CIRCUITS SYSTEMS SIGNAL PROCESSING
VOL. 25, NO. 5, 2006, PP. 649–660

DOI: 10.1007/s00034-005-0721-7

DESIGN OF DIGITAL FIR FILTERS USING DIFFERENTIAL EVOLUTION ALGORITHM*

Nurhan Karaboga[1] *and Bahadir Cetinkaya*[1]

Abstract. The differential evolution (DE) algorithm is a new heuristic approach with three main advantages: it finds the true global minimum of a multimodal search space regardless of the initial parameter values, it has fast convergence, and it uses only a few control parameters. The DE algorithm, which has been proposed particularly for numeric optimization problems, is a population-based algorithm like the genetic algorithms and uses similar operators: crossover, mutation, and selection. In this work, the DE algorithm has been applied to the design of digital finite impulse response filters, and its performance has been compared to that of the genetic algorithm and least squares method.
Key words: FIR filter design, differential evolution algorithm, genetic algorithm.

1. Introduction

Filtering is a process by which the frequency spectrum of a signal can be modified, reshaped, or manipulated according to some desired specifications. It may entail amplifying or attenuating a range of frequency components, rejecting or isolating one specific frequency component, etc. The digital filter is a digital system that can be used to filter discrete-time signals. The main advantages of digital filters are the traditional advantages associated with digital systems such as: (i) uncritical component tolerances, (ii) high accuracy, (iii) small physical size, and (iv) high reliability. A very important additional advantage of digital filters is the ease with which their characteristics can be changed or adapted by simply changing the contents of a finite number of registers. Because of this feature, digital filters are naturally suited for the design of programmable filters that can be used to perform a multiplicity of filtering tasks, and for the design of adaptive filters that can be used in a diverse range of applications, such as system identification, chan-

* Received July 21, 2005; revised February 13, 2006; published online October 16, 2006.
[1] Erciyes University, Faculty of Engineering, Department of Electronics Engineering, 38039, Melikgazi, Kayseri, Turkey. E-mail for Karaboga: nurhan_k@erciyes.edu.tr; E-mail for Cetinkaya: cetinkaya@erciyes.edu.tr

nel equalization, signal enhancement, and signal prediction. There are two main classes of digital filters: finite impulse response (FIR) filters and infinite impulse response (IIR) filters. FIR filters are simple to design and they are guaranteed to be bounded-input, bounded-output; namely, they are always stable filters. By designing the filter taps to be symmetrical about the center tap position, an FIR filter can be guaranteed to have linear phase [15].

Heuristic optimization algorithms such as genetic algorithms, tabu search, and simulated annealing algorithms have been widely used in the optimal design of digital filters [2]–[5], [7]–[14], [16], [20]–[23]. In the literature, there are several works about the application of genetic algorithms (GAs) to the digital FIR filter design [5], [13], [14], [22], [23]. In these works, it is emphasized that filters designed by a GA have the potential of obtaining near global optimum solutions. Although standard GAs perform well for finding the promising regions of the search space, they are not so successful at determining local minima in terms of convergence speed. In order to overcome this disadvantage of the GA in numeric optimization problems, a differential evolution (DE) algorithm has been introduced by Storn and Price [17]. The DE algorithm is a new heuristic approach having mainly three advantages; it can find the true global minimum of a multimodal search space regardless of the initial parameter values, it has fast convergence, and it uses only a few control parameters. The DE algorithm is a population-based algorithm like the GAs and uses similar operators: crossover, mutation, and selection. The studies on the design of optimal digital filters using the DE algorithm are not as common as those on the GAs. In the literature, there are only a few studies related to the application of the DE algorithm to digital filter design [9], [10], [16], [21]. In [16], the task of designing an IIR filter using a DE algorithm is investigated. In [21], the article proposes a solution to the problem of multicriterion filter design. In contrast to the standard method, the DE method allows one to solve problems of filter design with some sorts of limitations. The works presented in [9], [10] describe the design of digital FIR filters based on the DE algorithm.

In this work, a performance comparison of design methods based on the DE algorithm, GA, and least square algorithm (LSQ) is presented for digital FIR filters as the DE algorithm is very similar to, but much simpler than, the GA. The paper is organized as follows. Section 2 contains a detailed review of the DE algorithm. Section 3 describes the application of the DE algorithm to the design of digital FIR filters. Section 4 presents the simulation results and discussion.

