A Genetic-Based Algorithm for the Design of Multiplierless Halfband IIR Filters

Valentina Markova [1], Juha Yli-Kaakinen [2], and Tapio Saramäki [3]

Abstract – **This paper describes an efficient algorithm for designing multiplierless halfband IIR digital filters. The coefficient optimization is performed in two steps. The proposed procedure gives finite-precision solutions requiring fewer computations than previously reported implementations.**

Keywords – **IIR digital filters, halfband filters, multiplierless filters, genetic algorithms.**

I. INTRODUCTION

Halfband filters are important for applications to multirate digital signal processing. A halfband filter satisfies the passband and stopband symmetry conditions [1], [2]. Both infinite- impulse response (IIR) and finite-impulse response (FIR) filters can be used to realize these filters [3]. For applications requiring exact linear phase, FIR filters are used [4]. However, when the phase requirement is not as strict the IIR filters are the best choice as they require considerably fewer coefficients to meet the given magnitude criteria than their FIR filter counterpart [5], [6].

Odd-order IIR halfband filters can be implemented as a parallel connection of two allpass filters. These filters have turned out to be very efficient for constructing filter banks since all the computations can be performed at the lower sampling rate. Furthermore, this filter class is characterized by low coefficient sensitivity. The importance of such a structure is that if the effect of coefficient value deviation from the ideal value is small, then the short coefficient wordlength can be used with only slightly violating the infinite-precision filter specifications, resulting in a faster, smaller and less expensive hardware [7], [8].

In this paper an algorithm for designing elliptic halfband IIR digital filters with short coefficient wordlength is introduced. This algorithm is based on the following observation: Finding two elliptic halfband filters, one of which has the minimized transition bandwidth and the second one the maximized stopband attenuation such that the given criteria are still met enables one to generate a parameter space including the feasible space where the filter specifications are satisfied. After determining this larger space, all what is needed is to check whether in this space there exist the desired discrete values for the coefficient representations. In order to reduce the computational complexity, a genetic algorithm is applied for finding the solutions meeting the specifications within the parameter space. This strategy is general but particularly efficient for filters implemented as a parallel connection of two allpass filters due to the fact that for these filters only the denominator coefficients of the allpass sections have to be quantized. Furthermore, these coefficients are represented as simple combinations of powers of two, thereby provide a low complexity halfband filter. In this paper, we suggest the multiplierless design with considerably smaller implementation cost of the filter than previously reported realizations.

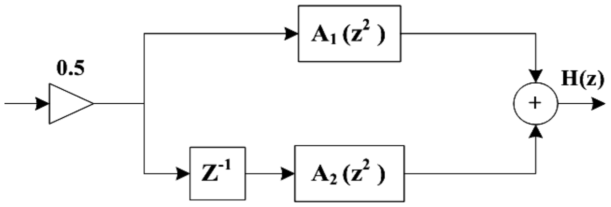

Fig 1. Realization of an odd-order halfband IIR filter.

II. PROPERTIES OF ODD-ORDER HALFBAND ELLIPTIC IIR FILTERS

It is well-known that odd-order lowpass halfband elliptic IIR filters are characterized by the following properties and restrictions (see, e.g., [2]). First, the design criteria for these filters should be stated as

$$\begin{aligned} 1-\delta_p \le \left|H(\Phi, e^{j\omega})\right| \le 1 \quad \text{for} \quad \omega\in\left[0,\omega_p\right] \\ \left|H(\Phi, e^{j\omega})\right| \le \delta_s \quad \text{for} \quad \omega\in\left[\omega_s,\pi\right], \end{aligned} \tag{1}$$

where ω_p and ω_s are restricted to be related through

$$\omega_p+\omega_s=\pi, \tag{2}$$

whereas the restrictions between δ_p and δ_s are expressible in terms of $A_p=-\log_{10}(1-(\delta_p)^2)$ and $A_s=-\log_{10}((\delta_p)^2)$, which are called as the passband variation in decibels and the stopband attenuation, respectively, as

$$A_p = 10\log_{10}\left(1+\frac{1}{10^{A_s/10}-1}\right) \tag{3}$$

or

$$A_s = -\log_{10}(1-10^{-A_p/10}). \tag{4}$$

Secondly, due to the fact the poles of the filter are restricted lie on the imaginary axis for a real-valued and stable transfer function, this function can be written as a parallel connection of two all-pass filters, according to Fig. 1, as

[1] Valentina I. Markova is with the Faculty of Telecommunications at Technical University of Varna, 1 Studentska Str, Varna 9010, Bulgaria, E-mail: via@tu-sofia.bg.

