Finite Wordlength Linear-Phase FIR Filter Design Using Babai's Algorithm

Andraž Božiček *andraz.bozicek@fri.uni-lj.si*
Faculty of Computer and Information Science
University of Ljubljana
Ljubljana, 1000, Slovenia

Abstract

Optimal finite linear-phase impulse response (FIR) filters are most often designed using the Remez algorithm, which computes so-called infinite precision filter coefficients. In many practical applications, it is necessary to represent these coefficients by a finite number of bits. The problem of finite wordlength linear-phase filters is not as trivial as it would seem. The simple rounding of coefficients computed by the Remez algorithm gives us a suboptimal filter. Optimal finite wordlength linear-phase FIR filters are usually designed using integer linear programming, which takes a lot of time to compute the coefficients. In this paper, we introduce a new approach to the design of finite wordlength FIR filters using very fast Babai's algorithm. Babai's algorithm solves the closest vector problem, and it uses the basis reduced by the LLL algorithm as an input. We have used algorithms which solve the problem in the L_2 norm and then added heuristics that improve the results relative to the L_∞ norm. The design method with Babai's algorithm and heuristics has been tested on filters with different sets of frequency-domain specifications.

Keywords: FIR filter design, finite wordlength coefficients, Babai's algorithm, LLL algorithm, closest vector problem.

1. INTRODUCTION

The design of optimal finite wordlength finite impulse response (FIR) filters can be formulated as the Chebyshev approximation problem [1]. It is viewed as a criterion that the weighted approximation error between the desired and the actual frequency response is spread evenly accross the passband and stopband of the filter. This criterion minimizes the maximum absolute error. There exist very good approximation algorithms (including the well-known Remez algorithm), which give the optimal polynomial coefficients in the L_∞ norm. Standard approximation algorithms yield unbounded or so-called "infinite precision" coefficients. In many practical situations, we want to use cheaper and faster fixed-point digital signal processors (DSPs).

There are many approaches which yield the finite wordlength linear-phase coefficients, but not all are optimal. The most simple approach is to round coefficients calculated by the Remez algorithm to a desired length. However, this results in poor frequency response of the filter and suboptimal coefficients [2]. Kodek [2] proposes the use of the mixed integer linear programming (MILP) technique in the design of finite wordlength linear-phase FIR filters to give optimal coefficients. The slowness is the only disadvantage of this technique. An approach which significantly speeds up the calculation of coefficients is represented in [3]. By knowing the lower bound of approximation error, the number of subproblems can be reduced. This also reduces the amount of calculation. Derivation of an improved lower bound that uses the LLL algorithm is given in [4].

As was mentioned earlier, we can formulate the problem of optimal finite wordlength linear-phase FIR filter design as a polynomial approximation. The polynomial approximation has been solved with approaches based on the lattice theory algorithms, i.e. the LLL algorithm and Babai's nearest plane algorithm [5]. The LLL algorithm is a polynomial-time lattice reduction algorithm, named after its three authors. The formal description of the algorithm is in [6], and the implementation of the algorithm, including the pseudo-code, can be found in [7]. The aim of the LLL algorithm is to reduce the lattice basis, so the new lattice is equally described with shorter and almost orthogonal vectors. The LLL algorithm has been successfully used in many areas, including practical applications such as cryptography, GPS navigation, and wireless communications. Babai's nearest plane algorithm solves the closest vector problem [8] and it uses the LLL reduced basis as the input.

2. OPTIMAL FINITE WORDLENGTH LINEAR-PHASE FIR FILTERS

The frequency response 1 of an optimal infinite precision linear-phase FIR digital filter of length N is equal to

$$H_r(\omega) = Q(\omega)P(\omega), \quad (1)$$
$$P(\omega) = \sum_{k=0}^{L} \alpha(k)\cos\omega k, \quad (2)$$

