Signal Processing 88 (2008) 2917–2926

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A neural network approach to FIR filter design using frequency-response masking technique

Xiao-Hua Wang [a,b,*], Yi-Gang He [b]

[a] Department of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, Hunan 410077, China
[b] Department of Electrical and Information Engineering, Hunan University, Changsha, Hunan 410082, China

ARTICLE INFO

Article history:
Received 15 November 2007
Received in revised form
11 June 2008
Accepted 24 June 2008
Available online 27 June 2008

Keywords:
Frequency-response masking
FIR digital filter
Neural network
Optimal design

ABSTRACT

This paper presents a new optimization method for the design of various frequency-response-masking (FRM)-based linear-phase finite-impulse response (FIR) digital filters. The method is based on a batch back-propagation neural network algorithm (NNA), which is taken as a variable learning rate mode. In order to reduce the complexity, the following two-step optimization technique is proposed. At the first step, an initial FRM filter is designed by alternately optimizing the sub-filters. This solution is then used as a start-up solution for further optimization. At the second step, the coefficients of overall sub-filters are optimized simultaneously by the NNA. Algorithm details for the design of basic and multistage FRM filters are presented to show that the proposed approach offers a unified design framework for a variety of FRM filters. Some examples taken from the literatures are included and the results show that the proposed algorithm can design better FRM filters than several existing methods.

Crown Copyright © 2008 Published by Elsevier B.V. All rights reserved.

1. Introduction

Digital signal-processing techniques are being applied to wireless communication systems, digital TVs, and multimedia system, which has resulted in a demand for narrow transition-width linear-phase finite-impulse response (FIR) digital filters. One of the most efficient approaches for synthesizing narrow transition band linear-phase FIR digital filters is the frequency-response masking (FRM) technique. This approach produces filters with very sparse coefficient vectors and hence reduces the number of multipliers and adders required for implementation tremendously when compared to those yielded by conventional direct design methods. As a result, in many cases, it has become the method of choice primarily because of the considerably reduced realization complexity it offers compared with the other available option [1–8].

Several design methods for FRM filters have been investigated in the past. In these methods [9–13] the sub-filters are optimized separately, and therefore produce only suboptimal solutions. Recently, several joint optimizations [14–19] have been developed. In [16,17], a weighted least squares (WLS) method is used for designing linear-phase FRM filters. In [16], the original least squares (LS) problem is decomposed into two linear LS problems when designing basic structure FRM filters. In [17], the original LS problem is decomposed into four linear LS problems when designing a modified basic structure FRM filters. Although the optimal design of multistage FRM filters was not addressed in [16,17], the WLS method appears to be applicable to such a design. However, as the number of filter stages increases, the original LS problem need to be decomposed into many linear LS problems. This will make the design process increasingly. In [18], Lu and Hinamomoto presented

* Corresponding author at: Department of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, Hunan 410077, China.
E-mail addresses: cslgwxh@yahoo.com.cn, cslgwxh@163.com (X.-H. Wang), hyghnu@yahoo.com.cn (Y.-G. He).

0165-1684/$ - see front matter Crown Copyright © 2008 Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2008.06.015

a semidefinite programing (SDP) approach in which the set of FRM filter coefficients of all sub-filters is treated as a single design vector and an optimal FRM filter can be designed through a sequence of linear updates for the design variables. The SDP method is readily extended to multistage FRM filters. However, a problem with the method is the large number of constraints that inevitably affects design efficiency and, in the case of high-order FRM filters, may cause numerical difficulties.

This paper introduces a joint optimization method for the design of linear-phase FIR digital filters using the FRM technique. The method is based on a neural network approach (NNA) in which the coefficients of overall sub-filters are optimized simultaneously. The NNA can be readily extended to multistage FRM filters design.

2. FRM filter structure and frequency response

As can be seen from Fig. 1, the linear-phase FIR filter transfer function of the basic FRM structure is constructed as follows:

$$H(z) = H_a(z^M)H_{ma}(z) + [z^{-0.5M(N_a-1)} - H_a(z^M)]H_{mc}(z) \tag{1a}$$

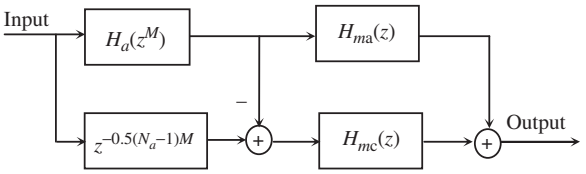

Fig. 1. Basic FRM filter structure.

The impulse-response coefficients $h_n^{(a)}$, $h_n^{(ma)}$, and $h_n^{(mc)}$ are symmetrical. If all sub-filters have linear-phase response with N_{ma} and N_{mc} either both even or both odd and $(N_a-1)M$ even, then the FRM filter has a linear-phase response.

The zero-phase frequency response of $H(z)$ (the group delay $D=(N_a-1)M/2+\max\{(N_{ma}-1)/2, (N_{mc}-1/2\}$ is omitted) can be express as

$$\begin{aligned} H(\omega) &= [\boldsymbol{a}_a^{\mathrm{T}}\boldsymbol{c}_a(\omega)][\boldsymbol{a}_{ma}^{\mathrm{T}}\boldsymbol{c}_{ma}(\omega) - \boldsymbol{a}_{mc}^{\mathrm{T}}\boldsymbol{c}_{mc}(\omega)] + \boldsymbol{a}_{mc}^{\mathrm{T}}\boldsymbol{c}_{mc}(\omega) \\ &= H_a(\omega)[H_{ma}(\omega) - H_{mc}(\omega)] + H_{mc}(\omega) \end{aligned} \tag{2a}$$

where

$$\boldsymbol{a}_a = \begin{cases} [h_a((N_a-1)/2) \quad 2h_a((N_a+1)/2) \quad \dots \quad 2h_a(N_a-1)]^{\mathrm{T}}, & \text{if } N_a \text{ odd} \\ 2[h_a(N_a/2) \quad \dots \quad h_a(N_a-1)]^{\mathrm{T}}, & \text{if } N_a \text{ even} \end{cases} \tag{2b}$$

$$\boldsymbol{c}_a(\omega) = \begin{cases} [1\ \cos(M\omega) \quad \dots \quad \cos[(N_a-1)M\omega/2]]^{\mathrm{T}}, & \text{if } N_a \text{ odd} \\ [\cos(M\omega/2) \quad \dots \quad \cos[(N_a-1)M\omega/2]]^{\mathrm{T}}, & \text{if } N_a \text{ even} \end{cases} \tag{2c}$$

$$\boldsymbol{a}_{ma} = \begin{cases} [h_{ma}((N_{ma}-1)/2) \quad 2h_{ma}((N_{ma}+1)/2) \quad \dots \quad 2h_{ma}(N_{ma}-1)]^{\mathrm{T}}, & \text{if } N_{ma} \text{ odd} \\ 2[h_{ma}(N_{ma}/2) \quad \dots \quad h_{ma}(N_{ma}-1)]^{\mathrm{T}}, & \text{if } N_{ma} \text{even} \end{cases} \tag{2d}$$

$$\boldsymbol{c}_{ma}(\omega) = \begin{cases} [1 \quad \cos\omega \quad \dots \quad \cos[(N_{ma}-1)\omega/2]]^{\mathrm{T}}, & \text{if } N_{ma} \text{ odd} \\ [\cos(\omega/2) \quad \dots \quad \cos[(N_{ma}-1)\omega/2]]^{\mathrm{T}}, & \text{if } N_{ma} \text{ even} \end{cases} \tag{2e}$$

