Design and Optimization of IIR Digital Filters with Non-standard Characteristics Using Continuous Ant Colony Optimization Algorithm

Adam Slowik and Michal Bialko

Department of Electronics and Computer Science, Koszalin University of Technology, Sniadeckich 2 Street, 75-453 Koszalin, Poland
aslowik@ie.tu.koszalin.pl

Abstract. In this paper method of design and optimization of stable IIR digital filters with non-standard amplitude characteristics using continuous ant colony optimization algorithm ACO_R is presented. In proposed method (named ACO-IIRFD) dynamical changes of parameters in designed filters are introduced. Due to these dynamical changes of filter parameters, design of IIR digital filters with small deviations between designed filter characteristics and assumed characteristics is possible. Three IIR digital filters with amplitude characteristics: linearly-falling, linearly-growing, and non-linearly-growing, which can have application in amplitude equalizers, are designed using proposed method.

1 Introduction

Methods based on evolutionary algorithms [1] are used to design of digital filters since several years. As an example we can mention following papers [2-5]. However, only few papers describe applications of ant colony optimization algorithm to design digital filters; one of them is article [6]. Generally, a very few number of articles in this subject is connected with fact, that ant colony optimization algorithms [7] concern global optimization with discrete domains. However, in the case of many optimization tasks, the domains are continuous (as for example in design of digital filters). To use the ant colony optimization algorithms to the problem of continuous optimization some modifications of these algorithms have been created in last years. These modifications are as follows: (1995) CACO (*Continuous Ant Colony Optimization*) [8], (2000) API [9], (2002) CIAC (*Continuous Interacting Ant Colony*) [10], and (2008) ACO_R [11]; the last algorithm (in opposition to other continuous ant colony optimization algorithms) maintains a general idea of ant colony optimization algorithms depending on step-by-step building of solution, and shows to be a very effective tool in continuous functions optimization, including multi-modal functions [11].

Generally, the transfer function $H(z)$ of designed IIR (n-th order) digital filter in z domain is described as follows:

$$H\left(z\right)=\frac{b_0+b_1\cdot z^{-1}+b_2\cdot z^{-2}+...+b_{n-1}\cdot z^{-(n-1)}+b_n\cdot z^{-n}}{1-\left(a_1\cdot z^{-1}+a_2\cdot z^{-2}+...+a_{n-1}\cdot z^{-(n-1)}+a_n\cdot z^{-n}\right)} \quad (1)$$

J. Darzentas et al. (Eds.): SETN 2008, LNAI 5138, pp. 395–400, 2008.

Stability of IIR digital filter depends on poles location of transfer function (1) in z domain. If all poles are inside unitary circle (it means $|z| = 1$), then filter is stable. The main goal in digital filter design is searching a such set of values of a_i, and b_i coefficients for given characteristic, that obtained filter fulfills design assumptions and is stable.

2 ACO-IIRFD Method

Method ACO-IIRFD (based on ACO_R algorithm [11]) is operating according to following steps: **In the first step**, the initial set T consisting of $SizeT$ potential solutions of given optimization problem is randomly created. Each solution t_i from the set T consists of $2 \cdot n + 1$ filter coefficients which are coded in the following order:

$$t_i = [b_0 \quad b_1 \quad b_2 \quad ... \quad b_{n-1} \quad b_n \quad a_1 \quad a_2 \quad ... \quad a_{n-1} \quad a_n] \tag{2}$$

that is: $t_{i,1}=b_0$, $t_{i,2}=b_1$, $t_{i,3}=b_2$, ..., $t_{i,2n}=a_{n-1}$, $t_{i,2n+1}=a_n$. The value of each filter coefficient variable $t_{i,j}$ ($j \in [1; 2 \cdot n+1]$) is in the range $t_{i,j} \in [Low_j; High_j]$; in this paper, it is assumed, that the initial values are: $High_j = 1$, $Low_j =$ -1. For each j-th in the i-th solution, its value is selected using following formula:

$$t_{i,j} = R \cdot (High_j - Low_j) + Low_j \tag{3}$$

where: R-pseudo-random number with uniform distribution in the range [0; 1). **In the second step**, each potential solution from the set T is evaluated using objective function $COST(.)$. The amplitude characteristics $H(f)$ is obtained:

$$H(f) = 20 \cdot log_{10} \left(\sqrt{H_{real}(f)^2 + H_{imag}(f)^2} \right) [dB] \tag{4}$$

