Expert Systems with Applications 38 (2011) 12671–12683

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

IIR system identification using cat swarm optimization

Ganapati Panda [a], Pyari Mohan Pradhan [a,*], Babita Majhi [b]

[a] School of Electrical Sciences, Indian Institute of Technology Bhubaneswar, India
[b] Department of Information Technology, ITER, SOA University Bhubaneswar, India

ARTICLE INFO

Keywords:
System identification
IIR system
Cat swarm optimization

ABSTRACT

Conventional derivative based learning rule poses stability problem when used in adaptive identification of infinite impulse response (IIR) systems. In addition the performance of these methods substantially deteriorates when reduced order adaptive models are used for such identification. In this paper the IIR system identification task is formulated as an optimization problem and a recently introduced cat swarm optimization (CSO) is used to develop a new population based learning rule for the model. Both actual and reduced order identification of few benchmarked IIR plants is carried out through simulation study. The results demonstrate superior identification performance of the new method compared to that achieved by genetic algorithm (GA) and particle swarm optimization (PSO) based identification.

1. Introduction

Adaptive IIR filtering is an active area of research for many years and has been used for applications in signal processing and communication. There has been substantial effort to establish adaptive IIR filters as alternative to adaptive FIR filters. In this filter the output feedback generates an infinite impulse response with only a finite number of parameters. Hence with same number of coefficients, an adaptive IIR filter performs better than an adaptive FIR filter. Alternatively, to achieve a particular level of performance, an IIR filter requires less number of coefficients than the corresponding FIR filter. This is because the desired response can be approximated more effectively by the output of a filter that has both poles and zeros compared to one that has only zeros (Shynk, 1989a). As a result an adaptive IIR filter with sufficient number of poles and zeros can exactly model an unknown pole-zero system, whereas an adaptive FIR filter with higher order can only approximate it.

A major concern in adaptive IIR filtering applications is that the error surface is usually non-quadratic and multimodal with respect to the filter coefficients. The gradient-based learning algorithms such as least mean square algorithm tries to find out the global minimum of the error surface by moving in the direction of negative gradient and hence can easily be struck at local minima and cannot converge to global minimum (Widrow & Strearns, 1985). In addition the IIR filters become unstable if the poles move outside the unit circle during the learning process. Therefore, for learning of higher order adaptive IIR filters, stability-monitoring is very essential. The properties of an adaptive IIR filter are considerably more complex than an adaptive FIR filter, and hence it is more difficult to predict the behavior of an adaptive IIR algorithm. Another drawback of adaptive IIR filters is slow convergence which needs further attention. In order to overcome these problems, several new structures and algorithms have been proposed in the literature (David, 1981; Regalia, 1992; Shynk, 1989b). Most of the adaptive IIR filters are realized in direct form due to its simple implementation and analysis. However, some disadvantages of the direct form such as finite-precision effects and the complexity of stability monitoring have led to the development of alternative structures like cascade (David, 1981), lattice (Regalia, 1992) and parallel (Shynk, 1989b). The computational complexity and convergence properties of adaptive algorithms depend on the filter realization.

In past few years several efforts have been made for studying alternative structures and algorithms for adaptive IIR filters. Recently evolutionary algorithms, such as GA, PSO, evolutionary programming (EP) and evolutionary strategies (ES) have received much attention for global optimization problems. These evolutionary algorithms are heuristic population-based search techniques and incorporate random search and selection principle to achieve the global optimal solution.

2. Related work

A number of adaptive system identification techniques have been reported in literature (Astriim & Eykhoff, 1971; Friedlander, 1982; Ljung, 1987; Siiderstrom, Ljung, & Gustavsson, 1978). An early attempt to implement adaptive IIR filter was made by White (1975). Johnson (1984) presented a tutorial on adaptive filtering in

* Corresponding author. Tel.: +91 9238000246.
E-mail addresses: ganapati.panda@gmail.com (G. Panda), pyarimohan.pradhan@gmail.com (P.M. Pradhan), babita.majhi@gmail.com (B. Majhi).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.054

which the concepts of adaptive control and adaptive filtering are unified. Another tutorial was also published by Shynk (1989a) which provides an overview of several methods, filter structures, and recursive algorithms used in adaptive IIR filtering. Soderstrom, Stoica and Nayeri have investigated on sufficient conditions for which there are no local minima for the system identification configuration (Nayeri, 1988; Soderstrom, 1975; Soderstrom & Stoica, 1982).