$$\boldsymbol{a}_{mc} = \begin{cases} [h_{mc}((N_{mc}-1)/2) \quad 2h_{mc}((N_{mc}+1)/2) \quad \dots \quad 2h_{mc}(N_{mc}-1)]^{\mathrm{T}}, & \text{if } N_{mc} \text{ odd} \\ 2[h_{mc}(N_{mc}/2) \quad \dots \quad h_{mc}(N_{mc}-1)]^{\mathrm{T}}, & \text{if } N_{mc} \text{even} \end{cases} \tag{2f}$$

$$c_{mc}(\omega) = \begin{cases} [1 \quad \cos\omega \quad \dots \quad \cos[(N_{mc}-1)\omega/2]]^{\mathrm{T}}, & \text{if } N_{mc} \text{ odd} \\ [\cos(\omega/2) \quad \dots \quad \cos[(N_{mc}-1)\omega/2]]^{\mathrm{T}}, & \text{if } N_{mc} \text{ even} \end{cases} \tag{2g}$$

where

$$H_a(z) = \sum_{n=0}^{N_a-1} h_a(n)z^{-n} \tag{1b}$$

$$H_{ma}(z) = \sum_{n=0}^{N_{ma}-1} h_{ma}(n)z^{-n} \tag{1c}$$

$$H_{mc}(z) = \sum_{n=0}^{N_{mc}-1} h_{mc}(n)z^{-n} \tag{1d}$$

Consider the design of a low-pass filter as an example. Given M, normalized passband edge ω_p and stopband edge ω_a, an initial design of sub-filters $H_a(z)$, $H_{ma}(z)$, and $H_{mc}(z)$ can be obtained using a standard method, such as Remez or WLS or the NNA as discussed in Section 3. The passband and stopband edges for $H_a(z)$, $H_{ma}(z)$, and $H_{mc}(z)$ are determined as follows [18].

(1) For $H_a(z)$, the passband edge θ and stopband edge ϕ are given by either (3) or (4) depending on which set of $\{\theta,\phi\}$ satisfies $0<\theta<\phi<\pi$:

$$\theta = \omega_p M - 2m\pi \tag{3a}$$

$$\phi = \omega_a M - 2m\pi \tag{3b}$$

$$m = \omega_p M/2\pi \tag{3c}$$

$$\theta = 2m\pi - \omega_a M \tag{4a}$$

$$\phi = 2m\pi - \omega_p M \tag{4b}$$

$$m = \omega_a M/2\pi \tag{4c}$$

where $\lfloor x \rfloor$ denotes the largest integer less than x, and $\lceil x \rceil$ denotes the smallest integer larger than x.

(2) If (3) is used to determine the values of θ and ϕ, then the passband and stopband edges of $H_{ma}(z)$ are given by $(2m\pi+\theta)/M$ and $[(2(m+1)\pi-\phi]/M$, respectively, and the passband and stopband edges of $H_{mc}(z)$ are given by $(2m\pi-\theta)/M$ and $(2m\pi+\phi)/M$, respectively.

(3) If (4) is used to determine the values of θ and ϕ, then the passband and stopband edges of $H_{ma}(z)$ are given by $[2(m-1)\pi+\phi]/M$ and $(2m\pi-\theta)/M$, respectively, and the passband and stopband edges of $H_{mc}(z)$ are given by $(2m\pi-\phi)/M$ and $(2m\pi+\theta)/M$, respectively.

3. Neural network design approach

3.1. Neural network algorithm

The optimization process is done on a dense grid frequencies ω_l, where $l = 1, 2, \ldots, L$. Using (2a), the vector of $\boldsymbol{H(\omega)}$ evaluated on the set of ω_l, $l = 1, \ldots, L$, can be expressed as

$$\begin{aligned}\boldsymbol{H} &= (\boldsymbol{a}_a^{\mathrm{T}}\boldsymbol{C}_a).^*(\boldsymbol{a}_{ma}^{\mathrm{T}}\boldsymbol{C}_{ma} - \boldsymbol{a}_{mc}^{\mathrm{T}}\boldsymbol{C}_{mc}) + \boldsymbol{a}_{mc}^{\mathrm{T}}\boldsymbol{C}_{mc} \\ &= \boldsymbol{H}_a.^*(\boldsymbol{H}_{ma} - \boldsymbol{H}_{mc}) + \boldsymbol{H}_{mc}\end{aligned} \tag{5a}$$

where ".*" is matrix element group multiplication operation in MATLAB, i.e., $\boldsymbol{A}.^*\boldsymbol{B} = [A_{ij}B_{ij}]_{N\times M}$, and

$$\boldsymbol{C}_a = [\boldsymbol{c}_a(\omega_1), \boldsymbol{c}_a(\omega_2), \ldots, \boldsymbol{c}_a(\omega_L)] \tag{5b}$$

$$\boldsymbol{C}_{ma} = [\boldsymbol{c}_{ma}(\omega_1), \boldsymbol{c}_{ma}(\omega_2), \ldots, \boldsymbol{c}_{ma}(\omega_L)] \tag{5b'}$$

$$\boldsymbol{C}_{mc} = [\boldsymbol{c}_{mc}(\omega_1), \boldsymbol{c}_{mc}(\omega_2), \ldots, \boldsymbol{c}_{mc}(\omega_L)] \tag{5c}$$

Define the error function

$$\boldsymbol{e} = \boldsymbol{B}.^*\boldsymbol{W}.^*(\boldsymbol{M}_{\mathrm{d}} - \boldsymbol{H}) \tag{6}$$

where $\boldsymbol{M}_{\mathrm{d}}$ is the vector of desired responses evaluated on ω_l, $l = 1,\ldots,L$, $\boldsymbol{B}$ and $\boldsymbol{W}$ are all the weighting coefficient vectors on ω_l, $l = 1,\ldots,L$, and $W(\omega_l)$ is one in the passband and δ_p/δ_s in the stopband (here δ_p and δ_s are the desired passband and stopband ripples, respectively), and $B(\omega_l) > 0$. Then performance index J is given by

$$J = \frac{1}{2}\sum_{l=1}^{L} e^2(l) = \frac{1}{2}||\boldsymbol{e}||_2^2 \tag{7}$$

To minimize the performance index J, a batch back-propagation neural network (BBPNN) model is chosen in Fig. 2. There, three layers are considered: input, hidden and output layers. The number of input neuron is L, and the weight of each input neuron is 1. The activation functions of the hidden neuron are composed of the three function vectors, such as $\boldsymbol{c}_a(\omega)$, $\boldsymbol{c}_{ma}(\omega)$, and $\boldsymbol{c}_{mc}(\omega)$, and the weighting vectors of the three activation function vectors are $\boldsymbol{a}_a^{\mathrm{T}}$, $\boldsymbol{a}_{ma}^{\mathrm{T}}$, and $\boldsymbol{a}_{mc}^{\mathrm{T}}$, respectively. Assume that N_a, N_{ma} and N_{mc} are all odd integers, then the number of the hidden neurons is $(N_a+N_{ma}+N_{mc}+3)/2$. The ouput lay includes $2L$ adders and L multipliers. For every frequency point ω_l, every iteration needs $(N_a+N_{ma}+N_{mc}+5)/2$ multiplications and $(N_a+N_{ma}+N_{mc}-1)/2$ additions to obtain $H(\omega_l)$. Then $\boldsymbol{a}$ is recursively updated using

$$\begin{aligned}\boldsymbol{a}_{a(k+1)} &= \boldsymbol{a}_{a(k)} - \eta\frac{\partial J}{\partial \boldsymbol{a}_{a(k)}} \\ &= \boldsymbol{a}_{a(k)} + \eta\boldsymbol{C}_a[\boldsymbol{W}^{\mathrm{T}}.^*\boldsymbol{B}^{\mathrm{T}}.^*\boldsymbol{e}_k^{\mathrm{T}}.^*(\boldsymbol{a}_{ma(k)}^{\mathrm{T}}\boldsymbol{C}_{ma} - \boldsymbol{a}_{mc(k)}^{\mathrm{T}}\boldsymbol{C}_{mc})^{\mathrm{T}}]\end{aligned} \tag{8a}$$

