Novel Design Method of Digital IIR Filters

PŘEMYSL ŽIŠKA, MILOŠ LAIPERT
Department of Circuit Theory
Czech Technical University in Prague
Technická 2, 166 27 Prague
CZECH REPUBLIC

Abstract: - A novel method of the IIR filter design using the genetic algorithm is described in the paper. Newly, a simplex direct search method is included to the algorithm to enable its faster convergence. The design procedure is implemented in the MATLAB program environment. The genetic algorithm applied is able to find a solution of the task successfully. Filter coefficients quantization process is included to the design procedure for realization in a digital signal processor with fixed point number presentation. This technique is illustrated on an example of a digital IIR pass-band filter design, which was simulated in the DSP TMS320C54x software simulator.

Key-words: - standard genetic algorithms, digital IIR filter, digital signal processors, magnitude frequency response

1 Introduction

The genetic and evolutionary algorithms were found out as a powerful tool in design and optimization of electrical circuits and systems. These algorithms are time-consuming, but, on the other hand, computationally robust and lead to the optimum results of the complicated design problems solution. Excellent results have been obtained e.g. in a transfer function approximation problem solution in many cases of design analog or digital filters. For example, in the recent paper [7] Storn has proposed a method for the design of a digital IIR filter with concurrent requirements for magnitude and group delay frequency response by using the Differential Evolutionary (DE) algorithms. Similarly, Martinek and Vondraš have dealt with the approximation of the analog filter transfer function with concurrent requirements for magnitude and group delay frequency response in the papers [2, 3].

Here, a simple method for IIR filter design using genetic algorithm will be presented. Such a method has been partly described in paper [8, 9], but now an efficiency of the method will be improved markedly using simplex direct search method. Efficiency of the described method will be compared with original standard genetic algorithm described in [9].

Generally the approximation of the filter transfer function represents a difficult mathematical problem. The commonly used design methods based on a usage of numerical methods for nonlinear equation system solving, need to estimate an initial approximation well to ensure convergence of the computed task. To avoid this, a simple method of the digital IIR filter design based on standard genetic algorithm is presented in this contribution. In general, the presented design method does not require the estimation of the initial condition.

Note that genetic algorithms are powerful global minimization algorithms which simulate an evolutional process in the nature, see [5]. They store a number of representations of solutions to a problem in the so called a 'population' matrix. The genetic algorithms can minimize not only a standard function, but also highly nonlinear and partly non-differentiable functions with many local minima. The disadvantage of genetic algorithms is the stochastic nature; it means, we are not able to predict the rate of convergence exactly, but some experiments show that the convergence process is relatively quick.

2 Improved Genetic Algorithm for Function Minimization

The standard genetic algorithm using floating-point number representation to find continuous parameters was proposed and programmed in the Matlab environment. The function of the algorithm can be described in the following block diagram in Fig. 1.

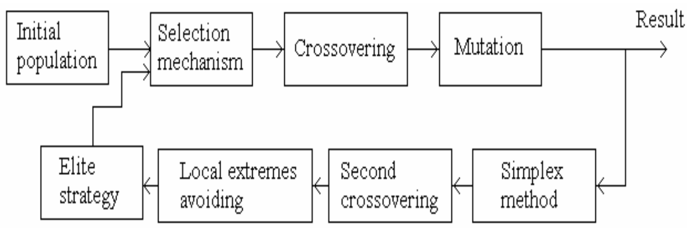

Fig. 1. Block diagram of the genetic algorithm

1. Generation of the initial population. This population is generated as NP x D matrix, where NP is number of members of the population and D is number of unknown

variables in the solved task. Each element of the matrix is generated as a random number with uniform probability distribution in the form:

$$x_{i,j} = \min + r \cdot (\max - \min) \quad (1)$$

where $i = 1...NP$, $j = 1...D$ and range of the variables is (min..max).

