

Evolutionary Synthesis of Low-Sensitivity Equalizers Using Adjacency Matrix Representation

Leonardo Bruno de Sá
Federal University of Rio de Janeiro, Brazil
C. Postal 68504 – CEP 21945-970
Rio de Janeiro – RJ – Brazil
Tel: 55 21 25628204
lbruno@openlink.com.br

Antonio Mesquita
Federal University of Rio de Janeiro, Brazil
C. Postal 68504 – CEP 21945-970
Rio de Janeiro – RJ – Brazil
Tel: 55 21 25628627
mesquita@coe.ufrj.br

ABSTRACT

An evolutionary synthesis method to design low-sensitivity IIR filters with linear phase in the passband is presented. The method uses a chromosome coding scheme based on the graph adjacency matrix. It is shown that the proposed chromosome representation enables to easily verify invalid individuals during the evolutionary process. The efficiency of the proposed algorithm is tested in the synthesis of a fourth-order linear phase elliptic lowpass digital filter.

Categories and Subject Descriptors: I.0 General

General Terms: Algorithms, Design

Keywords: Circuit, synthesis, digital filters, genetic algorithms.

1. INTRODUCTION

The distortion-free transmission of signals requires linear phase digital filters. Since digital filters with linear phase along the entire frequency response are unrealizable, the linear phase requirement is generally limited to the passband. According to the length of the impulse response digital filters are classified as Finite Impulse Response (FIR) or Infinite Impulse Response (IIR). The linear phase requirement can be easily implemented in FIR filters. However, FIR filters satisfying the same magnitude specifications tend to have higher order than its IIR counterpart. Consequently, the design of IIR filters with linear phase is more attractive for such applications.

There are two different approaches to design linear phase IIR filters. To design a single filter that realizes the magnitude and phase specifications simultaneously [1] or to design a filter that meets the desired magnitude followed by a cascade of allpass equalizer stages that correct the phase response [2-3]. In this paper, the last approach is adopted where the evolutionary algorithm is used to synthesize a cascade of second-order low-sensitivity allpass sections that correct the phase response of an IIR filter.

Low-sensitivity digital filters are of major practical interest since any actual implementation of a digital filter will involve

finite word length arithmetic operations and quantization errors. However, the design of low-sensitivity filters is not a straightforward task since there are no general analytical methods to derive the optimal filter configuration [4].

In the synthesis of a digital filter composed of multipliers, delays and adders, only the values of the multipliers should be optimized. In a large number of works found in the literature [5-11], a Genetic Algorithm (GA) is used just as an optimization method to find the multipliers values for a given fixed filter topology. On the other hand, among the works [12-17] addressing the evolutionary synthesis methods only a few deal with the low-sensitivity problem [16-17]. Several equivalent filter structures are able to realize the same transfer function. These structures can be obtained by different methods, varying significantly with respect to complexity and number of elements. A given structure may require a large number of multipliers and yet be relatively insensitive to the coefficients quantization errors while one with fewer elements may generate parasitic oscillations when signals are quantized [18]. Thus, low-sensitivity is related to the filter topology and it is expected that the evolutionary synthesis methods could find better solutions than the deterministic approaches [19-22] which are bounded to a limited set of filter structures.

The efficiency of an evolutionary algorithm requires suitable representations to detect the generation of anomalous individuals [23]. In digital filters domain, there are two possible anomalies: structurally invalid individuals and non-computable individuals containing delay-free loops. The first category consists of filters that have more than one or no component at all incident into a circuit node. The second category is composed of physically unrealizable filters that would demand digital processors with infinite bandwidth to operate properly. In this case, it is shown that the computability verification is easily implemented with the proposed representation requiring a small amount of computation. To verify the performance of the proposed evolutionary algorithm an equalizer for an IIR fourth-order lowpass filter is synthesized.

The content of the paper is outlined as follows. A brief introduction on equalizer design is presented in Section 2. The quantization effects in digital filters are discussed in Section 3. The proposed adjacency matrix representation for digital filters and the rules to construct valid individuals including the computability and the application of the genetic operators are discussed in Section 4. The heuristic of the fitness computation is described in Section 5. In Section 6, a numerical example to verify the performance of the proposed evolutionary synthesis method is presented. Concluding remarks are given in Section 7.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
GECCO '08, July 12–16, 2008, Atlanta, Georgia, USA.
Copyright 2008 ACM 978-1-60558-130-9/08/07...\$5.00.

2. Design of Allpass Transfer Function Equalizer

The transfer function of a digital filter is the ratio between the z-transform of the output node and the z-transform of the input node. Fig. 1 illustrates a first-order filter structure, where v_1 and v_3 are, respectively, the input and output nodes.

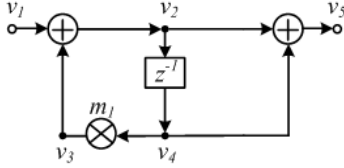


Figure 1. First-order digital filter structure.

The filter transfer function, $H_F(z)$, is given by:

$$H_F(z) = \frac{v_5}{v_1} = \frac{z+1}{z-m_1} \quad (1)$$

The design of linear-phase or constant group-delay digital filters is generally accomplished in two steps. In the first step, a filter satisfying the magnitude specifications is designed ignoring the phase. After that an allpass equalizer, which is placed in cascade with the filter, is designed to compensate for the filter's phase variations. Fig. 2 illustrates the block diagram of a digital filter with phase equalization, where $H_F(z)$ and $H_E(z)$ are the transfer functions of the filter and equalizer, respectively.