In the past many evolutionary algorithms have been employed for design and identification of IIR filter. In Ng, Leung, Chung, Luk, and Lau (1996), genetic algorithm along with the least mean square (LMS) algorithm are used in adaptive IIR filtering which gives faster convergence and global search capability in comparison to conventional LMS and genetic algorithm implementations. In Abe and Kawamata (1998a, 1998b), Abe, Kawamata, and Higuchi (1996), evolutionary digital filtering is used for IIR adaptive digital filters. In this approach, instead of using any gradient-based algorithms such as the LMS, several digital filters are used and the best output among them is selected. The filter coefficients are updated using evolutionary programming technique which makes it a computationally complex algorithm. Recently a new method for digital IIR system identification based on artificial bee colony optimization is proposed in Karaboga (2009). Adaptive IIR system identification based on PSO with quantum infusion is proposed in Luitel and Venayagamoorthy (2010). Majhi and Panda (2010) has also proposed an efficient identification of complex nonlinear dynamic plants using PSO and Bacterial Foraging Optimization (BFO).

The goal of this paper is to introduce CSO algorithm as an adaptive learning tool for IIR system identification. In Section 3, the mathematical formulation of IIR system identification is discussed. The basic steps of CSO algorithm along with the flowchart is presented in Section 4. The CSO based training of IIR parameters is dealt in Section 5. The simulation result and its analysis is provided in Section 6. The sensitivity of different parameters of CSO are explained in Section 7. Some concluding remarks and scope for future work are listed in Section 8.

3. IIR system identification

In the system identification configuration, the adaptive algorithm searches for the adaptive filter coefficients such that its input/output relationship matches closely to that of the unknown

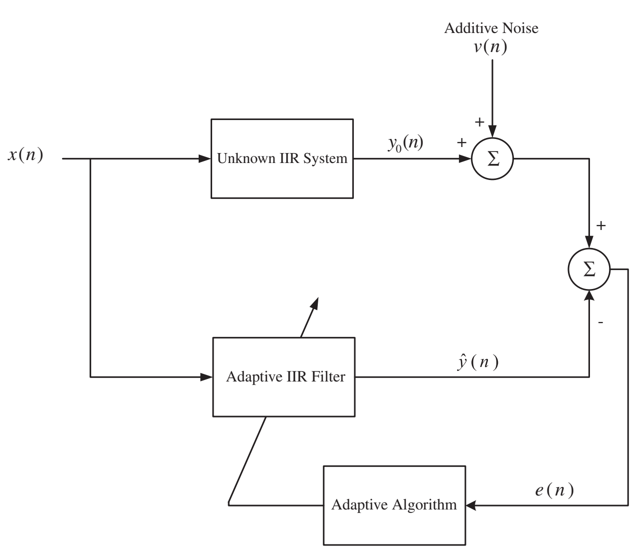

Fig. 1. Block diagram of IIR system identification.

system. The block diagram of an adaptive IIR system identification is shown in Fig. 1. An IIR system is described using a difference equation as

$$y_0(n) = H_P(z)x(n) \tag{1}$$

where $x(n)$ is the input signal and $y_0(n)$ is the output of the IIR plant. $H_P(z)$ is the transfer function of the unknown plant and is given as

$$H_P(z) = \left[\frac{A(z)}{B(z)}\right] \tag{2}$$

$A(z)$ and $B(z)$ are Z-domain feed-forward and feed-back coefficient polynomials of IIR plant respectively and is represented as

$$A(z) = \sum_{i=0}^{L} a_i z^{-i}$$

and

$$B(z) = 1 - \sum_{i=1}^{M} b_i z^{-i}$$

The feedback filter order M is greater than the feed forward filter order L. The overall output of the plant is given as

$$y(n) = y_0(n) + v(n) \tag{3}$$

where $v(n)$ is an additive white Gaussian noise. Combining (1) and (3), we get

$$y(n) = \left[\frac{A(z)}{B(z)}\right]x(n) + v(n) \tag{4}$$

The adaptive filter is governed by the difference equation:

$$\hat{y}(n) = H_M(z)x(n) \tag{5}$$

where $H_M(z)$ is the transfer function of the IIR model and is represented as

$$H_M(z) = \left[\frac{\widehat{A}(z)}{\widehat{B}(z)}\right] \tag{6}$$

$\widehat{A}(z)$ and $\widehat{B}(z)$ are feed forward and feedback coefficient polynomials of the adaptive filter respectively and are given as

$$\widehat{A}(z) = \sum_{i=0}^{L} \hat{a}_i z^{-i}$$

and

$$\widehat{B}(z) = 1 - \sum_{i=1}^{M} \hat{b}_i z^{-i}$$

The symbols $\hat{a}_i$ and $\hat{b}_i$ denote the estimated feed-forward and feedback coefficient of the model, respectively. The transfer function $H_P(z)$ of IIR plant is to be identified using the transfer function $H_M(z)$ of the adaptive filter. This task of identification is formulated as an optimization problem where mean square error (MSE) defined in (7) is used as the cost function

$$J = E[e^2(n)] \approx \frac{1}{N}\sum_{n=1}^{N} e^2(n) \tag{7}$$

where $e(n) = y(n) - \hat{y}(n)$ is the error signal and N is the number of input samples to be used and $E(\cdot)$ represents the statistical expectation operator.

