AN ALGORITHM FOR THE DESIGN OF MULTIPLIERLESS TWO-CHANNEL PERFECT RECONSTRUCTION ORTHOGONAL LATTICE FILTER BANKS

Juha Yli-Kaakinen, Tapio Saramäki, and Robert Bregović

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

ABSTRACT

This paper describes an algorithm for designing multiplierless two-channel perfect-reconstruction (PR) orthogonal lattice filter banks. The advantage of the lattice implementation is that the PR property is structurally ensured even after quantizing the coefficients into very simple representation forms. The coefficient optimization is performed in three basic stages. First, a simple design scheme is used for generating an initial solution for further optimization. Second, this initial solution is used as a start-up solution for the non-linear optimization algorithm being employed for determining a parameter space of the infinite-precision coefficients including the feasible space where the filter bank meets the given criteria. The third step involves finding the coefficients in this space so that the resulting filter bank meets the given criteria with simple coefficient representation forms. An example taken from the literature illustrates that the proposed algorithm results at least in a good suboptimal finite-precision solution in a fairly short time.

1. INTRODUCTION

DURING the last three decades, two-channel filter banks have been broadly examined due to their applications in many areas, such as subband coding of speech and images, communications, and short time spectral analysis [1]–[3]. In all applications the purpose of the two-channel filter bank is to separate in the frequency domain the signal under consideration into two signals or compose two signals into a single signal. This enables one to perform the necessary signal processing operation at a lower sampling rate or to transmit multiple signals through one channel. Moreover, a filter bank can be designed in such a way that the output signal of the filter bank is a delayed version of the input signal or at least a very good approximation of the input signal. The first case corresponds to the perfect-reconstruction (PR) filter banks whereas the second case corresponds to the nearly perfect-reconstruction (NPR) filter banks.

This paper concentrates on multirate alias-free PR orthogonal two-channel filter banks where finite-impulse response (FIR) filters are used for constructing the filter bank [4]. In such a filter bank, all the filters are derived from one filter by using proper transforms as it will be shown in Section 2. This fact significantly simplifies the filter bank design. Furthermore, PR orthogonal filter banks can be implemented in the direct form or by using a lattice structure. The disadvantage of the direct-form implementations is that the PR property is affected by the coefficient quantization. However, a lattice implementation structurally ensures the PR property for all values of the lattice coefficients [5]. Since the PR property is structurally ensured, it is only necessary to consider the magnitude response when optimizing the lattice coefficient values in the discrete coefficient space.

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

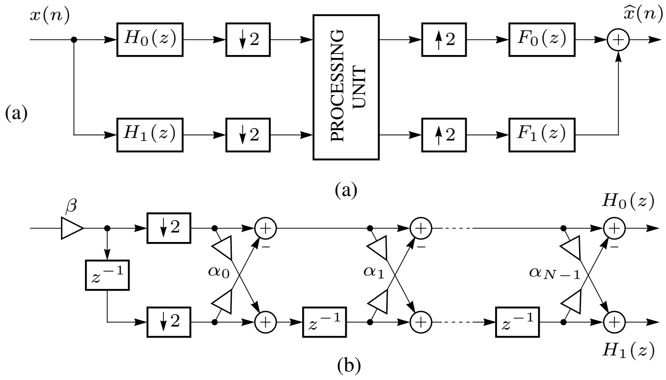

Fig. 1 (a) Two-channel filter bank. (b) Lattice structure for the analysis bank of the PR orthogonal filter bank.

This paper proposes an efficient algorithm for designing multiplierless PR lattice orthogonal filter banks. This algorithm is based on the following observation: Finding the smallest and largest values of the lattice coefficients in such a way that the given criteria are still met enables one to find a parameter space including the feasible space where the filter bank specifications are met. After finding this larger space, all what is needed is to check whether in this space there exist the desired discrete values for the lattice coefficient representations. In practice, the resulting space is too large to be searched in a sensible time even for filter banks constructed using filters of a moderate order. Therefore, this space is appropriately divided into smaller fragments and all the discrete coefficient value combinations inside these smaller parameter spaces are then evaluated. An example taken from the literature shows that the proposed algorithm results in the finite-precision solutions with a performance comparable with those ones achieved in [6].

