Analog Filter Design Based on Evolutionary Algorithms

PRAVOSLAV MARTINEK
Department of Circuit Theory
Czech Technical University, Faculty of Electrical Engineering
Technická 2, Praha 6, 166 27 CZ
CZECH REPUBLIC

DAŠA TICHÁ
Department of Telecommunications, FE
University of Žilina
Univerzitná, Žilina, 010 26 SK
SLOVAK REPUBLIC

Abstract: This paper introduces some results, obtained by experiments with evolutionary algorithms in synthesis of passive LC filters. The aim of the presented experiments was to verify, if heuristic algorithms were able to successfully solve filter design problem including some optimization requirements in the mathematical program environment. As shown in some examples, MAPLE program seems to be a convenient tool for this task.

Key-Words: - LC filter, Filter synthesis, Evolutionary algorithms, Differential evolution, Optimization

1 Introduction

Passive-LC filter design based on theory of resistively terminated non-dissipative two-ports is rigorous and gives excellent results, when filter designed operates under "standard" operating conditions [1], [2]. But the filter specification frequently requires to include other additional requirements into design conditions, e.g. the filter component values spread minimization, sensitivity and dynamics optimization, influence of element losses, technological restrictions,... In another word, the design should be performed as multi-criteria, using suitable optimization procedure. Many optimization procedures and strategies have been developed for this purpose, but lately the inconsiderable attention is paid to the "heuristic" algorithms, particularly genetic and evolution algorithms. These ones are time consuming, but robust and reliable in the selecting of the best solution. With respect to the contemporary computer technique it is possible to apply such algorithms successfully on the standard PC under acceptable calculating time, even for relatively exigent tasks.

To verify usability of genetic and evolutionary algorithms in analog filter design, some of these algorithms were implemented in the MATLAB and MAPLE mathematical programs and used in design procedure, for the first time in the case of passive-LC filter synthesis. The results obtained will be discussed in the following parts of our contribution.

2 Algorithms used

Evolutionary Algorithms (EA) simulate evolution in nature. Algorithms are “heuristic” and operate with a “population” of possible solutions. The key idea is “generate and test”. With respect to this, a general form of these algorithms can be expressed as follows:

1. *Generate and evaluate an initial population P (collection of candidate solutions).*
2. *Produce and evaluate a collection of new individuals P’ by making randomized changes to selected members of P.*
3. *Replace some of the members (individuals) of P with some of the members (individuals) of P’ → create new population.*

4. *Evaluate the new population and return to the 2nd step (unless some termination criterion has been reached).*

Each individual (candidate solution) is represented by a vector of object variables. The evolution is based on survival of the fittest. The individuals are evaluated and evaluation result (fitness) determines, if are able to be members of the following generation. The new individuals arise by mutation and crossover or recombination of the "parents", i.e. individuals from the previous generation. The basic form of evolutionary algorithm can be written in the form (see Ref.[4])

G = 0

initialize P_G : $P_G := \{\vec{x}_{1,G}, \vec{x}_{2,G}, \ldots, \vec{x}_{NP,G}\}$

evaluate P_G : $\{f(\vec{x}_{1,G}), f(\vec{x}_{2,G}), \ldots, f(\vec{x}_{NP,G})\}$

do (until termination criteria are met or $G = G_{\max}$)

$P'_G = mutate\ (P_G)$

$P''_G = recombine\ (P'_G)$

evaluate $P''_G = \{f(\vec{x}''_{1,G}), f(\vec{x}''_{2,G}), \ldots, f(\vec{x}''_{NP,G})\}$

select P_{G+1} *from* P''_G *and* P_G

$G = G + 1$

end do

Mutation and recombination are carried out under selected algorithm strategy; in the simplest case the mutation is done by replacement of randomly chosen individual's parameters by random numbers, and recombination (crossover) using formula

$$\begin{aligned}\vec{x}_{12} &= \left(r\vec{x}_1 + (1-r)\vec{x}_2\right)\\ \vec{x}_{21} &= \left(r\vec{x}_2 + (1-r)\vec{x}_1\right)\end{aligned} \tag{1}$$

where vectors $\vec{x}_1, \vec{x}_2$ represent "parents" and $\vec{x}_{12}, \vec{x}_{21}$ offspring. Symbol r labels random number, $r \in \langle 0,1 \rangle$. As mentioned, selection of the ongoing individual - vector of object variables - depends on its evaluated fitness ("cost").

