An Algorithm for the Design of Multiplierless IIR Filters as a Parallel Connection of Two All-Pass Filters

Valentina I. Anzova
Department of Telecommunications
Technical University of Sofia
Kliment Ohridiski 8, 1756 Sofia, Bulgaria
email: via@tu-sofia.bg

Juha Yli-Kaakinen and Tapio Saramäki
Institute of Signal Processing,
Tampere University of Technology
P. O. Box 553, FI-33101 Tampere, Finland
email: {ylikaaki, ts}@cs.tut.fi

***Abstract*— This paper describes an algorithm for designing multiplierless recursive digital filters as a parallel connection of two all-pass filters. The coefficient optimization is performed in two steps. First, simple closed-form algebraic expressions are used for determining 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 filter parameters in this space so that the resulting filter meets the given criteria with the simplest coefficient representation forms. Two implementation structures are under consideration, namely, the all-pass transfer functions are implemented either using Stoyanov-Kawamata or Gray-Markel all-pass sections. Comparisons with some other existing quantization schemes show that the proposed algorithm gives better finite-precision solutions in all examples taken from the literature.**

I. INTRODUCTION

In highly customized very large-scale integration (VLSI) implementations, the general multiplier element is very costly. Therefore, it is beneficial to carry out the multiplication of a data sample by each filter coefficient value using a sequence of shifts and adds or subtracts. The shifts are often hardwired and, therefore, essentially free. Thus, only a few adders or subtracters are required for implementing each coefficient. Such an implementation is usually called "multiplierless". In order to generate multiplierless filter implementations, it is very essential that a digital filter is realized using a low-sensitivity structure being very insensitive to variations in the filter coefficients. The importance of such a structure is that if the effect of the coefficient value deviation from the ideal value is small, then short coefficient wordlengths can be used with only slightly violating the infinite-precision filter specifications, resulting in a faster, smaller, and less expensive hardware.

One of the best structures for implementing recursive digital filters are the parallel connection of two all-pass filters [1]. This filter class is characterized by a low coefficient sensitivity. Furthermore, the number of multipliers required in the implementation is directly the filter order, unlike in some other implementation forms, such as in the canonic direct-form realizations requiring approximately twice the number of multipliers. In addition, these all-pass subfilters can be realized by using first- and second-order sections as basic building blocks. The resulting filter structures are highly modular, thereby making them suitable for VLSI implementations [2], [3], [4].

This paper describes an algorithm for designing recursive digital filters with short coefficient wordlength based on use of parallel all-pass filters. This algorithm is based on the following observation: Finding for an elliptic filter the minimum passband ripple, the maximum stopband attenuation, the maximum passband edge, and the minimum stopband edge such that the given criteria are still met

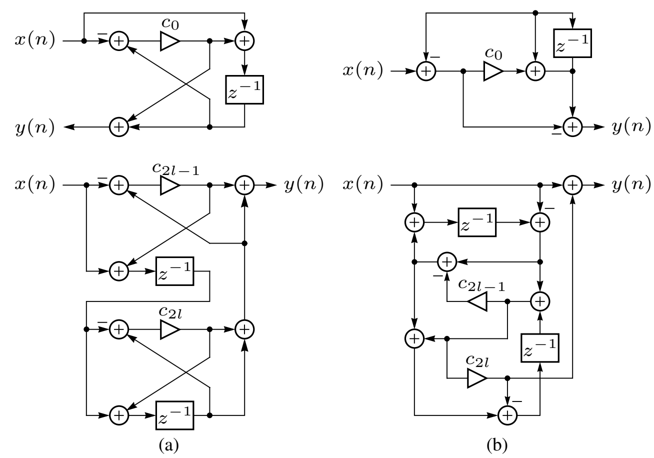

Fig. 1. First- and second-order all-pass sections under consideration. (a) Gray-Markel sections. (b) Stoyanov-Kawamata sections.

enables one to find 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 all-pass filters due to the fact that for these filters only the denominator coefficients of the all-pass sections have to be quantized. Several examples taken from the literature are included illustrating that in all the examples the proposed quantization scheme results in a better finite-precision solution than other existing quantization techniques.

II. TRANSFER FUNCTION UNDER CONSIDERATION

Let the transfer function of a recursive filter be given by

$$H(z) = \frac{1}{2}[A_1(z) + A_2(z)], \tag{1}$$

where $A_1(z)$ and $A_2(z)$ are real stable all-pass filters of orders M and N, respectively. This contribution concentrates on designing lowpass filters. In this case, $M = N - 1$ or $M = N + 1$ so that $M + N$, the overall order of $H(z)$, is odd.

