An International Journal of Optimization and Control: Theories & Applications
Vol.6, No.2, pp.129-139 (2016) © IJOCTA
ISSN: 2146-0957 eISSN: 2146-5703
DOI: 10.11121/ijocta.01.2016.00298
http://www.ijocta.com

Modeling of higher order systems using artificial bee colony algorithm

Aytekin Bagis [a], Halit Senberber [b]

[a,b] Erciyes University, Engineering Faculty,
Electrical & Electronics Engineering Department, 38039, Kayseri, Turkey
bagis@erciyes.edu.tr, halitsenberber@erciyes.edu.tr

(*Received January 26, 2016; in final form May 6, 2016*)

Abstract: In this work, modeling of the higher order systems based on the use of the artificial bee colony (ABC) algorithm were examined. Proposed model parameters for the sample systems in the literature were obtained by using the algorithm, and its performance was presented comparatively with the other methods. Simulation results show that the ABC algorithm based system modeling approach can be used as an efficient and powerful method for higher order systems.

Keywords: Artificial bee colony algorithm; system modeling; parameter optimization
AMS Classification: 68T01, 93C05, 68W99, 93A30, 68U20

1. Introduction

In design of a control system, the use of an accurate model structure has a great significance to obtain the correct characteristic values of the system to be controlled. A mathematical definition procedure to characterize the system behavior by using input-output data of the system can be called as the system modeling [1-6]. Determination of the unknown parameters in the system model is usually performed by using an appropriate adaptation algorithm. In order to obtain a good model description, the elimination of the error value between the model and actual outputs by the algorithm is aimed [1, 4].

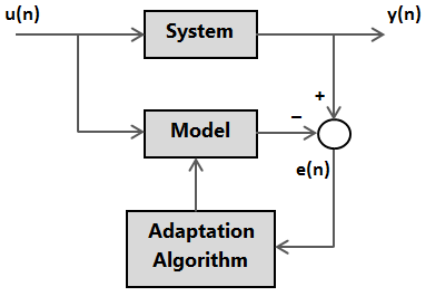

Figure 1. Basic block diagram for system modeling process

Basic block diagram of system modeling process is presented in Figure 1 [7]. In this figure, u(n), y(n) and e(n) are input, output and error signals, respectively.

Studies in recent years clearly prove that the artificial intelligence based algorithms can be widely and successfully used in the system modeling problems. Genetic algorithm (GA), differential evolution algorithm (DEA), clonal selection algorithm (CSA), particle swarm optimization (PSO) and artificial bee colony algorithm (ABC) are often used in the modeling of the systems [3-17]. In one of these studies presented by Zorlu and Ozer [3], a CSA is used in identification of nonlinear systems. In the study of Coelho and Pessoa [4], the authors use the GA in the nonlinear model identification process of an experimental ball-and-tube system. Luitel and Venayagamoorthy use the PSO approach for the modeling of the nonlinear systems [8]. In the study of Senberber and Bagis, the model parameters for time delay systems are estimated by using ABC algorithm [9]. In another study presented by Zorlu and Ozer [10], the parameter values of the proposed model

Corresponding Author. Email: halitsenberber@erciyes.edu.tr

structures for Box-Jenkins and a sample bilinear system problems are obtained by DEA and GA. Nam and Powers use the recursive least squares (RLS) method in the identification of the Volterra systems [11]. Performance of the ABC algorithm for time-delay model structure in the modeling of the unstable systems is examined by Senberber and Bagis [12]. A PSO based system modeling approach is presented by Deng [13]. In a study given by Bagis, the results of a system modeling approach based on the use of PSO method are presented in the definition of the higher order oscillatory systems by lower order models [14]. Ming and Dazi use new Luus-Jaakol algorithm (NLJ) for the identification of fractional systems in their work [15]. Least mean squares (LMS) algorithm is used for rapid system identification in the study given by Yu et al. [16]. In a detailed study conducted by the Senberber, different model structures have been proposed for the modeling of various systems with different characteristics and performance of the ABC algorithm in determination of the parameter values of these models have been presented as compared with the DEA and GA [17].

The main purpose of this paper is to examine the performance of the ABC algorithm in the modeling of higher order systems. For this reason, lower order model structures for the transfer functions of the various higher order systems given in the literature are taken into account. The parameter values of the proposed models are determined by using ABC algorithm. The simulation results are presented comparatively with the DEA, GA, and the model structures of the other methods given in the literature.

In the second section, the algorithms used in this study have been introduced briefly. The results of simulation studies obtained from different algorithms are given in the third section. The values of the model parameters, ISE error value, and computation times for the model parameters are presented in the tables in this section. Furthermore unit step responses of the systems and model results obtained by algorithms are also graphically given in this section. The conclusions of the study are presented in the final section.

2. The algorithms used for comparison

2.1 Artificial bee colony (ABC) algorithm

Artificial bee colony(ABC) algorithm utilizing the foraging behavior of honey bees was developed by Karaboğa in 2005 [17-25]. The basis of the algorithm is dependent on the proposed model by Tereshko and Loengarov to describe the foraging behavior of honey bee colonies [20]. This model has three main components; employed bees, unemployed bees and food sources [21]. Unemployed bees are divided into two groups, which are scout and onlooker.