$$\boldsymbol{a}_{ma(k+1)} = \boldsymbol{a}_{ma(k)} + \eta\boldsymbol{C}_{ma}(\boldsymbol{W}^{\mathrm{T}}.^*\boldsymbol{B}^{\mathrm{T}}.^*\boldsymbol{e}_k^{\mathrm{T}}.^*[\boldsymbol{a}_{a(k)}^{\mathrm{T}}\boldsymbol{C}_a]^{\mathrm{T}}) \tag{8b}$$

$$\boldsymbol{a}_{mc(k+1)} = \boldsymbol{a}_{mc(k)} + \eta\boldsymbol{C}_{mc}(\boldsymbol{W}^{\mathrm{T}}.^*\boldsymbol{B}^{\mathrm{T}}.^*\boldsymbol{e}_k^{\mathrm{T}}.^*[1 - \boldsymbol{a}_{a(k)}^{\mathrm{T}}\boldsymbol{C}_a]^{\mathrm{T}}) \tag{8c}$$

where $\eta > 0$ is the learning rate. We can prove the following convergence theorem.

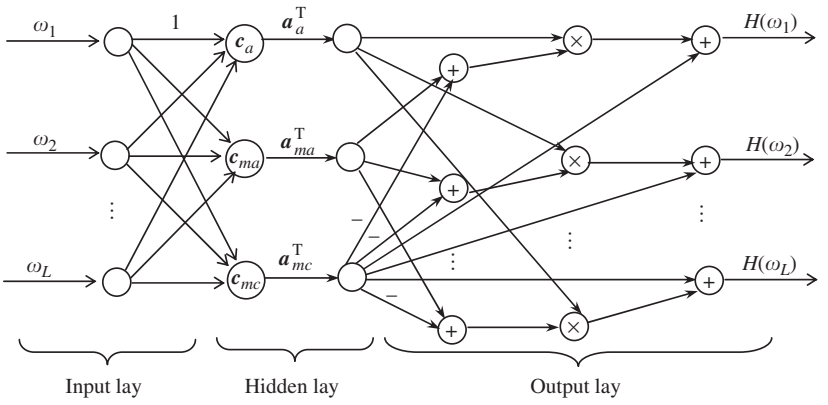

Fig. 2. Neural network model.

Convergence Theorem. If the learning rate satisfies $\eta < 2/(\mu^2 \max(\boldsymbol{B}^2)(\|\boldsymbol{C}_a\|_2^2+\|\boldsymbol{C}_{ma}\|_2^2+\|\boldsymbol{C}_{mc}\|_2^2))$, where $\mu = \max\{\boldsymbol{H}_a, \boldsymbol{H}_{ma}, \boldsymbol{H}_{mc}\}$, the neural is stable, and the algorithm is convergent to its global minimum.

3.2. Algorithm implementation

Step 1: Desired frequency response and weighting function $\boldsymbol{W}$: Consider the case of designing a low-pass FRM filter where each delay in $H_a(z)$ is replaced by M delays. The passband edge and stopband edge are ω_p and ω_a, respectively. The desired response $M_d(\omega_l)$ in this case is given by

$$M_d(\omega_l) = \begin{cases} 1, & \text{for } 0 \leqslant \omega_l \leqslant \omega_p \\ 0, & \text{for } \omega_a \leqslant \omega_l \leqslant \pi \end{cases} \tag{9}$$

The weighting function

$$W(\omega_l) = \begin{cases} 1, & \text{for} \leqslant \omega_l \leqslant \omega_p \\ \delta_p/\delta_s, & \text{for} \leqslant \omega_a \leqslant \pi \\ 0, & \text{elsewhere} \end{cases} \tag{10}$$

Step 2: Weighting function $\boldsymbol{B}$: Set initial weighting function $\boldsymbol{B}$ as

$$B(\omega_l) = \begin{cases} 1, & \text{for } 0 \leqslant \omega_l \leqslant \omega_p \\ 1, & \text{for } \omega_a \leqslant \omega_l \leqslant \pi \\ 0, & \text{elsewhere} \end{cases} \tag{11}$$

To design equiripple or quasi-equiripple FIR filters, the weighting function $B(\omega_l)$ needs to be updated as follows (see [16]):

$$B_{k+1}(\omega_l) = B_k(\omega_l)\frac{|W(\omega_l)(M_d(\omega_l) - H(\omega_l))|}{\sum_l |e_k(\omega_l)|} \tag{12}$$

Step 3: Initial design: For a given set of M, N_a, N_{ma}, N_{mc}, ω_p and ω_a, an initial design can be obtained by using a simple standard method, such as Remez or WLS or BBPNN algorithm as discussed in this section (in this paper, we use the BBPNN algorithm to design these filters). The coefficients of $H_a(z)$, $H_{ma}(z)$ and $H_{mc}(z)$ are used to form the initial parameter vectors of $\boldsymbol{a}_a$, $\boldsymbol{a}_{ma}$ and $\boldsymbol{a}_{mc}$, respectively.

Step 4: Placement of grid points and bound ε: Our design practice has indicated that relatively denser grid points should be placed in the regions near the band edges in both passband and stopband so as to avoid using unnecessarily large number of total grid points. The general practice is to have M times denser grid points in the frequency closer to the bandedge than that of others, where M is the number of zeros inserted in between two adjacent coefficients in FRM. For multiple-stage FRM techniques, the grid frequency will be $M_1 \times M_2 \times \ldots$ times accordingly. According to the rule, specify data length L. Obtain the desired amplitude response vector $\boldsymbol{M}_d$. This form the neural network training set. Then specify a suitable error bound ε.

Step 5: Produce the predicted output vector $\boldsymbol{H}$ of the neural network using (5a).

Step 6: Calculate the error vector $\boldsymbol{e}$ and performance index J via (6) and (7), respectively.

Step 7: Weight function $\boldsymbol{B}$: Update weight function $\boldsymbol{B}$ using (12).

Step 8: Learning rate: Update the learning rate according to above Convergence Theorem. In this paper, the learning rate was selected using $\eta = 1/(\mu^2 \max(\boldsymbol{B}^2)(\|\boldsymbol{C}_a\|_2^2+\|\boldsymbol{C}_{ma}\|_2^2+\|\boldsymbol{C}_{mc}\|_2^2))$.

Step 9: Update the filter coefficient vectors $\boldsymbol{a}_a$, $\boldsymbol{a}_{ma}$ and $\boldsymbol{a}_{mc}$ according to 8(a–c), respectively.

Step 10: If $J > \varepsilon$, go to Step (5); otherwise, stop.