where $Q(\omega)$ is a real function and $\alpha(k)$ are the coefficients of the filter depending on filter length (odd or even) and filter symmetry (positive or negative). There are exactly four types of linear-phase FIR filters. The upper limit L in the sum is $L = \frac{M-1}{2}$ for type 1 filters, $L = \frac{M-3}{2}$ for type 3 filters, and $L = \frac{M}{2} - 1$ for type 2 and type 4 filters. M is defined as the filter length. We also define the desired frequency response $H_{dr}(\omega)$ (which is defined to be unity in the passband and stopband of the filter) and a weighting function $W(\omega)$ (which allows us to choose the relative size of the approximation error in the passband and stopband of the filter). For mathematical convenience, a modified weighting function $\widehat{W}(\omega)$ and a modified desired frequency response are defined as

$$\widehat{W}(\omega) = W(\omega)Q(\omega) \quad (3)$$
$$\widehat{H}_{dr}(\omega) = \frac{H_{dr}(\omega)}{Q(\omega)} \quad (4)$$

The weighted approximation error is defined as

$$E(\omega) = W(\omega)[H_{dr}(\omega) - H_r(\omega)] = \widehat{W}(\omega)\left[\widehat{H}_{dr}(\omega) - \sum_{k=0}^{L} \alpha(k)\cos\omega k\right] \quad (5)$$

To determine the filter cofficients $\alpha(k)$, the following minimax approximation problem has to be solved

$$\min_{P(\omega)}\left[\max_{\omega\epsilon S}\left|\widehat{W}(\omega)\left[\widehat{H}_{dr}(\omega) - \sum_{k=0}^{L} \alpha(k)\cos\omega k\right]\right|\right] \quad (6)$$

The set S consists of the passbands and stopbands of the desired filter, e.g. $S = \Omega_p \cup \Omega_s$.
To make the finite wordlength constraint, the filter coefficients $\alpha(k)$ are b-bit integers from the set I_b, where

$$I_b = \{-2^{b-1}, \ldots, -1, 0, 1, \ldots, 2^{b-1}\}. \quad (7)$$

The most significant bit of the coefficient represents the sign and the other $b-1$ bits represent the magnitude.

3. POLYNOMIAL APPROXIMATION AND FINITE WORDLENGTH LINEAR FIR FILTER DESIGN USING BABAI'S ALGORITHM AND HEURISTICS

To represent an arbitrary non-trivial mathematical function on a computer, one usually uses its approximation. The approximation problem can be defined as a search for a function $g(x)$, which belongs to a given class of functions and is as close as possible to a function $f(x)$. Because there

exist efficient schemes of polynomial evaluation, the approximation function $g(x)$ usually belongs to a class of polynomials.
The typical approximation problem is to search the polynomial $g(x)$ degree $\leq n$ which is sufficiently close to $f(x)$. The distance between functions is defined using the norm. The quality of the approximation is measured with the norm of the remainder $\|f - g\|$. Different norms have different approximation functions. The most usual are the L_2 and L_∞ norms. If a continuous function $f(x)$ on an interval $[a, b]$ is assumed, then the approximation problem can be defined as

the L_2 norm searching the $g(x)$, which minimizes

$$\|f(x) - g(x)\|_2 = \sqrt{\int_a^b |f(x) - g(x)|^2 dx} \quad (8)$$

the L_∞ norm searching the $g(x)$, which minimizes

$$\|f(x) - g(x)\|_\infty = \max_{a \leq x \leq b} |f(x) - g(x)| \quad (9)$$

The approximation polynomial can be a trigonometric polynomial. Finite wordlength FIR filter design can be formulated as the problem of approximation by sums of cosines.

According to (5) and (6) the aim of the polynomial approximation is that the desired frequency response $\hat{H}_{dr}(\omega)$ and polynomial $P(\omega)$ are as close as possible. If $P(\omega)$ is represented as the vector $\vec{v}$ and $\hat{H}_{dr}(\omega)$ as the vector $\vec{y}$