[2] Juha Yli-Kaakinen was with the Department of Signal Processing at Tampere University of Technology, Finland He is now with VLSI Solution Oy, FIN-33720 Tampere, Finland, E-mail: juha.ylikaakinen@pp.inet.fi.

[3] Tapio Saramäki is with the Department of Signal Processing at Tampere University of Technology, P.O. Box 553, FIN-33101 Tampere, Finland, E-mail: ts@ cs.tut.fi.

$$H(z)=\frac{1}{2}\left[A_1(z^2)+z^{-1}A_2(z^2)\right], \tag{5}$$

where $A_1(z)$ and $A_2(z)$ can be expressed as

$$A_1(z)=\prod_{\ell=1}^{m}\frac{-c_\ell+z^{-1}}{1-c_\ell z^{-1}} \text{ and } A_2(z)=\prod_{\ell=m+1}^{m+n}\frac{-c_\ell+z^{-1}}{1-c_\ell z^{-1}}. \tag{6}$$

Because this contribution focuses on implementing $A_1(z)$ and $A_2(z)$ as a cascade of Gray-Markel lattice allpass sections [9], they are expressed in the above form. Equally well, the above form suits to be implemented as a lattice wave digital filter [6].

Thirdly, when considering Eqs. (5) and (6), the following attractive observations are made: 1) The second allpass filter $z^{-1}A_2(z^2)$ of $H(z)$ has a first-order section that is a pure delay. 2) Both allpass filters contain second-order terms that are obtained from a first-order sections in Eq. (6) by replacing z^{-1} by z^{-2}. These observations lead to the transfer function of order 2(m+n)+1, but it requires only m+n multipliers. The orders of the first and second allpass filters are $M=2n$ and $N=2m+1$, which should differ by one. This implies that either $n=m$ or $n=m+1$.

III. Statement of the Problem

Before stating the optimization problem, the transfer function of the halfband filter is denoted for later use by $H(\Phi, z)$, where Φ is the vector containing the adjustable filter parameters c_ℓ for $\ell=1,2,\ldots,m+n$. According to the discussion of Section II, given the minimum stopband attenuation A_s and the stopband edge angle ω_s, the magnitude specifications for the odd-order halfband elliptic filter are given as

$$\begin{aligned}0\le -20\log_{10}\left|H(\Phi,e^{j\omega})\right| &\le -10\log_{10}\left(1+\frac{1}{10^{A_s/10}-1}\right)\\ &\text{for}\quad \omega\in[0,\pi-\omega_s]\\ -20\log_{10}\left|H(\Phi,e^{j\omega})\right|\ge A_s &\quad\text{for}\quad \omega\in[\omega_s,\pi]\,.\end{aligned} \tag{7}$$

This work concentrates on coefficients quantization in fixed-point arithmetic. In VLSI implementations, where general multipliers are very costly, it is attractive to carry out the multiplication of a data sample by a filter coefficient value using a sequence of adds (subtracts) and shifts. For such a purpose, the coefficient values are expressed as

$$\sum_{r=1}^{R}a_r 2^{-P_r}\ , \tag{8}$$

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 aim is to find all the coefficient values so that, first, R, the number of powers-of-two terms, is made as small as possible, and, secondly, P_r, the maximum number of shifts, is made as small as possible.

An estimate for the implementation cost of the filter can be calculated as a sum of the number of the adders and subtracters used to implement all the filter coefficients, that is, the cost is given by

$$\sum\nolimits_{l=1}^{m+n}\sigma_\ell\ , \tag{9}$$

where the σ_ℓ's are the number of adders and subtracters required to implement the filter coefficients c_ℓ.

The optimization problem under consideration is the following:

Optimization Problem: Given A_s and ω_s, find m and n, and the adjustable parameter vector Φ such that, first, the criteria of Eq. (7) are met after quantizing the coefficient values 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. (9), is minimized.

IV. Filter Optimization

The solution to the stated optimization problem can be found in the following two-step procedure. In the first step, the stopband edge angle is minimized such that the resulting stopband attenuation achieves just the specified value in Eq. (7) and the stopband attenuation is maximized such that the resulting stopband edge angle achieves just the specified value in Eq. (7). This enables one to find the parameter space of the infinite-precision coefficients including the feasible space where the filter meets the requirements. The second step involves finding the filter parameters in this space using a genetic algorithm such that the resulting filter meets the given criteria with the simplest coefficient representation forms.

