Adaptive System Identification By Using Artificial Bee Colony Algorithm

MEHMET BAHADIR ÇETINKAYA
Department of Mechatronics Engineering
University of Erciyes
University of Erciyes, Faculty of Engineering, Department of Mechatronics Engineering, 38039, Melikgazi, Kayseri.
TURKEY
cetinkaya@erciyes.edu.tr

Abstract: - The theory and design of adaptive finite impulse response (FIR) filters are well developed and widely applied in practice due to their simple analytic description of error surfaces and intrinsic stable behavior. However, the studies on adaptive infinite impulse response (IIR) filters are not as common as adaptive FIR filters. The reason is that there are two main drawbacks in the design of adaptive IIR filters: stability during the adaptation process may not be ensured in some applications and the convergence to the optimal design is not always guaranteed because of their multi-modal error surface structures. In order to overcome these difficulties, global optimization based approaches are used in adaptive IIR filter design. One of the most recently proposed swarm intelligence based global optimization algorithms is the artificial bee colony (ABC) algorithm which simulates the intelligent foraging behavior of honeybee swarms. In this work, a novel approach based on artificial bee colony algorithm is described and applied to the design of adaptive IIR filters and its performance is compared to that of differential evolution (DE) and particle swarm optimization (PSO) algorithms.

Key-Words: - Artificial bee colony algorithm, Modified artificial bee colony algorithm, Particle swarm optimization algorithm, Differential evolution algorithm, Adaptive IIR filter design, System identification

1 Introduction

The design of the conventional linear non-adaptive filters require a priori information about the statistics of the data to be processed. The non-adaptive filter is optimum only when the statistical characteristics of the input data match the priori information on which the design of the filter is based. If the statistical characteristic of the input data varies with respect to time or there is no priori knowledge about the variation, adaptive filters are needed [1,2]. Adaptive filters find applications in a wide range of diverse fields such as system identification, noise cancellation, channel equalization, linear prediction, control, and modeling. There are two major classes of adaptive filter realizations, distinguished by the form of impulse response, finite impulse response (FIR) filters and infinite impulse response (IIR) filters. The theory and design of adaptive FIR filters are well developed and widely applied in practice due to their simple analytic description of error surfaces and intrinsic stable behavior [3,4]. Since the adaptive IIR filters require fewer number of coefficients than adaptive FIR filters to model the same system [5], they offer potential performance improvements and less computational cost than equivalent FIR filters [6]. On the other hand, an adaptive IIR filter gives a more general structure as it contains both poles and zeros in the transfer function, while an FIR filter has only zeros [6]. However, there are two main drawbacks in the design of adaptive IIR filters. They might have multi-modal and non-quadratic error surfaces which lead the filter to a local minimum instead of a global solution. A further problem is the possibility of the filter becoming unstable during the adaptation process. The unstability problem can be handled by restricting the parameter space in a suitable value range. As the error surface of adaptive IIR filters is usually multi-modal and non-quadratic with respect to the filter coefficients, gradient based learning algorithms can easily be stuck at local minima and cannot converge to the global optimum. In order to achieve the global optimum solution, approaches based on global optimization algorithms such as genetic algorithm (GA), simulated annealing (SA), tabu search (TS), differential evolution (DE), particle swarm optimization (PSO) and artificial bee colony (ABC) algorithms can be used for the design of adaptive IIR filters. Among these algorithms SA, TS, GA, DE and PSO based design methods have been described and applied to the adaptive IIR filter design [1,3,6-12]. However, to our best knowledge ABC algorithm has not been used to design adaptive IIR filters although it was employed for designing non-adaptive IIR filters [13]. In [13], basic ABC algorithm was employed for designing non-adaptive filter structures in which the signals are stationary. However, in the design of adaptive filters time-varying nonstationary signals are employed.

In this work, a new approach based on ABC algorithm for adaptive IIR filter design is introduced and the performance comparison of the design methods based on ABC, DE and PSO algorithms is presented. The paper is organized as follows. Section 2 presents a brief review to the artificial bee colony algorithm. Section 3 describes the adaptive IIR filter design problem. In section 4, the proposed approach is described and the simulation results are produced on the test problems considered and discussed.

2 Artificial Bee Colony Algorithm

2.1 Basic Artificial Bee Colony Algorithm

Swarm intelligence is the discipline that deals with collective behavior of natural and artificial systems composed of many individuals to solve combinatorial and numerical optimization problems [14-16]. In particular, it focuses on the collective behaviors that result from the local interactions of the individuals with each other and with their environment [16]. The classical examples of swarm are bee colony swarming around their hive; a colony of ants; a flock of birds; an immune system which is a swarm of cells and a crowd that is a swarm of people. In 2005, D. Karaboga introduced a bee swarm algorithm called artificial bee colony algorithm for numerical optimization problems [17]; and B. Basturk and D. Karaboga compared the performance of ABC with that of some other well-known population based optimization algorithms [18]. Moreover, ABC have been employed by several researchers to solve various problems in different research areas [19-24].

In ABC algorithm, the colony contains three groups of artificial bees: employed bees, onlookers and scouts. The first half of the colony consists of the employed bees and the other half includes the onlooker bees. The position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. Each food source is associated with only one employed bee. In other words, the number of the employed bees or the onlooker bees is equal to the number of solutions in the population. The employed bee whose food source has been exhausted by the other bees becomes a scout.

Initially, a population P is produced from the SN solutions randomly distributed within the search space, where SN denotes the size of employed or onlooker bees. If D is denoted as the number of optimization parameters, each solution $x_i \; (i = 1,2,...,SN)$ can be represented by a D-dimensional vector. Then, each food source in the initial population is randomly associated with an employed bee and their nectar amounts are calculated and memorized by the bees. At the initialization stage, after determining the initial population the nectar amounts of each solution is calculated and then the values of control parameters of the algorithm are assigned.