4. Cat swarm optimization

Chu and Tsai (2007) have proposed a new optimization algorithm which imitates the natural behavior of cats. Cats have a strong curiosity towards moving objects and possess good hunting skill. Even though cats spend most of their time in resting, they always remain alert and move very slowly. When the presence of a prey is sensed, they chase it very quickly spending large amount of energy. These two characteristics of resting with slow movement and chasing with high speed are represented by seeking and tracing, respectively. In CSO these two modes of operations are mathematically modeled for solving complex optimization problems.

4.1. Seeking mode

The seeking mode corresponds to a global search technique in the search space of the optimization problem. Some of the terms related to this mode are:

- Seeking Memory Pool (SMP): It is the number of copies of a cat produced in seeking mode.
- Seeking Range of selected Dimension (SRD): It is the maximum difference between the new and old values in the dimension selected for mutation.
- Counts of Dimension to Change (CDC): It is the number of dimensions to be mutated.

The steps involved in this mode are:

1. Create T copies of ith cat.
2. Based on CDC update the position of each copy by randomly adding or subtracting SRD percents the present position value.
3. Evaluate the fitness of all copies
4. Pick the best candidate from T copies and place it at the position of ith cat.

4.2. Tracing mode

The tracing mode corresponds to a local search technique for the optimization problem. In this mode, the cat traces the target while spending high energy. The rapid chase of the cat is mathematically modeled as a large change in its position. Define position and velocity of ith cat in the D-dimensional space as $X_i = (X_{i1}, X_{i2}, X_{iD})$ and $V_i = (V_{i1}, V_{i2}, V_{iD})$ where $d(1 \leqslant d \leqslant D)$ represents the dimension. The global best position of the cat swarm is represented as $P_g = (P_{g1}, P_{g2}, \ldots, P_{gD})$. The update equations are:

$$V_{id} = w * V_{id} + c * r * (P_{gd} - X_{id}) \tag{8}$$

$$X_{id} = X_{id} + V_{id} \tag{9}$$

where w is the inertia weight, c is the acceleration constant and r is a random number uniformly distributed in the range $[0,1]$.

4.3. Algorithm

The CSO algorithm reaches its optimal solution using two groups of cats, i.e. one group containing cats in seeking mode and other group containing cats in tracing mode. The two groups combine to solve the optimization problem. A mixture ratio (MR) is used which defines the ratio of number of cats in tracing mode to that of number of cats in seeking mode.

1. Randomly initialize the position of cats in D-dimensional space for the population, i.e. X_{id} representing position of ith cat in dth dimension.
2. Randomly initialize the velocity for cats, i.e. V_{id}.
3. Evaluate the fitness of each cat and store the position of the cat with best fitness as P_{gm} where $m = 1, 2, \ldots, D$.
4. According to MR, cats are randomly picked from the population and their flag is set to seeking mode, and for others the flag is set to tracing mode.
5. If the flag of ith cat is seeking mode, apply the cat to the seeking mode process, otherwise apply it to the tracing mode process. The steps of the corresponding modes are followed.
6. Evaluate the fitness of each cat and store the position of the cat with best fitness as P_{lm} where $m = 1, 2, \ldots, D$.
7. Compare the fitness of P_g and P_l, and update P_g.
8. Check the termination condition, if satisfied, terminate the program. Otherwise repeat steps 4–7.

The flowchart of the algorithm is shown in Fig. 2.

5. CSO based IIR weights update algorithm

1. Generate a random input signal $x(n)$ where $n = 1, 2, \ldots, N$.
2. The cat positions, representing the weights of the adaptive IIR filter, are initialized with random values in the range [0 1]. Hence a matrix P of order $L \times D$ is formed where L represents