Process steps of ABC algorithm can be given as follows [17-22]:

i) Scout bees randomly start the food searching process without any guidance.

ii) Nectar is moved to the hive from discovered food source by scout bees.

iii) Employed bees bring food to the hive, go back to the source or nectar source information is transferred to scout bees in hive via "information dance.

iv) Employed bee which consumes food, starts to work as a scout bee again and it searches the new food sources.

v) Information of food sources depending on the quality and frequency of the dance from employed bees is obtained by scout bees in the hive, and they are canalized to the food sources.

Considering this information, the basic steps of the ABC algorithm can be specified as the following [17, 21]:

1) *Produce an initial source*

2) *Send the employed bees to the food sources and determine the amount of nectar.*

3) *Calculate the selection probabilities of the sources by scout bees.*

4) *Selection of food sources by scouts bees according to the probability values.*

5) *Abandonment criteria: production of limit and scout bee*

6) *Number of Cycles: maximum number of cycles*

7) *Repeat the cycle until the stopping criteria is reached and return to step 2.*

ABC algorithm, that is a method based on swarm intelligence, quite closely simulates the actual foraging behavior of the bees. Main important features of the algorithm are that it has quite simple and flexible structure with few control parameters. Especially, the variations in the two important parameters such as colony size and limit value directly affect the quality of the solution. While global search is done by the scout

bees, local search is done by the employed bees and onlooker bees. These searches are carried out in parallel. A general statement produced by employed and onlooker bees can be given in the following form. This definition specifies the location of new food sources.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (1)$$

Assuming SN is the number of employed or onlooker bees, ABC algorithm starts to work with a random initial population P consisting of SN solutions. In Equation (1), k ($k \in \{1,2,\ldots..,SN\}$) a randomly chosen index that has to be different from i and j ($j \in \{1,2,\ldots..,D\}$) are randomly selected indice, and φ_{ij} is a random value in the interval [-1,1]. In here, D is the number of optimization parameters.

The i^{th} solution set in the population is defined by X_i. Thus, the size of variable 'i' is limited to the population or colony size. On the other hand, in Equation (1), the parameters of X_{ij} and X_{kj} can be considered as the current and neighbour solutions, respectively. And, a possible new solution is specified by the parameter of V_{ij} .

Quality value of each solution in the population is calculated by the employed bee of the related solution. When determining the new possible food source, a greedy selection method is used. This method makes the choice between the new source and the previous source in the memory.

If a position in the algorithm does not improve enough during the predetermined cycles, this means that the food source specified by the position is abandoned. This cycle size is an important and critical control parameter for the algorithm and it is called as limit. The value of this parameter is generally determined in proportion to the colony size and the number of parameters optimized. In this study, the limit parameter is defined according to the statement of [(colonies size x number of parameter) / 2]. Therefore, when the colony size for a sample system having a model with 5 parameters is selected as 10, the value of limit parameter is obtained as 25 ([(10x5) / 2]).

2.2 Differential evolution algorithm (DEA)

Differential evolution algorithm is a population-based heuristic algorithm to optimize the functions developed by Storn and Price [23, 24]. DEA introduced for solving numerical optimization problems is used the differences between solutions in the production of the new solutions during the simultaneous search at many points.

In the DEA benefiting from some main operators such as crossover, mutation, and selection to improve the solutions, the advancement in the solution quality is based on the use of mutation operator rather than crossover operation contrary to GA. The basis of this process is that the differences between the vectors defining possible solutions are multiplied by some coefficients such as scaling factor (F). Basic steps of the algorithm can be summarized as follows:

1. *Create the initial population*
2. *Evaluate the population members*
3. *Perform the following steps until the terminating criterion is achieved*
   - **3.a.** *Mutation*
   - **3.b.** *Crossover*
   - **3.c.** *Evaluation*
   - **3.d.** *Selection*

In the mutation process, the variation of the solutions is provided by using the differences between the population elements which are randomly selected. This difference is multiplied by a coefficient called as the scaling factor, and it is added to another population member. This situation for i^{th} element of the population can be defined as follows:

$$v_i^{(G+1)} = x_a^{(G)} + F(x_b^{(G)} - x_c^{(G)}) \quad (2)$$

Where, V is the mutation vector, G is the generation, Xa, Xb, Xc are different members of the population, and F is the scaling factor. In this study, in order to produce a new mutation vector Vi, the element of Xa is assumed as Xbest which is the best individual with high quality in the population.

Then, a recombination or crossover operation is applied to the population. In this paper, the binomial type crossover operation is used in the algorithm. According to this procedure, the new solution element $u_{i,j,G}$ of the population is obtained as follows:

$$u_{i,j,G+1} = \begin{cases} v_{i,j,G+1} & \text{if } rand_j(0,1) < CR \text{ or } j = j_{rand} \\ x_{i,j,G} & \text{other} \end{cases} \quad (3)$$

Where, $V_{i,j,G+1}$ is the mutant vector, randj is a uniform random number in the interval of (0,1), CR is the crossover rate ($CR \in [0, 1]$), j =1, 2, .., D is an integer number, and j_{rand} is a random number in [1,D].