2. DE algorithm

The DE algorithm is a method based on the principles of GAs, but with crossover and mutation operations that work directly on continuous-valued vectors [18]. The main difference in constructing better solutions is that GAs rely on crossover, whereas the DE algorithm relys on the mutation operation. This main operation, which has been shown to be effective for global optimization of reasonably complex functions and is very simple to implement, is based on the differences of randomly sampled pairs of solutions in the population. The DE algorithm uses mutation as a search mechanism and selection to direct the search toward the prospective regions in the search space. The DE algorithm also uses a nonuniform crossover that can take child vector parameters from one parent more often than it does from others. By using the components of existing population members to construct trial vectors, the recombination (crossover) efficiently shuffles information about successful combinations, enabling the search for a better solution space.

Initially, a population is constructed from the solutions randomly and uniformly distributed within the search space. The process of generating a proposal makes use of at least four mutually exclusive population members. First, one candidate in the population is chosen systematically (in some predefined order) for improvement. Then the vector difference between two randomly chosen members, weighted by a scalar parameter, F, is added to a third randomly chosen "parent". Some implementations add two or more such vector differences. A crossover operation takes place between this produced vector and the candidate vector, to obtain a proposal point. A function evaluation takes place at the proposal, and a test is then performed to determine whether this is better or not than the candidate solution. According to the outcome, either the proposal replaces the candidate or the population remains unchanged. In either case, a new candidate for replacement will be selected on the next iteration. Crossover is implemented here by selecting each element of the proposal vector from either the candidate or the new vector according to some prespecified probability P_C. Proposals outside the bounds of the search space are ignored, allowing a valid proposal to be obtained for each candidate [1], [6], [19].

Figure 1 illustrates this process in more detail: the weighted difference between two population members (1, 2) is added to a third population point (3). The result (4) is subject to crossover with the candidate for replacement (5) to obtain a proposal (6). The proposal is evaluated and replaces the candidate if it is found to be better. Note that the proposal could be identical to (4) or (5), depending on the outcome of the crossover [19].

The role of the vector addition is to provide proposals that are within, or near the boundaries of, the existing population. The scalar parameter F affects the rate at which the variance of the population is likely to decrease. Crossover prevents the population from becoming trapped in a subspace (local minima) and may exploit the relative independence between different directions of the search space.

An optimization task consisting of D parameters can be represented by a D-dimensional vector. Therefore, in the DE algorithm a population of NP solution vectors is randomly created at the start. This population is successfully improved by applying mutation, crossover, and selection operators.

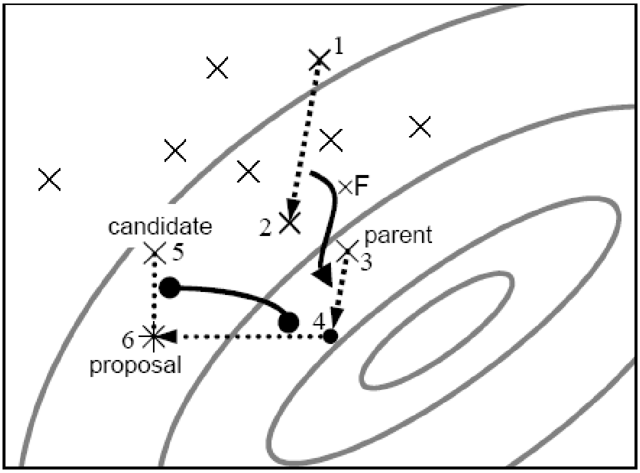

Figure 1. Obtaining a new proposal in differential evolution.

The main steps of a basic DE algorithm are given below:

Initialization
Evaluation
Repeat
  Mutation
  Recombination
  Evaluation
  Selection
Until (termination criteria are met)

2.1. Mutation

For each target vector $x_{i,G}$, a mutant vector is produced by

$$v_{i,G+1} = x_{i,G} + K \cdot (x_{r1,G} - x_{i,G}) + F \cdot (x_{r2,G} - x_{r3,G}), \tag{1}$$

where $i, r_1, r_2, r_3 \in \{1, 2, \ldots, NP\}$ are randomly chosen and must be different from each other. In equation (1), F is the scaling factor $\in [0, 2]$ affecting the difference vector $(x_{r2,G} - x_{r3,G})$, and K is the combination factor.

