012225	0.01002662	0.99
C45/8	0.00066997	0.01392602	0.00967561	0.00912290	0.00847374	0.93
C125/21	0.00000001	0.00000393	0.00000112	0.00000109	0.00000109†	1.00†
D35/9	0.01043321	0.02642507	0.02102677	0.02102677	0.01917670	0.91
D45/9	0.00223461	0.04796042	0.01282368	0.01282368	0.01282368	1.00
D125/22	0.00000004	0.00001250	0.00000124	0.00000124	0.00000124†	1.00†
E35/8	0.01760590	0.02507156	0.02293072	0.02231877	0.02200041	0.99
E45/8	0.00653752	0.01659247	0.01523235	0.01491765	0.01347661	0.90
E125/21	0.00000787	0.00001023	0.00000923	0.00000916	0.00000888†	0.97†

for variable t. Since $P^{(n+1)}(\omega)$ is known (31) can be rewritten

$$\min_t \max_{\omega\in\Omega} \left| W(\omega) P^{(n+1)}(\omega) \left(\frac{D(\omega)}{P^{(n+1)}(\omega)} - t \right) \right|. \tag{32}$$

This is a one variable minimax approximation problem. It follows from the Chebyshev equioscillation theorem (9) that there are two extremal points ω_0 and ω_1 at which the approximation error achieves its maximum

$$\left| W(\omega_i) P^{(n+1)}(\omega_i) \right| \left(\frac{D(\omega_i)}{P^{(n+1)}(\omega_i)} - t^* \right) = (-1)^i d, \quad i = 0, 1. \tag{33}$$

This is easy to solve with either the general Remez algorithm or its faster, simplified one variable version. Solution t^* gives a noninteger scaling factor $s^* = t^* s$ which gives the reduced approximation error. It is computed in steps 1 and 2 for all instances of E_{I_b} computation. The scaling factor s^* that gives the lowest $E_{I_b} = |d|$ is the result of our heuristic and was used in experiments that are given in Table 3. Note that this method of scaling factor computation was also used for the case of rounding to the nearest in column 3.

The variable scaling factors give approximation errors that are significantly lower than those from Table 2. Both telescopic rounding methods are again consistently better than the simple rounding to the nearest. The results show that the two coefficient telescoping rounding is within 90% of the optimal solution in all 15 examples and there are also four examples (A45/8, C125/21, D45/9, D125/22) in which they are equal.

The price for much lower approximation errors is the increase in the computing time. The variable scaling factor two coefficient telescoping rounding for all 15 filters took 523 s. Compare this with 5231 s that were needed to compute all 15 optimal finite wordlength filters with optimal scaling factor. Computing time for each of the $N = 125$ filters was again limited. The limit was 1200 s and these filters were not proved to be optimal. The times are much longer than the constant scaling factor times for examples from Table 2.

Integer set I_b was used in our experiments but similar results can be expected for other discrete sets I_b.

We also used the telescoping rounding to reduce the amount of computation in our branch-and-bound based algorithm for optimal finite wordlength FIR filter design. The reduction results from (1) having a better starting solution and (2) from using telescopic rounding on selected subproblems to "guess" a possible better solution before it would be found otherwise. The observed degree of reduction differs considerably from one problem to another. We found that it tends to be higher for smaller problems but nevertheless worth doing for all.

6. Conclusion

This paper presents a new rounding method for suboptimal finite wordlength FIR digital filter. The method is simple to implement and produces filters that are much better than those obtained by simple rounding of coefficients to their nearest finite wordlength representation. Design examples have confirmed its effectiveness.

References

[1] D.S. Chan, L.R. Rabiner, Analysis of quantization errors in the direct form for finite impulse response digital filters, IEEE Trans. Acoust. Speech Signal Process. 21 (1973) 354–366.
[2] A. Gersho, B. Gopinath, A.M. Odlyzko, Coefficient inaccuracy in transversal filtering, Bell System Tech. J. 58 (1979) 2301–2316.
[3] T.W. Parks, J.H. McClellan, A program for the design of linear phase finite impulse response filters, IEEE Trans. Audio Electroacoust. 20 (1972) 195–199.
[4] P.J. Davis, Interpolation and Approximation, Dover, New York, 1975, pp. 149–151.
[5] D.M. Kodek, Telescoping minimax approximation for finite wordlength FIR filter design, in: Proceedings of Conference on Information Sciences and Systems (CISS), Princeton, 1990, pp. 186–191.
[6] M.J.D. Powell, Approximation Theory and Methods, Cambridge Univ. Press, Cambridge, 1981, pp. 78–79.
[7] D.M. Kodek, K. Steiglitz, Comparison of optimal and local search methods for designing finite wordlength FIR digital filters, IEEE Trans. Circuits Syst. 28 (1982) 28–32.
[8] D.M. Kodek, K. Steiglitz, Filter-length wordlength tradeoffs in FIR digital filter design, IEEE Trans. Acoust. Speech Signal Process. 28 (1980) 739–744.
[9] D.M. Kodek, Design of optimal finite wordlength FIR digital filters, in: Proceedings of European Conference on Circuit Theory and Design ECCTD '99, vol. I, Stresa, Italy, 1999, pp. 401–404.

Dusan M. Kodek received the BEE degree in 1970, the MEE degree in 1973, and the PhD degree in 1975, all from the University of Ljubljana, Ljubljana, Slovenia. Since 1971 he has been with the Faculty of Electrical Engineering and from 1996 with the new Faculty of Computer and Information Science, where he is now a professor, teaching and conducting research in the computer architecture and digital signal processing areas. Professor Kodek is the author of three books on computer architecture and microprocessor system design and served as the first Dean of the faculty from 1996 to 2001. He is a recipient of three Slovenian awards for research and innovation.

Marjan Krisper received the BME degree in 1971, the MME degree in 1977, all from the University of Ljubljana, Ljubljana, Slovenia, and the PhD degree in 1990 from the University of Belgrade,

Belgrade, Yugoslavia. Since 1982 he has been with the Faculty of Electrical Engineering and from 1996 with the new Faculty of Computer and Information Science, where he is now an assistant professor. His research interests are in the area of information systems. Dr. Krisper is the author of two books on information systems development.