A significant improvement of the computation efficiency can be achieved using the Differential Evolution (DE) Algorithms. These algorithms presented by Price [4] use mutation with the differences of randomly sampled pairs of members P_G and different versions are distinguished by the crossover schemes they use.

The main features of DE algorithms can be characterized as follows:

1. Initial population is generated randomly using formula (2)

$$\begin{aligned}&P_G = \{\vec{x}_{1,G}, \vec{x}_{2,G}, \ldots, \vec{x}_{NP,G}\},\ NP \geq 4;\\ &\vec{x}_{i,G} = \{x_{1,i,G}, x_{2,i,G}, \ldots, x_{j,i,G}, \ldots, x_{D,i,G}\},\\ &x_{j,i,G} = x_j^{(lo)} + r_j\left(x_j^{(hi)} - x_j^{(lo)}\right);\\ &i = 1,2,\ldots NP;\ \ j = 1,2,\ldots D\end{aligned} \tag{2}$$

   where $\vec{x}_{i,G}$ labels the i[th] object variable vector containing D object variables, population *P* is created by NP members.

2. An auxiliary vector $\vec{v}_{i,G+1}$ corresponding to the mutation and crossover operation is gained by

$$\begin{aligned}&\vec{v}_{i,G+1} = \vec{x}_{r3,G} + F.\left(\vec{x}_{r1,G} - \vec{x}_{r2,G}\right);\\ &r1, r2, r3 \in \{1,2,\ldots,NP\};\ r1 \neq r2 \neq r3 \neq i\end{aligned} \tag{3}$$

   and the "trial" vector $\vec{u}_{i,G+1}$ is formed using criterion applied to all the object variables:

   if $rand_j(0,1) < CR$

   then $u_{j,i,G+1} = v_{j,i,G+1}$, else $u_{j,i,G+1} = x_{j,i,G}$.

3. A selection of the offspring into new population is accomplished using decision process

$$\vec{x}_{i,G+1} = \begin{cases}\vec{u}_{i,G+1} & \text{if } f\left(\vec{u}_{i,G+1}\right) \leq f\left(\vec{x}_{i,G}\right)\\ \vec{x}_{i,G} & otherwise\end{cases} \tag{4}$$

Note that *CR* and *F* are user-specified control variables. Schematic representation of the algorithm, published in Ref. [4], is shown in Fig. 1.

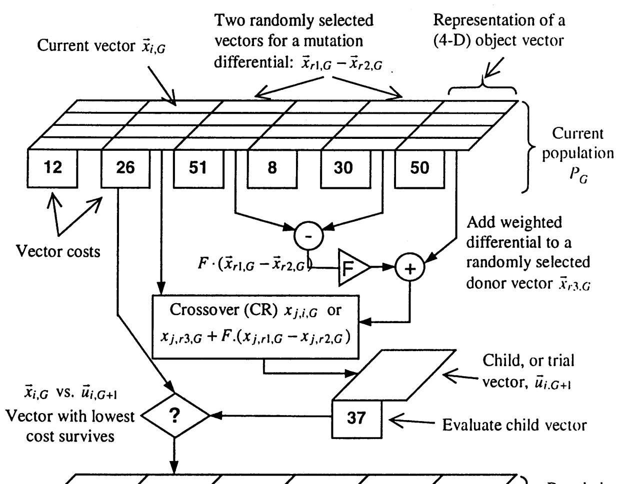

Fig. 1.: Schematic representation of DE algorithm.