A. Optimization of Infinite-Precision Filters

It has been turned out that the desired parameter space for the filter coefficients can be conveniently generated by designing two infinite-precision elliptic halfband filters, which satisfy the specifications as follows:

Design 1: The stopband edge angle is minimized in the criteria of Eq. 7 such that the stopband attenuation reaches just the specified value.

Design 2: The stopband attenuation is maximized in the criteria of Eq. 7 such the stopband edge angle reaches just the specified value.

These designs can be obtained by using simple closed-form algebraic expressions [3]. The parameter vectors containing the optimal infinite-precision filter parameters for Designs 1 and 2 are denoted by $\Phi^{(1)}$ and $\Phi^{(2)}$, respectively, whereas the corresponding sets of the coefficients values for the filter transfer function under consideration are denoted by $c_\ell^{(1)}$'s and $c_\ell^{(2)}$'s. Based on these set of coefficients, the smallest and largest values for the filter coefficients can be determined as

$$c_\ell^{(\min)}=\min\left\{c_\ell^{(1)},\,c_\ell^{(2)}\right\} \text{ and } c_l^{(\max)}=\max\left\{\,c_l^{(1)},\,c_l^{(2)}\right\} \tag{10}$$

for $\ell=1,2,\ldots,n+m$.

B. Optimization of Finite-Precision Filters

It has been experimentally proved that the parameter space defined above forms a very good approximation for the feasible space where the filter criteria are met. After finding this parameter space, all what is needed is to find in this space the existing combinations of the discrete coefficients values which satisfy the given requirements.

This search can be done in a straightforward manner [6], [7] by first finding the sets of powers-of-two numbers C_l for $\ell = 0, 1, \ldots, n+m$ between the smallest and largest values of each coefficient ($c_l^{(\min)}$ and $c_l^{(\max)}$).

The magnitude response is then estimated for each combination of the $c_l^{(k)}$ for $\ell = 1, 2, \ldots, n+m$ and $k = 0, 1, \ldots, K_\ell$ to check whether the filter meets the given criteria. Here, the number of powers-of-two values between $c_l^{(\min)}$ and $c_l^{(\max)}$ is denoted by K_ℓ, whereas the kth existing discrete value between these smallest and largest values is denoted by $C_l^{(k)}$ for $k = 0, 1, \ldots, K_\ell$. The number of discrete coefficient value combinations is thus given by

$$\prod_{\ell=1}^{n+m} K_\ell \ . \tag{11}$$

The number of discrete coefficient value combinations can be huge. For this reason it is beneficial to use a genetic algorithm for searching those discrete coefficient values with which the specifications are satisfied. This discrete-valued optimization problem can be efficiently solved by utilizing a genetic algorithm as follows: First, the indexes of the power-of-two numbers between the smallest and largest values of the coefficients are represented using a binary code. Furthermore, a lookup table containing the power-of-two values of the corresponding indexes is generated. The next step is to construct the chromosomes by concatenating all these binary strings. At the end, the fitness of the population is evaluated by decoding the chromosomes to their corresponding power-of-two coefficients values using the above-mentioned lookup table.

The fitness function to be maximized is given by

$$f = -\max\left\{\Delta_p / \delta_p, \Delta_s / \delta_s\right\} \ , \tag{12}$$

where Δ_p and Δ_s are the realized passband and stopband ripples, respectively. The solution meeting the given criteria is obtained when f becomes greater than or equal to minus unity.

V. Design Example

Consider the following criteria for an odd-order IIR halfband filter [8]: $\omega_p = 0.44\pi$ and $A_s = 46$ dB. The rest of the filter criteria are found using Eqs. (2) and (3), which give $\omega_s = 0.56\pi$ and $A_p = 1.1 \cdot 10^{-4}$ dB. These specifications are met by a ninth-order elliptic halfband IIR filter.

The infinite-precision coefficient values of the two elliptic halfband initial filter designs are given in Table 1. In this table, $\Phi^{(1)}$ and $\Phi^{(2)}$ are the optimal coefficient values for Designs 1 and 2, respectively. The corresponding magnitude responses for the initial filters are shown in Fig. 2.