After initialization, each cycle of the search consists of three stages [25] : i.) employed bees randomly determine a food source within the neighbourhood of the food source in their memory and evaluate the nectar amount of this candidate food source. Then, a greedy selection procedure is applied between the old and the candidate solutions. If the new produced candidate food source has an equal or better nectar amount than the old source, it is replaced with the old one in the memory. Otherwise, the bee keeps the present position in the memory. Then, employed bees share their information with onlooker bees. ii.) onlooker bees prefer a food source area depending on the nectar information taken from the employed bees. Onlooker bees are placed on the food sources with respect to roulette wheel selection method. As the nectar amount of a food source increases, the probability of that food source chosen also increases. Each onlooker bee produce a neighbour food source within the neighbourhood of the one to which she has been assigned and then evaluate the nectar amount of it. Then, as mentioned in the first stage, the better solution is selected by using the greedy selection method. iii.) the employed bee of the food sources whose nectar amount is abandoned by the bees becomes as a scout bee. Scout bees are the explorers of the colony and they do not use any prior knowledge while investigating the new food sources. A new food source is randomly produced by a scout bee and replaced with the abandoned one. In ABC algorithm, by means of the scout bees discovering the rich and entirely unknown food sources becomes possible. In our model, at each cycle only one of the employed bees is selected and classified as a scout bee. The selection of the scout bee is controlled by a control parameter called "*limit*". If a solution representing a food source can not be improved by a predetermined number of trials, it means that the associated food source has been exhausted by the bees and then the employed bee of this food source becomes a scout. The number of trials for releasing a food source is equal to the value of "*limit*" which is an important control parameter of ABC algorithm. The three stages mentioned above are repeated at each cycle of the search until the termination criteria are satisfied.

An artificial onlooker bee chooses a food source depending on the probability value associated with that food source, p_i , calculated by the following expression,

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n} \tag{1}$$

where, fit_i represents the fitness value of the solution i which is proportional to the nectar amount of the food source in the position i. The fitness values of the solutions are calculated by using the expression (2),

$$fit_i = \frac{1}{1+f_i} \tag{2}$$

In the equation above, f_i is the Mean Squared Error (MSE) value produced by the solution *i*. In order to produce a candidate food position from the old one in memory, the ABC uses the expression (3),

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{3}$$

where, $k \in \{1, 2, ..., SN\}$ and $j \in \{1, 2, ..., D\}$ are randomly chosen indexes. In order to provide the old and the candidate positions to be different from each other, k has to be different from i. ϕ_{ij} is a random number between [-1,1].

The food source of which the nectar is abandoned by the bees is replaced with a new food source by the scout bees. If x_i is assumed as the abandoned food source and $j \in \{1,2,...,D\}$, the new food source position discovered by the scout bees to be replaced with x_i can be defined as in (4),

$$x_i^j = x_{\min}^j + rand\ [0,1](x_{\max}^j - x_{\min}^j) \tag{4}$$

The pseudo-code of the basic ABC algorithm is given below:

1: Randomly generate an initial population of solutions x_i, $i = 1,2,...,SN$
2: Evaluate the fitness value of each solution x_i in the initial population
3: cycle = 1,
4: **REPEAT**
5: Produce new solutions v_{ij} within the neighbourhood of x_{ij} for the employed bees by using (3) and evaluate them by using (2)
6: Apply the greedy selection process between th x_i and v_i solutions of each employed bee and memorize the selected solutions.
7: Calculate the probability values p_i for the solutions x_i by using (1).
8: Produce the new solutions v_i for the onlooker bees from the solutions x_i selected depending on p_i and evaluate them by using (2)
9: Apply the greedy selection process between the x_i and v_i solutions of each onlooker bee and memorize the selected solutions.
10: Determine the abandoned solutions for the scout, if exist, replace it with a new randomly produced solution x_i by using (4).
11: Memorize the best solution obtained so far
12: cycle=cycle+1
13: **UNTIL** (termination criteria are met)

2.2 ModifiedArtificial Bee Colony Algorithm

Basic ABC algorithm has three control parameters: colony size (number of employed bees or food sources), maximum cycle number and the *limit* value. In basic ABC algorithm, the new position of a food source is produced by changing only one parameter of the present position. This process reduces the convergence speed of ABC during the initial phase of search. In order to avoid this undesirable characteristic of basic ABC, the new position of a food source might be determined by changing more than one parameter. The ABC producing neighbour solutions in this way is called modified ABC. Modified ABC has got one more control parameter compared to basic ABC, called modification rate, which controls the frequency of parameter change in the production of a neighbor solution. The recommended value for this control parameter is between [0,1]. Modified ABC has been used for constrained optimization problems [26] by Karaboga and Basturk in 2007. In this paper, the ABC and modified ABC is used for the design of adaptive IIR filters for the purpose of adaptive system identification which is an unconstrained optimization problem. warm intelligence is the discipline that deals with

3 Definition of the Problem

Many problems in the area of signal processing can be reduced to system identification process [27], where one might gather data from a system whose structure is initially unknown. In this work, adaptive IIR filters are employed for the system identification purpose. Fig. 1 represents the block diagram of a system identification process using an adaptive IIR filter.

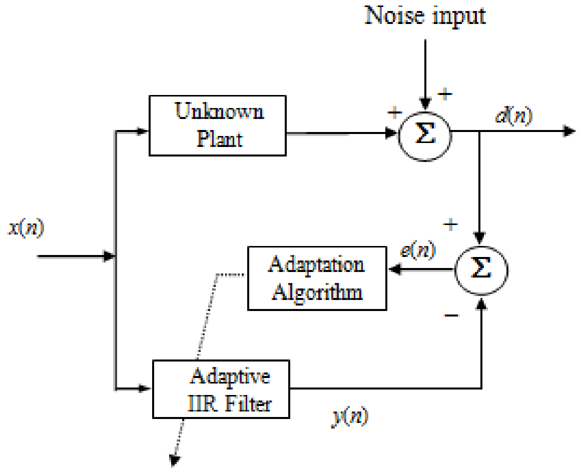


Fig. 1 Block diagram of system identification process using adaptive IIR filter

As seen from the figure, an adaptive IIR filter is used to model the behaviour of a physical dynamic system. Generally, the nature of the system is unknown and thus it may be regarded as unknown system. At each cycle, the coefficients of the filter are adaptively adjusted by adaptation algorithm so as to minimize the error between the outputs of the filter and the unknown system.