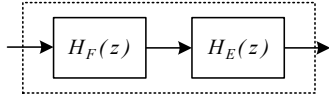


Figure 2. Structure of group delay equalizer

The filter and the equalizer group delays are respectively given by:

$$\tau_F(\omega) = -\frac{d\theta_F(\omega)}{d\omega} \quad (2a)$$

$$\tau_E(\omega) = -\frac{d\theta_E(\omega)}{d\omega} \quad (2b)$$

Where:

$$\theta_F(\omega) = \arg^{-1} \frac{\text{Im}\{H_F(e^{j\omega})\}}{\text{Re}\{H_F(e^{j\omega})\}} = \angle H_F(e^{j\omega}) \quad (3a)$$

$$\theta_E(\omega) = \angle H_E(e^{j\omega}) \quad (3b)$$

Thus:

$$\left| H_{FE}(e^{j\omega}) \right| = \left| H_F(e^{j\omega}) \right| \cdot \left| H_E(e^{j\omega}) \right| \quad (4a)$$

$$\tau_{FE}(\omega) = \tau_F(\omega) + \tau_E(\omega) \quad (4b)$$

According to (4), a constant group delay filter can be designed using an allpass equalizer that meets the following specifications:

$$\begin{cases} \left| H_E(e^{j\omega}) \right| = 1 & \text{for } 0 \leq \omega \leq \frac{\omega_s}{2} \\ \tau_E(\omega) = \tau - \tau_F(\omega) & \text{for } \omega_{p1} \leq \omega \leq \omega_{p2} \end{cases} \quad (5)$$

Where ω_s , ω_{p1} , ω_{p2} and τ are the sampling frequency, the lower limit of the passband, the upper limit of the passband and a given constant, respectively. To meet the first specification, the equalizer transfer function must be of the form:

$$H_E(z) = \prod_{i=1}^N \frac{a_{0i}z^2 + a_{1i}z + 1}{z^2 + a_{1i}z + a_{0i}} \quad (6)$$

Where, N is the number of second-order sections.

To insure the stability, the poles of the transfer function in (6) must be inside the unit circle. This is obtained by constraining the coefficients of each second-order denominator by the following relations:

$$\begin{cases} -1 < a_{0i} < 1 \\ -1 - a_{0i} < a_{1i} < 1 + a_{0i} \end{cases} \quad (7)$$

Replacing $z = e^{j\omega}$ in (6) and taking into account (2b), the equalizer group delay for the i^{th} second-order section can be written as:

$$\tau_{Ei}(\omega) = -\frac{2(a_{0i}-1)(a_{0i}+a_{1i}\cos\omega+1)}{2\cos\omega(2a_{0i}\cos\omega+a_{0i}a_{1i}+a_{1i})+(a_{0i}-1)^2+a_{1i}^2} \quad (8)$$

Therefore, the coefficients a_{0i} and a_{1i} of each section of the transfer function $H_E(z)$ must satisfy (7). They should be optimized so that the equalizer group delay (8) corrects the filter group delay, in such way that the overall structure achieves an approximately constant group delay in the passband.

3. Quantization Effects

Although the fact that a given transfer function can be realized by an unlimited number of structures, in general, only a few are sufficiently insensitive to the coefficients quantization errors. To illustrate the quantization effects in the transfer function, consider the first-order structure of Fig. 1. A practical implementation of this transfer function requires the coefficients quantization due to registers finite word length. After quantization the transfer function can be expressed as:

$$H_{F,Q}(z) = \frac{z+1}{z-[m_1]_Q} \quad (9)$$

Where $H_{F,Q}(z)$ and $[m_1]_Q$ are the quantized versions of $H_F(z)$ and coefficient m_1 respectively.

Replacing $z = e^{j\omega}$ in (9), the magnitude and the phase responses can be expressed as:

$$\begin{aligned} \left| H_{F,Q}(e^{j\omega}) \right| &= \sqrt{\frac{2 \cdot (\cos\omega + 1)}{[m_1]_Q^2 - 2[m_1]_Q \cos\omega + 1}} \\ \tau_{F,Q}(\omega) &= -\frac{[m_1]_Q^2 - 1}{2 \cdot ([m_1]_Q^2 - 2[m_1]_Q \cos\omega + 1)} \end{aligned} \quad (10)$$

Where $\left| H_{F,Q}(e^{j\omega}) \right|$ and $\tau_{F,Q}(\omega)$ are the quantized versions of $\left| H_F(e^{j\omega}) \right|$ and $\tau_F(\omega)$.

As a consequence, the transfer function differs from the ideal response and the poles and zeros are displaced from their original locations. If the poles are displaced to a location outside the unit circle in the z-plane, the quantized filter will be unstable. However, it is possible to minimize these effects by synthesizing filter structures that are inherently less sensitive to the coefficients quantization.

An evolutionary approach able to synthesize low-sensitivity digital filters is a compelling alternative to the restricted number of structures that may be synthesized by deterministic methods. However, such an algorithm should be able to overcome some of the drawbacks associated with the generation and detection of anomalous structures during the evolutionary process.

It will be shown that, when used to represent digital filters, the proposed chromosome coding scheme enables to devise computationally efficient procedures to identify such anomalous individuals.