$$\underbrace{\begin{pmatrix} \frac{\alpha_0}{2^{b-1}} + \frac{\alpha_1 \cos \omega_1}{2^{b-1}} + \cdots + \frac{\alpha_{L-1} \cos(L-1)\omega_1}{2^{b-1}} + \frac{\alpha_L \cos L\omega_1}{2^{b-1}} \\ \frac{\alpha_0}{2^{b-1}} + \frac{\alpha_1 \cos \omega_2}{2^{b-1}} + \cdots + \frac{\alpha_{L-1} \cos(L-1)\omega_2}{2^{b-1}} + \frac{\alpha_L \cos L\omega_2}{2^{b-1}} \\ \vdots \\ \frac{\alpha_0}{2^{b-1}} + \frac{\alpha_1 \cos \omega_l}{2^{b-1}} + \cdots + \frac{\alpha_{L-1} \cos(L-1)\omega_l}{2^{b-1}} + \frac{\alpha_L \cos L\omega_l}{2^{b-1}} \end{pmatrix}}_{\vec{v}} \text{ and } \underbrace{\begin{pmatrix} \widehat{H_{dr}}(\omega_1) \\ \widehat{H_{dr}}(\omega_2) \\ \vdots \\ \widehat{H_{dr}}(\omega_l) \end{pmatrix}}_{\vec{y}}, \quad (10)$$

then we wish that vectors $\vec{v}$ and $\vec{y}$ are as close as possible according to the L_∞ norm. Frequencies $\omega_1, \omega_2, \ldots, \omega_l$ represents equally discretized frequencies in the interval $[0, \pi]$. The transition band is, unlike to the Remez design method, observed in the interval. If $P(\omega)$ is rewritten as vectors

$$\alpha_0 \underbrace{\begin{pmatrix} \frac{1}{2^{b-1}} \\ \frac{1}{2^{b-1}} \\ \vdots \\ \frac{1}{2^{b-1}} \end{pmatrix}}_{\vec{v}_0} + \alpha_1 \underbrace{\begin{pmatrix} \frac{\cos \omega_1}{2^{b-1}} \\ \frac{\cos \omega_2}{2^{b-1}} \\ \vdots \\ \frac{\cos \omega_l}{2^{b-1}} \end{pmatrix}}_{\vec{v}_1} + \cdots + \alpha_{L-1} \underbrace{\begin{pmatrix} \frac{\cos(L-1)\omega_1}{2^{b-1}} \\ \frac{\cos(L-1)\omega_2}{2^{b-1}} \\ \vdots \\ \frac{\cos(L-1)\omega_l}{2^{b-1}} \end{pmatrix}}_{\vec{v}_{L-1}} + \alpha_L \underbrace{\begin{pmatrix} \frac{\cos L\omega_1}{2^{b-1}} \\ \frac{\cos L\omega_2}{2^{b-1}} \\ \vdots \\ \frac{\cos L\omega_l}{2^{b-1}} \end{pmatrix}}_{\vec{v}_L}, \quad (11)$$

then integer coefficients $\alpha_0, \ldots, \alpha_L$ that minimize

$$\|\vec{v} - \vec{y}\|_\infty = \|\alpha_0 \vec{v}_0 + \alpha_1 \vec{v}_1 + \cdots + \alpha_L \vec{v}_L - \vec{y}\|_\infty \quad (12)$$

have to be found.

Problem (11) can be formulated as the closest vector problem in the L_∞ norm. Kannan's algorithm solves this problem in L_∞, but its complexity is super-exponential [5]. In practice it is better to use Babai's algorithm, which solves the problem in the L_2 norm, and then use some heuristics to improve the results relative to the L_∞ norm.

Babai's algorithm returns the vector $\vec{v}$ as the result. In [5], heuristics similar to that used in this paper are described. The heuristics assume that the result in the L_∞ norm is close to the result in the L_2 norm. The heuristics search the neighborhood of the vector $\vec{v}$. Vectors which represent polynomial $P(\omega)$ (and are LLL-reduced) are added and subtracted from the vector $\vec{v}$. In this manner, we get candidates for a result which is, according to the L_∞ norm, better than the previous result. The L_∞ norm is calculated between the candidates in the desired frequency response, and if the L_∞ norm is lower than it was for the previous result than we keep the new result. This procedure is repeated until the result is improved. The heuristics are described with the pseudocode presented in Table 1.