Two implementation structures are under consideration in this contribution. In the first case, the all-pass transfer functions are implemented using so called Stoyanov-Kawamata low-sensitivity all-pass sections. In the second case, these transfer functions are implemented using Gray-Markel lattice all-pass sections. The implementations of these all-pass sections are depicted in Fig. 1.

If $A_1(z)$ and $A_2(z)$ are implemented as a cascade of first- and second-order all-pass sections and M and N are assumed to be an odd and even integer, respectively, then $A_1(z)$ and $A_2(z)$ are expressible in terms of the coefficients of Stoyanov-Kawamata sections as follows (see, e.g., [5]):

$$A_1(z) = \frac{-(1-c_0)+z^{-1}}{1-(1-c_0)z^{-1}} \cdot \prod_{l=1}^{m} \frac{(1-c_{2l})+(-2+2c_{2l-1}+c_{2l})z^{-1}+z^{-2}}{1+(-2+2c_{2l-1}+c_{2l})z^{-1}+(1-c_{2l})z^{-2}} \tag{2a}$$

and

$$A_2(z) = \prod_{l=m+1}^{m+n} \frac{(1-c_{2l})+(-2+2c_{2l-1}+c_{2l})z^{-1}+z^{-2}}{1+(-2+2c_{2l-1}+c_{2l})z^{-1}+(1-c_{2l})z^{-2}}, \tag{2b}$$

where $m = (M-1)/2$ and $n = N/2$. For Gray-Markel sections, the corresponding all-pass transfer functions are expressed as follows:

$$A_1(z) = \frac{-c+z^{-1}}{1-cz^{-1}} \prod_{l=1}^{m} \frac{-c_{2l-1}-c_{2l}(1-c_{2l-1})z^{-1}+z^{-2}}{1-c_{2l}(1-c_{2l-1})z^{-1}-c_{2l-1}z^{-2}} \tag{3a}$$

and

$$A_2(z) = \prod_{l=m+1}^{m+n} \frac{-c_{2l-1}-c_{2l}(1-c_{2l-1})z^{-1}+z^{-2}}{1-c_{2l}(1-c_{2l-1})z^{-1}-c_{2l-1}z^{-2}}. \tag{3b}$$

III. Statement of the Problem

Before stating the optimization problem, we denote the transfer function of the filter by $H(\Phi,\, z)$, where Φ is the vector containing the adjustable parameters of the filter. Given the passband and stopband edges ω_p and ω_s, respectively, as well as the passband and stopband ripples δ_p and δ_s, respectively, the magnitude specifications for the filter are stated as follows:[1]

$$1-\delta_p \le |H(\Phi,\, e^{j\omega})| \le 1 \quad \text{for } \omega \in [0,\, \omega_p] \tag{4a}$$

$$|H(\Phi,\, e^{j\omega})| \le \delta_s \quad \text{for } \omega \in [\omega_s,\, \pi]. \tag{4b}$$

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

$$\textstyle\sum_{r=1}^{R} a_r 2^{-P_r}, \tag{5}$$

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 and 2) P_R, the maximum number of shifts, is made as small as possible.

An estimate for the implementation cost of the filter is the number of adders and subtracters required to implement all the filter coefficients, that is, the implementation cost is given by

$$\textstyle\sum_{l=0}^{2(m+n)} \sigma_l, \tag{6}$$

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

The optimization problem under consideration is the following:

[1]These specifications are typical of most recursive filters built using all-pass filters as building blocks. In these cases, the filter structure constrains the maximum of the magnitude response to be unity.

Optimization Problem: Given ω_p, ω_s, δ_p, and δ_s, find M and N, and the parameter vector Φ in such a manner that, first, the criteria of Eq. (4) 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. (6), is minimized.

IV. Filter Optimization

The solution to the stated optimization problem can be found in the following two steps. In the first step, the minimum passband ripple, the maximum stopband attenuation, the maximum passband edge, and the minimum stopband edge are determined for an elliptic filter so that the given specifications are still 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 second step involves finding the filter parameters in this space using genetic algorithm so that the resulting filter meets the given criteria with the simplest coefficient representation forms.