2. The selection mechanism. This algorithm uses a tournament selection. Two members of the population are chosen randomly and their values of the objective function are compared. A member which has lower value of the objective function is chosen. The selected member is inserted into the new population. The selection mechanism runs till the necessary number of members for creating the new population is not found.

3. Crossovering. For this algorithm an arithmetical crossover operator was used, which can be defined in the form:

$$x' = r \cdot x + (1 - r) \cdot y \quad (2)$$
$$y' = (1 - r) \cdot x + r \cdot y \quad (3)$$

where x is a randomly chosen member of the population, y is the best member corresponding with the best solution of the problem in previous algorithm running, r is random number with uniform probability distribution in the range (0..1) and x',y' are new offspring. Member x is chosen with uniform probability P_c. The advantage of this operator is that it does not need to check exceeding of the range of the found variables if there is solved the task with the limited variables.

4. Mutation operator. This operator is defined as follows: elements of the offspring are chosen with uniform probability P_m and then each of them is substituted by a number with uniform probability in the form:

$$x'(i) = \min + r \cdot (\max - \min) \quad (4)$$

where i is the position in the matrix NP x D and range of the variables is (min..max).

5. The simplex search method, see [10]. This is a direct search method that does not use numerical or analytic gradients. If n is the length of x, a simplex in n-dimensional space is characterized by the n+1 distinct vectors that are its vertices. In two-space, a simplex is a triangle; in three-space, it is a pyramid. At each step of the search, a new point in or near the current simplex is generated. The function value at the new point is compared with the function's values at the vertices of the simplex and, usually, one of the vertices is replaced by the new point, giving a new simplex. This step is repeated until the diameter of the simplex is less than the specified tolerance. The method can often handle discontinuity, particularly if it does not occur near the solution. As a starting vector for this method is used the best member corresponding with the best solution of the problem in previous algorithm running.

6. Second crossovering. This operator is defined by formulae (2), (3). But, x is a vector obtained using simplex method, y is the best member corresponding to the best solution of the problem in previous algorithm running, r is random number with uniform probability distribution in the range (0..1) and x',y' are new offspring in this operator. Both new offspring are included to the population matrix instead of two vectors with the worst value of an objective function.

7. Avoiding of convergence to local extremes. The simplex method may only give local solutions. Therefore, a special functional block has to be included to the modified genetic algorithm, which enables to avoid convergence of the algorithm to the local extremes. It works by the following principle: if the best value of minimization process does not change during performance of a preselect number of generations then a population vector, which appertain to the best value, it is saved to special matrix *Z*. Thus, population vectors, which denote local extremes, are saved in the matrix. Subsequently, new population matrix is generated by the principle described above for block 1. Population vectors, which correspond to the vectors values saved in the matrix *Z*, are penalized during evaluation of an objective function. It will be described in greater detail on practical example.

8. Sometimes the best offspring dies from the previous generation as the result of operation: selection, crossovering and mutation. Therefore, proposed algorithm uses the elite strategy. This strategy saves the best offspring, which is always inserted into the new population.

This process is repeated until an acceptable solution has been found or a preselect number of generations has been performed.

3 Application Task

To illustrate usability of the described algorithm to the digital IIR filter design, an example of the "full" design of the IIR digital band-pass filter will be presented.

The filter should answer the requirements to magnitude frequency response defined by band-pass filter specification in Fig. 3. An implementation on a digital signal processor is presumed. A cascade structure is preferred as the design method. It is important to mention the necessity of the filter transfer function coefficients quantization with respect to the fixed-point digital signal processor, see [1]. Moreover, it is usually requested to implement the cascade structure applying minimal memory and computing time requirements.

Therefore, a design procedure starts from the solution of the approximation problem leading to the identification of the transfer function $H(z)$ defined as a product of the second order transfer functions of the cascaded biquadratic sections in the form:

$$H(z)=\prod_{i}^{N} H_i(z)=\prod_{i=1}^{N}\frac{b_{0i}+b_{1i}\cdot z^{-1}+b_{2i}\cdot z^{-2}}{1-a_{1i}\cdot z^{-1}-a_{2i}\cdot z^{-2}} \qquad (5)$$

The biquadratic section can be realized by the "standard" canonical structure corresponding to the block diagram in Fig. 2.