The selection process defines a criterion to propagate the fittest elements in the new generation for new produced solution vector. The performances of the trial vector $U_{i,G+1}$ and its parent $X_{i,G}$ are compared and the better one is

selected according to their fitness values *f( )*. The selection process can be expressed as follows:

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & if \quad f\left(u_{i,G+1}\right) \le f\left(x_{i,G}\right) \\ x_{i,G} & other \end{cases} \qquad (4)$$

DEA is a fast, easy to use, simple approach that can provide effective convergence with low computational cost for complex optimization problems. Operation of the DEA is continued by improving the individuals in the population at each iteration, and when the desired solution or predetermined iterations are reached, search of the algorithm is terminated.

2.3 Genetic algorithm (GA)

The foundations of genetic algorithm (GA), which is used for solving various optimization problems in many field of the engineering in today, were firstly introduced by J. Holland [25-30]. Basic principle of the GA that is a population-based heuristic algorithm is relied on the two keynotes: (i) protection of the best, and (ii) obtaining the new individuals with better quality by utilizing the old solutions generated. For this purpose, initially, algorithm starts to the operation by the evaluation of a population which consists of a certain number of possible solutions (chromosome). Then, GA tries to improve the solution quality by performing operations of reproduction, crossover and mutation in the population. The main aim in these processes depending on the crossover and mutation rates is to achieve the high quality solutions instead of the poor quality solutions. Basic steps of the algorithm can be summarized as follows:

1. *Create the initial population*
2. *Determine the fitness (quality) values of the population elements*
3. *Apply the mating, crossover and mutation processes*
4. *Create a new population, determine the individual with most appropriate solution*
5. *If the stopping criterion is satisfied stop, otherwise return to step 2.*

The main important element of the GA is the crossover operator which is used to explore the search space for high quality new solutions. In this paper, single point crossover operator is employed as shown in Figure 2. In this process, crossover point is randomly determined, and then, the solution parts of the parents are mutually moved. Thus, new possible solutions for new population are obtained. On the other hand, the main principle in the mutation is the randomly alteration of the genes on the chromosome. In this operation, according to mutation rate which is in the interval of [0-1], some genes on the chromosome are randomly determined, and the values of these elements are changed into a different value. A typical mutation process is given in Figure 3.

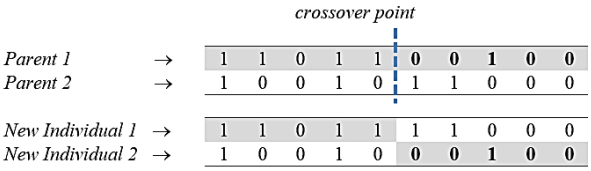

Figure 2. Single point crossover operation

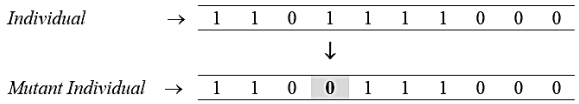

Figure 3. Typical mutation operation

Predictably, some critical decisions such as the coding of the solutions, crossover type, crossover and mutation rates directly affect the performance of the GA that can provide fast solutions depending on the population size used.

3. Simulation results

In this study, model structures for some examples of high-order systems in the literature are proposed, and the parameters of these model structures are obtained by using the ABC algorithm. This work is also repeated by using DEA and GA, and then, the results are comparatively presented. Four different high-order systems, which are previously given in the literature, are used in this study (Equations 5-8) [14, 31-39]. Time delay does not exist in the systems defined as G1 [31], G2 [14, 32-36], G3 [37, 38], and G4 [39]. The proposed model structure for these systems is stated as Gm in Eq.9. The number of parameters that must be obtained by the algorithms is 5 for the model considered.

Control parameters of algorithms that are used to obtain the proposed model parameters are presented in Table 1. The studies were repeated for 30 times at least in the model, and the control parameters in the studies with the values of the best model parameters are given in this table. In the simulations, Matlab program package and Intel Core 2 Duo E7500 2.93 GHz computer were used [40-42].