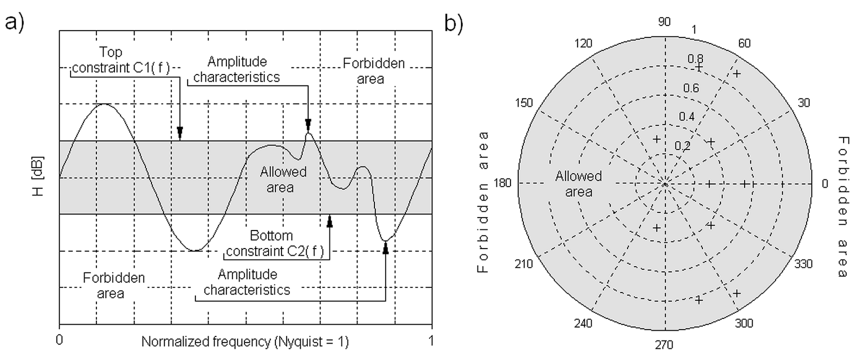

Fig. 1. Amplitude characteristics of designed digital filter with assumed constraints (a), poles of stable transfer function (b)

Accordingly to Figure 1, the objective function $COST(.)$ is defined as follows:

$$COST(.) = \sum_{i=1}^{k} Error\left(f_i\right) + \sum_{i=1}^{m} Stab_i \tag{5}$$

$$Error(f_i) = \begin{cases} \left|H(f_i) - C1(f_i)\right|, when \ H(f_i) > C1(f_i) \\ \left|H(f_i) - C2(f_i)\right|, when \ H(f_i) < C2(f_i) \\ 0, \qquad\qquad\qquad\quad when \ H(f_i) \in [C2(f_i); C1(f_i)] \end{cases} \tag{6}$$

$$Stab_i = \begin{cases} |z_i| \cdot w, when \ \ (|z_i| \geq 1) \\ 0, \qquad\quad when \ \ (|z_i| < 1) \end{cases} \tag{7}$$

where: m - number of poles of transfer function (m=n), k - number of frequency samples (k=256), w - value of penalty (assumed w=10^5), f_i - value of i-th normalized frequency, $|z_i|$ - module value of i-th pole of transfer function in z domain. The result of evaluation of i-th solution using function $COST(.)$ is stored in variable $eval_i$ assigned to each solution t_i from the set T. The ACO-IIRFD algorithm, presented in this paper minimizes the value of objective function $COST(.)$. **In the third step**, the solutions t_i from the set T are sorted according to $eval_i$ values. After sorting process, the best solution (with the smallest value of $eval$) is located under index number 1 in the set T, and the worst solution (with the highest value of $eval$) is stored under index number $SizeT$ in the set T. **In the fourth step**, the value of ω_i is computed for each solution t_i. This value determines, the selection chance of the i-th solution from the set T to creation of new solution (new set of a_i, b_i coefficients), and is higher when i-th solution is located under lower index in the set T. The value of ω_i is defined as follows:

$$\omega_i = \frac{1}{q \cdot SizeT \cdot \sqrt{2 \cdot \pi}} \cdot exp\left(-\frac{(i-1)^2}{2 \cdot q^2 \cdot SizeT^2}\right) \tag{8}$$

The variable ω_i is defined by a function with normal distribution for argument i, mean equal to 1.0, and standard deviation $q \cdot SizeT$, where $q \in [0;1]$ is the parameter of algorithm. In the case, when the value of q is small, then better solutions, that is, solutions having smaller values of indices after sorting process are stronger preferred (solutions located close to the best solution are selected more often), and in the case, when the value of q is large, then selection of solutions is taken from wider range with respect to the best solution. **In the fifth step**, each "ant" marked as an Ant_g ($g \in$[1; NoA], NoA represents number of ants, and it is a parameter of the algorithm) generates one new solution marked as the S_g. This operation is executed according to following schema, which is repeated separately for each "artificial ant":

a) choose one solution from the set T using roulette selection method [1] scaled by values ω_i, and remember its index in variable h;
b) compute value of standard deviation marked as the σ_j for each j-th variable of new created solution S_g using following formula:

$$\sigma_j = \xi \cdot \sum_{e=1}^{SizeT} \frac{|t_{e,j} - t_{e,h}|}{SizeT - 1} \tag{9}$$