2. PERFECT RECONSTRUCTION ORTHOGONAL LATTICE FILTER BANKS

Consider the two-channel filter bank shown in Fig. 1(a), where $H_0(z)$ and $H_1(z)$ represent the low-pass and high-pass filters in the analysis bank, respectively, and $F_0(z)$ and $F_1(z)$ are the corresponding synthesis filters. Let the length of each analysis and synthesis filter be $2N$.

Assuming that there are no coding, quantization, or channel degradations in the processing unit, $\widehat{x}(n) = x(n-2N+1)$, that is, the overall bank satisfies the PR property, is achieved by meeting the following conditions [7]:

Condition 1: $2N-1$ is an odd integer.

Condition 2: $F_0(z) = 2H_1(-z)$, $F_1(z) = -2H_0(-z)$, and $H_1(z) = -z^{-(2N-1)}H_0(-z^{-1})$.

Condition 3: $T(z) = [H_0(z)F_0(z) + H_1(z)F_1(z)]/2 = z^{-(2N-1)}$.

In Condition 3, $T(z)$ is the input-output transfer function for the overall PR system. Equivalently, Condition 3 can be expressed, after some manipulations, in terms of the low-pass analysis transfer function $H_0(z)$, as

$$H_0(z)H_0(z^{-1}) + H_0(-z)H_0(-z^{-1}) = 1. \quad (1)$$

The above equation implies that $H_0(z)$ satisfies the following power complementary property [5]:

$$|H_0(e^{j\omega})|^2 + |H_0(e^{j(\omega-\pi)})|^2 = 1. \quad (2)$$

The lattice implementation of a $(2N-1)$-th order PR analysis bank with N multipliers is shown in Fig. 1(b). Note that in this implementation the decimation by a factor of two is performed before implementing the lattice structure. The filters obtained using this lattice are structurally ensured to satisfy Eq. (2). For this structure, the analysis filters $H_0(z)$ and $H_1(z)$ for the N-stage lattice are given by

$$\begin{bmatrix} H_0(z) \\ H_1(z) \end{bmatrix} = \beta_N X(\alpha_{N-1}) Z X(\alpha_{N-2}) Z \cdots Z X(\alpha_0) \begin{bmatrix} 1 \\ z^{-1} \end{bmatrix}, \quad (3a)$$

where

$$\beta_N^2 = \frac{1}{2} \prod_{k=0}^{N-1} \frac{1}{1+\alpha_k^2}, \quad (3b)$$

$$X(\alpha_k) = \begin{bmatrix} 1 & -\alpha_k \\ \alpha_k & 1 \end{bmatrix}, \quad \text{and} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & z^{-2} \end{bmatrix}. \quad (3c)$$

3. STATEMENT OF THE PROBLEM

Before stating the optimization problem, the transfer function of the low-pass analysis filter is denoted by $H_0(\Phi, z)$, where Φ is the following adjustable parameter vector:

$$\Phi = [\alpha_0, \alpha_1, \ldots, \alpha_{N-1}]. \quad (4)$$

Given the stopband edge ω_s and the stopband ripple δ_s the magnitude specifications for the low-pass analysis filter are stated as follows:[1]

$$|H_0(\Phi, e^{j\omega})| \le \delta_s \quad \text{for } \omega \in [\omega_s, \pi]. \quad (5)$$

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

$$\sum_{r=1}^{R} a_r 2^{-P_r}, \quad (6)$$

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

As the implementation cost, the number of adders and subtracters required to implement all the lattice coefficients as well as the structural adders required to implement the lattice stages is used, that is, the implementation cost is given by

$$2N + \sum_{l=0}^{N-1} \sigma_l, \quad (7)$$

where σ_l for $l = 0, 1, \ldots, N-1$ is the number of adders and/or subtracters required to implement α_l. The optimization problem under consideration is the following.