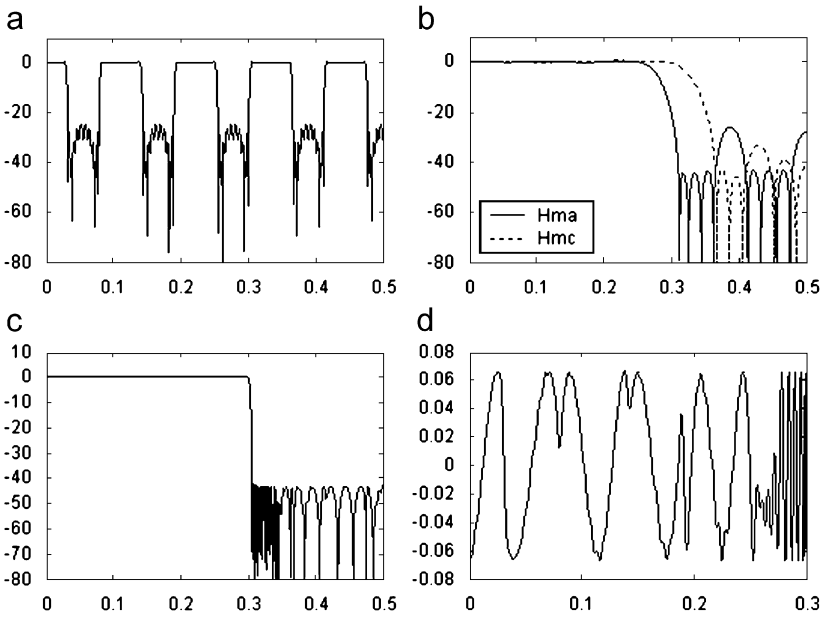

Fig. 3. Amplitude responses of (a) prototype filter $H_a(z^9)$, (b) masking filters $H_{ma}(z)$ and $H_{mc}(z)$, (c) FRM filter, and (d) passband ripples of the FRM filter; all in decibels.

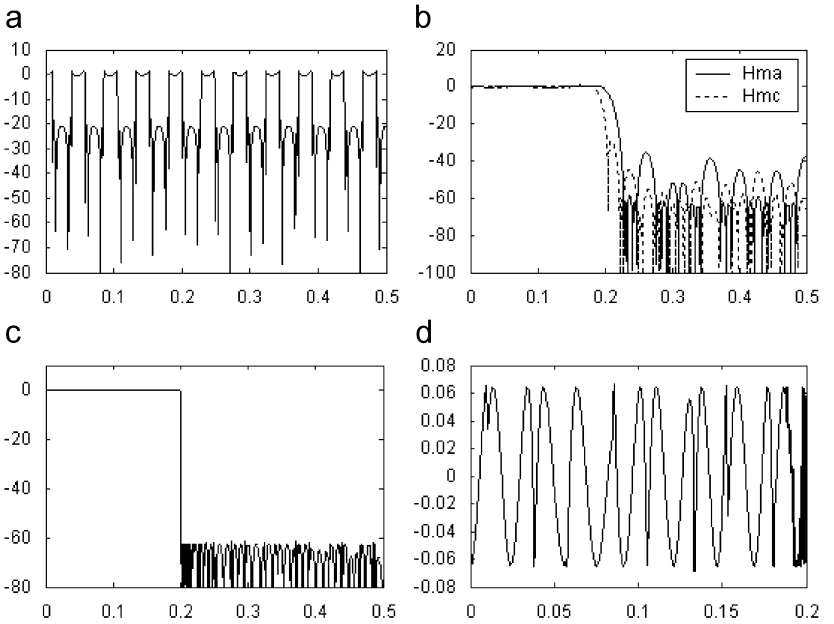

Fig. 4. Amplitude responses of (a) prototype filter $H_a(z^{21})$, (b) masking filters $H_{ma}(z)$ and $H_{mc}(z)$, (c) FRM filter, and (d) passband ripples of the FRM filter; all in decibels.

3.3. Design example

Example 1. The design is a linear-phase low-pass FRM filter with the same design parameters as the first example in [9,15,18], i.e., $M=9$, $\omega_p=0.6\pi$, and $\omega_a=0.61\pi$. Set $N_a=43$, $N_{ma}=37$, $N_{mc}=29$, $\delta_p/\delta_s=1.1$, $L=1000$, $\varepsilon=0.0062$. After 3500 training iterations (the computer used 5.2 s on a CPU of Pentium4), the algorithm converges to a FRM filter with the amplitude responses of its sub-filters $H_a(z^M)$, $H_{ma}(z)$, and $H_{mc}(z)$ shown in Fig. 3(a) and (b), respectively, and the amplitude response of the FRM filter and its passband ripples shown in Fig. 3(c) and (d), respectively. The maximum passband ripple is found to be 0.0672 dB and the minimum stopband attenuation 42.42 dB. Comparing with the literature [9,15,18] which all set $N_a=45$, $N_{ma}=41$, $N_{mc}=33$, we find that the passband ripple and stopband attenuation of the design in both [9,15] are 0.0896 and 40.96 dB, respectively, and in [18] they are 0.0674 and 42.25 dB, respectively.

A genetic algorithm (GA) [19] was also used to design this filter. The lengths of $H_a(z)$, $H_{ma}(z)$, and $H_{mc}(z)$ are 45, 28, and 22, respectively, and M is chosen to be 9. The GA technique achieved 38.0 dB stopband attenuation and 0.11 dB passband ripple. Using our approach, we achieved 39.10 dB stopband attenuation and 0.1027 dB passband ripple.

Example 2. This is a linear-phase low-pass FRM filter with the same design parameters as the first example in [14,16,18], i.e., $M=21$, $\omega_p=0.4\pi$, and $\omega_a=0.402\pi$. We set $N_a=99$, $N_{ma}=69$, $N_{mc}=85$, and define $\delta_p/\delta_s=10$, $L=2500$, $\varepsilon=0.005$. After 5500 training iterations (the computer used 36.9 s on a CPU of Pentium4), the algorithm converges to a FRM filter with the amplitude responses of its sub-filters $H_a(z^M)$, $H_{ma}(z)$, and $H_{mc}(z)$ shown in Fig. 4(a) and (b), respectively, and the amplitude response of the FRM filter and its passband ripples shown in Fig. 4(c) and (d), respectively. The maximum passband ripple is found to be 0.0689 dB, the minimum stopband attenuation is 61.54 dB and, the group delay $D=1071$. For the purpose of comparison, in [14,16,18], $N_a=123$, $N_{ma}=56$, $N_{mc}=78$, and the passband ripple and stopband attenuation of the design in [14,16] (the computer used 43 min on a SGI origin 2000) are 0.0864 and 60 dB, respectively, and in [18] they are 0.0855 and 60.93 dB, respectively, and the group delay are all $D=1319.5$ in [14,16,18]. In [11], the same filter was designed with $N_a=163$, $N_{ma}=71$, $N_{mc}=99$, $M=16$. We can find that the passband ripple and stopband attenuation of the one-stage design are 0.0864 and 60 dB, respectively, and the group delays $D=1345$. Therefore, our proposed approach offers better performance than the Remez algorithm in [11], the proposed two-step design approach in [14], the WLS method in [16] and the SDP method in [18].

4. Multistage FRM filter design

4.1. Design approach

The algorithm in Section 3 also applies to multistage FRM filters. For the sake of notation simplicity, in what follows we focus our attention on the two-stages case

shown in Fig. 5. The proposed approach can be extended for the design of FRM filters with more than two stages.