where: ξ-real number higher than zero, and having similar sense as a coefficient of pheromone evaporation in discrete ant colony optimization algorithms. When the value of coefficient ξ is higher - the convergence of the algorithm is slower;
c) determine mean value marked as a μ_j for each j-th variable of new created solution S_g (the value of j-th variable in the solution S_g will be selected using normal distribution with mean value μ_j and standard deviation σ_j), using following formula: $\mu_j = t_{h,j}$;
d) the value of the j-th variable for new created solution S_g is defined as follows: $S_{g,j} = randn\,(\mu_j, \sigma_j)$, where: $randn(\mu_j,\, \sigma_j)$-random value with normal distribution having mean value μ_j, and standard deviation σ_j. **In the sixth step**, each new solution S_g is evaluated using function $COST(.)$, and the obtained value is assigned to variable $eval_g$. **In the seventh step**, the new obtained solutions are included into the solution set T, which size is increased to the value $SizeT$+NoA. Next, all solutions are sorted with respect to the value of parameter *eval* (the best solution with the smallest value *eval* obtains index number 1). Then, *NoA* last solutions are deleted from set T, and the original size of this set equal to *SizeT* is restored. **In the eighth step**, it checked whether $COST(t_1)$=0. If yes, then solution t_i is remembered as the *TheBest* solution, and the objective function $COST(.)$ is modified by decreasing the value of accepted deviations of amplitude characteristics, and then the new value $eval_i$ is computed for each solution t_i. If $COST(t_1)\neq 0$, then ninth step of the proposed algorithm is executed. **In the ninth step**, it is checked whether the result stored in solution t_1 has not been improved by d iterations of the algorithm. If the result stored in solution t_1 has not been improved, then the result stored in *TheBest* is returned, and the operation of algorithm is stopped. If the result stored in solution t_1 has been improved before reaching of d iterations of algorithm, then the third step of the ACO-IIRFD algorithm is executed.

3 Description of Experiments

Proposed method has been tested by the design of the 10 order (n=10) three IIR digital filters with amplitude characteristics: linearly growing (for normalized frequency f=0 the value of gain was equal to -40 [dB], for normalized frequency f=1 the value of gain was equal to 0 [dB]), linearly falling (for normalized frequency f=0 the value of gain was equal to 0 [dB], for normalized frequency f=1 the value of gain was equal to -40 [dB]), and non-linearly growing (the attenuation is represented by quadratic characteristics; for normalized frequency f=0 the value of gain was equal to -40 [dB], and for normalized frequency f=1 the value of gain was equal to 0 [dB]. The values of parameters in the ACO-IIRFD method were as follows: filter order n=10, size of solutions set *SizeT*=800, number of ants *NoA*=100, q=0.01, ξ=0.085, d=200. At the start was assumed that the deviation values of the attenuation characteristics from ideal case can not be higher than 10 [dB] at any frequency point. Then, during the algorithm operation the values of assumed attenuation deviations were decreased by 1 [dB] step, and after achieving the value of deviation equal to 1 [dB], the values of attenuation deviations were decreased by 0.1 [dB] step. The main purpose of the experiment was a design of 10-order digital filters having possibly lowest deviations of their amplitude characteristics from ideal characteristics.

In Figures 2a - 2c, the differences (deviations) between ideal characteristics *H*, and characteristics *H'* obtained using ACO-IIRFD method for designed filters are presented and in Figures 2d - 2f, the number of generations required to obtain designed digital filters with prescribed deviations of attenuation values between obtained characteristics, and ideal characteristics is shown.