Optimization Problem: Given ω_s and δ_s find N and the parameter vector Φ in such a manner that, first, the criteria of Eq. (5) are met after quantizing the values of the lattice coefficients corresponding to the parameters included in Φ to achieve the above-mentioned form for their representations and, then, the implementation cost, as given by Eq. (7), is minimized.

4. FILTER BANK OPTIMIZATION

The solutions to the stated optimization problem can be found in the following three steps. In the first step, an initial filter is designed using a simple design scheme. In the second step, the smallest and largest values are determined for each adjustable parameter by reoptimizing the remaining unknowns in a parameter vector in such a manner that the given specifications are met. This enables one to find the parameter space of the infinite-precision coefficients including the feasible space where the filter meets the specifications. The third step involves finding the filter parameters in this space such that the resulting filter meets the given criteria with simple coefficient representation forms.

4.1. Algorithm for Finding an Initial Filter Bank

The design of $H_0(z)$ and $H_1(z)$ for the orthogonal filter bank in such a way that the magnitude response of $H_0(z)$ approximates in the minimax sense zero on $[\omega_s, \pi]$ can be accomplished in the following four steps [7]–[9]:

Step 1: Design an equiripple half-band filter of order $2N_0 = 2(2N-1)$ having the stopband edge at ω_s. The design of this filter can be performed with the aid of Remez algorithm by using the desired function equal to unity on $[0, \pi - \omega_s]$ and equal to zero on $[\omega_s, \pi]$ and by using equal weightings in the passband and stopband. Let the resulting transfer function having single zeros on the unit circle be $\widetilde{E}(z) = \sum_{n=0}^{2N_0} \widetilde{e}(n) z^{-n}$. The impulse response coefficients of this filter satisfies $\widetilde{e}(N_0) = 1/2$ and $\widetilde{e}(N_0 \pm 2r) = 0$ for $r = 1, 2, \ldots, (N_0-1)/2$. Furthermore, let the resulting stopband ripple be $\widehat{\delta}_s$.

Step 2: Determine with the aid of $\widetilde{E}(z)$ a half-band filter transfer function $E(z) = \sum_{n=0}^{2N_0} e(n) z^{-n}$ having double zeros on the unit circle. This is achieved by relating the $e(n)$'s to the $\widetilde{e}(n)$'s through $e(N_0) = \widetilde{e}(N_0)$ and $e(n) = \widetilde{e}(n)/(1 + 2\widehat{\delta}_s)$ for $n \ne N_0$. This gives $e(N_0) = 1/2$ and $e(N_0 \pm 2r) = 0$ for $r = 1, 2, \ldots, (N_0-1)/2$, as is desired.

Step 3: Form $H_0(z) = \sum_{n=0}^{N_0} h_0(n) z^{-n}$ such that it contains the zeros of $E(z)$ inside the unit circle and one each of the double zeros on the unit circle. Finally, scale $H_0(z)$ in such a way that $|H_0(1)| = 1$. This guarantees that the corresponding magnitude response takes on the value of unity at the zero frequency that is true for orthogonal filter banks. The corresponding $H_1(-z) = z^{-N_0} H_0(z^{-1})$ has the same zeros on the unit circle and contains those zeros of $E(z)$ being located outside the unit circle.

[1] Due to the fact that the filters obtained using the lattice are structurally ensured to satisfy the power complementary property, only the stopband magnitude response of $H_0(z)$ is required to be taken into account when optimizing these filters.

Step 4: Transform the resulting impulse-response values $h_0(n)$ and $h_1(n)$ for $n = 1, 2, \ldots, N_0$ to the lattice coefficient values α_n for $n = 0, 1, \ldots, N-1$ using the algorithm described in [2].