$$G1(s)=\frac{35s^7+1086s^6+13285s^5+82402s^4+278376s^3+511812s^2+482964s+194480}{s^8+21s^7+220s^6+1558s^5+7669s^4+24469s^3+46350s^2+45952s+17760} \quad (5)$$

$$G2(s)=\frac{18s^7+514s^6+5982s^5+36382s^4+122664s^3+222088s^2+185760s+40320}{s^8+36s^7+546s^6+4536s^5+22449s^4+67284s^3+118124s^2+109584s+40320} \quad (6)$$

$$G3(s)=\frac{6s^4+50s^3+196s^2+418s+434}{s^6+12s^5+71s^4+256s^3+575s^2+804s+585} \quad (7)$$

$$G4(s)=\frac{25.2s^2+21.2s+3}{s^5+16.58s^4+25.41s^3+17.18s^2+11.70s+1} \quad (8)$$

$$Gm(s)=\frac{ps+q}{as^2+bs+c} \quad (9)$$

The parameter values obtained by the algorithms in the proposed model for different systems are presented in Table 2. The information of the ISE (integrated squared error) error value, computation time for the algorithms and standard deviation are also given in the table. Unit step responses of the model obtained by the algorithms and iteration-error variations during the optimizations for all processes are collectively presented in Figure 4. Step response curves for G1, G2, G3, and G4 processes are illustrated in Figures 4(a), (c), (e), and (g). The iteration-error changes belonging to these curves are stated in Figures 4(b), (d), (f), and (h). Because it is sufficient to measure the model performance, the time intervals in the simulations are used as [0-6] sec for G1, G2, and G3 processes. This time period for G4 process is in the interval of [0-30] sec.

Table 1. Control parameters of the algorithms used in this study

Control Parameter		**DEA**	**GA**	**ABC**
Population (or colony) Size		10	10	10
Crossover Rate		0.9	0.9	-
Mutation Rate		-	0.3	-
Scaling Factor (F)		0.8	-	-
Combination Factor		0.8	-	-
Limit		-	-	25
Number of Generation (or iteration)	G1	300	300	300
	G2, G3, G4	100	100	100
Parameter Search Interval		[0-100]		

It can be clearly seen from Table 2 that the DEA based models have the lowest ISA error value nearly for all systems. For G1, G2, and G4 processes in the case of using DEA, the lowest error values such as 0.724, 4.090x10-4, and 3.290 as compared with other algorithms are obtained. Similarly, a very low ISE value as 7.52x10-4 can be achieved for G3 process by DEA again. On the other hand, it should be momentously noted that the error performance of the ABC algorithm is impressively good for all models. The error value obtained based on the use of ABC algorithm for G2, G3, and G4 is found almost equal to the DEA. The error value in the model of G1 is achieved at the low values as 1.224. Thus, it would not be wrong to argue that the error performances of the DEA and ABC algorithms are close to each other for all models examined. However, the modeling performance of the GA lies distinctly behind the other algorithms.

On the other hand, it can be clearly seen from Table 2 that an interesting feature of the ABC algorithm is noted in the table: computation time of the ABC algorithm for model parameters is dramatically longer than the other algorithms. This time none of the systems in the ABC approach is close to other algorithms, and moreover, it is three or four times more of the calculation costs in the other algorithms for the processes. For the modeling of the G1 and G2 processes, the computation times in DEA algorithm is obtained as 44 sec and 39 sec, at these times ABC algorithm is approximately noted as 135 sec and 184 sec, respectively.

Table 2. Parameter values of the models based on the algorithms used for different systems

System	Alg.	Model Parameters					Error (ISE)	Time (sec)	Standard Deviation
		p	q	a	b	c			
G1	DEA	10.079	93.631	0.249	0.448	8.763	0.724	44.349	0
	GA	99.711	71.669	0.949	8.776	6.819	29.540	42.658	12.349
	ABC	8.177	98.915	0.248	0.476	9.208	1.224	135.076	7.569
G2	DEA	68.779	20.419	4.021	28.050	20.728	4.090×10^{-4}	39.224	0
	GA	77.036	33.659	5.216	30.830	30.569	2.307×10^{-2}	43.590	5.610×10^{-2}
	ABC	19.697	5.889	1.152	8.024	5.957	4.150×10^{-4}	184.357	1.151×10^{-2}
G3	DEA	4.8102	58.988	13.293	39.450	79.828	7.52×10^{-4}	22.196	0.003509
	GA	50.613	46.124	54.565	63.249	79.427	42.81×10^{-4}	18.169	0.0015
	ABC	1.471	45.132	9.345	27.064	60.561	0.71×10^{-4}	45.420	0.003415
G4	DEA	3.91×10^{-4}	61.3966	32.3769	20.7600	22.0936	3.290	13.391	2.452×10^{-2}
	GA	9.6599	77.334	49.423	27.828	27.934	3.595	13.452	8.729×10^{-1}
	ABC	1.00×10^{-6}	88.899	46.422	31.015	31.956	3.291	30.440	2.632×10^{-2}

Modeling times of the G3 process for DEA, GA, and ABC methods occur as 22sec, 18sec, and 45sec, respectively again. In the modeling of G4, while the computation time is about 13 sec for the other algorithms, this time it is 30 sec for ABC algorithm. This situation is no doubt a negative feature that should not be ignored for the ABC algorithm.

From the results given in Table 2, it can be obviously show detected that the standard deviation values of the DEA indicate a striking accuracy to achieve the same ISE values. In the modeling of the other systems except for G1, the ABC algorithm exhibits satisfying standard deviation values. Although the order of the systems and the number of numerator-denumarator components in the processes of G1 and G2 is same, it is an interesting fact that while a high consistency is achieved in the modeling of G2 by the ABC, this consistency level in the modeling of G1 is undesirably low.

