Performance Limit of Finite Wordlength FIR Digital Filters

Dušan M. Kodek, *Senior Member, IEEE*

***Abstract*—In many practical situations, it is necessary to represent the coefficients of a finite impulse response (FIR) digital filter by a finite number of bits. This not only degrades the filter frequency response but also introduces a theoretical limit on the performance of the filter. Derivation of a lower bound on filter degradation is the purpose of this paper. We consider a general case of a length N filter with a discrete set of allowable coefficients. A theorem that gives the lower bound on the increase in minimax approximation error that is caused by the finite wordlength restriction is presented. Its extension and application to filter design cases is demonstrated. The importance of this bound is not only theoretical. Its practical effectiveness is shown in the algorithm for optimal finite wordlength FIR filter design where it significantly reduces the amount of computation.**

***Index Terms*—FIR digital filters, finite wordlength, minimax approximation, performance limits.**

I. Introduction

IT is often not practical to use the optimal finite impulse response (FIR) digital filter coefficients obtained by some “infinite precision” algorithm. The so-called infinite precision coefficients are typically 32-bit floating point numbers. Although the 32-bit wordlength is hardly infinite, it is much longer than practical finite wordlengths in which we are interested. One may, for example, wish to use a fixed point DSP processor which is usually cheaper and/or faster than a floating-point one. The number of bits b that can be used to represent the filter coefficients will in general depend on the filter length N, processor properties, and on signal quantization, but it is almost always true that filter coefficients with as short as possible wordlength b are desirable.

Replacing the optimal filter coefficients with the b-bit ones degrades filter’s frequency response. The number of bits b must therefore not be too short, or the filter will no longer be good enough. The designer faces the following question: Given the filter specifications, what is the lowest number of bits b that will give an acceptable finite wordlength filter? It is clear that this question cannot be answered by rounding the coefficients to b bits and computing the frequency response—rounding gives a suboptimal filter, which can be up to 30 dB worse than the optimal b-bit filter. What is needed is a frequency response of the optimal filter, and herein lies the problem. Designing an optimal finite wordlength filter requires a solution of an NP-complete approximation problem that is not easy to solve. Most designers would prefer to know in advance if the result is worth trying at all, and this paper is an attempt in this direction.

Many papers on the practical aspects of finite wordlength FIR filter design have been published in the literature. They typically use one of the two types of finite wordlength coefficient constraints: signed b-bit integers [1]–[3] or sums of a limited number of signed power-of-two terms [4]–[6]. The integer coefficients can be used, for example, with the fixed-point DSP processors, whereas the sums of power-of-two allow a multiplierless implementation. Various versions of integer programming techniques were used to solve the approximation problem.

It is perhaps surprising that so little is known about the theoretical aspects of the problem of finite wordlength FIR filter design. Some initial results were given in [7]–[9], where it was found that it is not possible to meet arbitrarily severe FIR filter specifications with the fixed b-bit wordlength by increasing the filter length N. A more difficult problem of estimating the increase in minimax (Chebyshev) approximation error that is caused by the b-bit constraint for a given filter of length N was left unanswered.

This paper presents a new method that solves this problem by deriving a lower bound theorem for the increase in minimax approximation error. This bound is a theoretical limit on the performance of a given b-bit FIR filter of length N. The motivation for developing the bound is, however, very practical. It can be used to significantly reduce the amount of computation in the algorithm for optimal finite wordlength FIR filter design.

II. Finite Wordlength Design Problem

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

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

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

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

Manuscript received December 29, 2003; revised July 31, 2004. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Anamitra Makur.

The author is with the Faculty of Computer and Information Science, University of Ljubljana, 1000 Ljubljana, Slovenia (e-mail: duke@fri.uni-lj.si).

Digital Object Identifier 10.1109/TSP.2005.849191

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

$$\min_{P(\omega)} \max_{a \le \omega \le b} |W(\omega)(D(\omega) - P(\omega))|. \tag{3}$$

The real function $D(\omega)$ is the desired frequency response, the weighting function $W(\omega)$ is by definition real and positive, and the interval $[a, b]$ is a subset (or a union of subsets) of the interval $[0, \pi]$.

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 described by Parks and McClellan [10]. 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.

We can, without loss of generality, make the finite wordlength constraint equal to requesting that the filter coefficients $h(k)$ are 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 b-bit numbers (sums of a limited number of power-of-two terms, 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 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 as $D_u(\omega), W_u(\omega)$, and $P_u(\omega)$ the original (unscaled) problem. The approximation problem can be rewritten as

$$\min_{P_u(\omega)} \max_{a \le \omega \le b} \left| \frac{W_u(\omega)}{s} (sD_u(\omega) - sP_u(\omega)) \right| = \min_{P(\omega)} \max_{a \le \omega \le b} |W(\omega)(D(\omega) - P(\omega))| \tag{4}$$

where

$$P(\omega) = sP_u(\omega) = \sum_{k=0}^{n} a_k \cos k\omega, \quad a_k \in I_b \tag{5}$$

is the finite wordlength polynomial, and $D(\omega) = sD_u(\omega)$ and $W(\omega) = W_u(\omega)/s$ are the scaled input functions. Observe that $a_k \in I_b$ in (5). 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

$$\begin{aligned} h(n) &= a_0, \quad n = (N-1)/2 \\ h(n-k) &= a_k/2, \quad k = 1, 2, \ldots, n. \end{aligned} \tag{6}$$

Obviously, if $h(k) \in I_b$, then a_k must be from the set of even numbers of twice the size of those in I_b for $k \ge 1$. Dividing the scaling factor s in (4) 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 (6) by

$$\begin{aligned} h(n) &= 2a_0, \quad n = (N-1)/2 \\ h(n-k) &= a_k, \quad k = 1, 2, \ldots, n. \end{aligned} \tag{7}$$

Note that the coefficient a_0 is a special case. Its values are constrained to the elements of I_b divided by 2, which is only a minor complication.