The basic structure of an IIR filter can be defined by the following difference equation:

$$y(n)+\sum_{i=1}^{N} a_i\, y(n-i) = \sum_{i=1}^{M} b_i\, x(n-i) \qquad (5)$$

where a_i and b_i are the adjustable coefficients of the model. $x(n)$ and $y(n)$ are the filter's input and output, respectively, and N($\geq$M) is the filter order. The input signal is usually chosen as a wideband signal in order to allow the adaptive filter to converge to a good model of the unknown system. For fixed filter coefficients the transfer function of the IIR filter can be written in the following general form:

$$H(z)=\frac{\sum_{i=0}^{M} b_i z^{-i}}{1+\sum_{i=1}^{N} a_i z^{-i}} \qquad (6)$$

The coefficients of the filter can be represented in the string form of $\mathbf{w}=\left[a_1\, a_2 ... a_N\; b_0\;\; b_1 ... b_M\right]^T$. The design of this adaptive IIR filter can be defined as a minimization problem of the cost function $J(\mathbf{w})$ by adjusting the coefficients at each cycle. The cost function (Mean Squared Error-MSE) is usually expressed as the time-averaged cost function defined by Equation 7,

$$\mathrm{J}(\mathbf{w}) = \frac{1}{N}\sum_{n=1}^{N}[d(n)-y(n)]^2 = \mathbf{E}[\, |e(n)|^2\,] \qquad (7)$$

where E denotes the statistical expected value, and $e(n)$ is the estimation error which is equal to the difference between the desired signal and the adaptive filter output,

$$e(n)=d(n)-y(n) \qquad (8)$$

4 Simulation Results

Simulation studies have been carried out on widely used three test problems defined for the purpose of system identification [11,27]. In this work, the simulations are realized by using the adaptation algorithms of ABC, modified ABC, DE and PSO which have global search ability, and then performances of the algorithms are compared. For each algorithm, the quality of the solution i in the population is calculated by using the following formula,

$$fit(i)=\frac{1}{1+J_i(w)} \qquad (9)$$

where $J_i(w)$ is the cost function value computed for the solution i. In order to calculate the quality of a solution a moving scheme is employed. The cost function defined by Equation 7 is calculated by using a block of N samples (N=100) and the data block is shifted by 1 sample after each cycle.

The control parameter values of the algorithms used in the simulations are given in Table 1. The values of the control parameters have a significant effect on the performance of the algorithms. In this work, for DE and PSO algorithms, the control parameter values suggested in the literature were used [10,28]. In ABC based algorithms, the optimum value of the limit parameter was chosen according to [29] and the value of the modification rate parameter was chosen as mentioned in [26]. For a fair comparison, the colony sizes and the evaluation numbers of the algorithms are chosen to be equal to each other. In the table, $X_{\min}$ and $X_{\max}$ represents the lower and upper bounds of the filter parameters.

Example 1: Low-dimensional (2 parameters), bimodal, no noise

In the first example [11], a second order system was being modeled with a first order adaptive IIR filter. The unknown plant and the adaptive filter had the following transfer functions, respectively,

$$H_D(z) = \frac{0.05 - 0.4z^{-1}}{1 - 1.1314z^{-1} + 0.25z^{-2}}$$

$$H(z) = \frac{b}{1 + a\,z^{-1}} \quad (10)$$

The system input was a uniform white sequence. The data length used in calculating the MSE was N = 100. Since the filter order is smaller than the system order, the MSE surface is multi-modal. The cost function has a global minimum at $w^{global} = [-0.311 \; -0.906]^T$ and a local minimum at $w^{local} = [0.114 \;\; 0.519]^T$.

Fig. 2 shows the evolution of the MSE averaged over 50 different runs of modified ABC, ABC, DE and PSO algorithms.

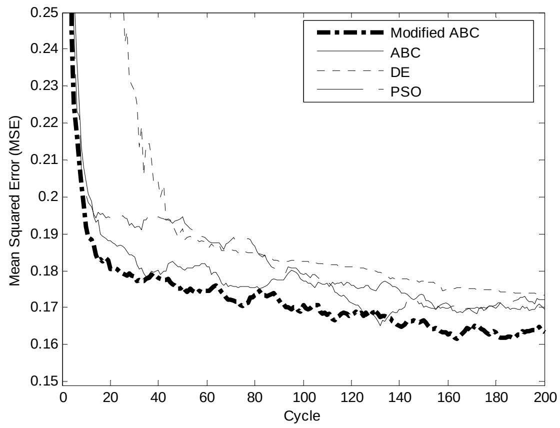


Fig. 2. Cost function value versus number of cycles averaged over 50 random runs for modified ABC, ABC, DE and PSO algorithms (Example 1)

As seen from the Fig. 2, PSO algorithm produces a similar but a bit better result than DE algorithm in terms of mean squared error. The evolution of MSE error for the basic and modified ABC algorithms is similar during first cycles. However, after around 80 cycles the MSE value of modified ABC gradually decreases and finally the lowest MSE value is obtained.

Fig. 3 also demonstrates the evolution of the parameters for the run in which the minimum MSE value is obtained.

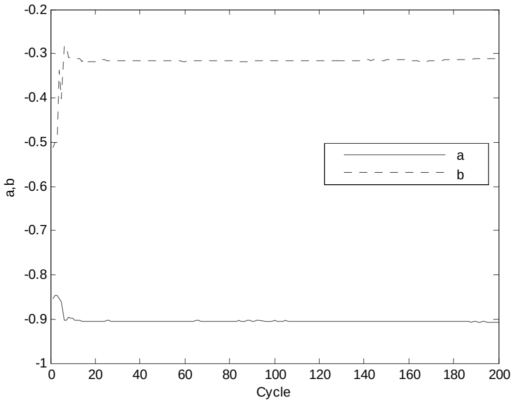
Fig. 4 demonstrates the positions of the poles and the zeros of the stable filters designed with the minimum MSE value by using the parameter values found by the algorithm given in in Table 2.

Table 1 Control parameter values of the algorithms used in the simulations

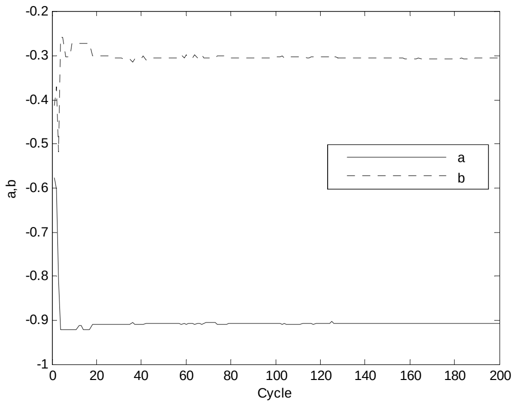
ABC	Modified ABC	DE	PSO
Colony size = 40	Colony size = 40	Population size = 40	Swarm size = 40
limit value = 180 (Example 1) limit value = 200 (Example 2,3) limit value = 400 (Example 4)	limit value = 180 (Example 1) limit value = 200 (Example 2,3) limit value = 400 (Example 4) Modification Rate = 0.7	Crossover rate = 1 Scaling factor (F) = 0.85	Inertia factor, ω = 0.5 Cognitive factor, c_1 = 1 Social factor, c_2 = 1
X_{max} = 1 X_{min} = -1	X_{max} = 1 X_{min} = -1	X_{max} = 1 X_{min} = -1	X_{max} = 1, X_{min} = -1 V_{max} =0.5, V_{min} =-0.5