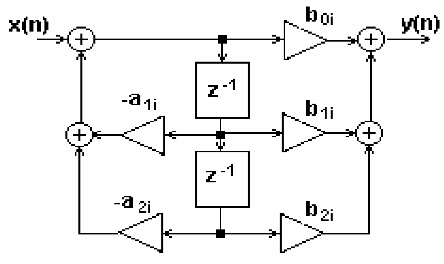

Fig. 2. Block diagram of the “biquadratic” section

The family of TMS320C54x digital signal processors was chosen for the IIR filter implementation. This processor uses an advanced Harvard-type architecture with common address space for the program memory and data memory. These C54x’ processors are 16-bit fixed point. Therefore, coefficient value must be converted to a *fraction* format of the number. If the number is expressed in twos complement form then maximum positive value is $1\text{-}2^{-k}$, where k is number of the used bits and minimum negative value is -1.

3.1 The digital IIR filter design procedure

As mentioned, the filter requirements are determined by the band-pass filter specifications under Fig. 3. These requirements to the magnitude frequency response can be accomplished using digital IIR filter with 6th-order transfer function proposed by applying the Cauer approximation.

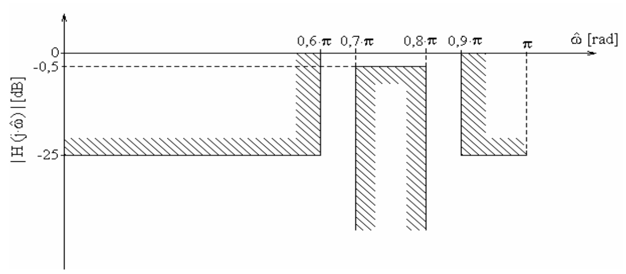

Fig. 3. The magnitude frequency response specification

Now, a design of the 6th-order digital filter transfer function using improved genetic algorithm will be proposed. A general 6th-order transfer function can be defined by formula:

$$H(z)=\frac{K\cdot\prod_{i=1}^{3}\left(1-z^{-1}\cdot r_{1i}\cdot e^{j\cdot\varphi_{1i}}\right)\cdot\left(1-z^{-1}\cdot r_{1i}\cdot e^{-j\cdot\varphi_{1i}}\right)}{\prod_{i=1}^{3}\left(1-z^{-1}\cdot r_{2i}\cdot e^{j\cdot\varphi_{2i}}\right)\cdot\left(1-z^{-1}\cdot r_{2i}\cdot e^{-j\cdot\varphi_{2i}}\right)} \qquad (6)$$

The resulting "realization form" of the IIR filter cascade structure transfer with the quantized coefficients is expressed by the formula (7)

$$H(z)=\prod_{i=1}^{3}\frac{b_{0i}+b_{1i}\cdot z^{-1}+b_{2i}\cdot z^{-2}}{1-a_{1i}\cdot z^{-1}-a_{2i}\cdot z^{-2}}\ . \qquad (7)$$

As evident, the cascade filter coefficients must be calculated at first and then these coefficients $a_{k,i}$, $b_{k,i}$ must be modified, because of coefficients quantization, by the terms

$$a_{k,i}=\frac{round\left(a_{k,i}\cdot 2^{k-1}\right)}{2^{k-1}};\quad b_{k,i}=\frac{round\left(b_{k,i}\cdot 2^{k-1}\right)}{2^{k-1}}. \qquad (8)$$

A fundamental of the presented procedure is the BP transfer function complex poles computation with respect to the prescribed magnitude frequency response under the filter specifications. The mathematical formulation of the objective function is

$$F(\underline{x})=H_1+H_2+H_3+H_4+\sum_{i=1}^{7}P_i\ , \qquad (9)$$

where $H_i,\ i=1,2,3,4$ are partial objective functions.