The magnitude responses of the resulting filters are orthogonal and exhibit equiripple performances in both the passbands and stopbands.

4.2. Optimization of Infinite-Precision Filter Bank

It has turned out that a very straightforward quantization scheme for optimizing the lattice coefficients is obtained as follows: For each lattice coefficient, the smallest and largest values are determined so that by reoptimizing the values of the remaining coefficients the given overall criteria, as given by Eq. (5), can still be met. This goal is achieved by solving $2N$ problems of the following form. Find the adjustable parameter vector Φ to minimize ψ subject to the condition of Eq. (5). Here, ψ is $-\alpha_n$ or α_n where α_n is one among the lattice coefficients α_n for $n = 0, 1, \ldots, N-1$. To solve these problems, the stopband region is discretized into the frequency points $\omega_i \in [\omega_s, \pi]$ for $i = 1, 2, \ldots, I$. The resulting discrete minimization problem is to find α_n for $n = 0, 1, \ldots, N-1$ to minimize ψ subject to the following condition

$$|H_0(\Phi, e^{j\omega_i})| - \delta_s \leq 0 \quad \text{for } i = 1, 2, \ldots, I. \tag{8}$$

The above problems can be solved using a nonlinear optimization algorithm `fmincon` from the optimization toolbox provided by the MathWorks Inc.

4.3. Optimization of Finite-Precision Filter Bank

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

This search can be done in a straightforward manner by first finding the sets of CSD numbers A_l for $l = 0, 1, \ldots, N-1$ between the smallest and largest values of each lattice coefficient, i.e., for $l = 0, 1, \ldots, N-1$

$$\left\{ A_l \in \mathbb{CSD}_{(R, P_R)} \mid \alpha_l^{(\min)} \leq A_l \leq \alpha_l^{(\max)} \right\}, \tag{9}$$

where $\alpha_l^{(\min)}$ and $\alpha_l^{(\max)}$ for $l = 0, 1, \ldots, N-1$, respectively, denote the smallest and largest values of the lattice coefficients obtained using the infinite-precision optimization. The stopband magnitude response is then evaluated for each combination of the A_l's to check whether the filter meets the given specifications.

In practice, the number of combinations is impractically large even for filter banks constructed using filters of a moderate order. This is because the difference between the largest and smallest values of the lattice coefficients is related to the absolute value of each coefficient, that is, for the lattice coefficients having a small absolute value, the difference between the largest and smallest values is small, whereas for the coefficients having a large absolute value, the difference is larger. However, it is well known, that for the optimal minimum-phase $H_0(z)$ the sign of α_l's alternates [2]. In addition, it has been observed that in the infinite-precision optimization, the lattice coefficients follow each others in such a manner that when the negative-valued lattice coefficients are minimized, then the positive-valued coefficients approximately achieve their maximum values. Similarly, when the positive-valued lattice coefficients are maximized, then the negative-valued coefficients achieve their minimum values. Therefore, it has been expected that also the finite-precision lattice coefficients follow each others in a similar manner. It should be pointed out that this is true only for the smallest order filter satisfying the given specifications. Based on these reasonings, the following algorithm has been developed for finding the finite-precision lattice coefficients:

Step 1: Set $r = 0$ and $\Delta_l = (\alpha_l^{(\max)} - \alpha_l^{(\min)})/D$ for $l = 0, 1, \ldots, N-1$, where D is a positive integer.

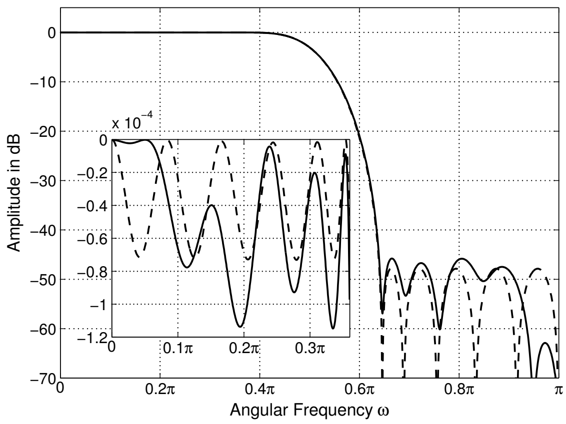

Fig. 2 Magnitude responses for the initial analysis low-pass filter (dashed line) and the optimized finite-precision filter with $R = 3$ and $P_R = 9$ (solid line).