Table 2 The best parameter values found by modified ABC, ABC, DE and PSO algorithms for Example 1

Coefficients	Global Minimum	Modified ABC	ABC	DE	PSO
b_0	-0.311	-0.3119	-0.3050	-0.3075	-0.3041
a_1	-0.906	-0.9066	-0.9073	-0.9109	-0.9121

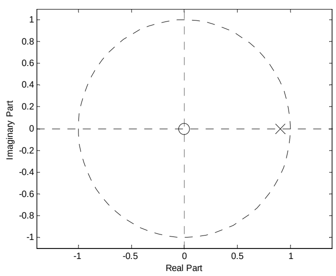


(a) Modified ABC

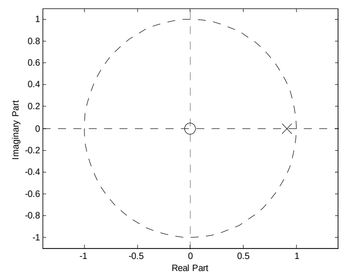


(b) ABC

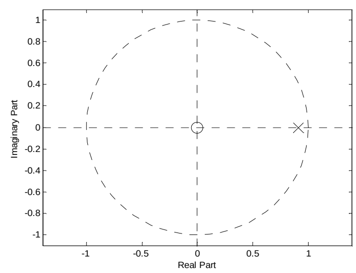
Fig. 3. Evolution of the parameters of the first filter for modified ABC and ABC algorithms



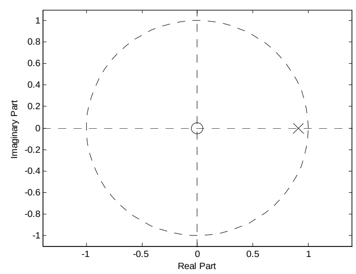
(a) Modified ABC



(b) ABC



(c) DE



(d) PSO

Fig. 4. Pole-zero diagrams of the best filters obtained for the first example by the algorithms

Example 2: High-dimensional (11 parameters), unimodal, SNR=30 dB

In this test problem, the unknown plant given was a fifth order low-pass Butterworth filter and the adaptive IIR filter to be designed was the same order with the plant [27],

$$H_D(z) = \frac{0.1084 + 0.5419z^{-1} + 1.0837z^{-2} + 1.0837z^{-3} + 0.5419z^{-4} + 0.1084z^{-5}}{1 + 0.9853z^{-1} + 0.9738z^{-2} + 0.3864z^{-3} + 0.1112z^{-4} + 0.0113z^{-5}}$$

$$H(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}+b_4z^{-4}+b_5z^{-5}}{1+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}+a_4z^{-4}+a_5z^{-5}} \qquad (11)$$

In the simulation, the system input *x(n)* was chosen as a uniform white sequence taking values in (-0.5, 0.5), and the signal-to-noise ratio (SNR) was 30 dB. In order to compare the algorithms in terms of the MSE evolution, Fig. 5 shows the average evolution of the best solutions over 50 random runs for four algorithms. From the figures drawn for two filters, it is seen that ABC and PSO algorithms produce similar performance for finding the optimum filters. The performance of the DE algorithm is the worst especially at the first cycles. Among four algorithms, the modified ABC is the best in terms of MSE and evolution speed performances in this example, too.

Fig. 6 and 7 present the evolution of the nominator and the denominator parameters for the ABC based algorithms for the best run. As seen from the figures given for basic ABC, the evolution might show instantaneous changes. However, the evolution process of the parameters for modified ABC seems more stable since the values are changing gradually.

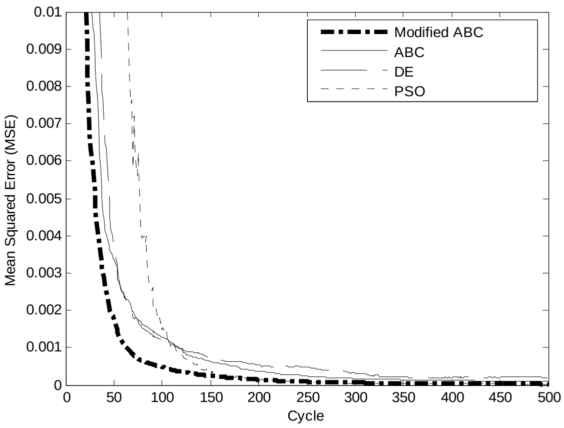
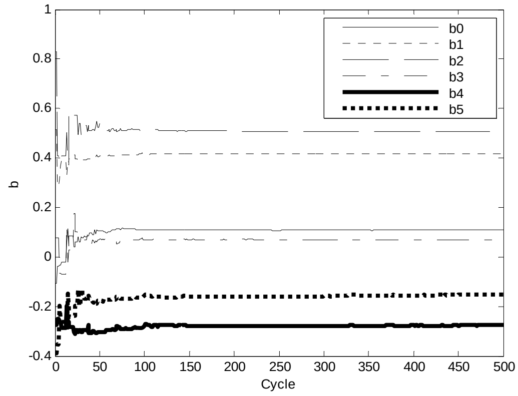
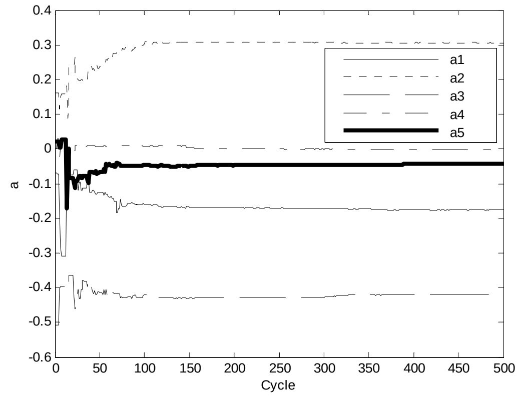


Fig. 5. Cost function value versus number of cycles averaged over 50 random runs for modified ABC, ABC, DE and PSO algorithms (Example 2)