4. The Adjacency Matrix Representation for Digital Filters

The adjacency matrix of an n-node oriented graph is defined as [24]:

Definition 1: Adjacency matrix

Let $G(V, E)$ be an oriented graph with no parallel edges and n vertices sorted out between v_1 and v_n , where V and E denote, respectively, the set of vertices and edges of the graph. The adjacency matrix $A = [a_{ij}]$ of the oriented graph G is the $n \times n$ matrix defined as:

$$a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in E \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

An edge is said to be incident out of its initial vertex and incident into its terminal vertex. A vertex is called an isolated vertex if no edge is incident on it.

In Fig. 3(a) the adjacency matrix A corresponding to the graph G of Fig. 3(b) is shown. In this matrix the main diagonal non-zero entries represent self-loops or short-circuits as, for example, the a_{33} entry of the matrix corresponding to the edge e_3 .

The adjacency matrices corresponding to the three basic elements of a digital filter are shown in Table 1. As shown in this table, a two-edge representation is used to represent the three-terminal adders. As a consequence, only the columns containing adders will have two entries due to the three-terminal nature of this element. The columns corresponding to the other elements will have just one entry.

4.1 Invalid Individuals

In an evolutionary process invalid individuals can be generated in the initial population or after the genetic operator's application. They can be classified in two categories: the structurally invalid individuals and the non-computable ones.

Since the fitness computation is the most time consuming task, the algorithm efficiency can be considerably increased if checking rules to prevent the evaluation of invalid circuits are

included [23]. Since in digital filters, the circuit components depend on the direction of the signal flow, the topology restrictions in a digital filter are particularly severe when compared with analog filters.

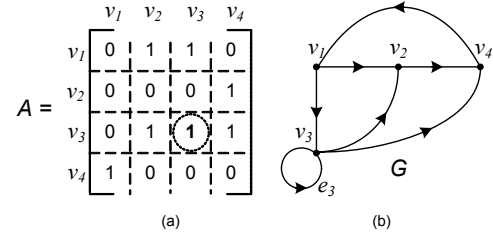


Figure 3. (a) Adjacency matrix (b) corresponding oriented graph G .

Table 1. Adjacency Matrix Representation for Basic Elements

	Oriented graph	Adjacency matrix
Adder		$\begin{matrix} & v_1 & v_2 & v_3 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{matrix}$
Multiplier		$\begin{matrix} & v_1 & v_2 \\ \begin{matrix} v_1 \\ v_2 \end{matrix} & \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix} \end{matrix}$
Delay		$\begin{matrix} & v_1 & v_2 \\ \begin{matrix} v_1 \\ v_2 \end{matrix} & \begin{bmatrix} 0 & 5 \\ 0 & 0 \end{bmatrix} \end{matrix}$

The invalid structures that may occur in a digital filter are listed in Table 2 with their corresponding adjacency matrix representation. The first structure in this table has a delay and a multiplier incident into the same node v_k as indicated by the two entries in the corresponding column of the matrix. The second anomalous structure is the short-circuiting self-loop in some node v_j which is represented by a nonzero entry in the corresponding position of the matrix main diagonal. Finally, the third anomalous structure is that containing an isolated node, such as node v_j in the third row of Table 2. This corresponds to an empty line in the adjacency matrix.

Therefore, in order to detect an invalid individual in the initial population and during the execution of the evolutionary synthesis it is sufficient to apply the following set of rules:

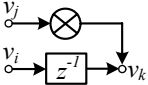
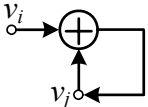
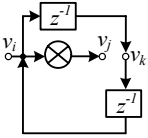
Rule 1: Each column of the adjacency matrix, except the first one, must contain only one type of element. With the proposed choice of element numbers, this rule implies that the sum of the entries of any column in a valid individual can not be different from 2 or 5.

Rule 2: Since the flow graphs of digital filters have no ground node, the main diagonal entries of the adjacency matrix must be zero.

Rule 3: Except for the last line, the adjacency matrix can not contain empty lines. Therefore, each node, except the output one, must have at least one edge incident out of it.

The last necessary step to validate an individual is to verify that it is a computable filter structure.

Table 2. Invalid Structures

Rule	Invalid Structure	Adjacency Matrix
1		$\begin{matrix} & v_i & v_j & v_k \\ v_i & 0 & 0 & 5 \\ v_j & 0 & 0 & 2 \\ v_k & 0 & 0 & 0 \end{matrix}$
2		$\begin{matrix} & v_i & v_j \\ v_i & 0 & 1 \\ v_j & 0 & 1 \end{matrix}$
3		$\begin{matrix} & v_i & v_j & v_k \\ v_i & 0 & 2 & 5 \\ v_j & 0 & 0 & 0 \\ v_k & 5 & 0 & 0 \end{matrix}$

4.2 Computable Digital Filters

To test for the computability of an individual using the proposed chromosome coding it will be necessary to introduce the concept of path matrix which is derived from the adjacency matrix. The path matrix is defined as:

Definition 2: Path Matrix

Let $G(V, E)$ be an oriented graph with n vertices v_1, v_2, \dots, v_n . Assuming the usual definition of an oriented path in an oriented graph [24], the path matrix of G is a $n \times n$ matrix $P = [p_{ij}]$ given by:

$$p_{ij} = \begin{cases} 1, & \text{if there is a directed path from } v_i \text{ to } v_j \\ 0, & \text{otherwise} \end{cases} \quad (18)$$

The path matrix is obtained from the adjacency matrix applying the Warshall algorithm [25].