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 (5) that scaling factor s can be interpreted as the filter gain. Scaling can also be used in the infinite precision case, where it does not affect the approximation error. Things are different in the finite wordlength design where approximation error changes with s. The choice of s is not trivial and is worth discussing a little more. Two different approaches are used in practice.

1) The scaling factor s is included in the minimax approximation problem [11] as a variable

$$\min_{s, P(\omega)} \max_{a \le \omega \le b} |W(\omega)(D(\omega) - sP(\omega))|. \tag{8}$$

This gives the optimal scaling factor and the lowest approximation error but is significantly more difficult to solve than (4).

2) A constant scaling factor s determined by some *ad hoc* method is used. A typical method is a positive integer s that is obtained by selecting the maximum s for which all products $s \cdot h^*(k)$ do not exceed the maximum element from I_b. Since the optimal b-bit $h(k)$ differs from $h^*(k)$, it is possible that one or more of the optimal $h(k)$ fall out of I_b. If this occurs, the initial s is reduced by 1, $D(\omega)$ and $W(\omega)$ redefined, and computation repeated. Several iterations may be needed.

A comparison between the optimal scaling factor and the above *ad hoc* integer scaling factor was done in [12]. The results show that the optimal scaling factor approximation error was on average about 8% (0.68 dB) lower than the one obtained by the integer scaling factor. The signed integer set I_b was used in these experiments, and it is possible that this difference is higher for other discrete sets.

In this paper, we assume that s is a known constant. Extension of the results to the case of optimal s is more complicated and is not presented here.

III. Lower Bound Derivation

Notation $P(\omega)$ will from here on denote a polynomial of degree n with b-bit coefficients from I_b, whereas $P^*(\omega)$ remains the optimal infinite precision polynomial. $D(\omega)$ and $W(\omega)$ are the scaled input functions in both cases. The well-known Chebyshev equioscillation theorem (also known as the alternation theorem) [13] gives the conditions for the optimal minimax approximation of degree n: There are at least $n+2$ so-called extremal points in $[a, b]$ at which the approximation error achieves its maximum. Let $\omega_i, a \leq \omega_0 < \omega_1 < \cdots < \omega_{n+1} \leq b$ be these extremal points. The following hold:

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

where $|d^*|$ is the optimal approximation error. No such property exists for b-bit $P(\omega)$. Approximation error $e(\omega)$ is simply

$$W(\omega)\left(D(\omega) - \sum_{k=0}^{n} a_k \cos k\omega\right) = e(\omega). \quad (10)$$

Since $P^*(\omega)$ is unique, the approximation error increases if $P(\omega) \neq P^*(\omega)$. There is

$$\epsilon = \max_{a \leq \omega \leq b} |e(\omega)| - |d^*| \quad (11)$$

where $\epsilon > 0$.

The problem we wish to address can be stated as follows: What is the minimum ϵ given the best possible coefficients $a_k \in I_b$? Stated differently, how much will $\max|e(\omega)|$ increase relative to $|d^*|$ because of the b-bit constraint. The lowest possible ϵ is needed to answer this question. This lowest ϵ is a theoretical limit on the performance of a given b-bit FIR digital filter of length N. Let us denote it as δ and define it formally as

$$\delta = \min_{b-\text{bit}\, P(\omega)} \epsilon = \min_{b-\text{bit}\, P(\omega)} \max_{a \leq \omega \leq b} |e(\omega)| - |d^*|. \quad (12)$$

To get a lower bound for δ, we must be able to express it as a function of differences $a_k^* - a_k, k = 0, 1, \ldots, n$. This will be done following an approach similar to the one used in [14] and [15]. Let us use the extremal points $\omega_i, i = 0, 1, \ldots, n+1$, and combine (9) and (10) into

$$e(\omega_i) = \sum_{k=0}^{n} W(\omega_i)(a_k^* - a_k) \cos k\omega_i + (-1)^i d^*. \quad (13)$$

These equations can be viewed as a system of $n+2$ equations with $n+2$ unknowns. The unknowns are $a_k^* - a_k$ and d^*. Note that the system's matrix is identical to the one in (9). Since (9) is already solved (to get a_k^* and d^*), it is clear that (13) is always invertible. The inverse can be written as

$$a_k^* - a_k = \sum_{i=0}^{n+1} g_{ki} \frac{e(\omega_i)}{W(\omega_i)}, \quad k = 0, 1, \ldots, n \quad (14)$$

$$d^* = \sum_{i=0}^{n+1} g_{n+1i} \frac{e(\omega_i)}{W(\omega_i)} \quad (15)$$

where g_{ki} are the elements of the inverted matrix. This matrix has some very useful properties that become visible if (13) is rewritten as

$$e(\omega_i) = W(\omega_i)(P^*(\omega_i) - P(\omega_i)) + (-1)^i d^*. \quad (16)$$