From the step responses of the models in Figure 4, it can be seen that there is a good agreement among the ABC, DEA, and actual process outputs in general. Admittedly, the proposed model structure is not sufficiently successful to define the process of G4. Howbeit, in order to see the power of the ABC algorithm, is an appropriate example. On the other hand, the modeling performance of the GA is quite behind the other algorithms in accordance with the error values given in Table 2, and thus, it is not successful in terms of the agreement of the model-actual outputs observed in the Figure 4.

As mentioned previously, from the results in Table 2, the DEA exhibits a better performance with less computational effort than those of ABC algorithm in terms of the modeling error. At this point, an interesting and exciting result is clearly observed when the curves of iteration-error change in Figure 4 are examined: the ABC algorithm can reduce the error at the remarkably lower iteration numbers than the other algorithms. For example, for all of the processes, while the ISE error is dramatically eliminated in the first 10 iterations by the ABC algorithm, DEA and GA need the use of high iteration numbers for the optimization. For a considerable reduction in the ISE error, 50 iterations approximately are spent by the DEA.

In the G2 and G3 processes, this algorithm does not come close to the error value achieved by the ABC during the 20 iterations at least. The similar situation is also present for the modeling of G4 process. For this process, the remarkable error reductions in the first 10 iterations of the ABC algorithm are hardly achieved at about 50 iterations by the DEA. It can be said that the error reduction performance of the ABC algorithm is highly impressive according to other methods.

To objectively evaluate the performances of the algorithms, the error values of the different approaches in the literature are presented in Table 3 as compared with the error values of the models optimized. In this table, the model structures that can be used for each process are shown as more clearly. From the values in the Table 3, it is possible to say that the error values of the ABC based model are much better than those of the other methods stated in the literature.