2.2. Crossover

The parent vector is mixed with the mutated vector to produce trial vector $u_{ji,G+1}$:

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if } (rnd_j \leq CR) \quad \text{or } j = rn_i \\ x_{ji,G} & \text{if } (rnd_j > CR) \quad \text{and } j \neq rn_i \end{cases} \tag{2}$$

where $j = 1, 2, \ldots, D$, $r_j \in [0, 1]$ is a random number, CR stands for the crossover constant $\in [0, 1]$, and $rn_i \in (1, 2, \ldots, D)$ is the randomly chosen index.

2.3. Selection

The performances of the trial vector and its parent are compared and the better one is selected. This method is usually called *greedy selection*. All solutions have the same chance of being selected as parents without the dependence of their fitness value. The better one of the trial solution and its parent wins the competition; this provides the significant advantage of converging performance over the GAs.

3. Application of DE algorithm to the problem

The transfer function of an FIR filter is given by equation (3),

$$H(z) = \sum_{n=0}^{N} a_n z^{-n}, \tag{3}$$

where a_n represents the filter parameters to be determined in the design process and N represents the polynomial order of the function. Factorizing the numerator polynomial, the zero description of the digital FIR filter can be obtained as

$$H(z) = K(z - z_1)(z - z_2) \ldots (z - z_i) \ldots (z - z_M), \tag{4}$$

where z_i represents the zero values of the function on the complex plane (z-domain). The zeros of the transfer function are the values of z for which $H(z)$ equals zero. The locations of zeros on the complex plane directly affect the frequency response. Therefore, the design process can be considered as an optimization process by which the zeros are placed into suitable locations. In this work, the DE algorithm and the GA are used for this purpose and their performances are compared.

In the design of an FIR filter using the GA or DE algorithm, first the solutions must be represented in a string form of parameters. The representation scheme used in this work is shown in Figure 2.

In order to evaluate the strings representing possible FIR filters, the least mean

a_{-L}	$a_{-(L-1)}$	.....	a_{-1}	a_0	a_1	.....	$a_{(L-1)}$	a_L

Figure 2. Representation of solutions in string form.

Table 1. Control parameter values of DE algorithm and GA

Differential Evolution Algorithm	Genetic Algorithm
Population size = 100	Population size = 100
Crossover rate = 0.8	Crossover rate = 0.8
Scaling factor $(F) = 0.8$	Mutation rate = 0.01
Combination factor $(K) = 0.8$	
Generation number = 500	Generation number = 500

squared (LMS) error is used. The strings that have higher evaluation values represent the better filters, i.e., the filters with better frequency response. The expression of the LMS function is given below:

$$LMS = \left\{ \sum_f [|H_I(f)| - |H_D(f)|]^2 \right\}^{\frac{1}{2}}, \tag{5}$$

where $H_I(f)$ is the magnitude response of the ideal filter and $H_D(f)$ is the magnitude response of the designed filter.

The fitness function to be maximized is defined depending on the LMS function as follows:

$$fitness_i = \frac{1}{LMS_i} \tag{6}$$

where $fitness_i$ is the fitness value of solution i and LMS_i is the LMS error value calculated when solution i is used for the filter.

4. Simulation results

The simulations have been realized for filters with the order of 8, 14, and 20. In the simulations, the sampling frequency was chosen as $f_s = 1$ Hz. Also, for all the simulations the sampling number was taken as 100.

It is known that the values of control parameters such as population size, crossover, and mutation rate and scaling and combination factors have a significant effect on the performance of the DE algorithm and the GA. In this work, the values suggested in the literature for these control parameters were