Inserting (16) into (14) and (15) gives

$$a_k^* - a_k = \sum_{i=0}^{n+1} g_{ki}\left(P^*(\omega_i) - P(\omega_i) + \frac{(-1)^i d^*}{W(\omega_i)}\right) \quad (17)$$

$$d^* = \sum_{i=0}^{n+1} g_{n+1i}\left(P^*(\omega_i) - P(\omega_i) + \frac{(-1)^i d^*}{W(\omega_i)}\right). \quad (18)$$

Setting $a_k = a_k^*$ for all k gives $P(\omega) = P^*(\omega)$ and

$$0 = \sum_{i=0}^{n+1} g_{ki} \frac{(-1)^i}{W(\omega_i)}, \quad k = 0, 1, \ldots, n \quad (19)$$

$$1 = \sum_{i=0}^{n+1} g_{n+1i} \frac{(-1)^i}{W(\omega_i)}. \quad (20)$$

This means that (17) and (18) can be simplified into

$$a_k^* - a_k = \sum_{i=0}^{n+1} g_{ki}(P^*(\omega_i) - P(\omega_i)), \quad k = 0, 1, \ldots, n \quad (21)$$

$$0 = \sum_{i=0}^{n+1} g_{n+1i}(P^*(\omega_i) - P(\omega_i)). \quad (22)$$

Two important properties of matrix elements g_{ki} now follow from (19)–(22).

1) For $k = 0, 1, \ldots, n$, at least two of g_{ki} are nonzero. This is easy to see because (21) holds for an arbitrary $P(\omega)$ of degree n. This means that it also holds for $P(\omega)$ with $a_k^* - a_k = u$, where u is an arbitrary nonzero number. Obviously, this is impossible if all g_{ki} are zero. Furthermore, it follows from (19) that at least two g_{ki} must be nonzero, or formally

$$g_{ki} \neq 0, \quad \text{for at least two } i, \quad k = 0, 1, \ldots, n. \quad (23)$$

2) For $k = n+1$, all of g_{n+1i} are nonzero, and their signs alternate

$$\text{sign}(g_{n+1i+1}) = -\text{sign}(g_{n+1i}), \quad i = 0, 1, \ldots, n. \quad (24)$$

This follows from a property of all functions that satisfy the Haar condition, of which cosine polynomial is a special case [16]. For any $P(\omega)$ of degree n and any set of $n+2$ distinct points ω_i, the nonzero numbers g_{n+1i} from (22) always satisfy (24). Using (20), we also get

$$g_{n+1i}(-1)^i > 0, \quad i = 0, 1, \ldots, n+1. \quad (25)$$

For any set of optimal coefficients a_k^*, there exist numbers a_{k+} and a_{k-}, both from I_b, that are the nearest upper and lower neighbors of a_k^*. In other words, a_{k+} is an element of I_b that gives the smallest positive difference $a_k - a_k^*$, and a_{k-} is an element of I_b that gives the smallest (in an absolute sense) negative difference $a_k - a_k^*$. Having a_{k+} and a_{k-}, we can now prove the following theorem.

Theorem 1: Let $P^*(\omega)$ be the optimal weighted minimax approximation to a real function $D(\omega)$ on the interval $[a, b]$, and let

$P(\omega)$ be a cosine polynomial with coefficients from I_b. Then, the increase in approximation error δ is bounded by

$$\delta \geq \max_{0\leq k\leq n} \min\left(\frac{a_{k+} - a_k^*}{f_{mk}}, \frac{a_{k-} - a_k^*}{f_{pk}}\right) \tag{26}$$

where the positive factors f_{mk} and the negative factors f_{pk} are defined as

$$f_{mk} = \max_{0\leq i\leq n+1}\left(\mathrm{sign}(d^*)\frac{g_{ki}}{g_{n+1i}}\right), \quad k = 0, 1, \ldots, n \tag{27}$$

$$f_{pk} = \min_{0\leq i\leq n+1}\left(\mathrm{sign}(d^*)\frac{g_{ki}}{g_{n+1i}}\right), \quad k = 0, 1, \ldots, n. \tag{28}$$

Proof: It follows from (12) that finding a lower bound on δ is equivalent to finding a lower bound on $\max|e(\omega)|$. This can be done with the help of (16) if we define the subset Z_P of extremal indices $i, i = 0, 1, \ldots, n+1$, as

$$i \in Z_P \quad \text{if } \mathrm{sign}(d^*)(-1)^i W(\omega_i)(P^*(\omega_i) - P(\omega_i)) \geq 0. \tag{29}$$

For $i \in Z_P$, the sign of $P^*(\omega_i) - P(\omega_i)$ equals the sign of $(-1)^i d^*$, and (16) gives the bound

$$\begin{aligned}\max_{a\leq\omega\leq b}|e(\omega)| &\geq \max_{i\in Z_P}|e(\omega_i)| \\ &\geq \max_{i\in Z_P} W(\omega_i)|P^*(\omega_i) - P(\omega_i)| + |d^*|\end{aligned} \tag{30}$$

where we used the fact that the maximum approximation error $|e(\omega)|$ on the interval $[a, b]$ cannot be lower than the maximum on its subset Z_P. For a given $P(\omega)$, (30) gives

$$\epsilon \geq \max_{i\in Z_P} W(\omega_i)|P^*(\omega_i) - P(\omega_i)|. \tag{31}$$