Description: Explore the neighborhood of the vector $\vec{v}$ to get close to the desired frequency response (vector $\vec{y}$) according to the L_∞ norm ***Input:*** vector $\vec{v}$, vector $\vec{y}$, LLL-reduced basis B' The vector $\vec{v}$ represents the frequency response of the filter and is according to the L_2 norm as close as possible to the desired frequency response of the filter. The vector $\vec{y}$ represents the desired frequency response. The LLL-reduced basis B' includes vectors $\vec{v}_0', \vec{v}_1', \ldots, \vec{v}_L'$. ***Output:*** vector $\vec{v}$. Set Z is filled with $2L+2$ vectors: $\vec{z}_0 = \vec{v} + \vec{v}_0', \vec{z}_1 = \vec{v} - \vec{v}_0', \vec{z}_2 = \vec{v} + \vec{v}_1', \vec{z}_3 = \vec{v} - \vec{v}_1', \ldots, \vec{z}_{2L} = \vec{v} + \vec{v}_L', \vec{z}_{2L+1} = \vec{v} - \vec{v}_L'$. Reference norm is calculated $ref = \lVert \vec{v} - \vec{y} \rVert_\infty$. For $i = 0$ to $2L+1$ calculate $\delta_i = \lVert \vec{z}_i - \vec{y} \rVert_\infty$. If exists $\min_{i=1\ldots 2L+1} \delta_i$ such that $\delta_i < ref$, then new $\vec{v} = \vec{z}_i$ and goto 2. Else return $\vec{v}$.

TABLE 1: Heuristics pseudocode

The disadvantage of the finite wordlength linear-phase FIR filter design method described is that the weighting function $\widehat{W}(\omega)$ cannot be observed in the design process.

4. RESULTS

Our design method has been tested on 25 filters with five different sets of frequency-domain specifications. The frequency domain specifications are given in Table 2.

Set	Band 1	Band 2	Band 3
A	$\Omega_p = [0;\ 0{,}40\pi]$ pass $W(\omega_p) = 1$	$\Omega_s = [0{,}50\pi;\ \pi]$ stop $W(\omega_s) = 1$	
B	$\Omega_p = [0;\ 0{,}40\pi]$ pass $W(\omega_p) = 1$	$\Omega_s = [0{,}50\pi;\ \pi]$ stop $W(\omega_s) = 10$	
C	$\Omega_{p1} = [0;\ 0{,}24\pi]$ pass $W(\omega_{p1}) = 1$	$\Omega_s = [0{,}40\pi;\ 0{,}68\pi]$ stop $W(\omega_s) = 1$	$\Omega_{p2} = [0{,}84\pi;\ \pi]$ pass $W(\omega_{p2}) = 1$
D	$\Omega_{p1} = [0;\ 0{,}24\pi]$ pass $W(\omega_{p1}) = 1$	$\Omega_s = [0{,}40\pi;\ 0{,}68\pi]$ stop $W(\omega_s) = 10$	$\Omega_{p2} = [0{,}84\pi;\ \pi]$ pass $W(\omega_{p2}) = 1$
E	$\Omega_p = [0{,}02\pi;\ 0{,}42\pi]$ pass $W(\omega_p) = 1$	$\Omega_s = [0{,}52\pi;\ 0{,}98\pi]$ stop $W(\omega_s) = 1$	

TABLE 2: Characteristics of filter test sets

Initially, filter A with 45 eight-bit coefficients was synthesized. As can be seen in Table 3, the design method with Babai's algorithm and heuristics did not give us the optimal filter, but the result was better than using rounding of coefficients calculated by the Remez algorithm. The calculation time of Babai's algorithm and heuristics was very short compared to the MILP method. The heuristics have also slightly improved the result of Babai's algorithm. Fig. 1 shows the magnitude response for the above filter designed using the MILP and Babai's algorithm with heuristics.

	Rounding	**Babai's algorithm**	**Babai's algorithm with heuristics**	**MILP**
Max. deviation	0.037059	0.032884	0.032668	0.028847
Passband	0.03125 (0.267279 dB)	0.029542 (0.252882 dB)	0.032668 (0.279217 dB)	0.028847 (0.247016 dB)
Stopband	0.037059 (28.622046 dB)	0.032884 (29.660432 dB)	0.031262 (30.099566 dB)	0.027691 (31.153227 dB)
Calculation time	0.064 s	0.192 s	0.210 s	44.431 s