Simultaneously to the presented basic form of DE algorithm many alternative options were developed. The main difference is usually in mutation and recombination scheme. The interesting versions used in our experiments differ each other in the form of auxiliary vector $\vec{v}_i$ construction:

A) The "basic" DE version under formula (3).

B) The "best" DE version, where

$$\vec{v}_i = \vec{x}_{best} + F.\left(\vec{x}_{r1} - \vec{x}_{r2}\right). \tag{5}$$

C) The generalized algorithm 1 (corresponding to the "*DE/current-to-rand/1"* version from [4])

$$\vec{v}_i = \vec{x}_i + K\left(\vec{x}_{r3} - \vec{x}_i\right) + F\left(\vec{x}_{r1} - \vec{x}_{r2}\right) \tag{6}$$

D) The modified algorithm C):

$$\vec{v}_i = \vec{x}_i + K\left(\vec{x}_{best} - \vec{x}_i\right) + F\left(\vec{x}_{r1} - \vec{x}_{r2}\right) \tag{7}$$

E) The second modification of algorithm C):

$$\vec{v}_i = \vec{x}_{best} + K\left(\vec{x}_i - \vec{x}_{best}\right) + F\left(\vec{x}_{r1} - \vec{x}_{r2}\right) \tag{8}$$

Some experience with them will be introduced in the following.

With respect to the filter design, object variables correspond to the filter circuit elements; fitness is evaluated using objective function, usually defined as a weighted sum of deviations of individual's frequency response (or other suitable parameters) from the given filter requirements.

3 Design Procedure

At first let us remind that the usage of evolutionary algorithms in digital IIR filter design under multi-criteria requirements has been presented firstly by Storn in [5]. Similar problems, concerning analog filter design, have been discussed and solved e.g. in Ref. [6], [7].

Contrary to the aforementioned works, the presented procedure solves only the synthesis stage of filter design and starts from the given transfer function. Such approach is more efficient and saves much computing time in comparison to the programs, forming the approximation stage and filter synthesis together as the only algorithmic block.

As mentioned, the idea of synthesis procedure concerns about a minimization of designed filter transfer function errors using EA and DE algorithms.

It starts from the given transfer function parameters and "blind" schematic diagram of filter designed. No initial conditions are premised, the initial population is chosen randomly. To make computations fast, symbolic transfer function of the filter designed is primarily found using the SYRUP library of MAPLE. This symbolic form is subsequently used as a basis for all the necessary numerical computations.

The key problem is in an effective evaluation of transfer function errors. Two ways are possible:

- an enumeration of transfer function poles and zeroes deviations,
- an enumeration of deviations of the numerator and denominator coefficients.

The second way requires higher accuracy of necessary computations to achieve acceptable result, but seems to be simpler and faster. Hence, it was preferred in our experiments.

The other significant question presents objective function composition. The fundamental part of objective function can be defined as a weighted sum of transfer function coefficient deviations. To respect other design requirements as well, the objective function contains yet another part created by penalty functions Px_j. With respect to this, a general notation of the objective function can be written in the form

$$fit = \sum_{k=1}^{m} w_k \left|d_k\right| + \sum_{j=1}^{p} Px_j \ , \tag{9}$$

or

$$fit = \sum_{k=1}^{m} w_k d_k^2 + \sum_{j=1}^{p} Px_j \ ; \tag{10}$$

$$d_k = \frac{a_{kc} - a_{ki}}{a_{ki}} . \tag{11}$$

Here a_{kc} means the k^{th} coefficient of the transfer function of evaluated individual, a_{ki} the corresponding coefficient of the given transfer function and d_k relative error. Coefficient errors are weighted by parameters w_k.