To get an estimate on ϵ over all b-bit $P(\omega)$, let us multiply and divide each term in (21) and (22) with $(-1)^i\mathrm{sign}(d^*)/W(\omega_i)$

$$\begin{aligned}a_k^* - a_k = \sum_{i=0}^{n+1} &\frac{\mathrm{sign}(d^*)(-1)^i g_{ki}}{W(\omega_i)} \\ &\cdot \mathrm{sign}(d^*)(-1)^i W(\omega_i)(P^*(\omega_i) - P(\omega_i))\end{aligned} \tag{32}$$

$$\begin{aligned}0 = \sum_{i=0}^{n+1} &\frac{\mathrm{sign}(d^*)(-1)^i g_{n+1i}}{W(\omega_i)} \\ &\cdot \mathrm{sign}(d^*)(-1)^i W(\omega_i)(P^*(\omega_i) - P(\omega_i)).\end{aligned} \tag{33}$$

These two contain the terms $\mathrm{sign}(d^*)(-1)^i W(\omega_i)(P^*(\omega_i) - P(\omega_i))$, which appear in (29). The proof is simplified if (33) is first multiplied by $\mathrm{sign}(d^*)$

$$\begin{aligned}0 = \sum_{i=0}^{n+1} &\frac{(-1)^i g_{n+1i}}{W(\omega_i)} \\ &\cdot \mathrm{sign}(d^*)(-1)^i W(\omega_i)(P^*(\omega_i) - P(\omega_i)).\end{aligned} \tag{34}$$

By multiplying (34) with an arbitrary factor f and subtracting it from (32), we get

$$a_k^* - a_k = \sum_{i=0}^{n+1} g'_{ki}\mathrm{sign}(d^*)(-1)^i W(\omega_i)(P^*(\omega_i) - P(\omega_i)) \qquad k = 0, 1, \ldots, n \tag{35}$$

where the new coefficients g'_{ki} are defined as

$$g'_{ki} = \frac{\mathrm{sign}(d^*)(-1)^i g_{ki} - f(-1)^i g_{n+1i}}{W(\omega_i)}. \tag{36}$$

Because of (25), all terms $(-1)^i g_{n+1i}$ are nonzero and positive. This means that it is always possible to find factors f which make all g'_{ki} negative or positive. Note that it follows from (19) and (23) that such factors can never be zero.

Let us examine the case $a_k = a_{k+}$, which gives $a_k^* - a_k \leq 0$ in (35). It is easy to see that a factor $f = f_{mk}$, which is defined by (27), is the smallest positive number that makes all $g'_{ki} \leq 0$. Since $g'_{ki} \leq 0, i = 0, 1, \ldots, n+1$, there must be at least one index i that is in Z_P. Furthermore, it is obvious that $\max_{i\in Z_P}|e(\omega_i)|$ is the lowest when all indices i are in Z_P. Equation (35) gives

$$\max_{i\in Z_P} W(\omega_i)|P^*(\omega_i) - P(\omega_i)| \geq \frac{a_k^* - a_{k+}}{\sum_{i=0}^{n+1} g'_{ki}}. \tag{37}$$

Using (19) and (20), it is possible to simplify the g'_{ki} sum into

$$\sum_{i=0}^{n+1} g'_{ki} = \sum_{i=0}^{n+1} \frac{\mathrm{sign}(d^*)(-1)^i g_{ki} - f_{mk}(-1)^i g_{n+1i}}{W(\omega_i)} = -f_{mk} \tag{38}$$

and the following bound follows from (31) and (37)

$$\epsilon \geq \frac{a_{k+} - a_k^*}{f_{mk}}. \tag{39}$$

Similarly, a negative factor $f = f_{pk}$, which is defined by (28), makes all $g'_{ki} \geq 0$. Using $a_k = a_{k-}$ gives $a_k^* - a_k \geq 0$, and the following bound is obtained:

$$\epsilon \geq \frac{a_{k-} - a_k^*}{f_{pk}}. \tag{40}$$

The numbers a_{k+} and a_{k-}, both from I_b, are defined as the nearest upper and lower neighbors of a_k^*. Obviously, no other choice of $a_k \in I_b$ can give ϵ that is lower than both (39) and (40). This means that δ, which is defined by (12), is bounded by

$$\delta \geq \min\left(\frac{a_{k+} - a_k^*}{f_{mk}}, \frac{a_{k-} - a_k^*}{f_{pk}}\right). \tag{41}$$

Since this holds for all $k, k = 0, 1, \ldots, n$, choosing the largest as in (26) gives the best lower bound. This completes the proof. ■

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

It is interesting to observe the lower bound behavior when $n \to \infty$. For a given finite k, the coefficient a_k^* becomes almost independent of n as it goes toward infinity. This means that the same is true for the corresponding differences $a_{k+} - a_k^*$ and $a_{k-} - a_k^*$ in (26). Things are different for the absolute values of denominators f_{mk} and f_{pk}, which tend to grow with n. This follows from (20) and (25), where g_{n+1i} must decrease with n, whereas this is not true for $g_{ki}, k \leq n$. We can expect that the lower bound will decrease for large n. The rate of decrease is, however, slower than the rate of decrease of $|d^*|$. The ratio of lower bound to $|d^*|$ therefore grows with n.

IV. Improved Lower Bound

The bound given by the *Theorem 1* is quite easy to compute. It can also be considerably improved because the theorem uses only one of the $n+1$ coefficient differences $a_k^* - a_k$, namely, the one that gives the maximum in (26). The other n differences play no part, which is identical to saying that only one of the coefficients a_k must be from I_b. Since this is not the case, an improved lower bound can be obtained by combining two or more of (35).