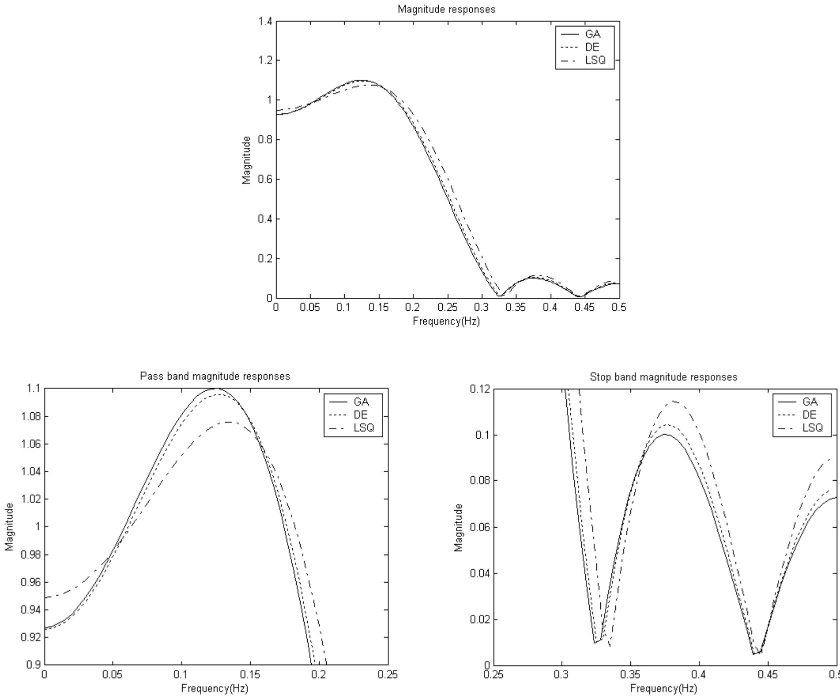

Figure 3. Magnitude responses of filters with the order of 8.

used [17]. The control parameter values employed for the DE algorithm and the GA are given in Table 1.

The magnitude responses of the designed digital FIR filters for the filter with the order of 8 are presented in Figure 3. As seen from Figure 3, the filters designed by the DE algorithm and the GA have similar magnitude responses for the passband and stopband regions. But, they have sharper transition band responses as compared to the filter designed by the LSQ algorithm. For the stopband region, the LSQ algorithm produces a filter whose response is a bit worse than the others.

The magnitude responses of the digital FIR filters designed using the GA and the DE and LSQ algorithms for the filter of 14th order are demonstrated in Figure 4. As seen from Figure 4, the filters designed by GA, DE, LSQ have similar magnitude responses for the passband region. The filters constructed using the DE algorithm and the GA have sharper transition band responses than that produced by the LSQ algorithm. For the stopband region, the LSQ algorithm produces a better response.

The magnitude responses of the digital FIR filters designed using the GA, and the DE and LSQ algorithms for the filter of 20th order are given in Figure 5. As

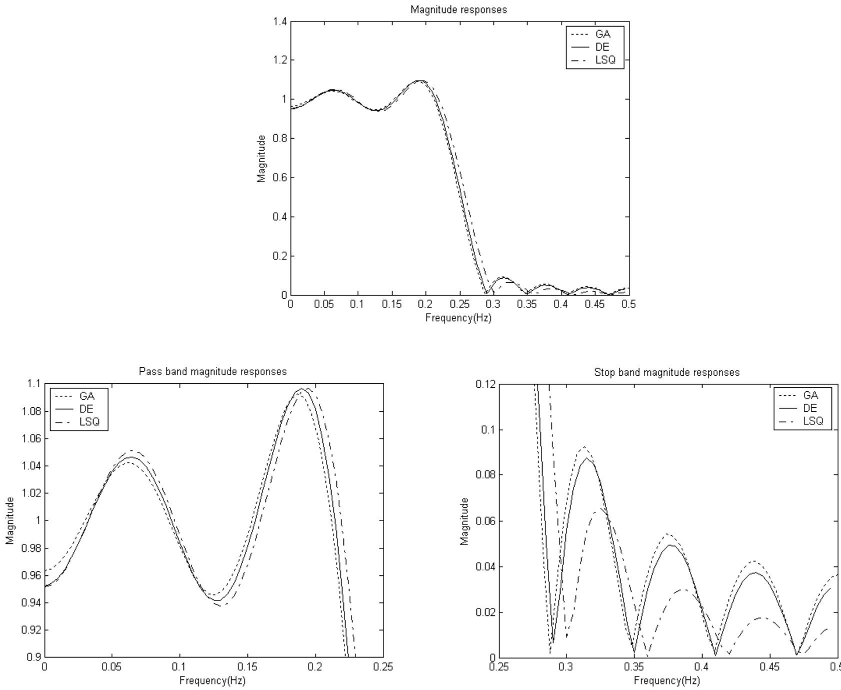

Figure 4. Magnitude responses of filters with the order of 14.

