Design of optimal finite wordlength FIR digital filters

Dušan M. KODEK*

Abstract

Design of optimal finite wordlength FIR digital filters requires a solution of an NP-complete approximation problem. This leads to computation times that can be many orders of magnitude slower than the infinite precision case. This, however, is not the only difficulty. The problem itself is not well defined – the solution depends on how the original approximation problem is redefined or scaled. Several scaling techniques can be used, each leading to its own optimal solution. The scaling possibilities and an algorithm for the optimal finite wordlength design are given in the paper.

1 Introduction

There are many situations when it is not practical to use "infinite precision" filter coefficients (typically 32-bit floating point numbers). One may, for example, wish to use a fixed point DSP processor because it is cheaper and faster than a floating point one. Another possibility is a multiplierless filter implementation. Rounding the coefficients to the nearest finite wordlength value is often used. It, however, produces a suboptimal filter which can be up to 40 dB worse than the optimal finite wordlength filter. Herein lies the problem. Design of optimal finite wordlength FIR digital filters is a difficult problem which can be many orders of magnitude slower than the infinite precision case. The question of its practicality has been open for a long time. It is the purpose of this paper to show that current algorithms are indeed practical and that they produce results in a very reasonable time in most design cases.

2 The finite wordlength design problem

Let us explain these ideas more precisely. The optimal infinite precision (i.e., filter coefficients $h^*_\infty(k)$ can be any real number) linear phase FIR digital filter can be defined as the cosine polynomial $P^*_\infty(\omega)$

$$P^*_\infty(\omega) = e^{j\omega(N-1)/2} \sum_{k=0}^{N-1} h^*_\infty(k) e^{-j\omega k}, \qquad (1)$$

which can also be written as

$$P^*_\infty(\omega) = Q(\omega) \sum_{k=0}^{n} a^*_{\infty k} \cos k\omega. \qquad (2)$$

$P^*_\infty(\omega)$ is a solution of the following approximation problem

$$\min_{P_\infty(\omega)} \left[\max_{a \le \omega \le b} |W(\omega)(D(\omega) - P_\infty(\omega))| \right]. \qquad (3)$$

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 $[0, \pi]$. The polynomial order n is related to the filter length N. Depending on N (odd or even) and filter symmetry (positive or negative) there are exactly 4 design cases and 4 different functions $Q(\omega)$.

The function $Q(\omega)$ is irrelevant from the point of view of the approximation problem. We will therefore assume $Q(\omega) = 1$. The problem of solving (3) has been considered an easy one ever since Parks and McClellan showed how to apply the Remez algorithm. The main reason for this is a well-known property of the optimal minimax approximation: there are exactly $n + 2$ so called extremal points in $[a, b]$ at which the approximation error achieves its maximum. Let $\omega_i, i = 0, 1, \cdots, n + 1$, be these extremal points. The following equations hold

$$W(\omega_i)(D(\omega_i) - \sum_{k=0}^{n} a^*_{\infty k} \cos k\omega_i) = (-1)^i e^*_\infty, \quad (4)$$

where $a^*_{\infty k}$ are the optimal infinite precision coefficients and e^*_∞ the approximation error.

The problem's complexity changes dramatically when the finite wordlength constraint is introduced. Constrained minimax approximation is an NP-complete problem 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. Note that the integers are chosen for convenience – any other finite set of numbers can be used instead. Let us denote this set as I_b and assume, again for convenience only, that it contains b-bit signed integers $\{-2^{b-1}, \cdots, -1, 0, 1, \cdots, 2^{b-1}\}$.

Constraining the coefficients $h(k)$ to the set I_b requires a redefinition or scaling of the original infinite precision approximation problem. The scaling is necessary to bring the coefficients within the

*Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25, 1000 Ljubljana, Slovenia, kodek@fri.uni-lj.si

range of numbers in I_b. It is usually done with the help of a scaling factor S; the original approximation problem (3) can be rewritten as

$$\begin{aligned}&\min_{P(\omega)}\left[\max_{a\le\omega\le b}|\frac{W(\omega)}{S}(SD(\omega)-SP(\omega))|\right]\\&=\min_{P'(\omega)}\left[\max_{a\le\omega\le b}|W'(\omega)(D'(\omega)-P'(\omega))|\right],\end{aligned}\quad(5)$$

where

$$P'(\omega)=SP(\omega)=\sum_{k=0}^{n}a_k\cos k\omega,\quad a_k\in I_b,\quad(6)$$

is the finite wordlength filter and $W'(\omega)$ and $D'(\omega)$ describe the new approximation problem. Finding the scaling factor S is not trivial and is worth investigating in greater detail. Before going into that note 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. The scaling factor S must be modified to make (6) correct. This follows from the formulas that relate the filter coefficients $h(k)$ to cosine polynomial coefficients a_k. For case 1 FIR filter (odd N, positive symmetry) the formula is