A. Optimization of Infinite-Precision Filters

It has turned out that a parameter space for the filter coefficients is obtained by designing four elliptic infinite-precision filters satisfying the specifications as follows:

Design 1: The passband and stopband criteria are just met and transition bandwidth is minimized such that the stopband edge is minimized.

Design 2: The passband and stopband criteria are just met and transition bandwidth is minimized such that passband edge is maximized.

Design 3: The passband criterion is just met and the stopband attenuation is maximized.

Design 4: The stopband criterion is just met and the passband ripple is minimized.

These designs can be obtained by using simple closed-form algebraic expressions [6]. The parameter vectors containing the optimal infinite-precision filter parameters for Designs 1, 2, 3, and 4 are denoted by $\Phi^{(k)}$ for $k = 1,2,3,4$, respectively, whereas the corresponding sets of coefficients values for the filter transfer function under consideration are denoted by $c_l^{(k)}$'s. Based on these set of coefficients, the smallest and largest values for the filter coefficients can then be determined as follows:

$$c_l^{(\min)} = \min_{k=1,2,3,4}\{c_l^{(k)}\} \quad \text{and} \quad c_l^{(\max)} = \max_{k=1,2,3,4}\{c_l^{(k)}\} \tag{7}$$

for $l = 0, 1, \ldots, M+N-1$.

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 specifications are satisfied. After finding this parameter space, all what is needed is to check whether in this space there exist combinations of the discrete 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 powers-of-two numbers C_l for $l = 0, 1, \ldots, M+N-1$ between the smallest and largest values of each coefficient, that is, determine for $l = 0, 1, \ldots, M+N-1$

$$\left\{ C_l \in \mathbb{POT}_{(R,P_R)} \;\middle|\; c_l^{(\min)} \le C_l \le c_l^{(\max)} \right\}. \tag{8}$$

Here, $\mathbb{POT}_{(R,P_R)}$ denotes the space of powers-of-two numbers for R, the given maximum number of powers-of-two terms and P_R, the maximum number of fractional bits [cf. Eq. (5)]. Denote by K_l the number of powers-of-two values between $c_l^{(\min)}$ and $c_l^{(\max)}$.

Furthermore, denote by $C_l^{(k)}$ for $k = 1, 2, \ldots, K_l$ the kth existing discrete value between these smallest and largest values.

The magnitude response is then evaluated for each combination of the $C_l^{(k)}$ for $l = 0, 1, \ldots, M + N - 1$ and $k = 1, 2, \ldots, K_l$ to check whether the filter meets the given specifications. The number of discrete coefficient value combinations is

$$\prod_{l=0}^{M+N-1} K_l. \qquad (9)$$

Due to the reason that the number of discrete coefficient value combinations can be huge it is beneficial to use genetic algorithm for searching those discrete coefficient values with which the specifications are met. This discrete-valued optimization problem can be efficiently solved using genetic algorithm as follows:

Step 1: Encode the indexes of the power-of-two numbers between the smallest and largest values of the coefficients as their binary representation. The number of bits needed to encode $C_l^{(k)}$'s is $\lceil \log_2 K_l \rceil$. In addition, generate a lookup table containing the power-of-two values of the corresponding indexes.

Step 2: Construct the chromosomes by concatenating all these binary strings, that is, the length of the cromosome is

$$\prod_{l=0}^{M+N-1} \lceil \log_2 K_l \rceil. \qquad (10)$$

Step 3: Evaluate the fitness of the population by decoding the chromosomes to their corresponding power-of-two coefficient value counterparts by using the above-mentioned lookup table.

The performance of the above encoding will be explained in more detail in connection with Example 1.

The fitness function to be maximized is given by

$$f = -\max\{\Delta_p/\delta_p,\, \Delta_s/\delta_s\}, \qquad (11)$$

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. Selecting the quantity to be maximized according to above equation is motivated by the fact that, in this case, the resulting relative error values with respect to the given values become the same for both the passband and stopband errors.

V. Numerical Examples

The purpose of this section is twofold. First, the performance of the proposed quantization scheme is illustrated by means of an example. Second, comparisons with some other existing quantization schemes show that the proposed algorithm gives better finite-precision solutions in all examples taken from the literature.