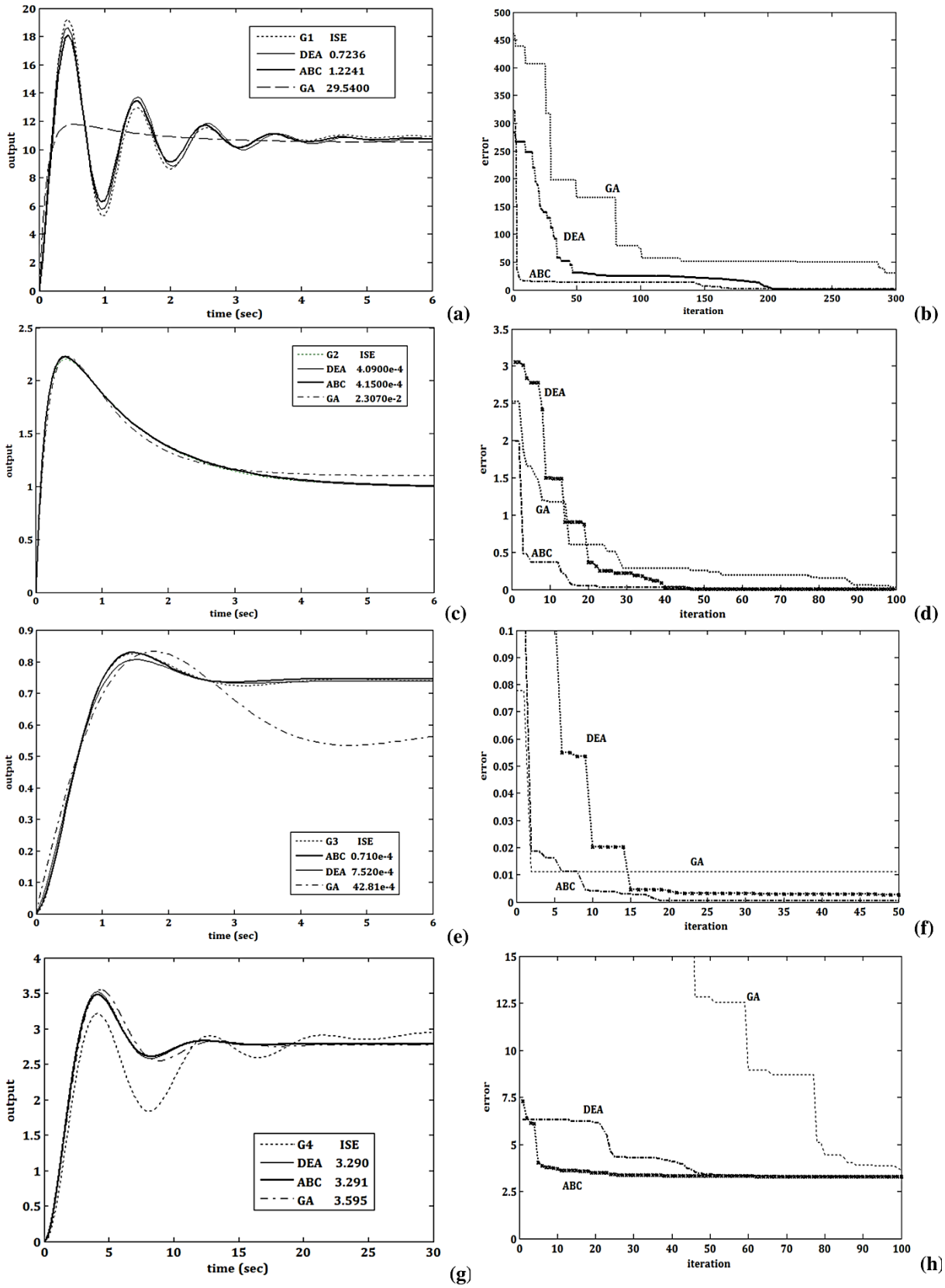

Figure 4. Unit step responses and iteration-error variation curves of the models obtained by the algorithms for each process

Table 3. Proposed model structures in the literature for the systems used

System	Method/Model/Error	System	Method/Model/Error
G1	**Manigandan [31]** $\frac{35s+401.21}{s^2+1.436s+36.63}$ ISE: 2.5021	G2	**Lucas [32]** $\frac{18s+5.603}{s^2+7.415s+5.603}$ ISE: 2.000×10^{-3}
	DEA $\frac{10.079s+93.631}{0.249s^2+0.448s+8.763}$ **ISE: 0.724**		**Krajewski [32]** $\frac{17.23s+6.375}{s^2+7s+6}$ ISE: 1.800×10^{-1}
	GA $\frac{99.711s+71.669}{0.949s^2+8.776s+6.189}$ ISE: 29.54		**Mukherjee [32]** $\frac{16.89s+5.279}{s^2+6.85s+5.279}$ ISE: 7.081×10^{-4}
	ABC $\frac{8.177s+98.915}{0.248s^2+0.476s+9.208}$ **ISE: 1.224**		**Shamash [33]** $\frac{6.779s+2}{s^2+3s+2}$ ISE: 7.3183
G3	**Parmar[37]** $\frac{5.275}{s^2+3.051s+7.109}$ ISE: 4.174×10^{-3}		**Prasad [34]** $\frac{17.986s+500}{s^2+13.25s+500}$ ISE: 18.430
	Layer [38] $\frac{6}{s^2+3.66s+7.78}$ ISE: 3.363×10^{-3}		**Mittal [35]** $\frac{7.091s+1.991}{s^2+3s+2}$ ISE: 6.9159
	DEA $\frac{4.8102s+58.988}{13.293s^2+39.45s+79.828}$ ISE: 7.52×10^{-4}		**Parmar [36]** $\frac{24.144s+8}{s^2+9s+8}$ ISE: 1.792
	GA $\frac{50.613s+46.124}{54.565s^2+63.249s+79.427}$ ISE: 4.281×10^{-3}		**Bagis [14]** $\frac{4.256s+1.263}{0.248s^2+1.735s+1.282}$ ISE: 4.176×10^{-4}
	ABC $\frac{1.471s+45.132}{9.345s^2+27.064s+60.561}$ **ISE: 0.71×10^{-4}**		**DEA** $\frac{68.779s+20.419}{4.021s^2+28.05s+20.728}$ **ISE: 4.090×10^{-4}**
G4	**DEA** $\frac{3.9059.10^{-4}s+61.3966}{32.3769s^2+20.76s+22.0936}$ **ISE: 3.290**		**GA** $\frac{77.036s+33.659}{5.216s^2+30.83s+30.569}$ ISE: 2.307×10^{-2}
	GA $\frac{9.6599s+77.334}{49.423s^2+27.828s+27.934}$ ISE: 3.595		**ABC** $\frac{19.697s+5.889}{1.152s^2+8.024s+5.957}$ **ISE: 4.150×10^{-4}**
	ABC $\frac{1.00.10^{-6}s+88.899}{46.422s^2+31.015s+31.956}$ **ISE: 3.291**		

4. Conclusions

In this study, an investigation about the definition of the higher-order systems by the lower order models is presented. For this aim, transfer functions of some higher-order systems in the literature are taken into consideration, and several second-order models with 5 parameters are proposed to describe these systems. Determination of the model parameters with the smallest error value is expected from the ABC algorithm. In order to provide a comparison with the literature, ISE error index is used as the performance criteria. Simulation studies are also repeated by using DEA and GA, and the results obtained are comparatively presented in the tables. Other model structures in the literature and error values are also included in the tables.