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

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 $k \geq 1$. A modification of the approximation problem is necessary if both $h(k)$ and a_k are to be from I_b. 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

$$\begin{aligned}h(n)&=2a_0, &n&=(N-1)/2,\\h(n-k)&=a_k, &k&=1,2,\ldots,n.\end{aligned}\quad(8)$$

The coefficient a_0 is a special case. Its values are constrained to the elements of I_b divided by 2 and must be used accordingly. This is a minor complication. Similar considerations apply to the case 2, 3, and 4 FIR filters; the difference is that S must be divided by 4 and not by 2. The net effect of dividing S by 2 or 4 is a unification of all 4 cases from the point of view of the approximation problem. It should be noted however, that this unification is more complicated for other (non-integer) sets I_b.

3 Scaling factor S

It follows from (5) that the scaling factor S changes the approximation problem and will therefore affect the approximation error[1]. The optimal scaling factor S_{opt} is defined as the one that gives the lowest approximation error. Filter with the lowest approximation error fits the desired frequency response best and is often a designer's choice. There are other considerations, however. Finding S_{opt} is not easy and may require much longer computation time. Note also that the scaling factor changes the gain of the finite wordlength filter. A division by S is necessary if a unity passband gain is desired. Selecting an S that is a power of 2 may be preferable if a division is to be avoided. Let us examine the types of scaling factor S that are used in practice:

1. ***S* is a power of 2**. This choice allows a simple, divisionless correction of the gain factor.

2. ***S* is a positive integer**. The advantage of this type of scaling is that a better match between finite wordlength coefficients a_k and the desired frequency response is possible. This will in most cases reduce the approximation error. Gain factor correction (if needed) is somewhat more complicated now.

   An initial guess for both power of 2 and integer scaling factor S can be obtained from the rounded b-bit coefficients. The maximum rounded coefficient $|h_r(k)|_{max}$ is found and S is computed as the largest multiplier (power of 2 or integer) that keeps $S|h_r(k)|_{max}$ within I_b. The redefined approximation problem (5) is solved and the optimal coefficients $h(k)$ are obtained. The optimal $h(k)$ are different from the rounded ones and one or more of $h(k)$ may fall out of I_b. If this occurs, the initial S must be reduced, $W'(\omega)$, and $D'(\omega)$ redefined and the computation repeated. This happens rarely with power of 2 scaling, but is not uncommon with the integer scaling. If a repetition is needed, one is usually sufficient.

3. **Optimal scaling factor**. The scaling described above is simple and is done before the approximation problem is solved. It is, however, not optimal and a scaling factor that is significantly better exists in most cases. A simple improvement of a power of 2 or integer S can be obtained, if the following approximation problem is solved for f

   $$\min_{f}\left[\max_{a\le\omega\le b}|W'(\omega)(D'(\omega)-fP'(\omega))|\right].\quad(9)$$

   This is a zero order minimax approximation problem and easy to solve. A new real-valued

[1] Note that this is not the case in the infinite precision design where the approximation error e^*_∞ is independent of S.

scaling factor $S_r = S/f$ is better than S. Finding the optimal factor S_{opt} is much more difficult, though. S must be included in the approximation problem as a variable. The problem of finding optimal S and optimal finite wordlength polynomial $P'(\omega)$) is now nonlinear. Solving it requires iterations of the original problem and is very slow. A good description of the difficulties and ways of improving the speed is given in [4].

Nine finite wordlength linear phase FIR filters with five different sets of frequency-domain specifications, denoted A through E, were used to demonstrate the influence of scaling on the approximation error. The frequency specifications are identical to those that were used in [5]. Notation A25/5 denotes the filter design problem for specification A, length 25 (13 independent coefficients), and $b = 5$ bits (sign included); similarly for A35/7, B25/7, and so on. Table 1 shows the effect of power of 2, integer, and optimal scaling factor S on optimal finite wordlength approximation error.

Table 1: Optimal finite wordlength approximation error for different scaling factors.