TABLE 3: Results for filter A, 45 eight-bit coefficients

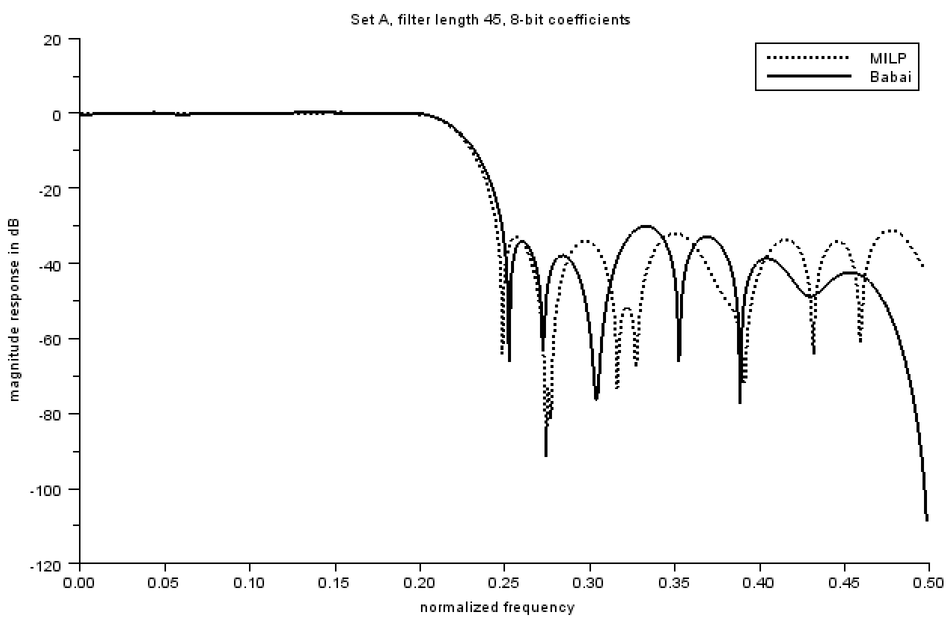

FIGURE 1: Magnitude response for filter from set A (length 45, 8-bit coefficients)

The non-unit weighting function could not be used in the design process. When designing filters from sets B and D, the magnitude response from the Remez algorithm was approximated. In these cases, the results were not very good. In all other cases, results were better than using the simple rounding technique, and in some cases optimal filter coefficients were calculated. The results can be seen in Table 4.

Set			Rounding	Babai's algorithm with heuristics	MILP
A	M=25, b=7	max. deviation	0.078125	0.065237	0.065237
	M=35, b=8	max. deviation	0.032668	0.032668	0.029966
	M=45, b=8	max. deviation	0.037059	0.032668	0.028847
	M=45, b=10	max. deviation	0.013872	0.010182	0.010182
	M=55, b=10	max. deviation	0.00979	0.009144	0.008296
B	M=25, b=8	passband deviation	0.140625	0.143501	0.154246
		stopband deviation	0.032914	0.039063	0.015234
	M=35, b=9	passband deviation	0.074219	0.066406	0.073276
		stopband deviation	0.015902	0.022931	0.007313
	M=45, b=9	passband deviation	0.03801	0.034366	0.052641
		stopband deviation	0.011719	0.015908	0.005681
	M=45, b=11	passband deviation	0.024179	0.025522	0.026655
		stopband deviation	0.006177	0.006282	0.002673
	M=55, b=11	passband deviation	0.010579	0.011861	0.01666
		stopband deviation	0.006234	0.004939	0.001643
C	M=25, b=7	max. deviation	0.041507	0.036676	0.036676
	M=35, b=8	max. deviation	0.046875	0.03125	0.016767
	M=45, b=8	max. deviation	0.030466	0.029409	0.016085
	M=45, b=10	max. deviation	0.00993	0.006843	0.004761
	M=55, b=10	max. deviation	0.010839	0.007636	0.004365
D	M=25, b=8	passband 1 deviation	0.0625	0.0625	0.078125
		stopband deviation	0.014408	0.014408	0.007966
		passband 2 deviation	0.0625	0.0625	0.078125
	M=35, b=9	passband 1 deviation	0.015987	0.023438	0.03125
		stopband deviation	0.012202	0.011957	0.003302
		passband 2 deviation	0.027344	0.023438	0.03125
	M=45, b=9	passband 1 deviation	0.01768	0.014144	0.022836
		stopband deviation	0.010906	0.011689	0.002552
		passband 2 deviation	0.018901	0.011097	0.025071
	M=45, b=11	passband 1 deviation	0.004244	0.003761	0.006396
		stopband deviation	0.003222	0.002954	0.000745
		passband 2 deviation	0.003906	0.004551	0.005306
	M=55, b=11	passband 1 deviation	0.002167	0.002167	0.00518
		stopband deviation	0.00371	0.00371	0.000618
		passband 2 deviation	0.004379	0.004379	0.005859
E	M=25, b=7	max. deviation	0.077633	0.062141	0.06071
	M=35, b=8	max. deviation	0.046934	0.033004	0.032888
	M=45, b=8	max. deviation	0.035784	0.035784	0.028987
	M=45, b=10	max. deviation	0.015263	0.011185	0.010104
	M=55, b=10	max. deviation	0.011802	0.009119	0.008199