In Fig. 9(a) the path matrix P corresponding to the graph G of Fig. 9(b) is shown. As can be seen in the figure, if $p_{ij} = 1$ and $i \neq j$ there is a path between vertices v_i and v_j otherwise, if $i = j$, there is a loop. Therefore, a non-zero entry in the main diagonal of matrix P indicates the existence of loops in the filter structure. As a consequence, if the delays are eliminated from the digital filter and the path matrix corresponding to the new adjacency matrix is obtained, the delay-free loops in the structure will be indicated by its non-zero main-diagonal entries.

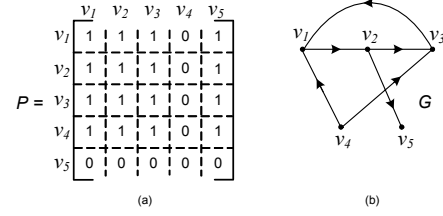


Figure 4. (a) Path matrix (b) corresponding oriented graph G .

As a consequence, using the adjacency matrix representation, the computability of digital filters is determined by the following algorithm:

Algorithm 2: Computability verification

S1. Generate the adjacency matrix following the rules of Section 4.2.

S2. In this adjacency matrix replace the delays by zeros.

S3. Obtain the path matrix P using the Warshall algorithm.

S4. If all main diagonal entries of matrix P are zeros the filter is computable, otherwise it is non-computable and the individual is not valid.

Fig. 10 illustrates the preceding steps. The first example is a computable structure as can be attested by the main diagonal entries of its path matrix P given in Fig. 10(d). The second example is a non-computable filter with a delay-free loop indicated by a dotted line in Fig. 10(e). The delay-free loop contains the nodes 2, 4 and 5 as indicated by the main diagonal entries of the path matrix in Fig. 10(h).

4.3 Genetic Operators Using the Adjacency Matrix

The crossover strategy is illustrated in Fig. 5. It consists in exchanging two submatrices with randomly chosen dimensions. Assume two adjacency matrices of dimensions m and n . If $m < n$, the coordinates of the crossover point (i, j) are chosen on the smaller matrix, where i and j are integers inside the intervals:

$$i \in [0, m-1] \quad (14a)$$

$$j \in [0, m-1] \quad (14b)$$

In order to define the dimensions of the submatrices to be exchanged in the crossover, two integers p and q are randomly chosen in the intervals:

$$p \in [1, m-i] \quad (15a)$$

$$q \in [1, m-j] \quad (15b)$$

Finally, the coordinates of the crossover point on the largest matrix are integers randomly chosen in the intervals:

$$k \in [0, n-p] \quad (15a)$$

$$l \in [0, n-q] \quad (15b)$$

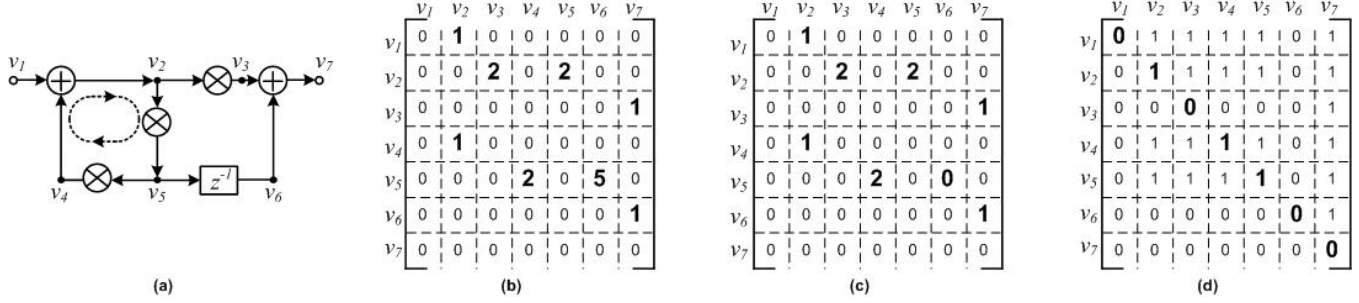


Figure 5. Verification of the computability for a first-order filters: (a) Digital filter structure (b) corresponding adjacency matrix representation (c) delay-free adjacency matrix and (d) path matrix.

In Fig. 5 the dimensions of the matrices, the coordinates of the crossover points and the dimensions of the submatrices are, respectively, $m=6$, $n=8$, $i=1$, $j=0$, $k=2$, $l=1$, $p=2$ and $q=4$.

The mutation operator simply replaces a randomly chosen matrix column by a new column containing an also randomly chosen element. Fig. 6 shows an example of the proposed mutation scheme. In this figure, the delay element that was incident out of v_3 became a multiplier incident out of v_5 , after the mutation.

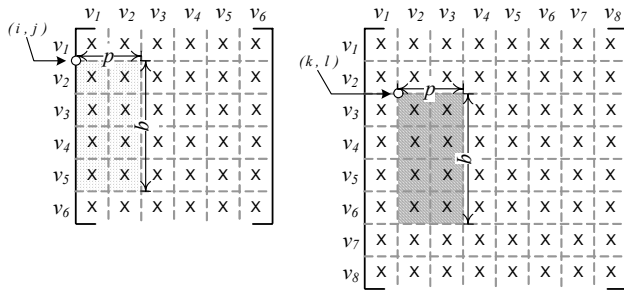


Figure 6. Crossover between two adjacency matrices.