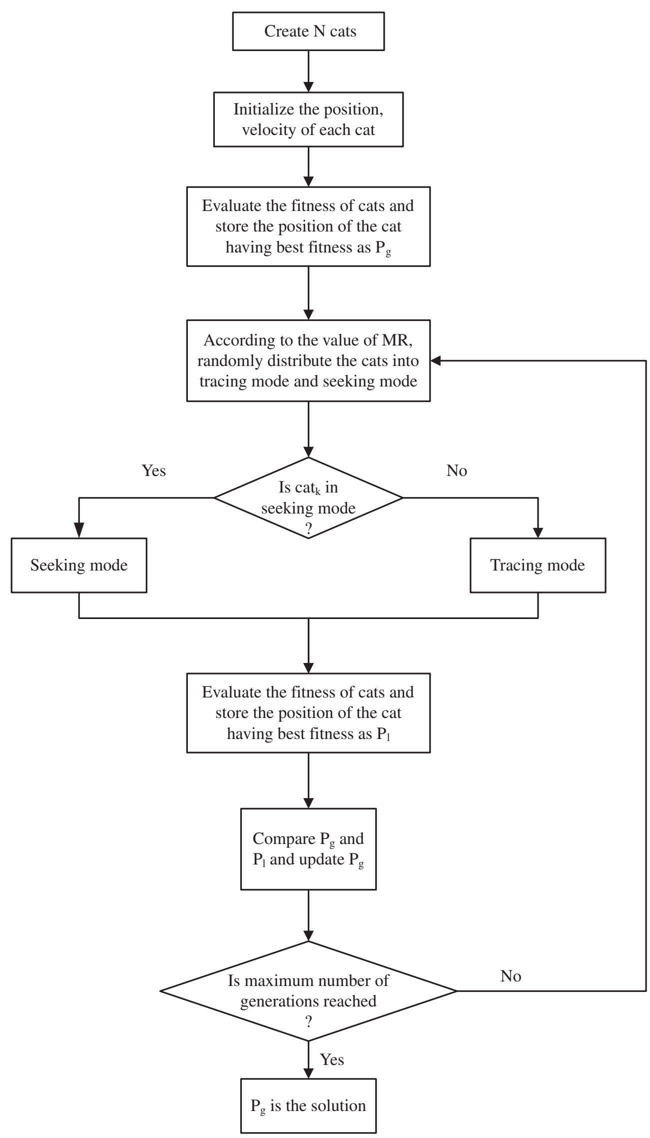

Fig. 2. Flowchart for cat swarm optimization.

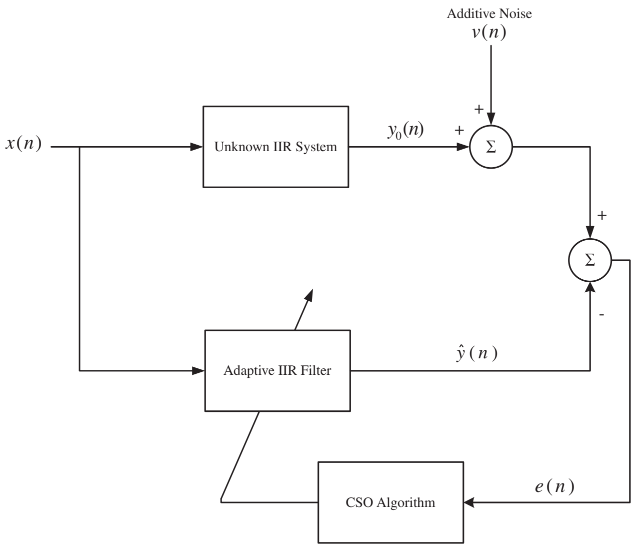

Fig. 3. Block diagram of CSO based IIR system identification.

the number of cats or population size. The position of each cat is composed of D dimensions representing the D weights of the adaptive IIR filter as shown in Fig. 3.

3. Similarly a matrix V of order $L \times D$ is formed where V_{ij} represents the velocity of ith cat in jth dimension.
4. For $t = 1$ to $ITER$ where $ITER$ represents the total number of iterations.
   (a) Generate unique integers T_q randomly in the range [1 L] where $q = 1, 2, \ldots, \frac{L}{1+MR}$. The elements of T represent the row indices of P or cats which will undergo the tracing process explained in Section 4. All other rows will undergo the seeking process.
   (b) For $k = 1$ to L
      i. Use the elements of kth row of P as the weights of the adaptive IIR filter.
      ii. Pass all N input samples of x through the system as well as model to obtain the error signal $e(n)$ where $n = 1,2,\ldots,N$.
      iii. Evaluate the fitness of the kth cat, $J_k = E[e^2(n)] = \frac{1}{N}\sum_{n=1}^{N} e^2(n)$. This also represents the MSE corresponding to kth solution.

   (c) The minimum value of MSE is stored as J_{min}.
   (d) Store the position of the cat corresponding to J_{min} as P_l.
   (e) If $t = 1$, then $P_g = P_l$ and $fit_t = J_{min}$ where P_g represents the best solution found so far and fit_t represents the minimum MSE found up to tth solution.
   (f) Else,
      i. If $J_{min} < fit_{t-1}$, then $fit_t = J_{min}$ and $P_g = P_l$.
      ii. Else, $fit_t = fit_{t-1}$ and $P_g = P_g$.