Table 1 The initial lattice coefficient values, the smallest and largest values of the infinite-precision lattice coefficients as well as the differences between the smallest and largest values of the lattice coefficients in the example of Section 5.

l	$\alpha_l^{(\mathrm{opt})}$	$\alpha_l^{(\min)}$	$\alpha_l^{(\max)}$	Δ_l
0	−3.726 207	−5.010 255	−3.231 103	1.779 152
1	1.206 685	1.044 245	1.557 822	0.513 578
2	−0.681 945	−0.851 333	−0.585 663	0.265 670
3	0.442 906	0.374 817	0.545 774	0.170 957
4	−0.300 285	−0.370 836	−0.248 317	0.122 518
5	0.203 235	0.162 543	0.254 749	0.092 206
6	−0.133 057	−0.171 415	−0.101 577	0.069 838
7	0.081 756	0.058 482	0.109 714	0.051 231
8	−0.045 440	−0.064 577	−0.029 598	0.034 980
9	0.021 585	0.012 000	0.033 332	0.021 332
10	−0.008 271	−0.015 278	−0.002 535	0.012 743

Step 2: Set $\alpha_l^{(\max)} = \alpha_l^{(\min)} + \Delta_l$ for $l = 0, 2, \ldots, 2\lfloor (N-1)/2 \rfloor$ and $\alpha_l^{(\min)} = \alpha_l^{(\max)} - \Delta_l$ for $l = 1, 3, \ldots, 2\lfloor N/2 \rfloor$.

Step 3: Evaluate the stopband magnitude response for each combination of the discrete coefficient values between the updated smallest and largest values of the lattice coefficients $\alpha_l^{(\min)}$ and $\alpha_l^{(\max)}$ for $l = 0, 1, \ldots, N-1$, respectively, to check whether the filter meets the given criteria.

Step 4: Set $r = r + 1$. If $r = D$, then stop. Otherwise, go to the next step.

Step 5: Set $\alpha_l^{(\min)} = \alpha_l^{(\max)}$ for $l = 0, 2, \ldots, 2\lfloor (N-1)/2 \rfloor$, $\alpha_l^{(\max)} = \alpha_l^{(\min)}$ for $l = 1, 3, \ldots, 2\lfloor N/2 \rfloor$, and go to Step 2.

In the above algorithm, the space between the smallest and largest values of each lattice coefficient is divided into D parts. The proper value of D is selected to be the largest integer for which there exist the discrete coefficient values between all the updated smallest and largest values of the lattice coefficient for $r = 0, 1, \ldots, D-1$. If no solutions satisfying the given specifications for the given coefficient representation and for the given value of D are found, then D is decreased by one and the algorithm is restarted. The division of the space between the smallest and largest values of each lattice coefficient into D fragments approximately reduces the computational workload by a factor of D^{N-1}.

5. NUMERICAL EXAMPLE

In order to compare the performance of the proposed algorithm with that presented in [6], the following specifications are considered: the stop-

Table 2 Updated smallest and largest values for the infinite-precision lattice coefficients with $D = 5$ in example of Section 5.