Filter	Scaling and approximation error		
	Power of 2	Integer	Optimal
A25/5	$S = 32$ 0.1012263	$S = 33$ 0.0981588	$S = 31.246$ 0.0905645
A35/7	$S = 128$ 0.0298382	$S = 140$ 0.0285714	$S = 134.806$ 0.0271478
B15/7	$S = 128$ 0.3068644	$S = 142$ 0.2968761	$S = 130.019$ 0.2877300
B25/7	$S = 128$ 0.1542276	$S = 142$ 0.1479446	$S = 141.496$ 0.1449093
B35/7	$S = 128$ 0.1171875	$S = 143$ 0.0919863	$S = 142.310$ 0.0875823
C25/5	$S = 16$ 0.1263982	$S = 26$ 0.0918509	$S = 24.345$ 0.0809789
C35/7	$S = 64$ 0.0375755	$S = 111$ 0.0218240	$S = 117.000$ 0.0170942
D25/7	$S = 64$ 0.1306068	$S = 118$ 0.0893562	$S = 114.469$ 0.0832647
E25/6	$S = 16$ 0.0879222	$S = 20$ 0.0793962	$S = 15.222$ 0.0778638

4 The finite wordlength algorithm

An algorithm that finds the optimal finite wordlength filter $P'(\omega)$ in the approximation problem (5) is required. A straightforward approach is to use one of the standard integer or mixed-integer programming packages that are available commercially. They are very general and are usually based on the linear programming algorithm. They are also very slow and not too suitable for practical filter design. A much better algorithm was developed that is tailored specifically to the minimax approximation problem (5).

The algorithm is based on the well-known branch-and-bound technique. Branch-and-bound offers several advantages. It is simple, various sets I_b can be used easily, and is also efficient as the early comparisons with other techniques have shown [2]. Remez algorithm, which is significantly faster than linear programming, can be used to solve the subproblems. It also allows a simple implementation of new theoretical results on the lower bound of the finite wordlength minimax approximation that were developed for the purpose of algorithm improvement.

Without going into details of branch-and-bound, which are available in standard textbooks like [1], we will first describe how the lower bound can be used to improve the speed. At each node in the branch-and-bound algorithm there is a subproblem for which the coefficients $a_{s+1}, a_{s+2}, \ldots, a_n$ are already in I_b while $a_0, a_1, \ldots, a_s$ are still infinite precision. Let e_s^* be the optimal minimax approximation error of this particular subproblem. The approximation error for all subproblems that branch from this one is $e_s^* + \epsilon$ where $\epsilon > 0$. The so called *level 1 lower bound* presented in [3] gives the lower bound for ϵ as

$$\epsilon \geq \min \left(\max_{0 \leq k \leq s} \frac{|a_k - a_{k+}|}{|f_{mk}|}, \max_{0 \leq k \leq s} \frac{|a_k - a_{k-}|}{|f_{pk}|} \right), \tag{10}$$

where a_{k+} and a_{k-} are the nearest upper and lower I_b neigbours of a_k. Factors f_{mk} and f_{pk} are easily computed as a byproduct of the subproblem solution.

It is extremely useful to know a lower bound for ϵ in the branch-and-bound. The particular node and all the subproblems that branch from it can be eliminated from the branch-and-bound tree if $e_s^* + \epsilon$ exceeds the current optimal approximation error. It is clear that they cannot lead to a solution with a lower approximation error and can therefore be eliminated without having to solve them first.

The lower bound (10) can be further improved to *level 2 lower bound* by combining two indices k. Its derivation is given in [6] and this improved bound was implemented in the algorithm. An additional improvement is possible if we consider the number of points in the frequency set Ω that replaces the interval $a \leq \omega \leq b$ in the Remez algorithm. In other words, the problem (5) is replaced by

$$\min_{P'(\omega)} \left[\max_{\omega \in \Omega} |W'(\omega)(D'(\omega) - P'(\omega))| \right]. \tag{11}$$

Ω is a so called dense set of frequency points. The number of points equal to $16n$ usually suffices, although the designer is free to choose any number. Increasing the number points slows the Remez algorithm, but produces a more accurate solution. The idea here is that solutions of the subproblems in branch-and-bound do not have to be as accurate as the final solution (which is always checked on the dense grid). Using a *rare set of points* Ω_r that is smaller than Ω can speed up the computation.

However, Ω_r must not be too small. A rare set of points results in an underestimated subproblem approximation error e_s^*. This in turn means that fewer nodes are eliminated from the branch-and-bound tree and that the number of subproblems which must be solved increases. The rare set Ω_r must be such that this increase is minimal.