5. P_g represents the final weights of the adaptive IIR filter.

6. Simulation results

Simulation study is carried out in MATLAB to demonstrate the potentiality of CSO algorithm for identification of IIR plants. The input is a white signal with zero mean, unit variance and uniform distribution. The additive noise is a Gaussian white signal with variance 10^{-3}. To compare the performance of the new method, the results of GA and PSO methods are also obtained through simulation. The initial population chosen for all the three algorithms is 50. The simulation parameters used for GA are as follows: single point crossover with probability 0.8, probability of mutation 0.1 and number of bits per dimension are 10. The simulation parameters for PSO are: inertia weight is linearly decreased from 0.9 to 0.4, both the acceleration constants are taken as 2 and the random numbers are chosen in the range [0 1]. The parameters for CSO are: SMP = 5, SRD = 20%, CDC = 80%, MR = 0.9, $C = 2$, inertia weight is linearly decreased from 0.9 to 0.4 and r lies in the range [0 1]. Different tests are carried out for identification of four benchmark IIR systems. Two performance measures, i.e. residual mean square error (RMSE) and mean square deviation (MSD) are used to compare the performance of CSO, PSO and GA based approaches. The RMSE is defined as the steady state MSE value and MSD is defined as

$$MSD = \frac{1}{Q}\sum_{i=0}^{Q-1}(\Phi(i) - \hat{\Phi}(i))^2 \tag{10}$$

where Φ is the desired parameter vector, $\widehat{\Phi}$ is the estimated parameter vector and Q represents the total number of parameters to be estimated.

An unknown plant can be modeled in two ways: (i) using a filter of order equal to the order of the plant and (ii) using a reduced order filter. The performance of an algorithm is mostly determined by its ability to model a plant using a reduced order model. For each standard test function, a reduced order model is used to assess the performance of GA, PSO and CSO. The results obtained in terms of convergence characteristics and the RMSE are provided in this Section both for actual and reduced order of the IIR plants. The estimated and actual parameters, the MSD and the computational time are also provided for actual order of the IIR plants.

6.1. Example 1

The transfer function of the plant is given by, Shynk (1989a)

$$H_S(z) = \left[\frac{0.05 - 0.4z^{-1}}{1 - 1.1314z^{-1} + 0.25z^{-2}}\right] \tag{11}$$

6.1.1. Case-1

This 2nd order plant can be modeled using a 2nd order IIR filter. Hence the transfer function of the model is given by

$$H_S(z) = \left[\frac{a_0 + a_1z^{-1}}{1 - b_1z^{-1} - b_2z^{-2}}\right] \tag{12}$$

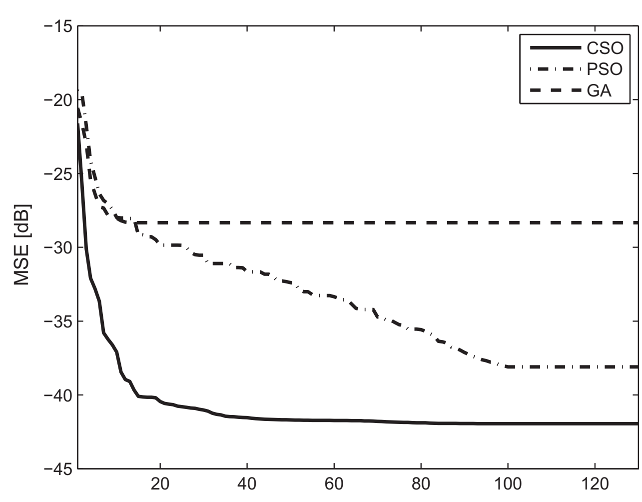

Fig. 4. Convergence characteristic for example-1 modeled using a 2nd order IIR filter.

Table 1
Parameter estimation for example-1 modeled using a 2nd order IIR filter.

Parameter	Actual value	Estimated value		
		GA	PSO	CSO
a_0	0.05	0.0877	0.0536	0.0493
a_1	−0.4	−0.4112	−0.4184	−0.4021
b_1	1.1314	1.182	1.0876	1.1248
b_2	−0.25	−0.305	−0.2077	−0.2433

Table 2
Results of RMSE metric for example-1 modeled using a 2nd order IIR filter.

RMSE	CSO	PSO	GA
Best	6.36395×10^{-5}	0.000101156	0.000264281
Worst	6.46289×10^{-5}	0.000274052	0.004922818
Average	$\mathbf{6.38494 \times 10^{-5}}$	0.000154906	0.001467056
Median	6.38062×10^{-5}	0.000145191	0.000824214
Std. dev.	2.89055×10^{-7}	5.18×10^{-5}	0.001548991

Table 3
Results of MSD metric for example-1 modeled using a 2nd order IIR filter.