Population members are represented by vector $\underline{\mathbf{x}}$ = [x_1 ,..,x_{13}]. In the case of transfer function $H(z)$ vector $\underline{\mathbf{x}}$ is mapped as

$x_1=r_{11}$	$x_7=\varphi_{11}$
$x_2=r_{12}$	$x_8=\varphi_{12}$
$x_3=r_{13}$	$x_9=\varphi_{13}$
$x_4=r_{21}$	$x_{10}=\varphi_{21}$
$x_5=r_{22}$	$x_{11}=\varphi_{22}$
$x_6=r_{23}$	$x_{12}=\varphi_{23}$
$x_{13}=K$	

The functions H_1, H_2, H_3, *and* H_4 are defined (for solved example, the whole magnitude frequency response is sampled at 128 equidistant points) as:

$$H_1 = \sum_{k=0}^{76} \begin{cases} \left|\left(H_{dB}(\hat{\omega}_k) - a_s\right)\right| & if \quad H_{dB}(\hat{\omega}_k) - a_s > 0 \\ 0 & otherwise \end{cases}, \quad (10)$$

$$H_2 = \sum_{k=90}^{102} \begin{cases} \left|\left(a_p - H_{dB}(\hat{\omega}_k)\right)\right| & if \quad a_p - H_{dB}(\hat{\omega}_k) > 0 \\ 0 & otherwise \end{cases}, \quad (11)$$

$$H_3 = \sum_{k=115}^{128} \begin{cases} \left|\left(H_{dB}(\hat{\omega}_k) - a_s\right)\right| & if \quad H_{dB}(\hat{\omega}_k) - a_s > 0 \\ 0 & otherwise \end{cases}, \quad (12)$$

$$H_4 = \sum_{k=76}^{115} \begin{cases} \left|H_{dB}(\hat{\omega}_k)\right| & if \quad H_{dB}(\hat{\omega}_k) > 0 \\ 0 & otherwise \end{cases}, \quad (13)$$

where a_p is passband loss and a_s is stopband loss. The magnitude frequency response is calculated using the term

$$H_{dB}(\hat{\omega}) = 20 \cdot \log_{10} \left|H(z)\right|_{z=e^{j \cdot \hat{\omega}}}, \quad (14)$$

where the variable $\hat{\omega}$ means normalized frequency

$$\hat{\omega} = \omega \cdot T. \quad (15)$$

T labels the sampling interval.

P_k are the penalty functions, which can be computed by:

$$P_1 = \sum_{i=1}^{6} \begin{cases} 30000 + 200 \cdot x_i & if \quad x_i \geq 1 \\ 0 & otherwise \end{cases} \quad (16)$$

$$P_2 = \sum_{i=1}^{6} \begin{cases} 30000 - 200 \cdot x_i & if \quad x_i < 0 \\ 0 & otherwise \end{cases} \quad (17)$$

$$P_3 = \sum_{i=7}^{12} \begin{cases} 30000 + 200 \cdot x_i & if \quad x_i > \pi \\ 0 & otherwise \end{cases} \quad (18)$$

$$P_4 = \sum_{i=7}^{12} \begin{cases} 30000 - 200 \cdot x_i & if \quad x_i < 0 \\ 0 & otherwise \end{cases} \quad (19)$$

$$P_5 = \begin{cases} 30000 + 200 \cdot x_{13} & if \quad x_{13} > 1 \\ 0 & otherwise \end{cases} \quad (20)$$

$$P_6 = \begin{cases} 30000 - 200 \cdot x_{13} & if \quad x_{13} < 0 \\ 0 & otherwise \end{cases} \quad (21)$$

$$P_7 = \sum_{j=1}^{D} \begin{cases} \sum_{i=1}^{13} 200 + 100 \cdot x_i & if \ \forall x_i, x_{z_i} - \tau < x_i < x_{z_i} + \tau, \underline{x}_z \in Z \\ 0 & otherwise \end{cases} \quad (22)$$

where D is number of vectors in the matrix Z, $\underline{x}_z$ is vector, which denotes local extreme, saved in matrix Z. The τ is tolerance value.