Since the sub-filters involved have linear-phase responses, they will be treated as zero-phase filters and the frequency response of the two-stage FRM filter can be expressed as

$$\boldsymbol{H} = \boldsymbol{H}_a^{(1)} .^*(\boldsymbol{a}_{1ma}^{\mathrm{T}}\boldsymbol{C}_{1ma} - \boldsymbol{a}_{1mc}^{\mathrm{T}}\boldsymbol{C}_{1mc}) + \boldsymbol{a}_{1mc}^{\mathrm{T}}\boldsymbol{C}_{1mc} \tag{13a}$$

where

$$\boldsymbol{H}_a^{(1)} = (\boldsymbol{a}_{2a}^{\mathrm{T}}\boldsymbol{C}_{2a}) .^*(\boldsymbol{a}_{2ma}^{\mathrm{T}}\boldsymbol{C}_{2ma} - \boldsymbol{a}_{2mc}^{\mathrm{T}}\boldsymbol{C}_{2mc}) + \boldsymbol{a}_{2mc}^{\mathrm{T}}\boldsymbol{C}_{2mc} \tag{13b}$$

The vectors $\boldsymbol{a}_{2a}$, $\boldsymbol{a}_{ima}$ and $\boldsymbol{a}_{imc}$ ($i = 1, 2$) in (13) are the coefficient vectors for filters $H_a^{(2)}(z)$, $H_{ma}^{(i)}(z)$ and $H_{mc}^{(i)}(z)$, respectively, and are defined in a way similar to vectors $\boldsymbol{a}_a$, $\boldsymbol{a}_{ma}$ and $\boldsymbol{a}_{mc}$ in (2), respectively. The cosine matrixes for the first-stage masking filters, namely $\boldsymbol{C}_{1ma}$ and $\boldsymbol{C}_{1mc}$, are

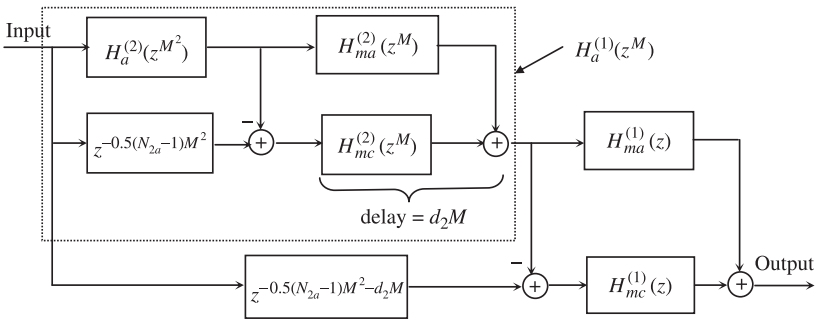

Fig. 5. Two-stage FRM filter structure.

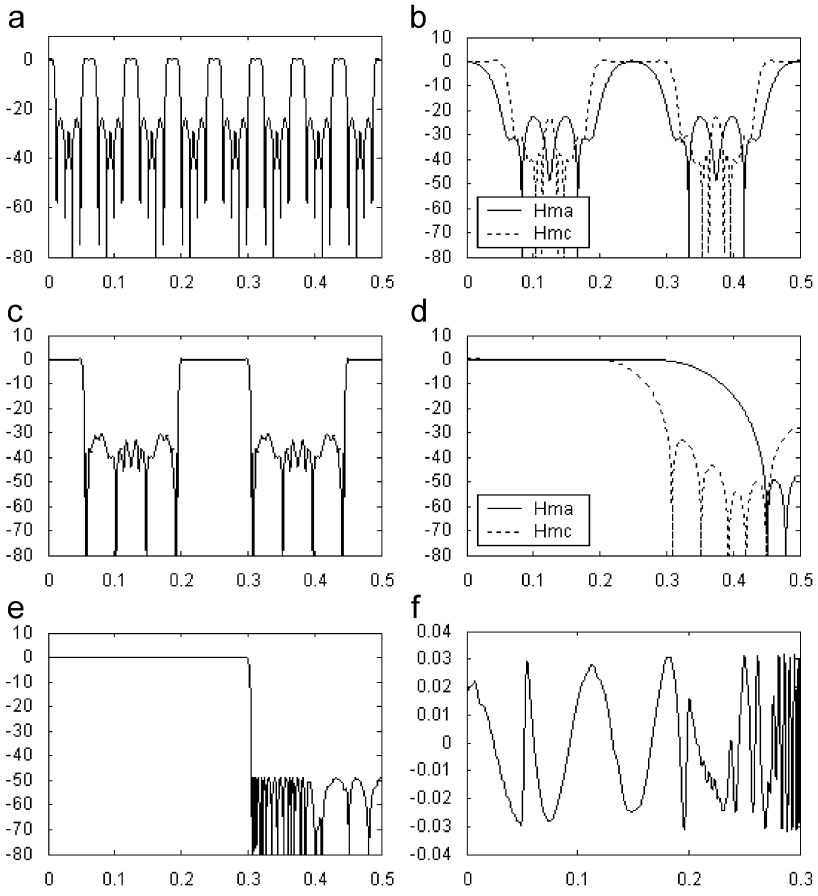

Fig. 6. Amplitude responses of (a) prototype filter $H_a^{(2)}(z^{16})$, (b) masking filters $H_{ma}^{(2)}(z^4)$ and $H_{mc}^{(2)}(z^4)$, (c) "prototype" filter $H_a^{(1)}(z^4)$, (d) masking filters $H_{ma}^{(1)}(z)$ and $H_{mc}^{(1)}(z)$, (e) FRM filter, and (f) passband ripples of the FRM filter; all in decibels.

defined in the same way as $\boldsymbol{C}_{ma}$ and $\boldsymbol{C}_{mc}$ in (2); and the cosine matrixes for the second-stage sub-filters, i.e., $\boldsymbol{C}_{2a}$, $\boldsymbol{C}_{2ma}$ and $\boldsymbol{C}_{2mc}$ are defined in a way similar to $\boldsymbol{C}_a$, $\boldsymbol{C}_{ma}$ and $\boldsymbol{C}_{mc}$ in (2) except that each ω is replaced by $M\omega$. The group delay of the FRM filter is given by [18]

$$D = \frac{(N_{2a}-1)M^2}{2} + d_2 M + d_1 \tag{14}$$

where N_{2a} is the length of $H_a^{(2)}(z)$ and $d_i = \max\{(N_{ima}-1)/2, (N_{imc}-1)/2\}$ for $i = 1, 2$ with N_{ima} and N_{imc} being the lengths of the masking filters $H_{ma}^{(i)}(z)$ and $H_{mc}^{(i)}(z)$, respectively. Define the error function $\boldsymbol{e}$ and performance index J in a way similar to (6) and (7), respectively. Then $\boldsymbol{a}$ is updated as

$$\boldsymbol{a}_{2a(k+1)} = \boldsymbol{a}_{2a(k)} + \eta \boldsymbol{C}_{2a}(\boldsymbol{W}^{\mathrm{T}} .* \boldsymbol{B}_k^{\mathrm{T}} .* \boldsymbol{e}_k^{\mathrm{T}} .* \boldsymbol{z}_{1(k)}^{\mathrm{T}} .* \boldsymbol{z}_{2(k)}^{\mathrm{T}}) \tag{15a}$$

$$\boldsymbol{a}_{2ma(k+1)} = \boldsymbol{a}_{2ma(k)} + \eta \boldsymbol{C}_{2ma}(\boldsymbol{W}^{\mathrm{T}} .* \boldsymbol{B}_k^{\mathrm{T}} .* \boldsymbol{e}_k^{\mathrm{T}} .* \boldsymbol{z}_{1(k)}^{\mathrm{T}} .* \boldsymbol{z}_{3(k)}^{\mathrm{T}}) \tag{15b}$$

$$\boldsymbol{a}_{2mc(k+1)} = \boldsymbol{a}_{2mc(k)} + \eta \boldsymbol{C}_{2mc}[\boldsymbol{W}^{\mathrm{T}} .* \boldsymbol{B}_k^{\mathrm{T}} .* \boldsymbol{e}_k^{\mathrm{T}} .* \boldsymbol{z}_{1(k)}^{\mathrm{T}} .* (1 - \boldsymbol{z}_{3(k)})^{\mathrm{T}}] \tag{15c}$$

$$\boldsymbol{a}_{1ma(k+1)} = \boldsymbol{a}_{1ma(k)} + \eta \boldsymbol{C}_{1ma}(\boldsymbol{W}^{\mathrm{T}} .* \boldsymbol{B}_k^{\mathrm{T}} .* \boldsymbol{e}_k^{\mathrm{T}} .* \boldsymbol{z}_k^{\mathrm{T}}) \tag{15d}$$