Table 2. LMS error values for different filter orders

Algorithms	The filter with the order of 8	The filter with the order of 14	The filter with the order of 20
LSQ	1.6744	1.3572	1.2943
GA	1.4790	1.1150	0.9961
DE	1.4740	1.1100	0.9900

seen from Figure 5, for the passband region, the LSQ algorithm produces a bit better response than the others. However, the filters designed by the GA and the DE algorithm have sharper transition band responses than that produced when the LSQ algorithm is used. For the stopband region, the filters designed by the three methods produce similar responses.

In order to compare the algorithms in terms of the convergence speed, Figure 6 shows the evolution of best solutions obtained when the GA and the DE algorithm are employed. From the figures drawn for three filters, it is seen that the DE algorithm is significantly faster than the GA for finding the optimum filters.

The performance of the algorithms can be compared in terms of the computa-

Table 3. Coefficients of the designed filters

(a) Filters with the order of 20

$a(n)$	GA	DE	LSQ
$a(1) = a(21)$	0.0001	−0.0047	−0.0166
$a(2) = a(20)$	−0.0355	0.0351	0.0260
$a(3) = a(19)$	−0.0003	0.0053	0.0181
$a(4) = a(18)$	0.0458	−0.0453	−0.0379
$a(5) = a(17)$	−0.0003	−0.0047	−0.0192
$a(6) = a(16)$	−0.0633	0.0635	0.0581
$a(7) = a(15)$	0.0000	0.0053	0.0201
$a(8) = a(14)$	0.1065	−0.1060	−0.1027
$a(9) = a(13)$	0.0001	−0.0047	−0.0206
$a(10) = a(12)$	−0.3181	0.3183	0.3172
$a(11)$	−0.5000	0.5053	0.5208

(b) Filters with the order of 14

a(n)	GA	DE	LSQ
$a(1) = a(15)$	−0.0452	−0.0454	−0.0384
$a(2) = a(14)$	0.0000	−0.0054	−0.0177
$a(3) = a(13)$	0.0641	0.0635	0.0584
$a(4) = a(12)$	−0.0000	0.0046	0.0185
$a(5) = a(11)$	−0.1058	−0.1061	−0.1029
$a(6) = a(10)$	0.0000	−0.0054	−0.0190
$a(7) = a(9)$	0.3187	0.3182	0.3172
$a(8)$	0.5000	0.5046	0.5192

(c) Filters with the order of 8

a(n)	GA	DE	LSQ
$a(1) = a(9)$	−0.0000	0.0042	0.0201
$a(2) = a(8)$	−0.1054	−0.1060	−0.1024
$a(3) = a(7)$	0.0000	−0.0058	−0.0207
$a(4) = a(6)$	0.3191	0.3183	0.3171
$a(5)$	0.5000	0.5042	0.5209

tion time, too. The DE algorithm requires about 5–6 seconds to design an optimal filter, whereas the GA needs approximately 70 seconds for 500 generations. In Table 2, the LMS error values obtained for the three examples are given. From the table it is clear that the performances of the DE algorithm and the GA are similar to each other in terms of LMS error. But their performance is significantly better