MSD	CSO	PSO	GA
Best	1.1998×10^{-6}	3.57×10^{-5}	0.000499034
Worst	2.81183×10^{-5}	0.005971312	0.028144788
Average	$\mathbf{1.30846 \times 10^{-5}}$	0.001869092	0.007996323
Median	1.37276×10^{-5}	0.001483938	0.005148268
Std. dev.	9.94175×10^{-6}	0.001798539	0.008993771

Table 4
Computational time (in seconds) required by each algorithm for example-1 modeled using a 2nd order IIR filter.

Computational time	CSO	PSO	GA
Best	29.3125	9.546875	65.703125
Worst	30.203125	9.875	67.015625
Average	29.8609375	**9.734375**	66.5359375
Median	30.1640625	9.78125	66.890625
Std. dev.	0.420583801	0.138975667	0.560715243

In this case the total number of fitness evaluations is set to 6500. The convergence characteristic for the three models, using GA, PSO and CSO, shown in Fig. 4 provides a qualitative measure of the performance of the three algorithms. It can be seen that GA falls into local optima very early and hence its convergence curve remains flat after 15 iterations. CSO converges to its minimum MSE after 50 iterations but PSO takes 100 iterations to reach its minimum MSE level. The convergence speed of CSO is much higher than that of GA and PSO.

The estimated parameters listed in Table 1 shows that CSO based learning rule is able to estimate the coefficients better than GA and PSO. Tables 2 and 3 provide a quantitative assessment of the performance of CSO in terms of RMSE and MSD. It shows that CSO provides the best average results with respect to RMSE and MSD.

Table 4 shows that the CSO requires a higher computation time in comparison to PSO. In PSO the complete swarm of particles start their flight with a global search and finish with a local search in the last iteration. All the particles undergo a similar process of velocity and position updation in each iteration. But in CSO, in each iteration, a part of the population perform global search (seeking mode) whereas the remaining perform local search (tracing mode). In the last iteration, most of the cats reach the final solution whereas a part of the population continues the global search expecting a better solution. Since local and global searches are carried out independently in each iteration, the CSO requires a higher computation time in comparison to the PSO.

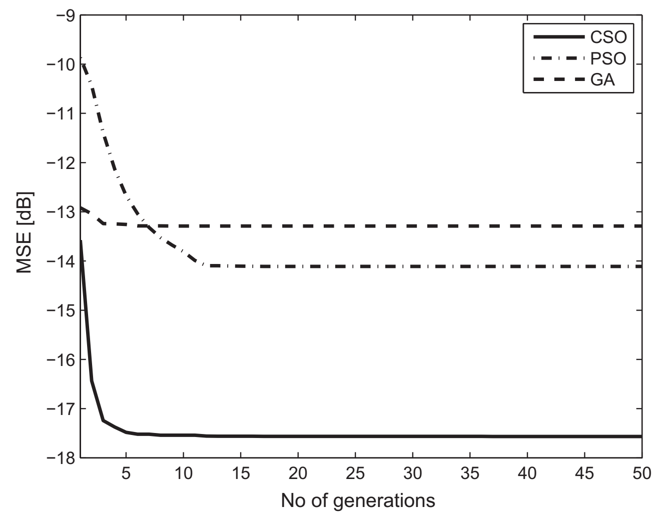

Fig. 5. Convergence characteristic for example-1 modeled using a 1st order IIR filter.

Table 5
Results of RMSE metric for example-1 modeled using a 1^{st} order IIR filter.

RMSE	CSO	PSO	GA
Best	0.0175154	0.0175154	0.027122122
Worst	0.0175154	0.055840586	0.057830381
Average	**0.0175154**	0.03880717	0.046893473
Median	0.0175154	0.055840586	0.057284912
Std. dev.	4.91×10^{-18}	0.020199147	0.013235689

6.1.2. Case-2

This 2nd order plant can also be modeled using a reduced order IIR filter. A 1st order IIR filter is used for modeling the 2nd order plant shown in (11). The cost function of this plant has a global minimum at $\{a_0, b_1\} = \{-0.311, -0.906\}$ and a local minimum at $\{a_0, b_1\} = \{0.114, 0.519\}$. The transfer function of the model is given by

$$H_S(z) = \left[\frac{a_0}{1 - b_1 z^{-1}}\right] \tag{13}$$

In this case the total number of fitness function evaluations carried out is 2500. Fig. 5 shows the convergence characteristic for the three models. It shows that CSO performs best in identifying a system using a reduced order model. The higher convergence speed of CSO and a smaller RMSE makes CSO one of the best choices for reduced order system modeling. Table 5 shows that the CSO provides the best average result in terms of the MSD.