$$\boldsymbol{a}_{1mc(k+1)} = \boldsymbol{a}_{1mc(k)} + \eta \boldsymbol{C}_{1mc}[\boldsymbol{W}^{\mathrm{T}} .* \boldsymbol{B}_k^{\mathrm{T}} .* \boldsymbol{e}_k^{\mathrm{T}} .* (1 - \boldsymbol{z}_k)^{\mathrm{T}}] \tag{15e}$$

where

$$\boldsymbol{z}_{1(k)} = \boldsymbol{a}_{1ma(k)}^{\mathrm{T}} \boldsymbol{C}_{1ma} - \boldsymbol{a}_{1mc(k)}^{\mathrm{T}} \boldsymbol{C}_{1mc} \tag{15f}$$

$$\boldsymbol{z}_{2(k)} = \boldsymbol{a}_{2ma(k)}^{\mathrm{T}} \boldsymbol{C}_{2ma} - \boldsymbol{a}_{2mc(k)}^{\mathrm{T}} \boldsymbol{C}_{2mc} \tag{15g}$$

$$\boldsymbol{z}_{3(k)} = \boldsymbol{a}_{2a(k)}^{\mathrm{T}} \boldsymbol{C}_{2a} \tag{15h}$$

$$\boldsymbol{z}_{(k)} = \boldsymbol{z}_{2(k)} .* \boldsymbol{z}_{3(k)} + \boldsymbol{a}_{2mc(k)}^{\mathrm{T}} \boldsymbol{C}_{2mc} \tag{15i}$$

The design procedure is similar to that in Section 3.2, except for some variety in initial design [18]:

(1) Use parameters M, N_{1ma}, N_{1mc}, ω_{p} and ω_a to obtain an initial design of $H_{ma}^{(1)}(z)$ and $H_{mc}^{(1)}(z)$.
(2) Use parameters M, ω_{p} and ω_a to obtain the values of θ and ϕ as the passband and stopband edges for $H_a^{(1)}(z)$, denote $\hat{\omega}_{\mathrm{p}} = \theta$ and $\hat{\omega}_a = \phi$, then use parameters M, N_{2a}, N_{2ma}, N_{2mc}, $\hat{\omega}_{\mathrm{p}}$ and $\hat{\omega}_a$ to obtain an initial design of $H_a^{(2)}(z)$, $H_{ma}^{(2)}(z)$ and $H_{mc}^{(2)}(z)$. The coefficients of $H_a^{(2)}(z)$, $H_{ma}^{(2)}(z)$, $H_{mc}^{(2)}(z)$, $H_{ma}^{(1)}(z)$ and $H_{mc}^{(1)}(z)$ are now used to form the initial parameter vectors of $\boldsymbol{a}_{2a}$, $\boldsymbol{a}_{2ma}$, $\boldsymbol{a}_{2mc}$, $\boldsymbol{a}_{1ma}$ and $\boldsymbol{a}_{1mc}$, respectively.

4.2. Design example

Example 3. For illustration purposes, we applied the optimization approach to design a two-stage FRM filter with the same design parameters as the 3rd example in [18], i.e., $M = 4$, $\omega_{\mathrm{p}} = 0.6\pi$, and $\omega_a = 0.61\pi$. Set $N_{2a} = 27$, $N_{2ma} = 11$, $N_{2mc} = 23$, $N_{1ma} = 13$, $N_{1mc} = 21$, $\delta_{\mathrm{p}}/\delta_{\mathrm{s}} = 1$, $L = 1000$, $\varepsilon = 0.0032$. It takes 3800 training iterations (the computer used 5.4 s on a CPU of Pentium4) to converge. The amplitude responses of the various sub-filters are shown in Fig. 6. The maximum passband ripple is found to be 0.0318 dB and the minimum stopband attenuation is 48.81 dB. Comparing with literature [18] which set $N_{2a} = 27$, $N_{2ma} = 13$, $N_{2mc} = 27$, $N_{1ma} = 15$, $N_{1mc} = 23$, we can find that the passband ripple and stopband attenuation are 0.0320 and 48.77 dB, respectively. Finally, we remark that although the design resulted in a slightly more economical FRM filter compared to the one in Example 1, the two-stage filter offers better performance in both passband and stopband as is evident from Fig. 6(e) and (f). Nevertheless, the performance gain is achieved at the cost of an increased group delay $D = 262$ compared to the group delay $D = 207$ for the filter in Example 1.

5. Conclusion

In this paper, a two-step optimization technique has been proposed for the design of various frequency-response-masking (FRM) based linear-phase finite-impulse response (FIR) digital filters. An initial FRM filter is generated at first step by alternately optimizing the sub-filters with the aid of a simple standard algorithm. At the second step, this start-up design is further improved by optimizing the coefficients of overall sub-filters simultaneously using the proposed neural network algorithm. Since the coefficients of overall sub-filters are optimized simultaneously, as a result, the joint optimization leads to improve design performance, and is readily able to handle designs of high order, multistage FRM filters with linear-phase responses. Because the batch back-propagation neural network algorithm, which utilizes a variable learning rate mode, is taken at the second step, the proposed approach can speed up the convergence. Some examples taken from the literatures have shown that the proposed algorithm can design better FRM filters compared to the several existing methods.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant no. 50677014, Doctoral Special Fund of Ministry of Education under Grant no. 20060532002, High-Tech Research and Development Program of China (No. 2006AA04A104), the Program for New Century Excellent Talents in University of China (NCET-04-0767), and the Foundation of Hunan Provincial Science and Technology of China under Grant no. 07JJ5076.

Appendix

Tables 1–11 in this appendix give the optimized impulse-response values for the proposed approach.

Table 1
Impulse-response values of the filter $H_a(z)$ in example 1 ($h_a(43-1-n)=h_a(n)$ for $n=0,1,\ldots,21$)

$h_a(0)=-0.00519487416179$	$h_a(8)=-0.01019336140953$	$h_a(16)=0.04331695834245$
$h_a(1)=-0.00456216886928$	$h_a(9)=0.01696817861631$	$h_a(17)=0.05378076812372$
$h_a(2)=0.00690931463271$	$h_a(10)=0.00769685306253$	$h_a(18)=-0.09395535307820$
$h_a(3)=-0.00212450888305$	$h_a(11)=-0.02681668889568$	$h_a(19)=-0.04683698513242$
$h_a(4)=-0.00770383951159$	$h_a(12)=0.00075179350460$	$h_a(20)=0.30504851044377$
$h_a(5)=0.00277494831138$	$h_a(13)=0.03661364468452$	$h_a(21)=0.56535091289033$
$h_a(6)=0.01094375624068$	$h_a(14)=-0.01537288759048$	
$h_a(7)=-0.01005220819436$	$h_a(15)=-0.04600342020476$	

Table 2
Impulse-response values of the filter $H_{ma}(z)$ in example 1 ($h_{ma}(37-1-n)=h_{ma}(n)$ for $n=0,1,\ldots,18$)

$h_{ma}(0)=0.00108229299255$	$h_{ma}(7)=0.00735654275290$	$h_{ma}(14)=0.05137933238313$
$h_{ma}(1)=-0.01033243723488$	$h_{ma}(8)=-0.01651606728706$	$h_{ma}(15)=-0.08389553158362$
$h_{ma}(2)=0.00060689950941$	$h_{ma}(9)=-0.00438566271966$	$h_{ma}(16)=-0.05692095942856$
$h_{ma}(3)=0.00848462744838$	$h_{ma}(10)=0.03659615659615$	$h_{ma}(17)=0.30600839862421$
$h_{ma}(4)=-0.00419627180678$	$h_{ma}(11)=-0.00966926101868$	$h_{ma}(18)=0.56156570240304$
$h_{ma}(5)=-0.00864603391386$	$h_{ma}(12)=-0.04396374471038$	
$h_{ma}(6)=0.00938834425024$	$h_{ma}(13)=0.03340922935613$	