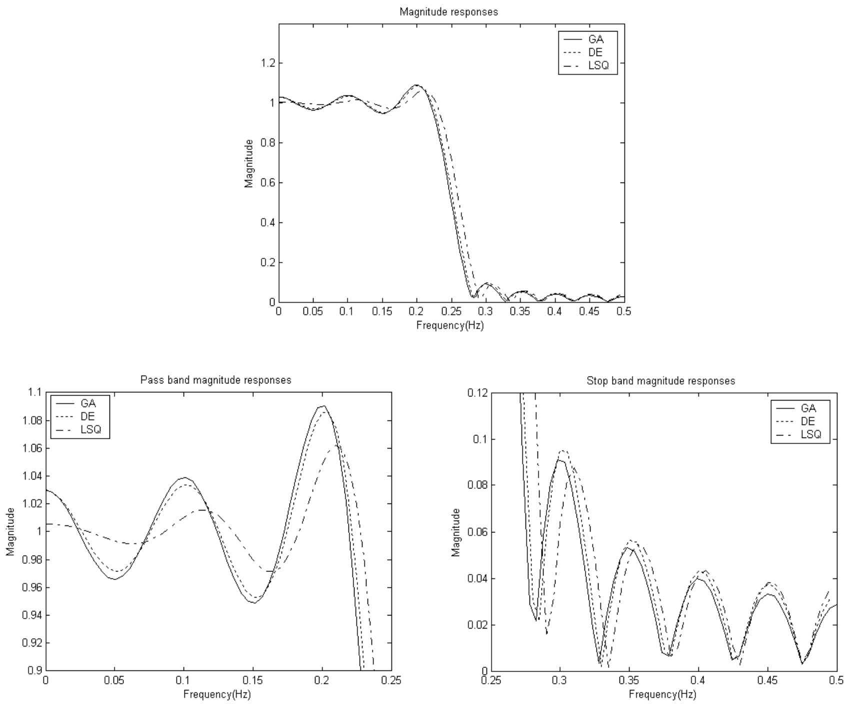

Figure 5. Magnitude responses of filters with the order of 20.

than that of the LSQ algorithm. The coefficients of the designed filters found by the three methods are given in Table 3.

5. Conclusion

The DE algorithm has been applied to the design of digital FIR filters with different orders. Although the DE algorithm has a more simple structure than the GA, for the same population size and generation number, the DE algorithm demonstrates a similar performance in terms of magnitude response and hence LMS error. Consequently, the DE algorithm has successfully designed FIR filters with desired magnitude responses and also found optimal filters much quicker than the GA. Therefore, the DE algorithm can be successfully used in digital FIR filter design.

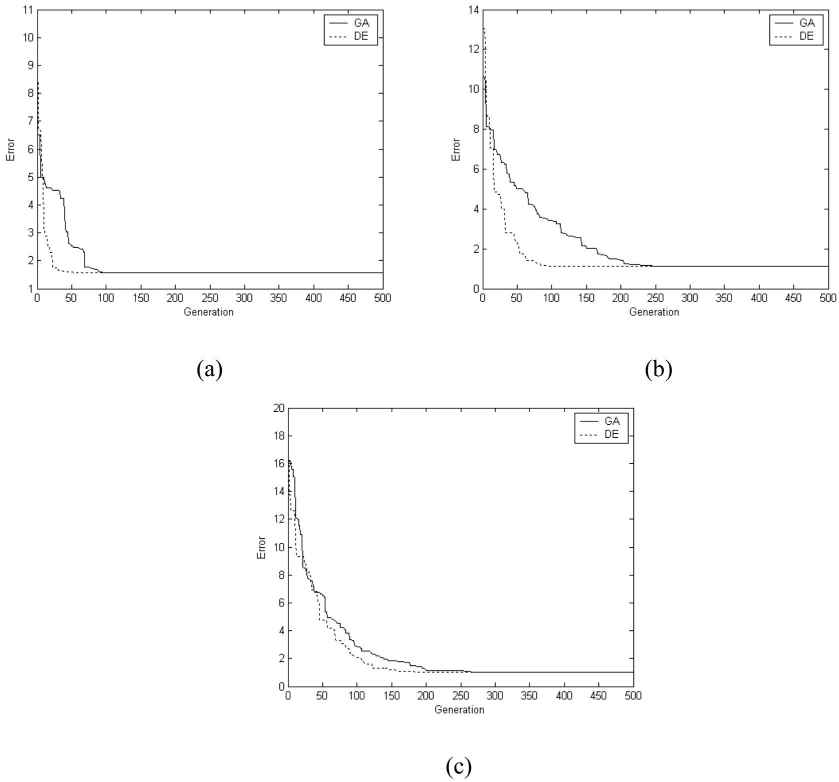

Figure 6. Performance comparison between DE algorithm and GA: (a) 8th order, (b) 14th order, (c) 20th order.

References

[1] H. A. Abbass, R. Sarker, and C. Newton, A pareto-frontier differential evolution approach for multi-objective optimization problems, *Proc. of the Congress on Evolutionary Computation*, Piscataway, NJ, 2001, 971–978.

[2] A. N. Belbachir, M. F. Belbachir, A. Fanni, S. Bibbo, and B. Boulerial, A new approach to digital FIR filter design using the tabu search, *IEEE NORSIG Sig. Process. Conf.*, Kolmärden, Sweden, 2000.