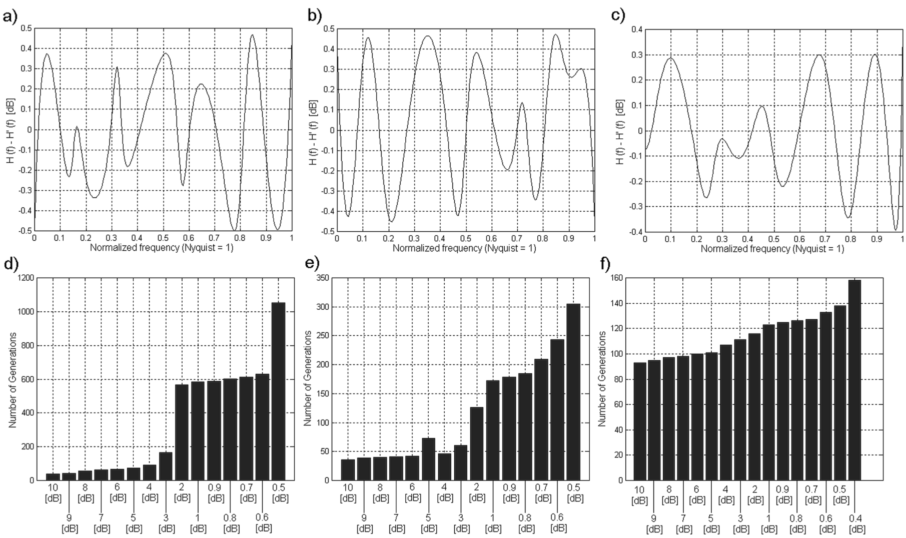

Fig. 2. Deviations between ideal characteristics, and characteristics obtained using ACO-IIRFD method for 10-order digital filters having characteristics: linearly growing (a), linearly falling (b), and non-linearly growing (c); Number of generations required to obtain designed digital filters with prescribed deviations of attenuation values of amplitude characteristics: linearly growing (d), linearly falling (e), and non-linearly growing (f)

It can be seen that in the case of filters with linearly growing (Figure 2a), and linearly falling (Figure 2b) characteristics, the deviations of attenuation values between obtained characteristics and ideal characteristics do not exceed +/- 0.5 [dB], however, in the case of filter with non-linearly growing characteristics (Figure 2c), these deviations of attenuation values do not exceed +/- 0.4 [dB]. All designed filters are stable (all poles are located inside unitary circle in the z plane). In performed experiments the ACO-IIRFD algorithm has been stopped after 1267 generations for linearly growing characteristics, after 728 generations for linearly falling characteristics, and after 452 generations for non-linearly growing characteristics. To obtain smaller deviations of attenuation values higher order filter is required.

4 Conclusion

It has been shown that it is possible to design and optimize digital filters with non-standard amplitude characteristics using the continuous ant colony optimization algorithm. Three digital filters designed using described ACO-IIRFD method fulfill all design assumptions and are stable. The full automation of the design and optimization process of digital filters with non-standard amplitude characteristics is possible with the use of proposed ACO-IIRFD method, and the expert knowledge concerning the filter design, and digital signal processing is not required. The ACO_R algorithm is newly developed technique (in year 2008) for continuous optimization based on ants colony. Because of that, this paper is probably the first application of this algorithm to design and optimization of stable IIR digital filters with non-standard amplitude characteristics.

References

1. Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs. Springer, Heidelberg (1992)
2. Erba, M., Rossi, R., Liberali, V., Tettamanzi, A.G.B.: Digital Filter Design Through Simulated Evolution. In: Proceedings of ECCTD 2001, Espoo, Finland, August 2001, vol. 2, pp. 137–140 (2001)
3. Slowik, A., Bialko, M.: Evolutionary Design of IIR Digital Filters with Non-Standard Amplitude Characteristics. In: 3rd National Conference on Electronics, Kolobrzeg, June 2004, pp. 345–350 (2004)
4. Nurhan, K.: Digital IIR filter design using differential evolution algorithm. EURASIP Journal on Applied Signal Processing 8, 1269–1276 (2005)
5. Nurhan, K., Bahadir, C., Tatyana, Y.: Performance comparison of genetic and differential Evolution algorithms for digital FIR filter design. In: Yakhno, T. (ed.) ADVIS 2004. LNCS, vol. 3261, pp. 482–488. Springer, Heidelberg (2004)
6. Karaboga, N., Kalinli, A., Karaboga, D.: Designing digital IIR filters using ant colony optimisation algorithm. Engineering Applications of Artificial Intelligence 17(3), 301–309 (2004)
7. Dorigo, M., Maniezzo, V., Colorni, A.: Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on SMC-B 26(1), 29–41 (1996)
8. Bilchev, G., Parmee, I.C.: The ant colony metaphor for searching continous design spaces. In: Fogarty, T.C. (ed.) AISB-WS 1995. LNCS, vol. 993, pp. 25–39. Springer, Heidelberg (1995)
9. Monmarche, N., Venturini, G., Slimane, M.: On how Pachycondyla apicalis ants suggest a new search algorithm. Future Generation Computer Systems 16, 937–946 (2000)
10. Dreo, J., Siarry, P.: A new ant colony algorithm using the heterarchical concept aimed at optimization of multiminima continous functions. In: Doringo, M., Di Caro, G., Samples, M. (eds.) Ant Algorithms 2002. LNCS, vol. 2463, pp. 216–221. Springer, Heidelberg (2002)
11. Socha, K., Doringo, M.: Ant colony optimization for continous domains. European Journal of Operational Research 185(3), 1155–1173 (2008)