5. Fitness Computation

The synthesis procedure is performed into two steps. In the first step, the equalizer transfer function is extracted from the desired filter transfer function. In this stage a genetic algorithm is used as an optimizer to adjust the coefficients of the equalizer transfer function (10). In the following, the evolutionary algorithm using the adjacency matrix representation is employed a low-sensitivity topology for the extracted equalizer transfer function. The input data for the first step are: the filter transfer function to be equalized and the desired group delay quality factor Q of the overall structure. The quality factor Q is a measure of the ripple in the passband group delay τ_{FE} being giving by [18]:

$$Q = \frac{100 \cdot (\max \tau_{FE} - \min \tau_{FE})}{(\max \tau_{FE} + \min \tau_{FE})} \quad (19)$$

In order to find the equalizer transfer function which meets the quality factor requirement, the algorithm runs an optimizer using a fixed length genetic algorithm. Starting with a single second-order section, the optimizer will eventually insert additional second-order sections until the Q requirement is fulfilled.

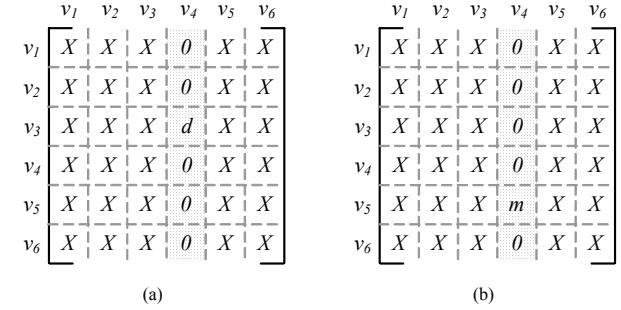


Figure 7. Adjacency matrix representation of an individual before (a) and after (b) mutation.

In the second step, the topology synthesis, a generational genetic algorithm with elitism is employed. The fitness computation verifies if an individual generated by the genetic process synthesizes the target equalizer transfer function.

As a fitness computation example, consider the target equalizer transfer function for a lowpass filter with normalized passband inside the interval $0 \leq \omega \leq 0.2$ given by:

$$H_E(z) = \frac{0.5z^2 + z + 1}{z^2 + z + 0.5} \quad (20)$$

Assume that the second-order direct form structure in Fig. 16(a) is generated by the evolutionary process. In the direct-form, the multipliers are the coefficients of the transfer function. The adjacency matrix corresponding to the given digital filter structure is shown in Fig. 16(b). Each column of the adjacency matrix, except the first one, represents one equation of the system in Fig. 16(c).

In the actual procedure, a symbolic mathematical package is used to perform the operations involved in the fitness computation, starting with the extraction of the symbolic equalizer transfer function from the text file containing the system of equations of Fig. 16(c).

The symbolic equalizer transfer function corresponding to the structure in Fig. 16(a) is given by:

$$H_E(z) = \frac{v_{10}}{v_1} = \frac{m_1 z^2 + z + 1}{z^2 - m_2 z - m_3} \quad (21)$$

The multipliers values ($m_1 = 0.50$, $m_2 = -1.00$, $m_3 = -0.50$) are obtained by solving the system of equations that results from equating the coefficients in (20) and (21). If this system has no real solution, zero fitness is assigned to the individual.

In the following step, the relative sensitivities [20] of the transfer function magnitude, $S_{m_j}^{|H_E|}$, and group delay, $S_{m_j}^{\tau_E}$ with respect to the multipliers are computed by replacing $z = e^{j\omega}$ in (21):

$$S_{m_j}^{|H_E|}(\omega) = \frac{m_j}{|H_E(e^{j\omega})|} \frac{\partial |H_E(e^{j\omega})|}{\partial m_j} \quad (22)$$

$$S_{m_j}^{\tau_E}(\omega) = \frac{m_j}{\tau_E(\omega)} \frac{\partial \tau_E(\omega)}{\partial m_j} \quad (23)$$

Where m_j is the j^{th} multiplier coefficient.

The previously obtained multipliers values are then replaced in equations (22) and (23) resulting in expressions for the sensitivities depending only on the frequency ω . Since no other symbolic computation is needed, the sensitivities values are sampled in 10,000 frequency points and stored in an array.

Fig. 17 illustrates, for the proposed example, the magnitude and the group delay relative sensitivities with respect to each multiplier versus the normalized frequency ω . Note that the magnitude sensitivity must be considered along the overall frequency range, whereas one needs to account for the group delay sensitivity only in the passband (shaded areas in Fig. 17).

The worst case magnitude and group delay sensitivities are, respectively, given by:

$$S_{wc}^{|H_E|} = \sum_{j=1}^{N_m} \max_{\omega} |S_{m_j}^{|H_E|}(\omega)| \quad (24)$$

$$S_{wc}^{\tau_E} = \sum_{j=1}^{N_m} \max_{\omega} |S_{m_j}^{\tau_E}(\omega)| \quad (25)$$

Where N_m is the number of multipliers.

The fitness is computed as the inverse of the function:

$$\varepsilon = 1 + w_{|H_E|} \cdot S_{wc}^{|H_E|} + w_{\tau_E} \cdot S_{wc}^{\tau_E} \quad (26)$$

Where ε is the objective function and the w_i 's are the weighting factors. The symbolic mathematical package returns to the evolutionary algorithm the value of the individual's fitness.