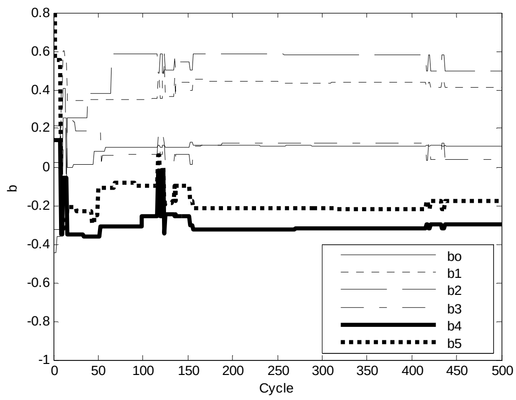


(a) Nominator parameters

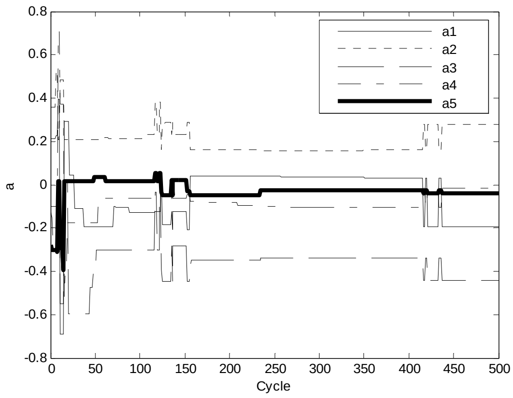


(b) Denominator parameters

Fig. 6. Evolution of the nominator and denominator parameters of the high order filter for modified ABC algorithm



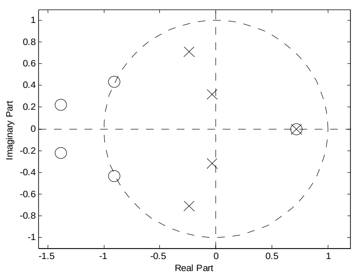
(a) Nominator parameters



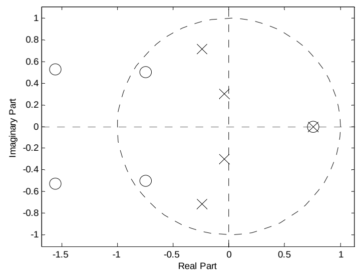
(b) Denominator parameters

Fig. 7. Evolution of the nominator and denominator parameters of the high order filter for ABC algorithm

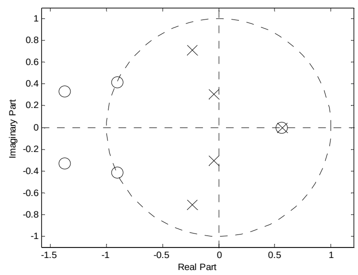
From the pole zero diagrams given in Fig.8, it is seen that all the poles are located in the unit circle and hence all the designs are stable.



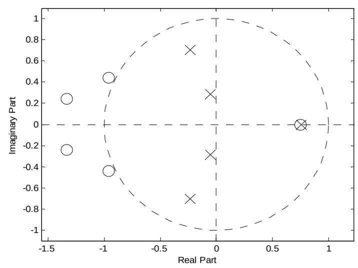
(a) Modified ABC



(b) ABC



(c) DE



(d) PSO

Fig. 8. Pole-zero diagrams of the best filters designed for the second example by the algorithms

The parameters values of the best designed adaptive IIR filters by the algorithms are given in Table 3.

Table 3 The parameter values of the best filters designed by modified ABC, ABC, DE and PSO algorithms for Example 2

Coefficients	Modified ABC	ABC	DE	PSO
a_1	-0.1759	-0.1940	-0.0096	-0.1768
a_2	0.3063	0.2798	0.3783	0.2554
a_3	-0.4183	-0.4418	-0.3002	-0.4192
a_4	-0.0029	-0.0162	-0.0003	-0.0298
a_5	-0.0424	-0.0398	-0.0301	-0.0353
b_0	0.1078	0.1071	0.1084	0.1086
b_1	0.4160	0.4120	0.4336	0.4158
b_2	0.5067	0.4992	0.5866	0.5002
b_3	0.0676	0.0384	0.2025	0.0454
b_4	-0.2770	-0.2988	-0.1721	-0.3080
b_5	-0.1533	-0.1776	-0.1209	-0.1668

Example 3: High-dimensional (9 parameters), bimodal, SNR=20dB

The unknown plant was a sixth order system and the adaptive IIR filter was a fourth order filter with the following transfer functions [27],

$$H_D(z) = \frac{1 - 0.4z^{-2} - 0.65z^{-4} + 0.26z^{-6}}{1 - 0.77z^{-2} - 0.8498z^{-4} + 0.6486z^{-6}}$$

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}}{1 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}} \qquad (12)$$

Since the system was a sixth order system and the filter fourth order, the error surface is bimodal as in the first example. Zero mean Gaussian pseudo-noise was added to the channel output to give a signal-to-noise ratio of 20 dB.

In order to compare the algorithms in terms of the MSE evolution for example 3, Fig. 9 shows the average evolution of the best solutions over 50 random runs with different initial solutions for four algorithms.

From the figures drawn for four algorithms, it is seen that modified ABC produces the best result in terms of evolution speed and it finds the lowest MSE value. Fig. 10 and 11 present the evolution of the nominator and the denominator parameters for the algorithms for the best run. It can be clearly seen that modified ABC shows more stable behaviour in terms of the evolution of parameters during the adaptation process.

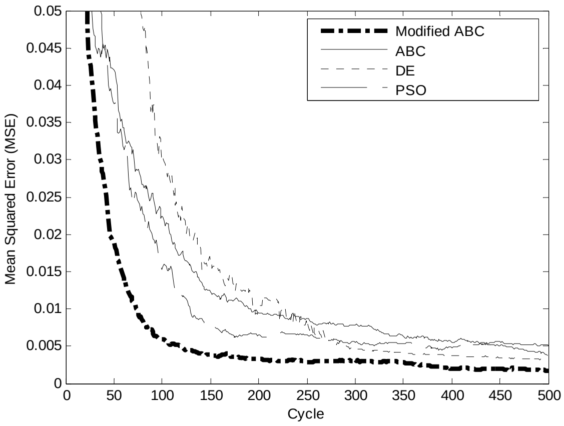
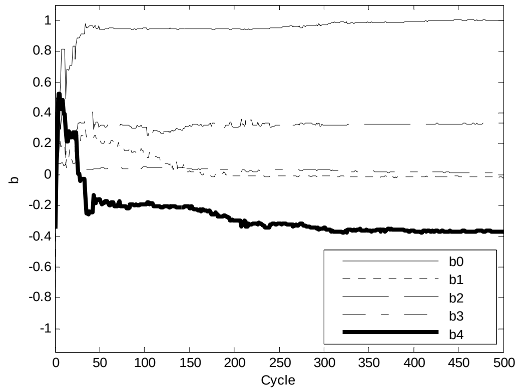
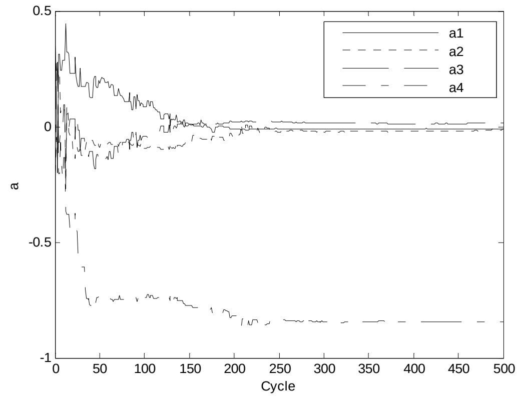