From the simulation results, the general evaluations given below can be done:

a. When considering the unit step input behavior, appropriate approaches through the use of second order low-order models to describe the higher order systems can be obtained. It is clear that, if different characteristic values of time and frequency domains are taken into consideration, the abilities to define the processes of these models can be improved and their accuracy properties can be increased.
b. It is important for us to investigate the system modeling performance of the ABC algorithm that presents the efficient results for numerical optimization problems especially. Error reduction performance of the ABC algorithm is satisfactory in the determination of model parameters. According to simulation results, it is possible to obtain the lower values of the error in very low iterations by using this approach.
c. In order to achieve the low error values, long computation time of the ABC algorithm as compared with the other algorithms is a negative feature of the method. For a good time cost-performance relationship, some improvements and time-recovering adjustments in the program used for the operation are clearly required.
d. When compared to the other algorithms in this study and the error values of the other models in the literature, it is possible to say that the modeling performance of the ABC algorithm is in a successful level. The performance results presented by the algorithm proved that ABC algorithm can be safely and effectively used in the modeling and controlling of more complex and more oscillatory systems for various purposes.

References

[1] Ljung, L., System Identification: Theory for The User, Prentice Hall, 1987.

[2] Söderström, T., Stoica, P., System Identification, Prentice-Hall, 1989.

[3] Zorlu, H., Özer Ş., Identification of Nonlinear Systems By Using Clonal Selection Algorithm, IEEE 17th Signal Processing and Communications Applications Conference (SIU-2009), Proceedings CD, Antalya, Turkey, 2009 (in Turkish).

[4] Coelho L. S. , Pessoa M. W., ‘‘Nonlinear Model İdentification Of An Experimental Ball-And-Tube System Using A Genetic Programming Approach’’, Mechanical Systems and Signal Processing, 23(5), 1434–1446, 2009.

[5] Pintelon R., and Schoukens J., ‘‘System Identification - A Frequency Domain Approach’’, IEEE Press, Piscataway, 2001.

[6] Bağış A., Özçelik Y., System Identification By Using Real Coded Genetic Algorithm, 12th Electrical, Electronics, Computer, Biomedical Engineering National Congress, Eskişehir, Turkey, 2007 (in Turkish).

[7] Zorlu, H., Özer Ş., Identification of Nonlinear Volterra Systems Using Differential Evolution Algorithm, Electrical, Electronics, and Computer Engineering Symposium (ELECO-2010), Proceedings CD, Bursa, Turkey, 2010 (in Turkish).

[8] Luitel B., Venayagamoorthy G.K., ‘‘Particle swarm optimization with quantum infusion for system identification’’, Engineering Applications of Artificial Intelligence, vol. 23, pp.635-649, 2010.

[9] Şenberber H., Bağış A., Performance Investigation of Artificial Bee Colony Algorithm in Modeling of Systems with Delay, Symposium on Innovations in Intelligent Systems and Applications (ASYU’2012), pp.164-168, Trabzon, Turkey, 2012 (in Turkish).

[10] Ozer, S., Zorlu, H., “Identification Of Bilinear Systems Using Differential Evolution

Algorithm", Sadhana-Academy Proceedings in Engineering Sciences, Vol. 36, Part 3, pp. 281–292. June 2011,

[11] Nam S.W., Powers, E.J., "Application Of Higher Order Spectral Analysis To Cubically Nonlinear System Identification", IEEE Trans. on Signal Processing, 42, 1746-1765, 1994.

[12] Şenberber H., Bağış A., Determination of Model Parameters of Unstable Systems By Using Artificial Bee Colony Algorithm, Automatic Control National Meeting (TOK'2012), pp.812-816, Niğde, Turkey, 2012 (in Turkish).

[13] Deng X., "System Identification Based on Particle Swarm Optimization Algorithm", International Conference on Computational Intelligence and Security (CIS'09), pp. 259-263, 2009.

[14] Bağış A., Performance Investigation of Partical Swarm Optimization in Higher Order Oscillatory Systems Modeling, Automatic Control National Meeting (TOK'2010), pp.168-173, Kocaeli, Turkey, 2010 (in Turkish).

[15] Ming Z., Dazi L., "Fractional System Identification Based on Improved NLJ Algorithm", Control and Decision Conference (CCDC), 24th Chinese, pp. 1057-1061, 2012.

[16] Yu R., Song Y., Rahardja S., "Lms In Promınent System Subspace For Fast System Identıficatıon", 2012 IEEE Statistical Signal Processing Workshop (SSP), pp. 209-212, 2012.

[17] Şenberber H., Artificial Bee Colony Algorithm Performance Investigation of System Modeling, M.Sc. Thesis, Erciyes University, Graduate School of Natural and Applied Sciences, Kayseri, Turkey, 2012 (in Turkish).

[18] Karaboğa, D., "An Idea Based On Honey Bee Swarm For Numerical Optimization, Tecnical Report TR06", Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[19] Karaboğa, D., Öztürk, C., 'A Novel Clustering Approach: Artificial Bee Colony (ABC) Algorithm',Applied Soft Computing, Vol 11, 652–657, 2011.

[20] Karaboğa, D., ''Artificial Bee Colony Algorithm'' Scholarpedia, 5(3):6915, 2010.

[21] Karaboğa, D., Akay, B., ''A Comparative Study Of Artificial Bee Colony Algorithm'', Applied Mathematics and Computation, Vol 214, 108–132, 2009.

[22] Akay, B., Karaboğa D,. ''A modified Artificial Bee Colony algorithm for real-parameter optimization'', Information Sciences, Vol. 192, pp.120–142, 2012.