Although the original standard genetic algorithm does not need penalty functions, usage of its combination with simplex method leads to needfulness of application these penalty functions.

The penalty function P_1 is included into the objective function to guarantee stability of the IIR digital filter. As is known, the radii of the complex poles must be located inside the unit circle in the complex plane of the variable *z*. The penalty function P_2 ensure positive values of radii and the penalty functions P_3, P_4 ensure the range of the phases of the poles. The penalty functions P_5, P_6 ensure correct value of amplification. The penalty function P_7 is designated for avoiding of the algorithm convergence to the local extremes. If the whole vector $\underline{\mathbf{x}}$ corresponds with vectors $\underline{x}_z$ saved in Z then such vector $\underline{\mathbf{x}}$ is penalized using the function P_7. Here we have used partly modified penalty functions published in [7].

The vector $\underline{\mathbf{x}}_{opt}$, for which the objective function $F(\underline{\mathbf{x}}) = 0$, is the wanted solution of the digital IIR filter design problem.

3.2 Results

The optimization task stated above for transfer function $H(z)$ was solved using the initial settings: NP=150, P_c=0.8, P_m=0.1, variables range: $x_1 - x_6$, $x_{13} \in (0..1)$, $x_7 - x_{12} \in (0..\pi)$, $\tau = 0.5$. Simplex method is always started after performing of first 40 generations. Then, it is started every time after 20 next performed generations (for instance 40, 60, 80, etc.). If the algorithm converges to the local extreme then the “Local extremes avoiding” block is run firstly. In that case, simplex method is started every time after 2 next performed generations (for example 202, 204, 206, etc.). The original standard genetic algorithm (described in [9]) found solution of the mentioned task after 46978 generations. New algorithm, which is described in this paper, is able to find solution much faster. The genetic algorithms exhibit stochastic nature. Therefore, the algorithm has been re-run repeatedly. It has been performed 100 experiments. The solution has been found successfully in all accomplished experiments. The fastest obtained solution, which has been found after 81 generations, is following:

$x_1=r_{11}=0.09648870641035$	$x_7=\varphi_{11}=0.80411123686141$
$x_2=r_{12}=0.76519031536948$	$x_8=\varphi_{12}=2.90680611153180$
$x_3=r_{13}=0.24816888604443$	$x_9=\varphi_{13}=1.33580857708929$
$x_4=r_{21}=0.92978008629822$	$x_{10}=\varphi_{21}=2.19435853956300$
$x_5=r_{22}=0.84162098398208$	$x_{11}=\varphi_{22}=2.38205436961830$
$x_6=r_{23}=0.92741595600869$	$x_{12}=\varphi_{23}=2.52162355501998$
$x_{13}=K=0.01907126335150$	

The slowest acquired solution has been found after 2097 generations. We have obtained after accomplishing of 100 experiments average value 336 generations needed for finding solution of the task stated above. It is in comparison with result from paper [9] very good value.

Important note: we have tried to use described algorithm without block denoted as “Local extremes avoiding”. We have obtained only 71 successfully found solution of 100 performed experiments in this example. Therefore, usage of this block is important.

The filter was simulated in the C54x’ signal processor software simulator where the filter was programmed in an assembly language. The evaluated magnitude frequency response of the proposed filter is shown in Fig. 4.