For the proposed example, the sensitivity values $S_{wc}^{|H_E|} = 4.3$ and

$S_{wc}^{\tau_E} = 1.5$ were obtained for $w_{|H_E|} = w_{\tau_E} = 1$ resulting in a fitness of 0.1471.

Besides the filter transfer function the inputs of the evolutionary process are: the maximum number of nodes, the desired interval for the multipliers values, the weighting factors and the usual genetic algorithm control parameters (population size, number of generations, crossover and mutation rates). The choice of the weighting factors is generally based on some kind of heuristics depending on expert knowledge of the design problem [26].

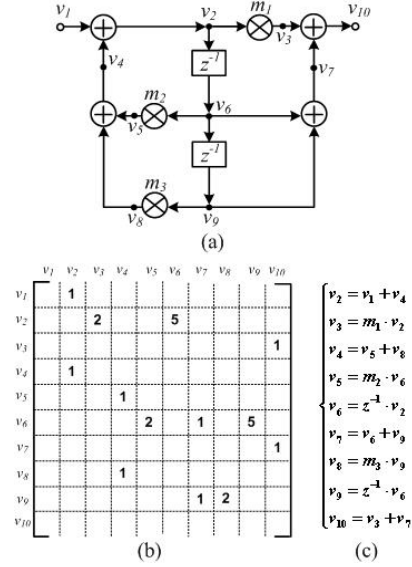


Figure 8. (a) Direct-form realization of the second-order filter of the example (b) corresponding adjacency matrix representation and (c) equations system derived from the adjacency matrix.

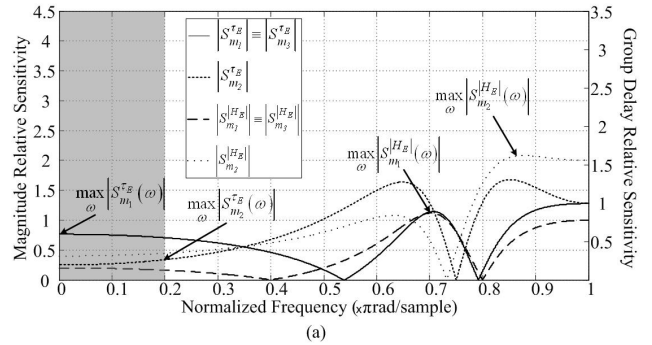


Figure 9. (a) Magnitude and Group delay relative sensitivity versus normalized frequency.

6. Numerical Results

Consider the design of a lowpass filter which must satisfy the magnitude specifications described in Table 3 and a quality factor lower than 2.50 in the passband. Using conventional methods [18], a fourth-order elliptic transfer function which meets the magnitude specifications is obtained:

$$H(z) = 1.1727 \times 10^{-2} \frac{z^4 - 2.8048z^3 + 3.8418z^2 - 2.8048z + 1}{z^4 - 3.6048z^3 + 4.9787z^2 - 3.1156z + 0.7447} \quad (27)$$

The magnitude and the group delay of the transfer function (27) are shown in Fig. 18, where the shaded areas indicate the passband. The quality factor Q , without equalization, is 64.61 in the passband.

The first step of the synthesis procedure, consisting of an optimizer, is run. Three sections of second-order allpass transfer functions with the coefficients described in Table 4 are cascaded with the elliptic filter, resulting in an overall quality factor of 2.10.

Table 3. Specifications for the elliptic filter

Passband ripple	1dB
Stopband attenuation	40dB
Passband edge	0.10 rad/s
Stopband edge	0.15 rad/s
Sampling frequency	2.00 rad/s

Table 4 - Coefficients of the second-order transfer functions

	Section 1	Section 2	Section 3
a_0	0.856502	0.861843	0.882595
a_1	-1.828910	-1.854456	-1.820200

The second step of the synthesis procedure which consists of the search for low-sensitivity structures using the adjacency matrix representation is employed. Table 5 shows the input parameters of the second step synthesis. The multipliers values are constrained to the interval $(-4, +4)$ in order to increase the number of bits representing their fractional part after the quantization process. Fig. 19 shows the average fitness and the best fitness throughout the generations for three sections of the equalizer. As shown in the figure, the algorithm responds well to genetic operators, increasing considerably the fitness along the generations. This suggests that the checking rules included for preventing the evaluation of invalid individuals after the genetic operators do not constrain the search space and do not reduce significantly the overall evolutionary algorithm performance.

The synthesized topologies after 50 generations are shown in Fig. 20. These structures are said to be canonic since the number of delays is equal to the order of the transfer function. Moreover, it uses a minimal number of multipliers that represents an important feature since the multipliers are the most expensive components in any digital filter implementation.

A comparison of the results obtained by the direct-form realization and the proposed topologies are given in Table 6. It can be observed that the smallest sensitivities were obtained with the topologies synthesized by the proposed evolutionary algorithm.

Fig. 21 shows the transfer function of (27) with a zoom in the passband after equalization using direct-form and synthesized topologies realizations for 12-bits fixed-point format. In the magnitude response, a nonlinear loss has been added to the passband in the direct-form realization and a slight deviation of the poles can be also observed. In the group delay response, the quality factors are 2.47 and 4.10 for synthesized topology and direct-form realizations, respectively. Therefore, due to the low-sensitivity of the synthesized topologies, the quality factor still meeting the desired quality factor specification.