[23] Price, K., Storn, R., "Differential Evolution: Numerical Optimization Made Easy", Dr. Dobb's J. 78, 18–24, 1997.

[24] Storn, R., Price, K., "Differential Evolution: A Simple and Efficient Heuristic For Global Optimization Over Continuous Spaces", Journal of Global Optimization, Vol 11, 341-359, 1997.

[25] Karaboğa, D., Artificial Intelligence Optimization Algorithms, Nobel Publ., ISBN: 975-6574-37-2, 2004 (in Turkish).

[26] Bağış, A., Savaşçıhabeş, A., Determination of PID Controller Parameters By Using Binary and Real Coded Genetic Algorithm, Symposium on Innovations in Intelligent Systems and Applications (ASYU'2008), pp.198-202, Isparta, Turkey, 2008 (in Turkish).

[27] Pham, D.T., Karaboğa, D., "Intelligent Optimization Techniques: Genetic Algorithms. Tabu Search, Simulated Annealing and Neural Networks", Springer– Verlag, 2000.

[28] Goldberg, D.E., Genetic Algorithms in Search, Optimisation and Machine Learning, Addison-wesley Publ. Co. Inc., USA, 1989.

[29] Goldberg, D.E., Lingle, R., Alleles, Loci and the Travelling Salesman Problem in Grefenstette J. J., (ed) Proceedings of the 1st International Conference on Genetic Algorithms and their Applications, Carneige-Mellon University, Pittsburgh, Lawrence Erlbaum Assoc, Publishers, 1985.

[30] Barrios A., Ballestin F., Valls V., "A Double Genetic Algorithm Fot The MRCPSP/max", Computer & Operations Research, Vol 38, 33-43, 2011.

[31] Manigandan T. , Devarajan N. , Sivanandam S., "Desing of PID Controller Using Reduced Order Model", AOI Journal, Vol. 15, 2005.

[32] Mukherjee S., Satakshi, Mittal, R.C., "Linear time invariant system order reduction using multipoint step response matching",

International Journal of Systems Science, Vol. 38, No. 3, 211–217, 2007.

[33] Shamash Y., "Linear system reduction using Pade approximation to allow retention of dominant modes", International Journal of Control, Vol. 21, No. 2, 257-272, 1975

[34] Prasad, R. and Pal, J., "Stable reduction of linear systems by continued fractions", Journal of the Institution of Engineers (India), Part EL, 72, 113–116. 1991

[35] Mittal A.K., Prasad R., and Sharma S.P., "Reduction of linear dynamic systems using an error minimization technique", Journal of Institution of Engineers IE(I) Journal – EL, Vol. 84, pp. 201-206, March 2004.

[36] Parmar G., Mukherjee S., Prasad R., "System reduction using Eigen spectrum analysis and Padé approximation technique", International Journal of Computer Mathematics, Vol.84, No.12, pp. 1871-1880, 2007.

[37] Parmar G., Mukherjee S., Prasad R., "Relative Mapping Errors of Linear Time Invariant Systems Caused By Particle Swarm Optimized Reduced Order Model", World Academy of Science, Engineering and Technology, Vol. 28, pp. 336-342, 2007.

[38] Layer E., "Mapping Error of Linear Dynamic Systems Caused by Reduced-Order Model", IEEE Transactıons On Instrumentatıon And Measurement, Vol. 50, No. 3, 2001.

[39] Kaya İ., "Parameter Estimation for Integrating Processes Using Relay Feedback Control under Static Load Disturbances", Ind. Eng. Chem. Res., 45, 4726-4731, 2006.

[40] Dorf R.C., Bishop R.H., Modern Control Systems-8th Edt., Addison-Wesley Longman, Inc., 1998.

[41] MathWorks, Control System Toolbox for Use with Matlab-Computation, Visualization, Programming, Using the Control System Toolbox-Version 1, The MathWorks, Inc., 2000.

[42] MathWorks, Control System Toolbox Documentation Center, http://www.mathworks.com/help/control/index.html (Accessed: 02/2014)

Aytekin Bagis *was born in Bartin, Turkey, on October 04, 1973. He received his Ph.D. degree from the Department of Electronic Engineering, Erciyes University, Kayseri, Turkey in 2001. He is currently an associate professor at Electrical and Electronic Engineering Department, Erciyes University, Kayseri, Turkey. His main research interests include PID controller design, intelligent control systems based on fuzzy logic, artificial intelligence optimization algorithms, system modeling, fuzzy modeling and control.*

Halit Senberber *was born in Kayseri, Turkey, on June 01, 1986. He received the B.S. and M.S. degrees from the Electrical and Electronic Engineering Department, Erciyes University, Kayseri, Turkey in 2009 and 2012, respectively. He has been an assistant of Electrical and Electronic Engineering Department, Erciyes University, Kayseri, Turkey, since 2010. His research interests concern artificial intelligence optimization algorithms, system modeling, and PID controller tuning.*