A. Example 1

This example is included to illustrate in detail the proposed overall synthesis scheme. It is desired to design a low-pass filter with the passband and stopband edges at $\omega_p = 0.05\pi$ and at $\omega_s = 0.07\pi$, respectively. The maximum allowable passband ripple and the required stopband attenuation are $\delta_p = 0.1$ (0.9151 dB) and $\delta_s = 1.4 \cdot 10^{-3}$ (57.08 dB), respectively. The minimum odd-order of an elliptic filter to meet the given amplitude criteria is seven ($M = 3$ and $N = 4$).[2]

The infinite-precision coefficient values of the four elliptic initial filter designs for the Stoyanov-Kawamata sections are shown in Table I. In addition, the corresponding smallest and largest values for the coefficient values, denoted by $c^{(\min)}$ and $c^{(\max)}$, respectively, are

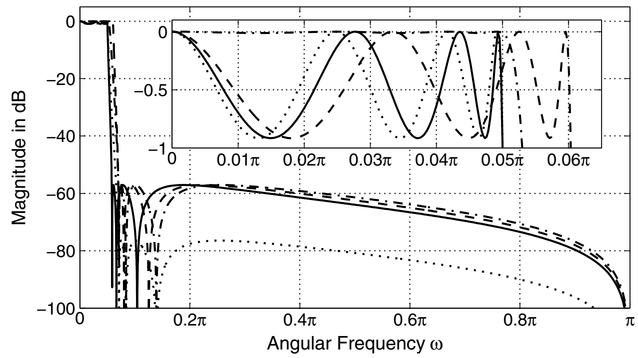

Fig. 2. Magnitude responses for the initial filters in Example 1. The solid, dashed, dotted, and dot-dashed lines, respectively, give the responses for Designs 1, 2, 3, and 4, respectively. In addition, the passband details are shown for these designs.

TABLE I
Infinite-Precision Coefficient Values for the Initial Designs

	$c^{(1)}$	$c^{(2)}$	$c^{(3)}$	$c^{(4)}$	$c^{(\min)}$	$c^{(\max)}$
c_0	0.045 18	0.039 07	0.102 18	0.054 31	0.039 07	0.102 18
c_1	0.009 78	0.008 97	0.011 78	0.014 19	0.008 97	0.014 19
c_2	0.028 18	0.034 06	0.078 84	0.033 86	0.028 18	0.078 84
c_3	0.004 60	0.003 69	0.007 70	0.006 65	0.003 69	0.007 70
c_4	0.064 18	0.062 06	0.151 94	0.076 90	0.062 06	0.151 94
c_5	0.012 26	0.012 22	0.014 17	0.017 81	0.012 22	0.017 81
c_6	0.007 11	0.010 17	0.022 88	0.008 56	0.007 11	0.022 88

TABLE II
Permissible Powers-of-Two Numbers
Between the Smallest and Largest Values of c_6

k	$C_6^{(k)}$	$b_{\text{idx}}^{(k)}$	k	$C_6^{(k)}$	$b_{\text{idx}}^{(k)}$
1	$2^{-7} = 0.007\,81$	000	5	$2^{-6} = 0.015\,63$	100
2	$2^{-7} + 2^{-9} = 0.009\,77$	001	6	$2^{-6} + 2^{-9} = 0.017\,58$	101
3	$2^{-6} - 2^{-8} = 0.011\,72$	010	7	$2^{-6} + 2^{-8} = 0.019\,53$	110
4	$2^{-6} - 2^{-9} = 0.013\,67$	011	8	$2^{-5} - 2^{-7} - 2^{-9} = 0.021\,48$	111

shown in this table. The magnitude responses as well as the passband details for the initial designs are shown in Fig. 2.

For the corresponding finite-precision overall filter three powers-of-two terms ($R = 3$) and nine fractional bits ($P_R = 9$) are required to meet the given specifications.[3] The number of permissible discrete values between the smallest and largest values of c_l for $l = 0, 1, \ldots, 6$ for this coefficient representation form are 29, 3, 26, 2, 40, 3, and 8, respectively. The number of bits needed to encode these discrete values are 5, 2, 5, 1, 6, 2, and 3, respectively, that is, the length of the cromosome is 24 bits. For clarity, Table II shows the permissible powers-of-two numbers between the smallest and largest values of c_6 for this coefficient representation form as well as the corresponding binary representation of the indexes denoted by b_{idx}.

The control parameters of the genetic algorithm such as crossover rate and mutation rate has been adjusted by running this algorithm 100 times with different parameter settings and then selecting the most optimal ones. Table III shows some typical optimization results when normalized geometric selection is used as a reproduction oper-

[2]It is well known that the odd-order elliptic filter is the most selective low-pass filter being implementable as a parallel connection of two all-pass filters (see, e.g., [5]).