Although the filter element values (i.e. object variables) are limited when initial population is generated, it is necessary to cross-check it after mutation and crossover operations. For this purpose each auxiliary vector $\vec{v}_i$ is tested and unsuitable object variable values are replaced by random number, generated under formula (2).

4 Results Achieved

The discussed design procedure was applied to the more normalized LP filter specifications from the 3^{rd} to the 10^{th}-order, corresponding to the Chebyshev or Cauer approximation function. Testing transfer functions were generated using the MAPLE library SYNTFIL [3]. The first experiments using simple form of Evolutionary Algorithm were not successful; acceptable results were obtained in the case of lower-order filters only. As the highest, the 5^{th}-order Chebyshev LP filter was designed. In this case, the result was gained after 10000 populations, NP = 40, coefficient errors $\delta_k \approx 10^{-4}$.

On this account, the following experiments were oriented to the use of DE algorithms. Here design of the same filter using the "basic" version A) required only 950 populations, $NP = 20$, coefficient errors $\delta_k \approx 10^{-7} - 10^{-9}$. This version was successfully used up to the 10^{th}-order of Chebyshev transfer function. The proper design required 8000 populations, $NP = 60$ and control parameter setting $F = CR = 0.8$. Under these conditions the coefficient errors dropped below $\delta_k < 10^{-4}$.

An important role in the convergence rate plays a form of objective function. The best results were obtained using notation (10), corresponding to the sum of quadratic deviations. In comparison to the version (9) the "quadratic" objective function equalizes coefficient error values and leads to the faster convergence.

The excellent results were obtained using the version B) of DE algorithm. For filter order $n = 10$ the required number of populations dropped to $G = 3000$ under the same conditions as in the "basic" version.

For illustration, the following Table 1. shows the comparison of the 10^{th}-order Chebyshev filter element values computed by "exact" design procedure and using DE algorithm under the same filter operating conditions ($R_G = 1$, $R_L = 0.376$).

Element	***catalogue value***	***DE Algorithm***	***δ*** (%)
C1	*2.1840*	*2.183698204*	*-0.0138*
L2	*1.1210*	*1.121421741*	*0.0376*
C3	*3.1290*	*3.128978136*	*-0.0007*
L4	*1.1930*	*1.193359385*	*0.0301*
C5	*3.1890*	*3.188774565*	*-0.0071*
L6	*1.1990*	*1.198908058*	*-0.0077*
C7	*3.1740*	*3.173662332*	*-0.0106*
L8	*1.1760*	*1.176303737*	*0.0258*
C9	*2.9820*	*2.982112425*	*0.0037*
L10	*0.8201*	*0.8209745128*	*0.1066*

Table 1.: Design results comparison

As evident, the differences of element values between the both procedures are negligible.

Similar results were obtained using D) and E) DE algorithm versions. Both the versions show approximately the same results from computation efficiency point-of-view, but require careful setting of constants F and CR. Convergence process is demonstrated in Fig. 2, showing minimization of objective function in dependence on the number of iterations (generated populations).

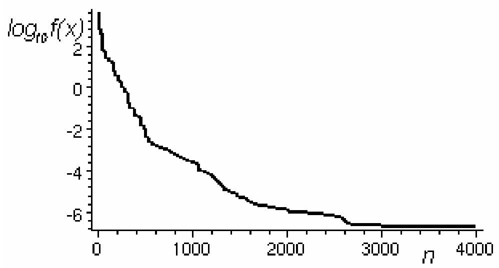

Fig.2.: Objective function minimization

A trouble-free design is likewise in the case transfer functions with finite transfer zeroes, e.g. corresponding to the Cauer or Inverse Chebyshev approximations. Solved example of the 6^{th}-order Cauer filter C 06 20b 49 (under filter catalogue) led to full minimization of objective function and transfer function coefficient errors $\delta_k < 10^{-9}$. Design procedure is insensitive to the non-standard filter termination, naturally in the boundaries given by physical principles of signal-power-transfer. With respect to the computation efficiency, approximately equal results were obtained using B), D) and E) versions of algorithms tested; version C) worked slowly, with relatively high residual value of objective function. A typical design parameters: $NP = 60$, $G = 2500$, $\delta_k < 10^{-9}$ for setting $K = \text{rand}(0.1 .. 0.95)$, $F = 0.65$, $CR = 0.9$.