7. Conclusions

An evolutionary algorithm able to synthesize low-sensitivity digital filters was proposed. It is shown that applying simple rules and requiring minimal computational effort, the proposed coding scheme representation has the property of detecting the generation of structurally invalid individuals and non-computable ones during the evolutionary process.

In order to test the algorithm, a forth-order lowpass digital filter was used as example and it was observed that the results

obtained by the proposed approach compare favorably with the straightforward direct-form realization.

8. REFERENCES

- [1] T. Inukai, "A unified approach to optimal recursive digital filter design," *IEEE Trans. Circuits Syst.*, vol. CAS-27, pp. 646-649, 1980.
- [2] C. Charalambous and A. Antoniou, "Equalization of recursive digital filters," in *Proc. Inst. Elec. Eng.*, vol. 127, part. G, no. 4, pp. 219-225, 1980.
- [3] A. G. Deczky, "Synthesis of recursive digital filters using the minimum p-error criterion," *IEEE Trans. Audio Electroacoust.*, vol. AE-32, pp.949-967, 1984.
- [4] T. Schnier, X. Yao, P. Liu, "Digital filter design using multiple pareto fronts", *Soft Computing* vol. 8, pp.332-343, Springer-Verlag, 2003.
- [5] N. E. Mastorakis, I. F. Gonos, and M. S. Swamy, "Design of two-dimensional recursive filters using genetic algorithms," *IEEE Trans. Circ. and Syst. I*, vol. 50, no. 5, pp. 634-639, May 2003.
- [6] R. Thumvichai, T. Bose, and R. L. Haupt, "Design of 2-D multiplierless IIR filters using the genetic algorithm," *IEEE Trans. Circ. and Syst. I*, vol. 49, no. 6, pp. 878-882, Jun. 2002.
- [7] M. Oner and M. Askar, "Incremental design of high complexity FIR filters by genetic algorithms," *Proc. Int. Symp. Signal Processing and Its Applications*, vol. 3, pp. 1005-1008, Brisbane Australia, 22-25 Aug. 1999.
- [8] A. Lee, M. Ahmadi, G. A. Jullian, W. C. Miller, and R. S. Lashkari, "Design of 1-D FIR filters using genetic algorithms," *Proc. IEEE Int. Symp. Circ. and Syst.*, vol. 3, pp. 295-298, 30 May-2 Jun. 1999.
- [9] K. S. Tang, K. F. Man, S. Kwong, and Z. F. Liu, "Design and optimization of IIR structure using hierarchical genetic algorithms," *IEEE Trans. Industrial Electronics*, vol. 45, No. 3, pp 481-487, Jun. 1998.
- [10] S. Kwong, Q. H. He, K. F. Man, K. S. Tang, and C. W. Chau, "Parallel genetic-based hybrid pattern matching algorithm for isolated word recognition," *Int. Jour. of Pattern Recog. and Artif. Intel.*, vol. 12, no. 5, pp. 573-594, Aug. 1998.
- [11] T. Arslan and D. H. Horrocks, "A Genetic algorithm for the design of finite word length arbitrary response cascaded IIR digital filters," *Proc. IEE Conf. of Genetic Algo. in Engineering Syst: Innov. and Applications*, pub. no. 414, pp. 276-281, 12-14 Sep. 1995.
- [12] M. Erba, R. Rossi, V. Liberali, and A. G. B. Tettamanzi, "Digital Filter Design through Simulated Evolution," *European Conference on Circuit Theory and Design*, vol. 2, pp. 137-140, 2001.
- [13] G. Tufte and P. C. Haddow, "Evolving an Adaptive Digital Filter", *Proc. of the 2000 NASA/DoD Conference on Evolvable Hardware*, IEEE Computer Press, pp. 143 - 150, 2000.
- [14] T. Kalganova and J. Miller, "Evolving more efficient digital circuits by allowing circuit layout evolution and multi-objective fitness," in *Proc. NASA/DoD Workshop Evolvable Hardware*, pp. 54-63, 1999.
- [15] J. Miller, "Digital filter design at gate-level using evolutionary algorithms," *Proc. of the Genetic and Evol. Comp. Conf. (GECCO'99)*, pp. 1127-1134, 1999.
- [16] K. Uesaka and M. Kawamata, "Evolutionary Synthesis of Digital Filter Structures Using Genetic Programming," *IEEE Trans. Cir. and Syst. II*, vol. 50, pp. 977-983, Dec. 2003.
- [17] K. Uesaka and M. Kawamata, "Synthesis of low-sensitivity second-order digital filters using genetic programming with automatically defined functions," *IEEE Signal Processing Lett.*, vol. 7, pp. 83-85, Apr. 2000.
- [18] A. Antoniou, "Digital Signal Processing: Signals, Systems, and Filters", McGraw-Hill, 2005.
- [19] R. C. Agarwal and C. S. Burrus, "New recursive digital filter structures having very low sensitivity and roundoff noise," *IEEE Trans. Circ. and Syst.*, vol. 22, pp. 921-927, Dec. 1975.