TABLE 4: Results

5. CONCLUSION

A new method using Babai's algorithm and heuristics for the design of finite wordlength linear-phase FIR filters was presented in this paper. Heuristics were used to improve the Babai's algorithm result and to bring the result closer to the L_∞ norm. Testing showed that the heuristics improve the result of Babai's algorithm. The major advantage of this design method is the speed of the algorithm. Major disadvantages are that we do not get always the optimal filter coefficients and we cannot design filters with non-unit weighting functions. These will be the main focus of our future research, and our aim is to minimize those disadvantages.

Our future research will also attempt to include some other concepts. The problem of searching the closest point in a lattice is in communications community referred as the sphere decoding. Using results of [9], [10] some of the improvements over the Babai's algorithm could be reached. In [11], [12] the peak constrained least squares method is introduced. This method balances the minimax and the squared error criteria. This approach could be useful framework to pose the filter design problem.

6. REFERENCES

[1] J. G. Proakis and D. G. Manolakis, *Introduction to Digital Signal Processing*., pp. 614-726.

[2] D. M. Kodek, "Design of optimal finite wordlength FIR digital filters using integer programming techniques," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 3, pp. 304- 308, junij 1980.

[3] D. M. Kodek, "Performance Limit of Finite Wordlength FIR Digital Filters," *IEEE Transactions on Signal Processing*, vol. 53, no. 7, pp. 2462- 2469, julij 2005.

[4] D. M. Kodek, "LLL Algorithm and the Optimal Finite Wordlength FIR Design," *IEEE Transactions on Signal Processing*, vol. 60, no. 3, pp. 1493-1498 , March 2012.

[5] N. Brisebarre and S. Chevillard, "Efficient polynomial L-approximations," *Computer Arithmetic, 2007. ARITH '07. 18th IEEE Symposium on*, pp. 169-176, junij 2007.

[6] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 264, no. 4, pp. 515-634, december 1982.

[7] N. P. Smart, *The algorithmic resolution of diophantine equations*.: Cambridge University Press, 1998, pp. 59-76.

[8] L. Babai, "On Lovász' Lattice Reduction and the Nearest Lattice Point Problem," in *Proceedings of the 2nd Symposium of Theoretical Aspects of Computer Science (STACS '85)*, London, 1985, pp. 13-20.

[9] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Transactions on Information Theory*, vol. 48, no. 8, pp. 2201-2214, August 2002.

[10] A. D. Murugan, H. El Gamal, M. O. Damen, and G. Caire, "A unified framework for tree search decoding: rediscovering the sequential decoder," *IEEE Transactions on Information Theory*, vol. 52, no. 3, pp. 933-953 , March 2006.

[11] J. W. Adams and J. L. Sullivan, "Peak-constrained least-squares optimization," *IEEE Transactions on Signal Processing*, vol. 46, no. 2, pp. 306-321, February 1998.

[12] I. W. Selesnick, M. Lang, and C. S. Burrus, "Constrained least square design of FIR filters without specified transition bands," *IEEE Transactions on Signal Processing*, vol. 44, no. 8, pp. 1879-1892, August 1996.