[3] A. N. Belbachir, M. F. Belbachir, A. Fanni, S. Bibbo, and B. Boulerial, Evaluation of the iterative least square method in digital filter design, *7th IASTED Int. Conf. on Sig. and Image Process.*, Las Vegas, NV, 2000.

[4] N. Benvenuto, M. Marchesi, and A. Uncini, Application of simulated annealing for the design of special digital filters, *IEEE Trans. Signal Process.*, 40, 323–332, 1992.

[5] R. Cemes and D. Ait-Boudaoud, Genetic approach to design of multiplierless FIR filters, *Electron. Lett.*, 29, 2090–2091, 1993.

[6] I. L. Cruz, L. G. V. Willigenburg, and G. V. Straten, Efficient differential evolution algorithms for multimodal optimal control problems, *Appl. Soft Computing*, 3, 97–122, 2003.

[7] M. Erba, R. Rossi, V. Liberali, and A. G. B. Tettamanzi, Digital filter design through simulated evolution, *European Conference on Circuit Theory and Design*, Finland, 2001, 137–140.
[8] R. V. Kacelenga, P. J. Graumann, and L. E. Turner, Design of digital filters using simulated annealing, *IEEE Sym. on Circuits and Systems*, New Orleans, LA, 1990, 642–645.
[9] N. Karaboga and B. Cetinkaya, Performance comparison of genetic and differential evolution algorithms for digital FIR filter design, *Lecture Notes in Computer Science*, 3261, 482–489, 2004.
[10] N. Karaboga and B. Cetinkaya, Efficient design of fixed point digital FIR filters by using differential evolution algorithm, *Lecture Notes in Computer Science*, 3512, 812–819, 2005.
[11] S. S. Kidambi, Comments on "2-D FIR filters design using least square error with scaling—free McClellan transformation," *IEEE Trans. Circuits and Systems Part II: Analog and Digital Signal Process.*, 50, 493, 2003.
[12] M. C. Lang, Weighted least squares IIR filter design with arbitrary magnitude and phase responses and specified stability margin, *IEEE Sym. on Advances in Digital Filtering and Signal Process.*, Victoria, Canada, 1998, 82–86.
[13] A. Lee, M. Ahmadi, G. A. Jullien, W. C. Miller, and R. S. Lashkari, Digital filter design using genetic algorithm, *IEEE Int. Symp. Circuits and Systems*, Monterey, CA, 1998, 34–38.
[14] A. Lee, M. Ahmadi, G. A. Jullien, W. C. Miller, and R. S. Lashkari, Design of 1-D FIR filters with genetic algorithms, *IEEE Int. Symp. on Circuits and System*, Orlando, FL, 1999, 295–298.
[15] L. Litwin, FIR and IIR digital filters, *IEEE Potentials*. 0278-6648, 2000, 28–31.
[16] R. Storn, Differential evolution design of an IIR-Filter with requariments for magnitude and group delay, *IEEE Int. Conf. on Evolutionary Computation*, Nagaya, Japan 1996, 268–273.
[17] R. Storn and K. Price, Differential Evolution—A simple and efficient adaptive scheme for global optimization over continious spaces, Technical Report TR-95-012, ICSI, 1995, ftp.icsi.berkeley.edu.
[18] M. Strens, M. Bernhardt, and N. Everett, Markov chain Monte Carlo sampling using direct search optimization, *Proc. of the 19th Int. Conf. on Machine Learning*, Sydney, Australia, 2002.
[19] M. Strens and A. W. Moore, Policy search using paired comparisons, *J. Machine Learning Research*, 3, 921–950, 2002.
[20] S. Traferro, F. Capparelli, F. Piazza, and A. Uncini, Efficient allocation of power-of-two terms in FIR digital filter design using tabu search, *IEEE Int. Symposium on Circuits and Systems*, Orlando, FL, 1999, 411–414.
[21] J. Vondras and P. Martinek, Multi-criterion filter design via differential evolution method for function minimization, *1st IEEE International Conference on Circuits and Systems for Communications*, 2002, 106–109.
[22] G. Wade, A. Roberts, and G. Williams, Multiplier-less FIR filter design using a genetic algorithm, *IEE Proc. Vision, Image and Signal Process.*, 1994, 175–180.
[23] M. Xiaomin and Y. Yixian, Optimal design of FIR digital filter using genetic algorithm, *J. China Univ. Posts Telecom.*, 5, 12–16, 1998.