Let us select any two of the $k = 0, 1, \ldots, n$ equations in (35) and denote the corresponding indices as j and l. Equation j is multiplied by a factor γ and subtracted from equation l, which gives

$$\begin{aligned} &a_l^* - a_l - \gamma(a_j^* - a_j) \\ &\quad = \sum_{i=0}^{n+1} t_i \operatorname{sign}(d^*)(-1)^i W(\omega_i)(P^*(\omega_i) - P(\omega_i)) \end{aligned} \tag{42}$$

where the new coefficients t_i are equal to

$$t_i = \frac{\operatorname{sign}(d^*)(-1)^i(g_{li} - \gamma g_{ji}) - f(-1)^i g_{n+1i}}{W(\omega_i)}. \tag{43}$$

Following the same approach as before, we define a positive factor $f = f_{mjl}$ that makes all $t_i \le 0$ and a negative factor $f = f_{pjl}$ that makes all $t_i \ge 0$

$$f_{mjl} = \max_{0 \le i \le n+1} \left(\operatorname{sign}(d^*) \frac{g_{li} - \gamma g_{ji}}{g_{n+1i}} \right) \tag{44}$$

$$f_{pjl} = \min_{0 \le i \le n+1} \left(\operatorname{sign}(d^*) \frac{g_{li} - \gamma g_{ji}}{g_{n+1i}} \right). \tag{45}$$

Because of $(-1)^i g_{n+1i} > 0$, factors f_{mjl} and f_{pjl} exist for any j, l, and γ. Furthermore, they are always nonzero. This is easy to see if (42) is rewritten as

$$a_l^* - a_l - \gamma(a_j^* - a_j) = \sum_{i=0}^{n+1} (g_{li} - \gamma g_{ji})(P^*(\omega_i) - P(\omega_i)). \tag{46}$$

Using the same argument as in (23), it is obvious that a γ giving $g_{li} - \gamma g_{ji} = 0$ for all i cannot exist, which in turn means that f_{mjl} and f_{pjl} are nonzero.

Assume now that a_j and a_l, both from I_b, are known and define a function ϵ_{jl}

$$\epsilon_{jl} = \max\left(\frac{a_l^* - a_l - \gamma(a_j^* - a_j)}{-f_{mjl}}, \frac{a_l^* - a_l - \gamma(a_j^* - a_j)}{f_{pjl}} \right). \tag{47}$$

Comparison with (39) and (40) shows that this corresponds to the following bound:

$$\epsilon \ge \epsilon_{jl}. \tag{48}$$

One of the terms in (47) is always positive and the other one always negative. The negative term does not contribute to the bound; the problem is that it is not known which one is negative because they both depend on γ, which has yet to be determined. The value of γ must be chosen so that the function ϵ_{jl} is as high as possible. To find such γ, it is necessary to solve the following optimization problem:

$$\max_{\gamma, f_{mjl}, f_{pjl}} \epsilon_{jl} \tag{49}$$

subject to

$$\operatorname{sign}(d^*)(-1)^i(g_{li} - \gamma g_{ji}) - f_{mjl}(-1)^i g_{n+1i} \le 0 \tag{50}$$

$$\operatorname{sign}(d^*)(-1)^i(g_{li} - \gamma g_{ji}) - f_{pjl}(-1)^i g_{n+1i} \ge 0 \tag{51}$$

$$i = 0, 1, \ldots, n+1$$

where the constraints (50) and (51) imply (44) and (45). The objective function ϵ_{jl} is nonlinear, whereas the constraints are linear. This type of optimization problem can be solved by the gradient-projection method [17], which becomes quite simple in this special case because the constraints restrict the domain of feasible points to $f_{mjl} > 0$ and $f_{pjl} < 0$. It follows from (47) that, on this domain, ϵ_{jl} is a monotone function along all three variables. Its maximum is always at the lowest (in an absolute sense) feasible value of either f_{mjl} or f_{pjl}.

The exact position of this point depends on γ and is straightforward to find. Let us assume that γ is known (in addition to a_j and a_l). Depending on the sign of $a_l^* - a_l - \gamma(a_j^* - a_j)$, either f_{mjl} or f_{pjl} gives the maximum in (47). In either case, there exists an index $i = i_0$ that gives f_{mjl} or f_{pjl} in (44) or (45) as

$$f_{mjl} \quad \text{or} \quad f_{pjl} = \operatorname{sign}(d^*) \frac{g_{li_0} - \gamma g_{ji_0}}{g_{n+1i_0}}. \tag{52}$$

Using (52), the objective function (47) can be simplified into

$$\epsilon_{jl} = \operatorname{sign}(d^*) g_{n+1i_0} \frac{a_l^* - a_l - \gamma(a_j^* - a_j)}{g_{li_0} - \gamma g_{ji_0}}. \tag{53}$$

The first derivative of (53) is equal to

$$\frac{d\epsilon_{jl}}{d\gamma} = \operatorname{sign}(d^*) g_{n+1i_0} \frac{(a_l^* - a_l) g_{ji_0} - (a_j^* - a_j) g_{li_0}}{(g_{li_0} - \gamma g_{ji_0})^2}. \tag{54}$$