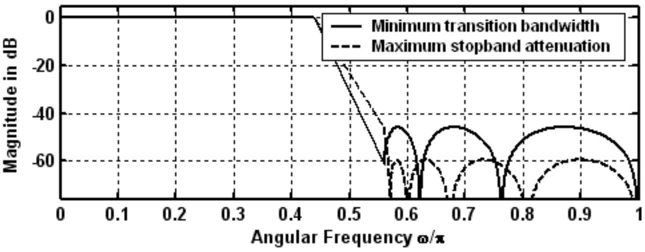

Fig. 2. Magnitude responses for the initial filters.

TABLE I
INFINITE-PRECISION COEFFICIENT VALUES FOR THE INITIAL DESIGNS

	$\Phi^{(1)} = \Phi^{(\min)}$	$\Phi^{(2)} = \Phi^{(\max)}$
c_1	−0.4653	−0.3440
c_2	−0.9204	−0.8686
c_3	−0.1535	−0.1022
c_4	−0.7342	−0.6145

For the corresponding finite-precision filter only three powers-of-two terms ($R = 3$) and eight fractional bits ($P_R = 8$) are required to fulfill the given criteria. The number of bits needed to encode all permissible discrete values between the smallest and largest values of c_ℓ for $\ell = 1, 2, \ldots, 4$ for this coefficient representation form are 5, 3, 4 and 4, respectively, that is, the length of the chromosome is 16. For more details of the above encoding, see [7].

TABLE II
PERFORMANCE OF THE GENETIC ALGORITHM

P_s=0.04				
P_x	P_m	f_{mean}	f_{std}	N_{hit}
0.6	0.04	−1.4871	0.4393	9
0.6	0.05	−1.1253	0.0882	18
0.6	0.06	−1.0929	0.1706	21
0.7	0.04	−1.2308	0.2788	11
0.7	0.05	−1.0834	0.1260	31
0.7	0.06	−1.0659	0.1129	26
0.8	0.04	−1.1977	0.1445	14
0.8	0.05	−1.1042	0.1723	28
0.8	0.06	−1.0820	0.1100	33
P_s = 0.045				
0.6	0.04	−1.2661	0.2876	10
0.6	0.05	−1.0956	0.183	21
0.6	0.06	−1.0674	0.1085	25
0.7	0.04	−1.2401	0.2764	11
0.7	0.05	−1.0705	0.0956	30
0.7	0.06	−1.0568	0.0873	35
0.8	0.04	−1.2035	0.2693	12
0.8	0.05	−1.0861	0.1189	26
0.8	0.06	−1.0603	0.1067	29

TABLE III
OPTIMIZED FINITE-PRECISION COEFFICIENTS VALUES OF THE ELLIPTIC HALFBAND IIR FILTER

$A_1(z)$		$A_2(z)$	
c_1	$-2^{-1}+2^{-3}$	c_3	$-2^{-3}+2^{-7}$
c_2	$-1+2^{-3}-2^{-8}$	c_4	$-2^{-1}-2^{-3}-2^{-6}$

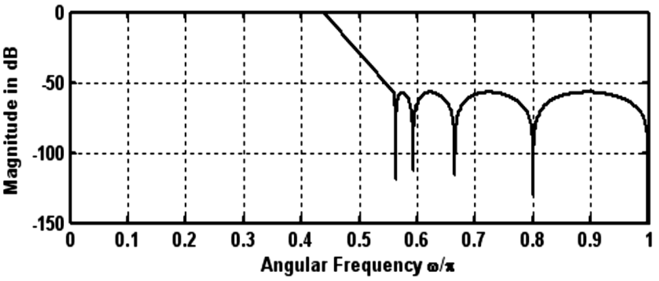

Fig. 3. Magnitude response of the resulting multiplierless halfband IIR filter.

The control parameters of the genetic algorithm such as crossover and mutation rates have been adjusted by, first, running this algorithm 100 times with different parameter settings and, then, selecting the most optimal ones. Some of them are shown in Table II. These results were obtained under the following circumstances. The normalized geometric selection was used as a reproduction operator, the population size was 150, and the number of generations was 250. In this table, P_s denotes the selection probability, P_x and P_m are the crossover and mutation rates, respectively, whereas f_{mean}, f_{std} and N_{hit} give, after these 100 runs, the mean fitness, the standard deviation of the fitness, and the number of solutions meeting the requirements, respectively. The fitness value of the best solution after these 100 runs was always the same that is, $f = -0.932$. The CPU time required for running one run was approximately 18 seconds when using the genetic algorithm optimization toolbox [10] in MATLAB 6.5 on a 3 GHz Pentium 4. The CPU-time required to evaluate all the possible coefficient value combinations was approximately 12 minutes.