	$r=0$		$r=1$		$r=2$		$r=3$		$r=4$	
l	$\alpha_l^{(\min)}$	$\alpha_l^{(\max)}$	$\alpha_l^{(\min)}$	$\alpha_l^{(\max)}$	$\alpha_l^{(\min)}$	$\alpha_l^{(\max)}$	$\alpha_l^{(\min)}$	$\alpha_l^{(\max)}$	$\alpha_l^{(\min)}$	$\alpha_l^{(\max)}$
0	−5.0103	−4.6544	−4.6544	−4.2986	−4.2986	−3.9428	−3.9428	−3.5869	−3.5869	−3.2311
1	1.4551	1.5578	1.3524	1.4551	1.2497	1.3524	1.1470	1.2497	1.0442	1.1470
2	−0.8513	−0.7982	−0.7982	−0.7451	−0.7451	−0.6919	−0.6919	−0.6388	−0.6388	−0.5857
3	0.5116	0.5458	0.4774	0.5116	0.4432	0.4774	0.4090	0.4432	0.3748	0.4090
4	−0.3708	−0.3463	−0.3463	−0.3218	−0.3218	−0.2973	−0.2973	−0.2728	−0.2728	−0.2483
5	0.2363	0.2547	0.2179	0.2363	0.1994	0.2179	0.1810	0.1994	0.1625	0.1810
6	−0.1714	−0.1574	−0.1574	−0.1435	−0.1435	−0.1295	−0.1295	−0.1155	−0.1155	−0.1016
7	0.0995	0.1097	0.0892	0.0995	0.0790	0.0892	0.0687	0.0790	0.0585	0.0687
8	−0.0646	−0.0576	−0.0576	−0.0506	−0.0506	−0.0436	−0.0436	−0.0366	−0.0366	−0.0296
9	0.0291	0.0333	0.0248	0.0291	0.0205	0.0248	0.0163	0.0205	0.0120	0.0163
10	−0.0153	−0.0127	−0.0127	−0.0102	−0.0102	−0.0076	−0.0076	−0.0051	−0.0051	−0.0025

Table 3 Optimized finite-precision lattice coefficient values for a 21th-order PR orthogonal filter bank with $R = 3$ and $P_R = 9$.

$\alpha_0 = -2^{+2} + 2^{-5}$	$\alpha_6 = -2^{-3} - 2^{-6} + 2^{-9}$
$\alpha_1 = 1 + 2^{-2} + 2^{-5}$	$\alpha_7 = 2^{-3} - 2^{-5} - 2^{-7}$
$\alpha_2 = -1 + 2^{-2} + 2^{-5}$	$\alpha_8 = -2^{-4} + 2^{-6}$
$\alpha_3 = 2^{-1} - 2^{-5} - 2^{-8}$	$\alpha_9 = 2^{-5} - 2^{-7}$
$\alpha_4 = -2^{-2} - 2^{-4} - 2^{-9}$	$\alpha_{10} = -2^{-7}$
$\alpha_5 = 2^{-2} - 2^{-5} - 2^{-7}$	

band edge is at $\omega_s = 0.64\pi$ ($\omega_p = 0.36\pi$) and the required stopband attenuation is 45.45 dB ($\delta_s = 0.005\,339$). The minimum order of an orthogonal lattice filter bank to meet the given magnitude criteria is 21 ($N = 11$). The stopband attenuation of the infinite-precision filter is 47.85 dB. The initial infinite-precision values of the lattice coefficients $\alpha_l^{(\mathrm{opt})}$ for $l = 0, 1, \ldots, 10$ are shown in Table 1 and the magnitude response of the initial low-pass analysis filter is shown in Fig. 2 using a dashed line.

In this case, the number of infinite-precision optimization problems is 22. The overall CPU-time required for solving all these problems with $I = 150$ is approximately 200 s when using a MATLAB 6.5 code running on a 500 MHz AlphaServer. The resulting smallest and largest values of the lattice coefficients as well as the Δ_l's being the differences between these values are also shown in Table 1. If three powers-of-two terms and nine fractional bits ($R = 3$ and $P_R = 9$) are used for the coefficient representations, then the number of discrete values between $\alpha_l^{(\min)}$ and $\alpha_l^{(\max)}$ for $l = 0, 1, \ldots, 10$ are 144, 46, 25, 50, 30, 30, 28, 23, 18, 11, and 6, respectively. Hence, in the $D = 1$ case, the overall number of coefficient value combinations is $5.7 \cdot 10^{15}$.