For nonzero derivative, it is clear that if γ is replaced by $\gamma + \Delta\gamma$ so that

$$\frac{d\epsilon_{jl}}{d\gamma} \Delta\gamma > 0 \tag{55}$$

the objective function ϵ_{jl} will increase. The size of $\Delta\gamma$ must be limited to the point where a change of index i_0 that gives f_{mjl} or f_{pjl} in (44) or (45) occurs. This follows from (54) because the sign of the derivative can change only when i_0 changes, or in other words, $\Delta\gamma$ must not be too large. It is easy to see from (52) that i_0 will change when the following equality is reached for some i:

$$\frac{g_{li_0} - (\gamma + \Delta\gamma) g_{ji_0}}{g_{n+1i_0}} = \frac{g_{li} - (\gamma + \Delta\gamma) g_{ji}}{g_{n+1i}} \tag{56}$$

where the sign of $\Delta\gamma$ must conform to (55). This gives the maximum allowed $|\Delta\gamma|$

$$|\Delta\gamma_{\max}| = \min_{\substack{0 \le i \le n+1 \\ i \ne i_0}} \operatorname{sign}\left(\frac{d\epsilon_{jl}}{d\gamma} \right) \left(\frac{\frac{g_{li_0}}{g_{n+1i_0}} - \frac{g_{li}}{g_{n+1i}}}{\frac{g_{ji_0}}{g_{n+1i_0}} - \frac{g_{ji}}{g_{n+1i}}} - \gamma \right) \tag{57}$$

where indices i that give negative values in the search for minimum must be ignored. Let us denote the index that gives the minimum as $i = i_1$ and use the fact that $\Delta\gamma$ is simply $|\Delta\gamma_{\max}|$ multiplied by the sign of derivative

$$\Delta\gamma = \operatorname{sign}\left(\frac{d\epsilon_{jl}}{d\gamma}\right)\left(\frac{\frac{g_{li_0}}{g_{n+1i_0}} - \frac{g_{li_1}}{g_{n+1i_1}}}{\frac{g_{ji_0}}{g_{n+1i_0}} - \frac{g_{ji_1}}{g_{n+1i_1}}} - \gamma\right). \tag{58}$$

The algorithm for finding γ that maximizes ϵ_{jl} in our optimization problem (49)–(51) can now be described. It consists of the following steps.

1) Start with $\gamma = 0$, and use (44)–(47) to compute ϵ_{jl}. Exchange indices j and l, and repeat the computation of ϵ_{jl}. Keep the original indices j and l if the first ϵ_{jl} is greater or equal to the second one—keep the exchanged indices otherwise. This step is important because it ensures that γ will always be finite.

   The greater of ϵ_{jl} is the starting solution. Its computation also gives the index $i = i_0$, as described by (52) and (53).
2) Compute the derivative (54). Stop if it is zero, else keep its sign.
3) Use (57) and (58) to compute $\Delta\gamma$ and the minimal index i_1.
4) Replace γ with the new value

$$\gamma \leftarrow \gamma + \Delta\gamma \tag{59}$$

   and compute new ϵ_{jl} using (53).
5) Replace index i_0 by the new value

$$i_0 \leftarrow i_1. \tag{60}$$

6) Compute the new derivative (54), and stop if it is zero or if its sign differs from the previous one. Otherwise, return to step 3 for the next iteration.

The algorithm is fast and needs only a small number of iterations before the optimal γ and ϵ_{jl} are found. Note however, that ϵ_{jl} is valid only for a given pair a_j, a_l and is not the lower bound on the increase in approximation error δ in which we are interested.

To get a lower bound on δ, it is necessary to repeat the computation of optimal ϵ_{jl} for all possible combinations of a_j, a_l from I_b. In addition, to get the best lower bound, all possible pairs of indices j and l can be tried. In other words, we must find

$$\delta \geq \max_{\substack{0\leq j\leq n \\ j+1\leq l\leq n}} \left[\min_{\substack{a_j, a_l \\ a_j \in I_b, a_l \in I_b}} \epsilon_{jl} \right]. \tag{61}$$

The search through all pairs j, l is straightforward and must be repeated $n(n+1)/2$ times. This number can be greatly reduced if only the most promising indices are used as j and l. Such a lower bound is obtained much faster and is typically just a little lower.

TABLE I
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	

Searching through all values of a_j and a_l from I_b appears more difficult because the size of I_b can be large, but it is really not necessary to try all values from I_b. The search space becomes much smaller with the help of *Theorem 1* if we note that (26)–(28) can also be used to compute the lower bound for a given a_k. We have

$$\delta \geq \begin{cases} \dfrac{a_k - a_k^*}{f_{mk}}, & \text{for } a_k - a_k^* \geq 0 \\ \dfrac{a_k - a_k^*}{f_{pk}}, & \text{for } a_k - a_k^* < 0. \end{cases} \tag{62}$$

Beginning with $a_j = a_{j+}$ and $a_l = a_{l+}$, for example, a starting optimal ϵ_{jl} is computed using the algorithm. The next higher a_j from I_b need be tried only if δ from (62) is lower than ϵ_{jl}. It is ignored otherwise, and the search in this direction is terminated since increasing a_j obviously cannot lead to a smaller lower bound. The same approach is repeated to search in the negative direction of a_j (starting with a_{j-}) and the positive and negative direction of a_l. The total number of values a_j and a_l that must be tried is quite reasonable and is typically much smaller than the size of I_b.

As mentioned before, these ideas can in principle be extended to three or more of (35). This extension, however, is much more complicated than in the case of two equations. Not only must the search be performed in three or more dimensions, but the algorithm that computes the optimal γ (two or more of them) becomes more complicated. This extension was not pursued after initial attempts.