[3]In this case, eight fractional bits is the shortest wordlength for which there exist discrete values between the smallest and largest values of the coefficient values. However, for this coefficient representation, there is no solutions satisfying the specifications.

TABLE III

PERFORMANCE OF THE GENETIC ALGORITHM FOR THE VARIOUS VALUES OF $P_{\rm S}$, $P_{\rm X}$, AND $P_{\rm M}$

$P_s = 0.025$					$P_s = 0.035$				
P_x	P_m	f_{mean}	f_{std}	N_{hit}	P_x	P_m	f_{mean}	f_{std}	N_{hit}
0.60	0.05	−1.1073	0.1541	32	0.60	0.05	−1.2706	0.2558	12
0.60	0.06	−1.0793	0.1025	21	0.60	0.06	−1.1914	0.1927	13
0.60	0.07	−1.0537	0.0910	32	0.60	0.07	−1.0797	0.1057	25
0.70	0.05	−1.0845	0.1292	34	0.70	0.05	−1.2558	0.2053	12
0.70	0.06	−1.0750	0.0990	21	0.70	0.06	−1.1448	0.1418	21
0.70	0.07	−1.0706	0.0938	25	0.70	0.07	−1.0661	0.1081	39
0.80	0.05	−1.1288	0.1554	24	0.80	0.05	−1.2701	0.2068	10
0.80	0.06	−1.0718	0.0967	27	0.80	0.06	−1.1063	0.1303	29
0.80	0.07	−1.0662	0.0872	25	0.80	0.07	−1.0522	0.0777	34

TABLE IV

OPTIMIZED FINITE-PRECISION COEFFICIENT VALUES OF THE STOYANOV-KAWAMATA SECTIONS FOR THE FILTER IN EXAMPLE 1

$A_1(z)$	$A_2(z)$
	$c_3 = 2^{-8}$
$c_0 = 2^{-4}$	$c_4 = 2^{-3} - 2^{-5} - 2^{-8}$
$c_1 = 2^{-7} + 2^{-9}$	$c_5 = 2^{-6} - 2^{-9}$
$c_2 = 2^{-5} + 2^{-7} + 2^{-9}$	$c_6 = 2^{-6} - 2^{-8}$

ator, the population size is 150, and the number of generations is 500. In this table, P_s denotes the selection probability, P_x and P_m are the crossover rate and mutation rate, 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 specifications, respectively. The fitness value of the best solution after these 100 runs is every time the same, that is, $f = -0.9943$. As can be concluded from this table, ten runs is usually enough to find a solution meeting the specifications. The CPU time required for running one run is approximately 40 seconds when using the genetic algorithm optimization toolbox [7] in MATLAB 6.5 on a 1.4 GHz Pentium-M. The CPU-time required to evaluate all the possible coefficient value combinations [c.f. Eq. (9)] is approximately 23 minutes.

A total of only seven adders and/or subtracters are required to implement all the multipliers for this coefficient representation form for the Stoyanov-Kawamata sections. The optimized coefficient values are shown in Table IV. The passband ripple and the stopband attenuation for this structure are 0.910 dB and 60.30 dB, respectively. For the Gray-Markel sections, the specifications are also met by $R = 3$ and $P_R = 9$. In this case 14 adders and/or subtracters are needed to implement all the multipliers.

B. Example 2

In [8], the passband and stopband edges are $\omega_p = 0.52\pi$ and $\omega_s = 0.58\pi$, respectively. The passband ripple and the stopband attenuation for the optimized 11th-order ($M = 6$ and $N = 5$) finite-precision filter in [8] are 0.426 dB and 36.36 dB, respectively. The maximum number of power-of-two terms is four ($R = 4$) and the number of fractional bits is 12 ($P_R = 12$) whereas the total number of adders and/or subtracters needed to implement all the filter coefficients is 29.[4] For the proposed technique, these specifications are met by seventh-order filter ($M = 3$ and $N = 4$). Only two power-of-two terms and six fractional bits ($R = 2$ and $P_R = 6$) are needed to meet the specifications. The number of adders and/or subtracters required to implement all the filter coefficients for the Stoyanov-Kawamata sections is only five, that is, the implementation complexity is less than 20 percent compared to that required by the optimized filter in [8]. The passband ripple and the stopband attenuation are 0.354 dB and 38.36 dB, respectively. The optimized finite-precision coefficient values are shown in Table V.