Fig. 9. Cost function value versus the number of cycles averaged over 50 random runs for modified and basic ABC, DE and PSO algorithms (Example 3)

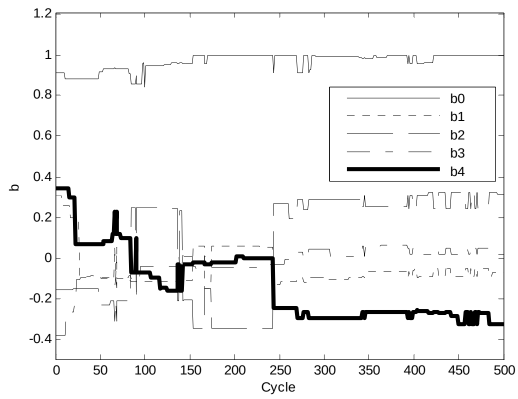


(a) Nominator parameters

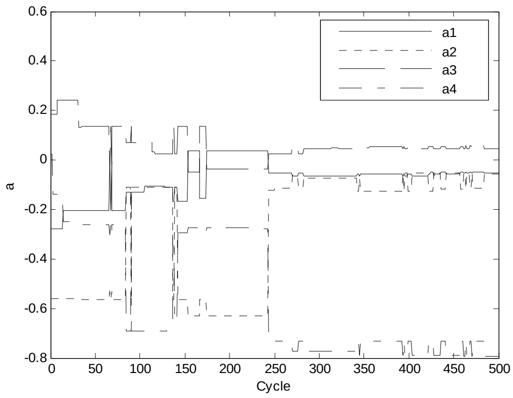


(b) Denominator parameters

Fig. 10. Evolution of the parameters of the third filter for modified ABC algorithm



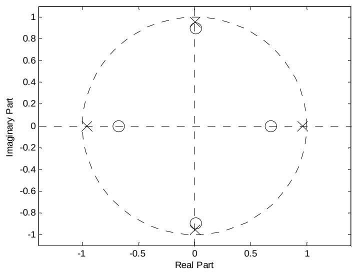
(a) Nominator parameters



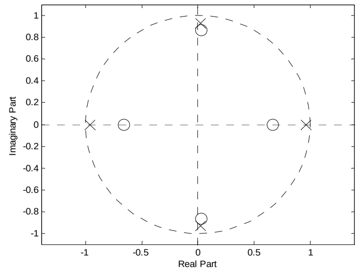
(b) Denominator parameters

Fig. 11. Evolution of the parameters of the third filter for ABC algorithm

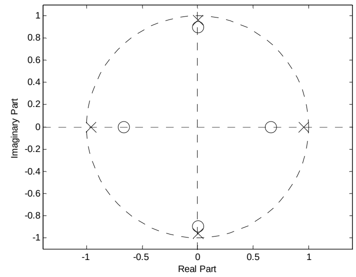
Also, Fig. 12 demonstrates the positions of the poles and the zeros of the stable filters designed by algorithms with the minimum MSE value, by using the parameter values given in Table 4.



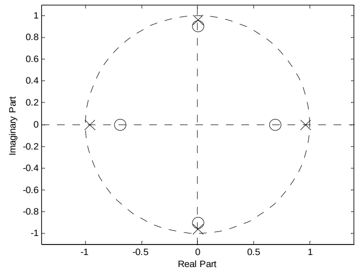
(a) Modified ABC



(b) ABC



(c) DE



(c) PSO

Fig. 12. Pole-zero diagrams of the filters with the minimum MSE value designed for the third example

Table 4 The parameter values of the best filters designed by the modified and basic ABC, DE and PSO algorithms in the Example 3

Coefficients	Modified ABC	ABC	DE	PSO
a_1	-0.0107	-0.0542	-0.0022	-0.0059
a_2	-0.0115	-0.0564	-0.0010	-0.0040
a_3	0.0142	0.0440	0.0024	0.0043
a_4	-0.8437	-0.7959	-0.8443	-0.8535
b_0	0.9962	1.0000	0.9909	1.0000
b_1	-0.0205	-0.0732	-0.0025	-0.0058
b_2	0.3362	0.3135	0.3639	0.3423
b_3	0.0096	0.0184	0.0013	0.0016
b_4	-0.3676	-0.3279	-0.3511	-0.3906

Example 4: High-dimensional (17 parameters), multi-modal, no noise

The unknown plant was chosen as an elliptic filter with the order of 10 and the adaptive IIR filter to be designed was an eighth order filter. Both filters had the transfer function as the following form,

$$H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \ldots + b_{(M-1)} z^{-(M-1)} + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_{(N-1)} z^{-(N-1)} + a_N z^{-N}} \quad (13)$$

The coefficients of the unknown plant are given in the first two columns of Table 5. Since the order of the unknown plant is higher than the adaptive IIR filter to be designed, the error surface is multi-modal. The system input was chosen as a uniform white sequence and the data length used in calculating the MSE was $N = 100$.

Fig. 13 represents the cost function value versus number of cost function evaluations averaged over 50 different runs of modified ABC, ABC, DE and PSO algorithms. As seen from the figure, ABC based approaches design the optimal filters quicker than the DE and PSO algorithms. The performance of the modified ABC algorithm is fairly better than the other three algorithms and the worst performance is performed by the DE algorithm. As seen from the pole-zero diagrams given in Fig. 14, all the adaptive IIR filters designed are stable.