V. RESULTS

Fifteen filters with five different sets of frequency-domain specifications, which are denoted A through E, were used for testing. The frequency specifications are identical to those used in [3] and are given in Table I. A is a lowpass 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

TABLE II
LOWER BOUNDS ON THE INCREASE IN MINIMAX APPROXIMATION ERROR FOR 15 DESIGN CASES

Filter	$\|d^*\|$	Theorem 1 lower bound	Improved lower bound	Optimal b-bit deviation
A25/8	0.039717	0.000708	0.001249	0.049053
A35/8	0.015946	0.000162	0.000464	0.029838
A45/8	0.007128	0.001008	0.001616	0.029623
B25/9	0.122890	0.002054	0.002336	0.136470
B35/9	0.052719	0.001807	0.003038	0.077095
B45/9	0.021048	0.002582	0.003824	0.056790
C25/8	0.012831	0.001457	0.001853	0.024841
C35/8	0.002629	0.000532	0.000803	0.017871
C45/8	0.000670	0.000213	0.000474	0.016090
D25/9	0.048086	0.001380	0.003144	0.062464
D35/9	0.010433	0.001691	0.001854	0.032528
D45/9	0.002235	0.000860	0.001122	0.026122
E25/8	0.040038	0.001350	0.001532	0.049084
E35/8	0.017606	0.001441	0.001770	0.032991
E45/8	0.006538	0.001192	0.001418	0.028877

in all bands, whereas D has a weighting of 10 in the stopband. E is a lowpass filter whose passband and stopbands do not include $\omega = 0$ or π.

We denote by A25/8 the filter design problem for specification A, length $N = 25$ ($n = 13$ independent coefficients), and $b = 8$ bits (sign included) and similarly for A35/8, B25/9, and so on. Table II shows a summary of the results, comparing the infinite precision deviation d^*, the lower bound on δ given by *Theorem 1*, and the improved lower bound given by (61) (using all pairs j, l). The last column contains the optimal b-bit approximation error. Integer set I_b and a constant scaling factor $s = 2^{b-1}$ were used in all design cases. This scaling was chosen for simplicity as well as to allow easier comparison with the older results.

As expected, the improved lower bound is consistently better than the simple bound given by *Theorem 1*, and it also grows with the filter length N when measured relative to $|d^*|$. It grows, for example, from 14% of $|d^*|$ for C25/8 to 71% for C45/8. This indicates that it can be effective in the algorithm for the optimal finite wordlength design.

The improved lower bound (61) was implemented in a program for optimal finite wordlength FIR filter design. The program is based on the branch-and-bound algorithm, which requires solutions of a large number of subproblems. Each branch-and-bound subproblem is a redefined infinite precision problem of the form (3).

TABLE III
RESULTS OF THE LOWER BOUND EFFECTIVENESS IN THE ALGORITHM FOR OPTIMAL FINITE WORDLENGTH FIR FILTER DESIGN

Filter	Number of subproblems		Computing time	
	With bound	Without bound	With bound	Without bound
A25/8	299	719	0.06	0.11
A35/8	797	1901	0.23	0.38
A45/8	5400	13445	1.64	2.94
B25/9	627	1355	0.12	0.21
B35/9	2855	7163	0.84	1.44
B45/9	7192	17447	2.55	4.74
C25/8	341	971	0.06	0.13
C35/8	2332	5960	0.47	0.88
C45/8	37036	112694	8.09	18.08
D25/9	514	1220	0.09	0.17
D35/9	14033	35387	2.84	5.86
D45/9	133802	333836	40.72	87.00
E25/8	385	896	0.06	0.13
E35/8	1534	3887	0.39	0.71
E45/8	5743	14873	1.75	3.23

Let us examine a subproblem i, which can be written as

$$\min_{P^{(i)}(\omega)} \max_{a \le \omega \le b} \left| W(\omega)\left(D^{(i)}(\omega) - P^{(i)}(\omega)\right)\right| \tag{63}$$

where $D^{(i)}(\omega)$ is defined as

$$D^{(i)}(\omega) = D(\omega) - \sum_{k=r+1}^{n} a_k^{(i)} \cos k\omega. \tag{64}$$

Coefficients $a_k^{(i)}, k = r+1, \ldots, n$ are already in I_b. Solution of (63) is a cosine polynomial of degree $r, r < n$

$$P^{(i)*}(\omega) = \sum_{k=0}^{r} a_k^{(i)*} \cos k\omega \tag{65}$$

with the optimal infinite precision coefficients $a_k^{(i)*}$ and the optimal approximation error $|d^{(i)*}|$. The lower bound on the increase in approximation error is computed as described above and added to $|d^{(i)*}|$. If the sum exceeds the current best b-bit solution, this subproblem obviously cannot lead to a better b-bit solution and can be removed from the list of subproblems. This in turn means that all subproblems emanating from this one need not be solved.

The price for this reduction is the time needed to compute the lower bound. This time must be small compared to the time that is needed to solve (63), or most of the gain will be lost. Experiments have shown that using all pairs j, l in (61) takes too much time. This time is greatly reduced when only pairs from the four indices out of $r + 1$ are chosen for each subproblem. The following four indices are used.

1) Index $k = r$ is selected because the variable a_r will be constrained to I_b next. This creates two new subproblems

(with $a_r = a_{r-}$ and $a_r = a_{r+}$), and a different lower bound can be computed for each one.

2) Indices k that give the lowest (in an absolute sense) f_{mk} and f_{pk} in (27) and (28) are the obvious choice since they promise the highest lower bound.

3) Index k that gives the maximum lower bound in (26) must be included to ensure that the improved lower bound is better than the one from *Theorem 1*.