Table 3
Impulse-response values of the filter $H_{mc}(z)$ in example 1 $h_{mc}(29-1-n)=h_{mc}(n)$ for $n=0,1,\ldots,14$)

$h_{mc}(0)=0.00034930814077$	$h_{mc}(5)=-0.00521529895702$	$h_{mc}(10)=0.07344202524222$
$h_{mc}(1)=0.01307635879031$	$h_{mc}(6)=-0.01948392394483$	$h_{mc}(11)=-0.00921979022659$
$h_{mc}(2)=-0.00589076148002$	$h_{mc}(7)=0.02958060643368$	$h_{mc}(12)=-0.13485852003937$
$h_{mc}(3)=-0.00998361020924$	$h_{mc}(8)=-0.00724066572077$	$h_{mc}(13)=0.27942116750101$
$h_{mc}(4)=0.01776810804508$	$h_{mc}(9)=-0.03913551117702$	$h_{mc}(14)=0.66032345081668$

Table 4
Impulse-response values of the filter $H_a(z)$ in example 2 $h_a(99-1-n)=h_a(n)$ for $n=0,1,\ldots,49$)

$h_a(0)=0.00016727541415$	$h_a(17)=-0.00455509374660$	$h_a(34)=0.01711110048710$
$h_a(1)=-0.00104013827231$	$h_a(18)=0.00200543074078$	$h_a(35)=-0.01260785995375$
$h_a(2)=-0.00118995269759$	$h_a(19)=0.00704509647153$	$h_a(36)=-0.02757010929494$
$h_a(3)=0.00000253698952$	$h_a(20)=0.00204686043078$	$h_a(37)=-0.00030513761847$
$h_a(4)=0.00143724345505$	$h_a(21)=-0.00691271027929$	$h_a(38)=0.03375345317638$
$h_a(5)=0.00098801335159$	$h_a(22)=-0.00615667259855$	$h_a(39)=0.01974381854680$
$h_a(6)=-0.00148271593384$	$h_a(23)=0.00486723370670$	$h_a(40)=-0.03087172369375$
$h_a(7)=-0.00274702814218$	$h_a(24)=0.01017462893769$	$h_a(41)=-0.04343205941143$
$h_a(8)=-0.00056280554980$	$h_a(25)=0.00023310284906$	$h_a(42)=0.01521812087143$
$h_a(9)=0.00269597069287$	$h_a(26)=-0.01170704877388$	$h_a(43)=0.06828632546260$
$h_a(10)=0.00232863746624$	$h_a(27)=-0.00674765548646$	$h_a(44)=0.02158369778272$
$h_a(11)=-0.00162280546926$	$h_a(28)=0.01003024120100$	$h_a(45)=-0.08828728581935$
$h_a(12)=-0.00342595529091$	$h_a(29)=0.01366208916705$	$h_a(46)=-0.09513854972136$
$h_a(13)=0.00007701964968$	$h_a(30)=-0.00407868022859$	$h_a(47)=0.08895200802375$
$h_a(14)=0.00436446809560$	$h_a(31)=-0.01836872652657$	$h_a(48)=0.28266313636000$
$h_a(15)=0.00255205655434$	$h_a(32)=-0.00551533200328$	$h_a(49)=0.43450154779924$
$h_a(16)=-0.00338948313883$	$h_a(33)=0.01862169636450$	

Table 5
Impulse-response values of the filter $H_{ma}(z)$ in example 2($h_{ma}(69-1-n)=h_{ma}(n)$ for $n=0,1,\ldots,34$)

$h_{ma}(0)=-0.00323181307881$	$h_{ma}(12)=-0.00472480039567$	$h_{ma}(24)=0.01378850629414$
$h_{ma}(1)=-0.00701466936395$	$h_{ma}(13)=0.00599997736100$	$h_{ma}(25)=-0.02912076282393$
$h_{ma}(2)=-0.00444236194810$	$h_{ma}(14)=0.00996423119316$	$h_{ma}(26)=-0.03872275687506$
$h_{ma}(3)=0.00160363043830$	$h_{ma}(15)=-0.00097849066507$	$h_{ma}(27)=0.00732618383436$
$h_{ma}(4)=0.00687856097651$	$h_{ma}(16)=-0.01224189461046$	$h_{ma}(28)=0.05144984169580$
$h_{ma}(5)=0.00324156586138$	$h_{ma}(17)=-0.00621333502890$	$h_{ma}(29)=0.02000361660376$
$h_{ma}(6)=-0.00330712039253$	$h_{ma}(18)=0.01054277566705$	$h_{ma}(30)=-0.06552150986586$

Table 5. (*continued*)

$h_{ma}(7) = -0.00389720147805$	$h_{ma}(19) = 0.01286674504562$	$h_{ma}(31) = -0.07650339445844$
$h_{ma}(8) = 0.00211730094772$	$h_{ma}(20) = -0.00571050794682$	$h_{ma}(32) = 0.07618206912062$
$h_{ma}(9) = 0.00595103765993$	$h_{ma}(21) = -0.01423238803916$	$h_{ma}(33) = 0.30734301856859$
$h_{ma}(10) = 0.00079589323555$	$h_{ma}(22) = 0.00654953477520$	$h_{ma}(34) = 0.41927037883698$
$h_{ma}(11) = -0.00685486665541$	$h_{ma}(23) = 0.02725647517501$	

Table 6
Impulse-response values of the filter $H_{mc}(z)$ in example 2($h_{mc}(85-1-n) = h_{mc}(n)$ for $n = 0, 1,\ldots,42$)

$h_{mc}(0) = -0.00152439523700$	$h_{mc}(15) = 0.00676413502646$	$h_{mc}(30) = 0.02258633340671$
$h_{mc}(1) = -0.00255505201900$	$h_{mc}(16) = 0.00184616356698$	$h_{mc}(31) = 0.01966772250299$
$h_{mc}(2) = -0.00292245182741$	$h_{mc}(17) = -0.00658817558909$	$h_{mc}(32) = -0.01197257886759$
$h_{mc}(3) = -0.00112811920503$	$h_{mc}(18) = -0.00687039040737$	$h_{mc}(33) = -0.03280630043641$
$h_{mc}(4) = 0.00136992603166$	$h_{mc}(19) = 0.00208764694449$	$h_{mc}(34) = -0.00966668594835$
$h_{mc}(5) = 0.00189245296672$	$h_{mc}(20) = 0.00895927026607$	$h_{mc}(35) = 0.03632471081624$
$h_{mc}(6) = -0.00036395547750$	$h_{mc}(21) = 0.00180085566489$	$h_{mc}(36) = 0.04368708430392$
$h_{mc}(7) = -0.00279288014539$	$h_{mc}(22) = -0.01086132957458$	$h_{mc}(37) = -0.01405006369558$
$h_{mc}(8) = -0.00129672124074$	$h_{mc}(23) = -0.01309644717076$	$h_{mc}(38) = -0.07815503702838$
$h_{mc}(9) = 0.00325272191123$	$h_{mc}(24) = 0.00171743334999$	$h_{mc}(39) = -0.04972486584857$
$h_{mc}(10) = 0.00486566486298$	$h_{mc}(25) = 0.01558413895038$	$h_{mc}(40) = 0.10370723607640$
$h_{mc}(11) = 0.00083621918302$	$h_{mc}(26) = 0.00963199574147$	$h_{mc}(41) = 0.29574653448209$
$h_{mc}(12) = -0.00475046253520$	$h_{mc}(27) = -0.01105703968130$	$h_{mc}(42) = 0.38213538017950$
$h_{mc}(13) = -0.00409035870432$	$h_{mc}(28) = -0.01959940705002$	
$h_{mc}(14) = 0.00272345701565$	$h_{mc}(29) = -0.00159375539320$	