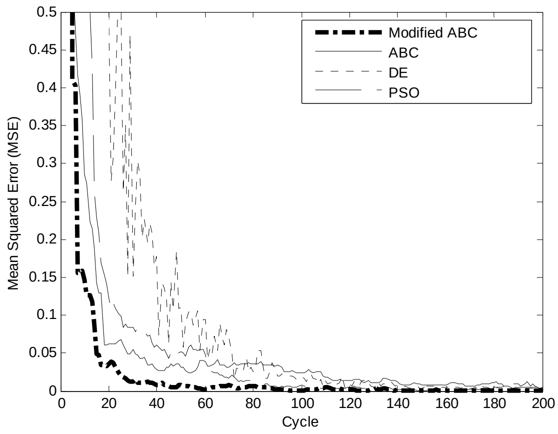
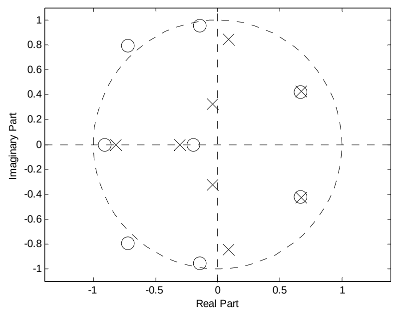
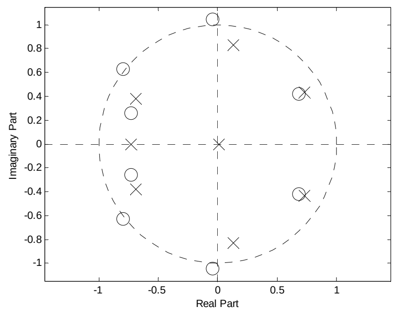


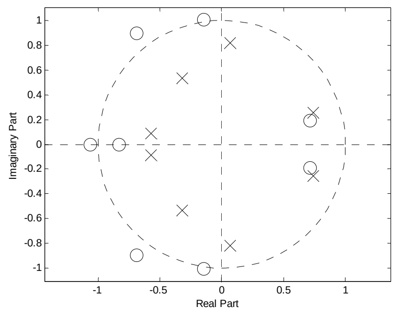
Fig. 13. Cost function value versus the number of cycles averaged over 50 random runs for modified and basic ABC, DE and PSO algorithms (Example 4)



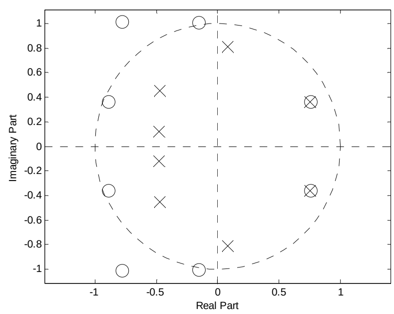
(a) Modified ABC



(b) ABC



(c) DE



(d) PSO

Fig. 14. Pole-zero diagrams of the best filters designed by the algorithms for the fourth example

The coefficients of the unknown plant and the adaptive IIR filters designed by the algorithms are given with Table 5.

Table 5 The parameter values of the unknown plant and the adaptive IIR filters with the minimum MSE value designed by the algorithms in the Example 4

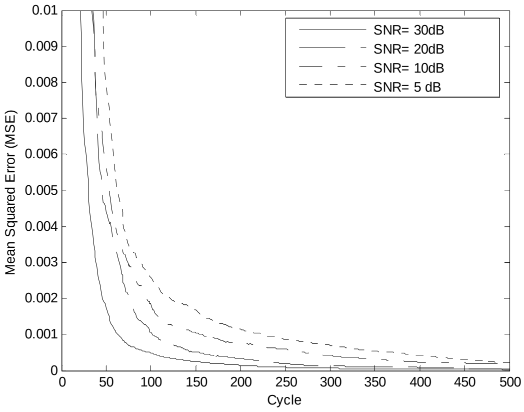
Coefficients	Unknown plant (elliptic filter)	Modified ABC	ABC	DE	PSO
a_1	-0.3771	-0.3165	0.3773	0.1773	0.2256
a_2	4.1167	0.2039	-0.1974	0.0599	0.0038
a_3	-1.4008	0.3004	0.2585	-0.1130	-0.0461
a_4	6.5731	-0.3202	0.0976	-0.2823	-0.0852
a_5	-1.9431	0.2490	-0.0972	-0.0685	-0.0176
a_6	5.0176	0.0946	0.2794	0.0824	0.1865
a_7	-1.1919	0.0353	0.2296	0.1310	0.1805
a_8	1.7830	0.0120	-0.0019	0.0536	0.0484
a_9	-0.2726	-	-	-	-
a_{10}	0.2217	-	-	-	-
b_0	0.2226	0.2262	0.2792	0.2116	0.2092
b_1	0.3241	0.3421	0.4964	0.4485	0.4455
b_2	1.1701	0.3253	0.4664	0.4598	0.5353
b_3	1.2975	0.0899	0.2523	0.1392	0.1646
b_4	2.3973	-0.0252	0.0513	-0.2573	-0.1570
b_5	1.9469	-0.0162	-0.1002	-0.3116	-0.2200
b_6	2.3973	0.0688	0.0660	-0.1609	-0.0112
b_7	1.2975	0.1517	0.2342	0.1223	0.2368
b_8	1.1701	0.0272	0.1212	0.1344	0.2292
b_9	0.3241	-	-	-	-
b_{10}	0.2226	-	-	-	-

For all examples, when the average MSE values obtained by the algorithms after 50 random runs are examined, it is clear that the modified ABC algorithm produces the minimum averaged MSE values. Hence, it can be said that ABC algorithm is more robust than DE and PSO algorithms and its tuning ability is better than these algorithms. It means that its performance is not so dependent on initialization process.

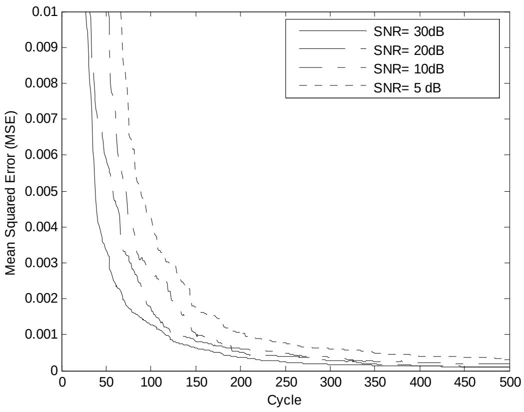
SNR Test : To show the effect of signal to noise ratio (SNR) on the performance of algorithms, the Example 2 was realized for three different SNR values of 5, 10, 20 and 30 dB, then the results were compared in terms of convergence speed as demonstrated in Fig. 15. From the figure, it is seen that the higher the SNR value, the quicker the convergence is. In terms of final MSE value the modified ABC is superior on other algorithms.