- [20] P. S. R. Diniz and A. Antoniou, "Low-sensitivity digital-filter structures which are amenable to error spectrum shaping," *IEEE Trans. Circuits Syst.*, vol. 32, pp. 1000-1007, Oct. 1985.
- [21] Y. V. R. Rao and C. Eswaran, "A Pole-Sensitivity Based Method for the Design of Digital Filters for Error-Spectrum Shaping," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1017-1020, Jul. 1989.
- [22] C. Barnes, "On the design of optimal state-space realizations of second-order digital filters," *IEEE Trans. Circuits Syst.*, vol. 31, pp. 602 - 608, Jul. 1984.
- [23] Sapargaliyev, Y.; Kalganova, T., "Constrained and Unconstrained evolution of LCR lowpass filters with oscillating length representation", IEEE Congress on Evolutionary Computation, pp. 1529-1536, 16-21 July 2006.
- [24] M. N. S. Swamy and K. Thulasiraman, *Graphs, Networks and Algorithms*, John Wiley & Sons, 1981.
- [25] S. Warshall, "A Theorem on Boolean Matrices", *Journal ACM*, vol. 9, pp. 11-12, 1962.
- [26] Zebulum, R. S.; Pacheco, M. A. C.; Vellasco, M. M. B. R.; "Evolutionary Electronics - Automatic Design of Electronic Circuits and Systems by Genetic Algorithms", CRC Press, 2001.

Table 5. Parameters of the Evolutionary Process

Population	200	Multipliers interval	(-4, +4)
Generations	50	Max. n° nodes	21
Crossover rate	70	$w H_E $	1
Mutation rate	20	w_{τ_E}	0.5

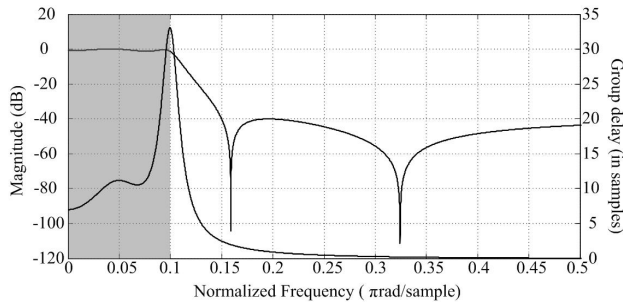


Figure 10. Magnitude and group delay response of the fourth-order elliptic filter.

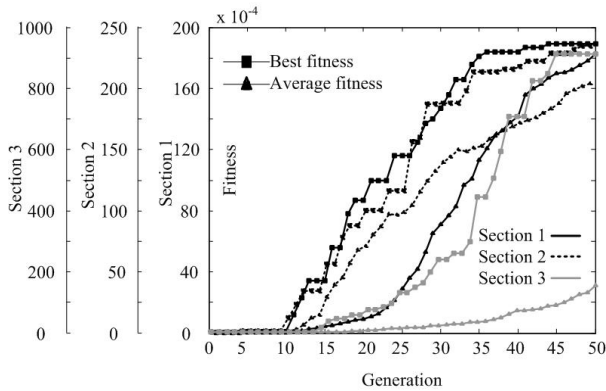


Figure 11. The average fitness and the best fitness of the proposed evolutionary algorithm along the generations.

Table 6. Sensitivities of the Equalizer Sections

	Section 1		Section 2		Section 3	
	$S_{wc}^{ H_E }$	$S_{wc}^{\tau_E}$	$S_{wc}^{ H_E }$	$S_{wc}^{\tau_E}$	$S_{wc}^{ H_E }$	$S_{wc}^{\tau_E}$
Direct-form	592.2	117.0	1470.9	371.7	182.0	92.2
Top. Fig. 12	45.9	5.9	33.7	8.0	7.8	3.4

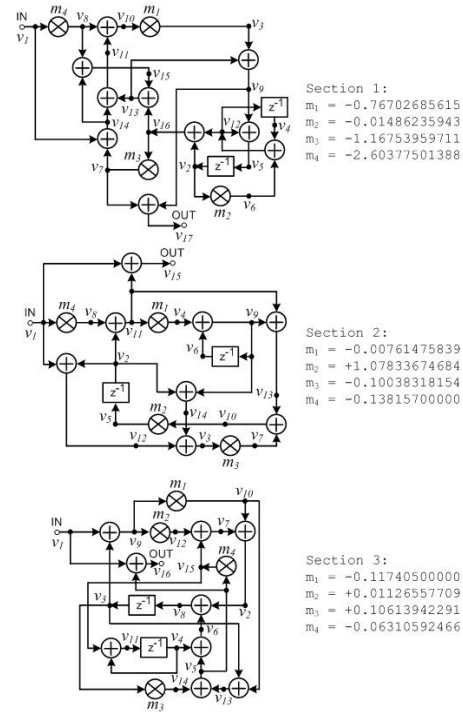


Figure 12. Topology synthesized by the proposed evolutionary process.

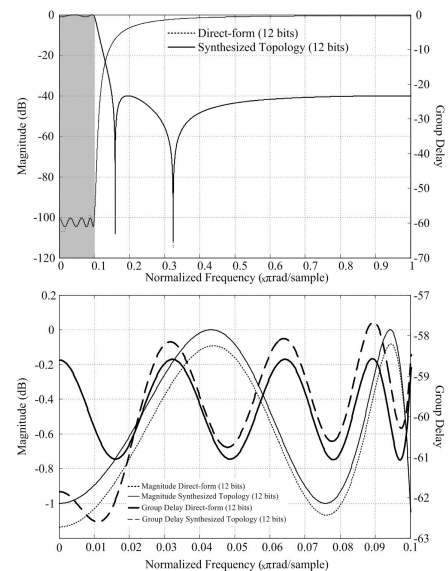


Figure 13. Transfer function responses.