A measure of the approximation error underestimation that is caused by the rare set is needed here. The infinite precision optimal minimax error function $e(\omega)$ can be written approximately as

$$e(\omega - \omega_i) = e_\infty^* \cos \frac{\pi(\omega - \omega_i)}{\omega_{i+1} - \omega_i}, \ \omega_i \le \omega \le \omega_{i+1}, \tag{12}$$

where ω_i are the extremal frequencies from (4). Equation (12) is almost exact for all ω that do not fall in transition bands. Let us assume that frequency points in Ω_r in the range $\omega_i \le \omega \le \omega_{i+1}$ are equidistant and that the distance between them is $\Delta\omega_r$. Then the approximation error computed on points from Ω_r is underestimated by Δe that is bounded by

$$\begin{aligned} \Delta e &\le e_\infty^* - e\left(\frac{\Delta\omega_r}{2}\right) \\ &\le e_\infty^* \left(1 - \cos\frac{\pi\Delta\omega_r}{2(\omega_{i+1} - \omega_i)}\right). \end{aligned} \tag{13}$$

This gives the following upper bound on the density of points in the rare set Ω_r

$$\Delta\omega_r \le \frac{2}{\pi}(\omega_{i+1} - \omega_i) \arccos\left(1 - \frac{\Delta e}{e_\infty^*}\right). \tag{14}$$

If Δe is known, the density of points in Ω_r can be computed. Δe should be selected as a fraction of the optimal finite wordlenght approximation error e^*. Now e^* is only known after the problem is solved. A rough estimate for e^* can be computed using the rounded coefficients. A better estimate can be obtained if a short local search is performed before the branch-and-bound is started. This is used in our algorithm and Δe is selected as 1% of the estimated e^*. Since $\Delta\omega_r$ in (14) depends on $\omega_{i+1} - \omega_i$ this is taken into account in the Ω_r construction. In other words, different $\Delta\omega_r$ is used along the ω axis.

Note also that for a particular subproblem the number of extremal points ω_i equals $s + 2$ and is therefore a variable. The difference $\omega_{i+1} - \omega_i$ increases for $s < n$, but so does e_s^*. Both effects compensate each other and $\Delta\omega_r$ in (14) is roughly independent of s.

The effectiveness of both rare set Ω_r and lower bound ϵ is demonstrated in table 2 on the same set of nine filters as in the table 1. Power of 2 scaling was used. The computer times are for a 450 MHz Pentium II.

Table 2: The effectivness of rare set and lower bound in the finite wordlength design algorithm.

Filter	Computer times in seconds			
	Ω_r no, ϵ no	Ω_r no, ϵ yes	Ω_r yes, ϵ no	Ω_r yes, and ϵ yes
A25/5	1.45	0.74	0.60	0.39
A35/7	11.10	6.01	5.12	3.04
B15/7	0.54	0.25	0.32	0.16
B25/7	2.26	1.27	1.04	0.74
B35/7	40.01	20.35	13.85	7.77
C25/5	1.09	0.63	0.34	0.26
C35/7	51.25	24.39	15.09	8.87
D25/7	7.65	3.90	2.25	1.43
E25/6	2.79	1.54	0.99	0.68

References

[1] C.H.Papadimitrou and K.Steiglitz, *Combinatorial optimization*. Prentice-Hall, Englewood Cliffs, N.J., pp.433-453, 1982.

[2] D.M.Kodek, *"Design of optimal finite wordlength FIR digital filters using integer programming techniques,"* IEEE Trans. on Acoustics, Speech and Signal Processing, vol.ASSP-28, pp.304-308, June 1980.

[3] D.M.Kodek, *"Limits of finite wordlength FIR digital filter design,"* Proceedings of ICASSP'97, vol III., pp.2149-2152, Apr. 1997.

[4] Y.C.Lim, *"Design of discrete-coefficient-value linear phase FIR filters with optimum normalized peak ripple magnitude"*, IEEE Trans. Circuit Syst., vol.CAS-37, pp.1480-1486, Dec. 1990.

[5] D.M.Kodek, K.Steiglitz, *"Comparison of optimal and local search methods for designing finite wordlength FIR digital filters,"* IEEE Trans. Circuit Syst., vol.CAS-28, pp.28-32, Jan. 1981.

[6] D.M.Kodek, *"A theoretical limit for finite wordlength FIR digital filters,"* Proceedings of CISS'98, vol.2, pp.836-841, Princeton, 1998.