4 Conclusion

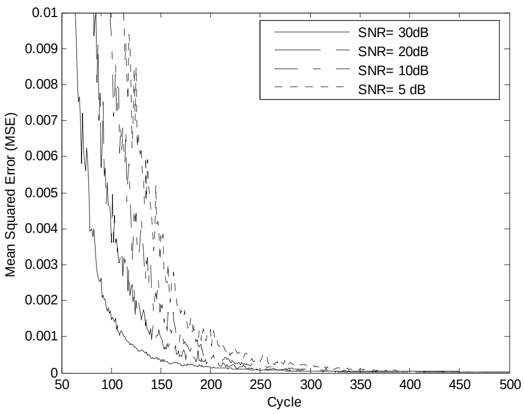
In this work, a novel approach based on ABC algorithm for adaptive IIR filter design was described and its performance compared with DE and PSO algorithms which have been also recently introduced global search algorithms for the purpose of system identification. The performance of the algorithms was examined in the case of uni-modal, multi-modal, with noise and without noise cases. Simulation results show that, the performance of ABC algorithm in terms of the evolution speed and final mean squared error is similar or better than DE and PSO algorithms although ABC is as simple as these two algorithms. It can be concluded that, ABC algorithm based approach can successfully be used for designing adaptive IIR filters with desired specifications.



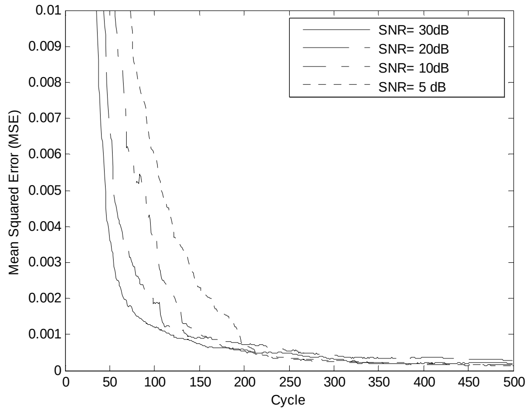
(a) Modified ABC algorithm



(b) ABC algorithm



(c) DE algorithm



(d) PSO algorithm

Figure 15. Performance comparison in terms of SNR values.

References:

[1] A. Kalinli, N. Karaboga, A new method for adaptive IIR filter design based on tabu search algorithm, *International. Journal of Electronics and Communications*, Vol.59, No.2, 2004, pp. 1-7.

[2] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 2002.

[3] D.J. Krusienski, W.K. Jenkins, Design and performance of adaptive systems based on structured stochastic optimization strategies, *IEEE Circuits and Systems Magazine*. Vol.5, No.1, 2005, pp. 8-20.

[4] L. Pasquato, İ. Kale, System identification via hybrid FIR-IIR adaptive filtering, IEEE Instrumentation and Measurement Technology Conference, 1999, pp. 1064-1069.

[5] M.J. Bruno, J.E. Cousseau, P.D. Donate, On reduced complexity IIR adaptive filters, IEEE International Symposium on Circuits and Systems, 2005, pp. 4329-4332.

[6] S.C. Ng, S.H. Leung, C.Y. Chung, A. Luk, W.H. Lau, The genetic search approach: A new learning algorithm for adaptive IIR filtering, *IEEE Signal Processing Magazine,* Vol.13, No.6, 1996, pp. 38-46.

[7] K.S. Tang, K.F. Man, S. Kwong, Q. He, Genetic algorithms and their applications, *IEEE Signal Processing Magazine,* Vol.13, No.6, 1996, pp. 22-37.

[8] N. Karaboga, Digital IIR filter design using differential evolution algorithm, *EURASIP Journal on Advances in Signal Processing*, Vol.2005, No.8, 2005, pp. 1269-1276.

[9] D.J. Krusienski, W.K. Jenkins, Adaptive filtering via particle swarm optimization, 37th Asilomar Conference on Signals Systems and Computers, 2003, pp. 571-575.

[10] D.J. Krusienski, W.K. Jenkins, Particle swarm optimization for adaptive IIR filter structures, Congress on Evolutionary Computation, 2004, pp. 965-970.

[11] S. Chen, B.L. Luk, Adaptive simulated annealing for optimization in signal processing applications, *Signal Processing*, Vol.79, No.1, 1999, pp. 117-128.

[12] S. Chen, R. Istepanian, B.L. Luk, Digital IIR filter design using adaptive simulated annealing, *Signal Processing*, Vol.11, No.3, 2001, pp. 241-251.

[13] N. Karaboga, A new design method based on artificial bee colony algorithm for digital IIR filters, *Journal of the Franklin Institute*, Vol.346, No.4, 2009, pp. 328-348.

[14] D. Karaboga, B. Basturk, On the performance of artificial bee colony (ABC) algorithm, *Applied Soft Computing*, Vol.8, No.1, 2008, pp. 687-697.

[15] A. P. Engelbrecht, *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons Publication, 2005.

[16] M. Dorigo, M. Birattari, *Swarm Intelligence*, Scholarpedia, Vol.2, No.9, 2007, pp. 1462.

[17] D. Karaboga, An idea based on honey bee swarm for numerical optimization. Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[18] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, *Journal of Global Optimization*, Vol.39, No.3, 2007, pp. 459-471.

[19] D. Karaboga, C. Ozturk, A novel clustering approach: Artificial Bee Colony (ABC) algorithm, *Applied Soft Computing*, Vol.11, No.1, 2011, pp. 652-657.

[20] T. Kurban, E. Besdok, A Comparison of RBF neural network training algorithms for inertial sensor based terrain classification, *Sensors*, Vol.9, No.8, 2009, pp. 6312-6329.

[21] A. Singh, An artificial bee colony algorithm for the leaf-constraint minimum spanning tree problem, *Applied Soft Computing*, Vol.9, No.2, 2009, pp. 625-631.

[22] R. S. Rao, S.V.L. Narasimham, M. Ramalingaraju, Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm, *International Journal of Electrical Power and Energy Systems Engineering*, Vol.1, No.2, 2008, pp. 116-122.

[23] P.W. Tsai, J.S. Pan, B.Y. Liao, S.C. Chu, Interactive artificial bee colony algorithm, International Symposium on Intelligent Informatics, 2008, pp. 247-251.

[24] S. Hemamalini, S.P. Simon, Economic load dispatch with valve-point effect using artificial bee colony algorithm, 32th National Systems Conference, 2008, pp. 17-19.

[25] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, *Applied Mathematics and Computation*, Vol.214, No.1, 2009, pp. 108-132.

[26] D. Karaboga, B. Basturk, Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, 2007, pp. 789-798.

[27] M.S. White, S. J. Flockton, *Evolutionary Algorithms in Engineering Applications*, Springer Verlag, 1997.

[28] R. Storn, Differential evolution design of an IIR-filter, Congress on Evolutionary Computation, 1996, pp. 268-273.

[29] B. Akay, D. Karaboga, Parameter tuning for the artificial bee colony algorithm, LNAI: 1st International Conference on Computational Collective Intelligence, 2009, pp. 608-619.