For the optimized finite-precision filter in [8], four powers-of-two terms ($R = 4$) and eight fractional bits ($P_R = 8$) are required to meet the given filter specifications and the number of adders and/or subtracters needed to implement all the coefficients for the Gray-Markel sections is eight. The proposed algorithm results in the optimized filter, which requires only six adders and/or subtracters to implement all the multipliers. The optimized coefficient values are shown in Table III. The magnitude response of the resulting multiplierless halfband IIR filter is displayed in Fig. 3. The corresponding realization is shown in Fig. 4.

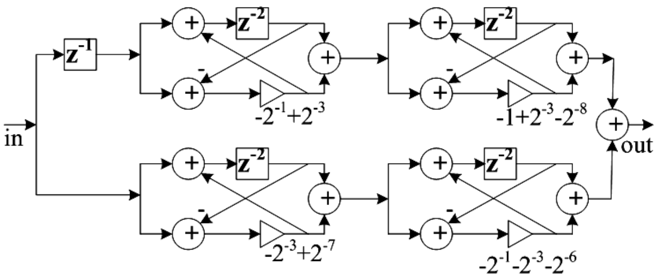

Fig. 4. Realization for the resulting multiplierless halfband IIR filter.

VI. CONCLUSION

A straightforward two-step scheme has been developed for designing multiplierless halfband IIR digital filters. The first step determines a parameter space of the infinite-precision coefficients including the feasible space where the filter meets the given criteria. The second step uses genetic algorithm for finding the coefficients in this space such that the given criteria are met by the simplest representation forms. The efficiency of the proposed procedure has been demonstrated by means of an example.

REFERENCES

[1] M. D. Lutovac and Lj. D. Milić, “Design of Multiplierless Elliptic IIR Halfband Filters and Hilbert Transformers,” EUSIPCO’98, Conference Proceedings, pp. 291–294, Rhodes, Greece, Sept. 1998.

[2] M. D. Lutovac, D. V. Tošic, and B. Evans, *Filter Design for Signal Processing using Matlab and Mathematica,*Upperside River, New Jersey: Prentice Hall, 2000.

[3] H. W. Schüβler and P. Steffen, “Halfband Filters and Hilbert Transformers,” Circuits, Systems, Signals Processing, vol. 17, no. 2, pp. 137-164, Feb. 1998.

[4] Y. C. Lim, Y. J. Yu, and T. Saramäki, “Optimum Masking Levels and Coefficient Sparseness for Hilbert Transformers and Halfband Filters Designed using the Frequency-Response Masking Technique,” IEEE Trans. Circuits. Syst. I: Regular Papers, vol 12, no. 6, pp. 2444–2453, Nov. 2005.

[5] S. A. Samad, A. Hussain and D. Isa, “Wave Digital Filters with Minimum Multiplier for Discrete Hilbert Transformer Realization,” Signal Processing Journal, pp. 3761–3768, 2006.

[6] J. Yli-Kaakinen and T. Saramäki, “A Systematic Algorithm for the Design of Multiplierless Lattice Wave Digital Filters,” IEEE Trans. Circuits Syst. I: Regular Papers, vol. 54, no. 8, pp. 3761–3768, Aug. 2004.

[7] V. I. Anzova, J. Yli-Kaakinen and T. Saramäki, “An Algorithm for the Design of Multiplierless IIR Filters as a Parallel Connection of Two All-pass Filters” IEEE APCCAS 2006, Conference Proceedings, pp. 744–747, Singapore, 2006.

[8] M. D. Lutovac and Lj. D. Milić, “Design of multiplierless elliptic IIR filters with a small quantization error,” IEEE Trans. Signal Processing, vol. 47, pp .469–479, Feb. 1999.

[9] A. H. Gray and J. D. Markel, “Digital Lattice and Ladder Filter Synthesis,” IEEE Trans. Audio Electroacoust., vol. AU-21, no. 6, pp. 491–499, Dec. 1973.

[10] C. Houck, J. Joines, and M. Kay, “A Genetic Algorithm for Function Optimization: A Matlab Implementation,” NCSU-IE Technical Report 95-09, North Carolina State University, 1995.