[4]It should be pointed out that in [8] the all-pass filters are implemented as a cascade connection of first- and second-order direct-form sections.

TABLE V

OPTIMIZED FINITE-PRECISION COEFFICIENT VALUES OF THE STOYANOV-KAWAMATA SECTIONS FOR THE FILTER IN EXAMPLE 2

$A_1(z)$		$A_2(z)$	
	$c_1 = 1 - 2^{-3}$	$c_3 = 2^{-1}$	$c_5 = 1 + 2^{-4}$
$c_0 = 2^{-1} + 2^{-3}$	$c_2 = 2^{-2} - 2^{-6}$	$c_4 = 2^{-1} + 2^{-4}$	$c_6 = 2^{-4}$

C. Example 3

Consider the half-band filter specifications [9] $\omega_p = 0.44\pi$ and $A_s = 46$ dB. Due to the properties of half-band IIR filters, $\omega_s = \pi - \omega_p = 0.56\pi$ and $\delta_p \approx \delta_s^2/2$, giving $A_p = 1.1 \cdot 10^{-4}$ dB (see, e.g., [10]). For the optimized finite-precision filter of order nine in [9], $P_R = 8$, $R = 4$, and the number of adders and/or subtracters needed to implement all the coefficients is eight. The proposed algorithm results in the optimized filter with $P_R = 8$, $R = 3$, and requiring only six adders and/or subtracters to implement all the coefficients for the Gray-Markel sections.

VI. CONCLUSIONS

A two step algorithm has been developed for designing multiplierless recursive digital filters as a parallel connection of two all-pass filters. The efficiency and the robustness of the proposed algorithm has been demonstrated by means of several examples. It has been shown that significant saving in the implementation cost are achieved using the proposed technique. In addition, it has been demonstrated that for the proposed technique the computational complexity is significantly smaller than for the technique where all the coefficient value combinations are systemically evaluated.

REFERENCES

[1] A. Fettweis, “Wave digital filters: Theory and practice,” *Proc. IEEE*, vol. 74, pp. 270–327, Feb. 1986.

[2] T. Saramäki and T. Ritoniemi, “Optimization of digital filter structures for VLSI implementation,” *Automatica*, vol. 34, pp. 111–116, 1993.

[3] J. Yli-Kaakinen and T. Saramäki, “A systematic algorithm for the design of multiplierless lattice wave digital filters,” in *Proc. 2004 IEEE Int. Symp. Control., Commun., Signal Process.*, Hammamet, Tunisia, Mar. 21–24 2004, pp. 415–418. [Online]. Available: http://www.cs.tut.fi/~ylikaaki

[4] V. Anzova, K. Ivanova, and G. Stoyanov, “Sensitivity investigations of cascaded realizations of IIR digital filters based on tapped identical allpass subfilters,” in *TELECOM’05*, Varna, Bulgaria, Oct. 6–7 2005.

[5] L. Gazsi, “Explicit formulas for lattice wave digital filters,” *IEEE Trans. Circuits Syst.*, vol. CAS-32, pp. 68–88, Jan. 1985.

[6] M. Lutovac, D. Tošić, and B. Evans, “Advanced filter design,” in *IEEE Asilomar Conf. Signal, Systems, Computers*, vol. 1, Pacific Grove, CA, Nov. 2–5 1997, pp. 710–715.

[7] C. Houck, J. Joines, and M. Kay, “A genetic algorithm for function optimization: A Matlab implementation,” North Carolina State University,” NCSU-IE Technical Report 95–09, 1995.

[8] L. Liang, M. Ahmadi, and M. A. Sid-Ahmed, “Design of complementary filter pairs with canonical signed-digit coefficients using genetic algorithm,” in *IEEE Int. Conf. Electr., Circuits and Syst. (ICECS’04)*, Dec. 13–15 2004, pp. 611–614.

[9] M. D. Lutovac and L. Milić, “Design of computationally efficient elliptic IIR filters with a reduced number of shift-and-add operations in multipliers,” *IEEE Trans. Signal Processing*, vol. 45, pp. 2422–2430, Oct. 1997.

[10] W. Wegener, “Wave digital directional filters with reduced number of multipliers and adders,” *AEÜ–Int. J. Electron. Commun.*, vol. 33, pp. 239–243, June 1979.