6.2. Example 2

The transfer function of the plant is given by

$$H_S(z) = \left[\frac{-0.2 - 0.4z^{-1} + 0.5z^{-2}}{1 - 0.6z^{-1} + 0.25z^{-2} - 0.2z^{-3}}\right] \tag{14}$$

6.2.1. Case-1

This 3rd order plant is modeled using a 3rd order IIR filter. Hence the transfer function of the model is given by

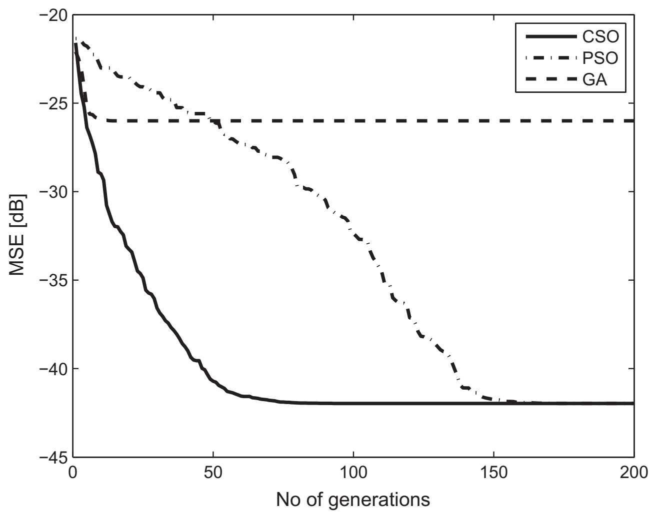

Fig. 6. Convergence characteristic for example-2 modeled using a 3rd order IIR filter.

Table 6
Parameter estimation for example-2 modeled using a 3rd order IIR filter.

Parameter	Actual value	Estimated value		
		GA	PSO	CSO
a_0	−0.20	−0.2258	−0.2105	−0.2050
a_1	−0.40	−0.2717	−0.3778	−0.3927
a_2	0.50	0.4643	0.4670	0.5038
b_1	0.60	0.7742	0.6123	0.6077
b_2	−0.25	−0.4379	−0.3134	−0.2519
b_3	0.20	0.3206	0.2249	0.2031

Table 7
Results of RMSE metric for example-2 modeled using a 3rd order IIR filter.

RMSE	CSO	PSO	GA
Best	6.35201×10^{-5}	6.35202×10^{-5}	0.000732034
Worst	6.35201×10^{-5}	6.35206×10^{-5}	0.006152866
Average	$\mathbf{6.35201 \times 10^{-5}}$	6.35203×10^{-5}	0.002510992
Median	6.35201×10^{-5}	6.35202×10^{-5}	0.002192451
Std. dev.	1.68717×10^{-18}	1.47673×10^{-10}	0.001485081

Table 8
Results of MSD metric for example-2 modeled using a 3rd order IIR filter.

MSD	CSO	PSO	GA
Best	1.22363×10^{-5}	1.21551×10^{-5}	0.004256485
Worst	1.22363×10^{-5}	1.23941×10^{-5}	0.026032614
Average	$\mathbf{1.22363 \times 10^{-5}}$	1.22585×10^{-5}	0.012835566
Median	1.22363×10^{-5}	1.22488×10^{-5}	0.010356046
Std. dev.	8.24073×10^{-12}	7.30148×10^{-8}	0.008089536

Table 9
Computational time (in seconds) required by each algorithm for example-2 modeled using a 3rd order IIR filter.

Compuational time	CSO	PSO	GA
Best	66	21.3125	171.25
Worst	67.71875	21.796875	172.328125
Average	66.4609375	**21.678125**	171.7609375
Median	66.390625	21.71875	171.7734375
Std. dev.	0.462296431	0.13665301	0.299200504

$$H_S(z) = \left[\frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2} - b_3 z^{-3}}\right] \tag{15}$$

In this case the total number of fitness function evaluations is set to 10,000. The convergence characteristic shown in Fig. 6 shows that the minimum MSEs obtained using CSO and PSO are equal but the CSO converges much faster than the PSO. The CSO takes 80 iterations to reach its minimum MSE level whereas the PSO needs 170 iterations to reach the same MSE value. The estimated parameters in Table 6 indicate that the CSO performs better than the GA and PSO in estimating the parameters of the system. The performance measures shown in Tables 7 and 8 reveal that the CSO and PSO provide the same quality of solution. In this case the higher convergence speed of CSO makes it a better choice for 3rd order system modeling. Table 9 shows that the CSO requires higher computation time to its PSO counterpart.