The maximum number of pairs is $4 \cdot 3/2 = 6$ and is often lower because the same index can appear in two (or more) of the above cases. The corresponding lower bound was found to be on average only between 5% and 10% lower than the bound computed by using all pairs. The computing time, however, is reduced much more.

This version of the improved lower bound was implemented in a program. Table III shows a summary of the results, comparing the number of branch-and-bound subproblems that must be solved when the lower bound is used and when it is not. The corresponding computing times in seconds on a 2.4 GHz Pentium 4 are also given. The results show that the number of subproblems was reduced by a factor that is about 2.5 on average. The total computing time reduction factor was 2.1. The simple lower bound given by *Theorem 1* was also tested giving the corresponding reduction factors 2.0 and 1.6.

A similar reduction in computing time can be expected for other discrete sets I_b as well as for the case of parallel machine implementation that was reported in [18].

VI. Conclusion

A new method that allows computation of a lower bound for the increase in minimax approximation error that is caused by the finite wordlength constraint was presented in this paper. This bound gives a theoretical limit on the performance of a given b-bit FIR filter of length N and can also be used to significantly reduce the amount of computation in the algorithm for optimal finite wordlength FIR filter design. Design examples have confirmed its effectiveness.

References

[1] D. M. Kodek, "Design of optimal finite word-length FIR digital filters using integer programming techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, pp. 304–308, Jun. 1980.

[2] Y. C. Lim, S. R. Parker, and A. G. Constantinides, "Finite word length FIR filter design using integer programming over a discrete coefficient space," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-30, pp. 661–664, Aug. 1982.

[3] D. M. Kodek and K. Steiglitz, "Comparison of optimal and local search methods for designing finite wordlength FIR digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-28, no. 1, pp. 28–32, Jan. 1982.

[4] Y. C. Lim and S. R. Parker, "FIR filter design over a discrete power-of-two coefficient space," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, pp. 583–591, Jun. 1983.

[5] C. L. Chen and A. N. Wilson, "A Trellis search algorithm for the design of FIR filters with signed power-of-two coefficients," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 1, pp. 29–39, Jan. 1999.

[6] Y. C. Lim, R. Yang, D. Li, and J. Song, "Signed power-of-two term allocation scheme for the design of digital filters," *IEEE Trans. Circuits Syst. II*, vol. 46, no. 5, pp. 577–584, May 1999.

[7] D. M. Kodek and K. Steiglitz, "A theoretical performance bound on the performance of direct-form finite wordlength FIR digital filters," in *Proc. 14th Annu. Conf. Inf. Sci. Syst.*, Princeton, NJ, Mar. 26–28, 1980, pp. 369–371.

[8] ——, "Filter-length word-length tradeoffs in FIR digital filter design," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, pp. 739–744, Dec. 1980.

[9] W. P. Niedringhaus, K. Steiglitz, and D. M. Kodek, "An easily computed performance bound for finite wordlength direct-form FIR digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-29, pp. 191–193, Mar. 1982.

[10] T. W. Parks and J. H. McClellan, "A program for the design of linear phase finite impulse response filters," *IEEE Trans. Audio Electroacoust.*, vol. AE-20, pp. 195–199, Aug. 1972.

[11] Y. C. Lim, "Design of discrete-coefficient-value linear phase FIR filters with optimum normalized peak ripple magnitude," *IEEE Trans. Circuits Syst.*, vol. 37, no. 12, pp. 1480–1486, Dec. 1990.

[12] D. M. Kodek, "Design of optimal finite wordlength FIR digital filters," in *Proc. Eur. Conf. Circuit Theory Design ECCTD*, vol. I, Stresa, Italy, Aug. 29–31, 1999, pp. 401–404.

[13] P. J. Davis, *Interpolation and Approximation*. New York: Dover, 1975, pp. 149–151.

[14] D. M. Kodek, "A lower bound for the increase of finite wordlength minimax approximation error," in *Proc. Elect. Comput. Sci. Conf.*, vol. B, Portorož, Slovenia, Sep. 28–30, 1992, pp. 3–6.

[15] ——, "Limits of finite wordlength FIR digital filter design," in *Proc. ICASSP Int. Conf. Acoust., Speech, Signal Process.*, vol. III, Munich, Germany, Apr. 21–24, 1997, pp. 2149–2152.

[16] M. J. D. Powell, *Approximation Theory and Methods*. Cambridge, U.K.: Cambridge Univ. Press, 1981, pp. 97–99.

[17] T. L. Saaty and J. Bram, *Nonlinear Mathematics*. New York: McGraw-Hill, 1964, pp. 133–137.

[18] Y. C. Lim, Y. Sun, and Y. J. Yu, "Design of discrete coefficient FIR filters on loosely connected parallel machines," *IEEE Trans. Signal Process.*, vol. 50, no. 6, pp. 1409–1416, Jun. 2002.

Dušan M. Kodek (M'79–SM'83) was born in Ljubljana, Slovenia, in 1946. He received the B.E.E. degree in 1970, the M.E.E. degree in 1973, and the Ph.D. degree in 1975, all from the University of Ljubljana.

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 and systems areas. From 1978 to 1979, he was a Fulbright visiting professor with the Department of Electrical Engineering and Computer Science, Princeton University, Princeton, NJ. He 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.

Dr. Kodek is a regular reviewer for several journals and has been on the Program Committee for several Workshops and Conferences. He is a recipient of three Slovenian awards for research and innovation.