Table 7
Impulse-response values of the filter $H_a^{(2)}(z)$ in example 3($h_a(27-1-n) = h_a(n)$ for $n = 0, 1,\ldots,13$)

$h_a(0) = 0.00215334952988$	$h_a(5) = 0.02559999082384$	$h_a(10) = 0.02801109270316$
$h_a(1) = -0.00563362628528$	$h_a(6) = 0.02121228917298$	$h_a(11) = 0.13826624427797$
$h_a(2) = -0.00978963469134$	$h_a(7) = -0.01609417385825$	$h_a(12) = 0.27861128039942$
$h_a(3) = -0.00795415219051$	$h_a(8) = -0.06093956197449$	$h_a(13) = 0.35120155799912$
$h_a(4) = 0.00841518613334$	$h_a(9) = -0.05935182201881$	

Table 8
Impulse-response values of the filter $H_{ma}^{(2)}(z)$ in example 3($h_{ma}(11-1-n) = h_{ma}(n)$ for $n = 0, 1,\ldots,5$)

$h_{ma}(0) = -0.02379804525796$	$h_{ma}(2) = 0.03202327961072$	$h_{ma}(4) = 0.23867142377061$
$h_{ma}(1) = -0.00254150530486$	$h_{ma}(3) = 0.11603704005915$	$h_{ma}(5) = 0.26313142326784$

Table 9
Impulse-response values of the filter $H_{mc}^{(2)}(z)$ in example 3($h_{mc}(23-1-n) = h_{mc}(n)$ for $n = 0, 1,\ldots,11$)

$h_{mc}(0) = -0.00213197201466$	$h_{mc}(4) = -0.02498028963524$	$h_{mc}(8) = -0.09867591938146$
$h_{mc}(1) = 0.01270235949775$	$h_{mc}(5) = 0.02799070289236$	$h_{mc}(9) = 0.01806759200015$
$h_{mc}(2) = 0.00733231243429$	$h_{mc}(6) = 0.04850058651147$	$h_{mc}(10) = 0.31022737380513$
$h_{mc}(3) = -0.01186831662473$	$h_{mc}(7) = -0.01473173001110$	$h_{mc}(11) = 0.49051315091856$

Table 10
Impulse-response values of the filter $H_{ma}^{(1)}(z)$ in example 3($h_{ma}(13-1-n) = h_{ma}(n)$ for $n = 0, 1,\ldots, 6$)

$h_{ma}(0) = 0.01615809921621$	$h_{ma}(3) = 0.04254963371348$	$h_{ma}(6) = 0.73352923122855$
$h_{ma}(1) = -0.02882876487731$	$h_{ma}(4) = -0.14322060329209$	
$h_{ma}(2) = 0.01171383173868$	$h_{ma}(5) = 0.23551985351120$	

Table 11
Impulse-response values of the filter $H_{mc}^{(1)}(z)$ in example 3($h_{mc}(21-1-n) = h_{mc}(n)$ for $n = 0, 1, \ldots, 10$)

$h_{mc}(0) = -0.00327071868033$	$h_{mc}(4) = -0.00860238590092$	$h_{mc}(8) = -0.00999168268147$
$h_{mc}(1) = 0.01044526526407$	$h_{mc}(5) = 0.04931841806332$	$h_{mc}(9) = 0.31950606798212$
$h_{mc}(2) = 0.00680134104499$	$h_{mc}(6) = 0.01081928443769$	$h_{mc}(10) = 0.51215043686825$
$h_{mc}(3) = -0.01794307507701$	$h_{mc}(7) = -0.08944395693150$	

References

[1] Y.J. Yu, K.L. Teo, Y.C. Lim, G.H. Zhao, Extrapolated impulse response filter and its application in the synthesis of digital filters using the frequency-response masking technique, Signal Processing 50 (March 2005) 581–590.

[2] L. Cen, Y. Lian, Hybird genetic algorithm for the design of modified frequency-response masking filters in discrete space, Circuits Systems Signal Process. 25 (2) (April 2006) 153–174.

[3] Y. Lian, The optimal design of half band filter using multi-stage frequency-response masking technique, Signal Processing 44 (July 1995) 369–372.

[4] M.B. Furtado, P.S.R. Diniz, S.L. Netto, Optimized prototype filter based on the FRM approach for consine-modulated filter bank, Circuits Systems Signal Process. 22 (2) (March/April 2003) 193–210.

[5] Y.C. Lim, R. Yang, On the synthesis of very sharp decimators and interpolators using the frequency-response masking technique, IEEE Trans. Signal Process. 53 (April 2005) 1387–1397.

[6] Y.C. Lim, Y.J. Yu, Synthesis of very sharp Hilbert transformer using the frequency-response masking technique, IEEE Trans. Signal Process. 53 (July 2005) 2595–2597.

[7] Y.C. Lim, Y.J. Yu, T. Saramäki, Optimum masking levels and coefficient sparseness for Hilbert transformer and half band filters designed using the frequency response masking technique, IEEE Trans. Circuits Systems I 52 (November 2005) 2444–2453.

[8] Y.C. Lim, Y.J. Yu, T. Saramäki, K.L. Teo, FRM based FIR filters with optimum finite word-length performance, IEEE Trans. Signal Process. 55 (June 2007) 2914–2924.

[9] Y.C. Lim, Frequency-response-masking approach for the synthesis of sharp linear phase digital filters, IEEE Trans. Circuits Systems CAS-33 (April 1986) 357–364.

[10] Y.C. Lim, Y. Lian, The optimum design of one- and two-dimensional FIR filters using the frequency response masking technique, IEEE Trans. Circuits Systems II 40 (February 1993) 88–95.

[11] T. Saramäki, Y.C. Lim, Use of Remez algorithm for designing FRM based FIR filters, Circuits Systems Signal Process. 22 (2) (March/April 2003) 77–97.

[12] Y. Lian, L. Zhang, C.C. Ko, An improve frequency response masking approach for designing sharp FIR filters, Signal Processing 81 (December 2001) 2573–2581.

[13] Y.C. Lim, Y. Lian, Frequency-response-masking approach for digital filter design: Complexity reduction via masking filters factorization, IEEE Tran. Circuits Systems II 41 (August 1994) 518–525.

[14] T. Saramäki, J.Y. Kaakinen, H. Johansson, Optimization of frequency-response masking based FIR filters, J. Circuits Systems Comput. 12 (May 2003) 563–591.

[15] W.S. Lu, T. Hinamoto, Optimal design of FIR frequency-response-masking filters using second-order cone programming, in: Proceedings of the IEEE International Symposium on Circuits and Systems, vol. III, Bangkok, Thailand, May 2003, pp. 878–881.

[16] W.R. Lee, V. Rehbock, K.L. Teo, A weight least-square-based approach to FIR filter design using the frequency- response masking technique, IEEE Signal Process. Lett. 11 (July 2004) 593–596.

[17] W.R. Lee, V. Rehbock, K.L. Teo, L. Caccetta, A weight least-square-based approach to FIR filters synthesized using the modified frequency response masking structure, IEEE Trans. Circuits Systems II 53 (May 2006) 379–383.

[18] W.S. Lu, T. Hinamoto, Optimal design of frequency-response - masking filters using semidefinite programming, IEEE Trans. Circuits Systems I 50 (April 2003) 557–568.

[19] Y.J. Yu, Y.C. Lim, Genetic algorithm approach for the optimization of multiplierless sub-filters generated by the frequency-response masking technique, in: Proceedings of the IEEE International Conference on Electronics Circuits and Systems, vol. III, Dubrovnik, Croatia, September 2002, pp. 1163–1166.