The largest value of D for which there exist the discrete coefficient values between all the updated smallest and largest values of the lattice coefficients is six. In this case, the number of coefficient value combinations are 1 658 880, 423 360, 15 396 480, 4 457 376, 3 981 312, and 10 608 000 for $r = 0, 1, \ldots, 5$, respectively. However, none of these combinations satisfies the given specification. The CPU-time required for evaluating the stopband attenuation with $I = 40$ for all these combinations is 549 s when using a Fortran 95 program on a 500 MHz AlphaServer.

Table 2 shows in the $D = 5$ case the updated smallest and largest values of the lattice coefficient for $r = 0, 1, \ldots, 4$. In this case, the number of coefficient value combinations are 9 123 840, 11 468 800, 92 017 296, 63 221 760, and 109 699 920 for $r = 0, 1, \ldots, 4$, respectively. The CPU-time required for evaluating all these combinations is 1.14 h. In this case, solutions satisfying the given criteria are found. The optimized finite-precision lattice coefficient for the solution maximizing the stopband attenuation with the lowest implementation cost are given in Table 3. The magnitude response for this optimized low-pass analysis filter is shown in Fig. 2 using a solid line.

Table 4 compares the performance of the proposed algorithm with that of the three search algorithm proposed in [6]. In this table, A_s denotes the minimum stopband attenuation in decibels and N_A is the number of adders and/or subtracters required to implement the analysis filter bank, that is, the number of adders and/or subtracters required to implement the α_l's as well as the structural adders required to implement the lattice stages [cf. Eq (7)]. For comparison purposes, also the $P_R = 8$ case is shown in this table. As can be seen from this table, the proposed algorithm results in a comparable solution much faster than the algorithm proposed in [6]. It should be pointed out that the proposed algorithm does not result in the optimal solution. However, it provides a reasonable good suboptimal solution in a fairly short time.

Table 4 Summary of filter bank designs.

	N	A_s	R	P_R	N_A	CPU
Tree search [6]	14	45.45 dB	2	10	56	∼ 55 h
Proposed algorithm	11	45.78 dB	3	9	56	1.14 h
	11	44.92 dB	3	8	52	2.34 h

REFERENCES

[1] H. S. Malvar, *Signal Processing with Lapped Transforms*. Boston, MA, USA: Artech House, 1992.

[2] P. P. Vaidyanathan, *Multirate Systems and Filter Banks*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[3] N. J. Fliege, *Multirate Digital Signal Processing*. Chichester: John Wiley & Sons, 1994.

[4] R. Bregović and T. Saramäki, "A general-purpose optimization approach for designing two-channel FIR filter banks," *IEEE Trans. Signal Processing*, vol. 51, pp. 1783–1791, July 2003.

[5] P. P. Vaidyanathan and P.-Q. Hoang, "Lattice structures for optimal design and robust implementation of two-channel perfect reconstruction QMF banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 1, pp. 81–93, Jan. 1988.

[6] Y. C. Lim and Y. J. Yu, "A width-recursive depth-first tree search approach for the design of discrete coefficient perfect reconstruction lattice filter bank," *IEEE Trans. Circuits Syst. II*, vol. 50, no. 6, pp. 257–266, June 2003.

[7] R. Bregović, "Optimal design of perfect-reconstruction and nearly perfect-reconstruction multirate filter banks," Dr. Tech. dissertation, Dept. of Inform. Tech., Tampere Univ. of Tech, Finland, Tampere, Finland, Aug. 2003.

[8] F. Mintzer, "Filters for distortion-free two-band multirate filter banks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 33, no. 3, pp. 626–630, June 1985.

[9] M. Smith and T. Barnwell III, "Exact reconstruction techniques for tree-structured subband coders," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 34, no. 3, pp. 434–441, June 1986.