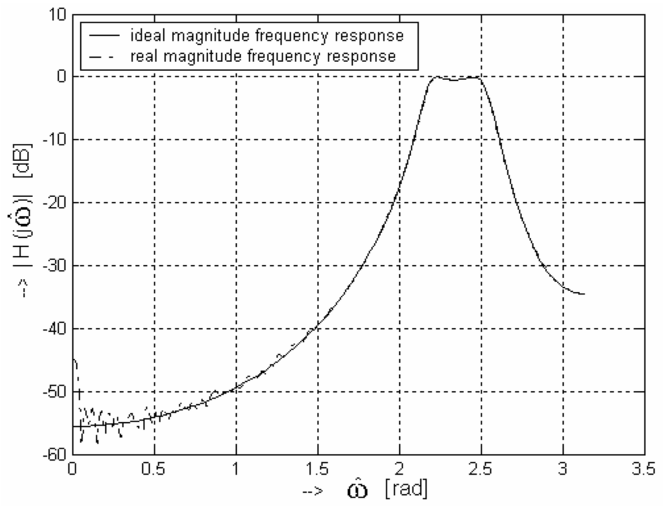

Fig. 4. The magnitude frequency response of filter with transfer function H(z)

The figure contains graph of ideal magnitude frequency response achieved as result of the IIR filter simulation in the MATLAB environment, and the real magnitude frequency response, which was achieved by mentioned filter simulations in the C54x’ signal processor software simulator. As can be seen, the real magnitude frequency response fully corresponds to the ideal magnitude frequency response.

4 Acknowledgement

The work reported in the paper has been supported by the grant GAČR No. 102/02/1067.

5 Conclusion

Usage of the combination of the genetic algorithm and simplex search method is new unconventional design method of the digital IIR filter. The band pass digital filter transfer function was proposed successfully. Resultant magnitude frequency response fully satisfies the pre-assigned tolerance band.

The described method is able to solve this class of the tasks effectively, even if the application of other numerical methods can lead to the convergence problems.

References:

[1] V. Davídek, M. Laipert, M. Vlček, *Analog and Digital filters, (Analogové a číslicové filtry in Czech)*, ČVUT Praha, 2000, pp. 180.

[2] P. Martinek, J. Vondraš, New Approach to Filters and Group Delay Equaliser Transfer Function Design, In: *ICECS 2001 - The 8th IEEE International Conference on Electronics, Circuits and Systems,* St. Julian’s: ICECS 2001, 2001, vol. 1, p. 70.

[3] P. Martinek, J. Vondraš, Multi-criterion Filter Design via Differential Evolution Method for Function Minimization, In: *ICCSC'02 1st IEEE International Conference on Circuits and Systems for Communications – Proceedings,* St. Petersburg: Saint-Petersburg State Technical University, 2002, vol. 1, p. 106-109, ISBN 5-7422-0260-1.

[4] V. Mařík, O. Štěpánková, J. Lažanský a kolektiv, *Artificial Intelligence (4), (Umělá inteligence (4) in Czech),* Vyd. 1., Academia Praha, 2003, pp. 172-192.

[5] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs,* 3rd edn, Springer-Verlag, Berlin Heidelberg, New York (1996).

[6] I. Zelinka, *Artificial Intelligence in Global Optimization Problems (Umělá inteligence v problémech globální optimalizace-in Czech),* Vyd. 1., BEN-technická literatura, Praha 2002, pp. 71-87.

[7] R. Storn, Differential Evolution Design of an IIR-Filter with Requirements for Magnitude and Group Delay, *Technical Report* TR-95-026, ICSI, May 1995.

[8] P. Žiška, M. Laipert, Design of Digital IIR Filters using Standard Genetic Algorithm, In *Proceedings of IFAC Workshop on Programmable Devices and Systems.* Gliwice: Silesian Technical University, 2004, s. 417-422. ISBN 83-908409-8-7.

[9] P. Žiška, P. Martinek, Nonstandard Design Method of Digital IIR Filter and its Comparison with Design of Digital IIR Filter using Cauer Approximation, In *Digital Technologies 2004*. Žilina: Technical University of Žilina, 2004, vol. 1, s. 38-43. ISBN 80-8070-334-5.

[10] J.C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions, *SIAM Journal of Optimization*, Vol. 9 Number 1, pp. 112-147, 1998.