Presented simple versions of design algorithms make easily possible optimization with respect to the filter element losses. The losses are expressed using Q-factor defined by known formula

$$Q_L = \frac{wL}{R}; \qquad Q_C = wCR, \tag{12}$$

where R characterizes losses, ω corresponds to the passband corner frequency (LP, HP), or passband center frequency (BP). For the prescribed losses it is simple to express R values as a function of Q and L (or C). This approach does not increase number of object variables, when calculation of transfer function coefficients is firstly made in symbolic form. Some examples of Chebyshev LP filters from the 5^{th}- to the 10^{th}-order proved ability of algorithm used to minimize losses influence in wide range; transfer function errors achieved were under $\delta_k \approx 10^{-5} - 10^{-9}$. As an example, design of the 6^{th}-order Chebyshev LP, $a_p = 1$dB, with prescribed inductor losses $Q_L = 10$ finished with coefficient errors $\delta_k < 10^{-9}$ after 1940 iterations, when $NP = 28$, $CR = F = 0.8$ (algorithm B) used).

As mentioned, an optimum efficiency of DE algorithms depends on a suitable setting of user-defined control constants and a population size. Systematic testing revealed relative insensitivity to the constant *CR* and *F* setting in the case of algorithms A), B). "Standard" values of *F* were in the range $F \in (0.6;0.9)$, similarly for *CR*.

Algorithms C), D), E) are more sensitive to the control variables setting. The best results give $CR \to 1$ in combination with $F \in (0.65;0.8)$. Higher values of *F* slows convergence rate down, lower values leads to the convergence stagnation without optimum solution reaching. In general, lower values of *CR* significantly increase number of iterations necessary for optimum result acquirement.

The control parameter ***K*** should be chosen randomly for each evaluation, $K \in (0;1)$.

The population size is important from the effectivity of computations as well. The optimum size was found in the range $NP \in (7D;12D)$. Higher size causes weak convergency and increases number of iterations; lower behaves similarly, but, in addition, leads to the higher probability of computation stagnancy.

5 Conclusion

The aim of our work has been primarily oriented to the systematic verification and comparison of the particular options EA and DE algorithms efficiency in analog filter synthesis, optimum setting of user-defined control variables and a formulation of an appropriate form of objective function. The results achieved made out good applicability and wide extent of solved tasks. Note that some of them are hardly solvable using "conventional" methods, e.g. design of filters with distributed losses [8]. Relative disadvantage of heuristic algorithms, which is computing time demandingness, can be partially suppressed using main procedure translation into C- or other similar “machine” language.

References:

[1] R.Schaumann, M.S.Ghausi, K.R.Laker, *Design of Analog Filters. Passive, Active RC, and Switched Capacitor,* Prentice Hall, 1990.

[2] M.Hasler, J.Neirynck, *Electric Filters,* ARTECH HOUSE, Inc., 1986

[3] J.Hospodka, J.Bičák, SYNTFIL – Synthesis of Electric Filters in MAPLE, *Proc. of the Maple Summer Workshop 2004*, Maplesoft, Waterloo, ON, 2004.

[4] D.Corne, M.Dorigo, F.Glover, *New Ideas in Optimization,* Mc Graw Hill, 1999.

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

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

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

[8] K.Hájek, J.Sedláček, B.Sviezeny, Improving of the Active RC and SC Filters by Use of Goal-Directed Lossy LC Ladder Prototypes, *Proceedings of ECCTD'03 – European Conference on Circuit Theory and Design.* Cracow, Poland, 2003, vol.1, p.I-381 – I-384