6.2.2. Case-2

The 3rd order plant shown in (14) is modeled using a 2nd order IIR filter. Hence the transfer function of the model is

$$H_S(z) = \left[\frac{a_0 + a_1 z^{-1}}{1 - b_1 z^{-1} - b_2 z^{-2}}\right] \tag{16}$$

In this case the total number of fitness function evaluations is set to 10,000 Fig. 7 indicates that the CSO converges much faster than the GA but performs almost equally like PSO. The values of MSD for CSO and PSO shown in Table 10 reveal that the quality of solution obtained using the PSO and CSO are almost equal. Although the CSO provides a smaller standard deviation, but the difference between the standard deviation of the PSO and CSO is negligible.

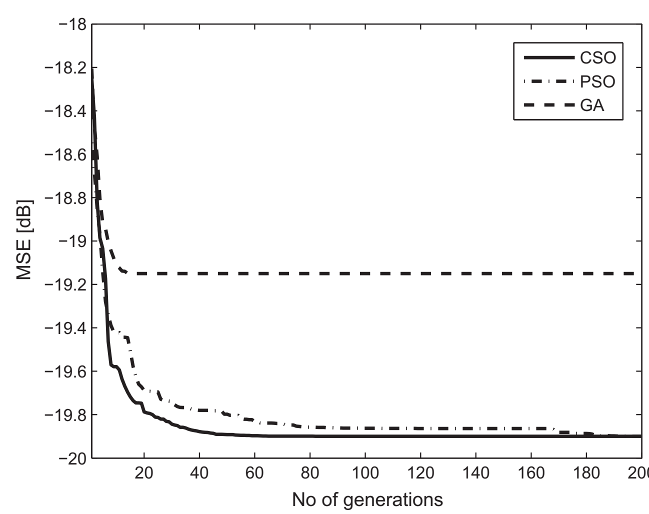

Fig. 7. Convergence characteristic for example-2 modeled using a 2nd order IIR filter.

Table 10
Results of RMSE metric for example-2 modeled using a 2nd order IIR filter.

RMSE	CSO	PSO	GA
Best	0.001393846	0.001393846	0.016505341
Worst	0.001393846	0.001393846	0.066687034
Average	**0.001393846**	0.001393846	0.032599686
Median	0.001393846	0.001393846	0.024584869
Std. dev.	1.0842×10^{-19}	2.96921×10^{-19}	0.016104742

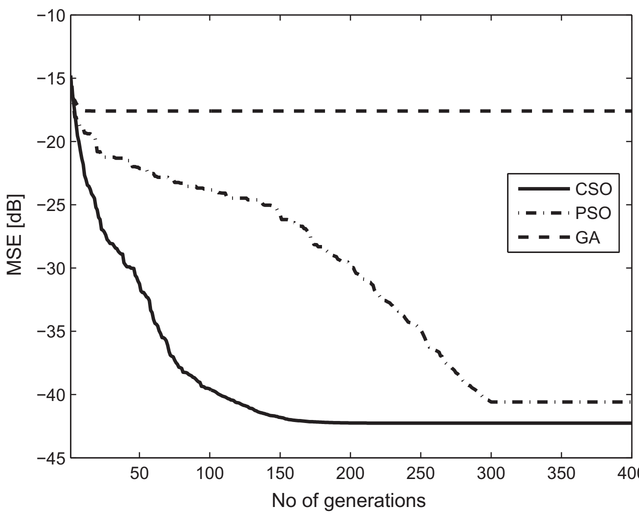

Fig. 8. Convergence characteristic for example-3 modeled using a 4th order IIR filter.

Table 11
Parameter estimation for example-3 modeled using a 4th order IIR filter.

Parameter	Actual value	Estimated value		
		GA	PSO	CSO
a_0	1.00	1.0670	1.1587	0.9951
a_1	−0.90	−0.7493	−0.6562	−0.8839
a_2	0.81	0.7214	0.3380	0.8206
a_3	−0.729	−0.4350	−0.9309	−0.7253
b_1	−0.04	−0.2308	−0.6264	−0.0506
b_2	−0.2775	−0.3064	−0.6618	−0.2930
b_3	0.2101	0.1065	0.5165	0.1962
b_4	−0.14	−0.0489	−0.0067	−0.1461

6.3. Example 3

The transfer function of the plant is given by

$$H_S(z) = \left[\frac{1 - 0.9z^{-1} + 0.81z^{-2} - 0.729z^{-3}}{1 + 0.04z^{-1} + 0.2775z^{-2} - 0.2101z^{-3} + 0.14z^{-4}}\right] \tag{17}$$

6.3.1. Case-1

This 4th order plant can be modeled using a 4th order IIR filter. Hence the transfer function of the model is given by

$$H_S(z) = \left[\frac{a_0 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3}}{1 - b_1z^{-1} - b_2z^{-2} - b_3z^{-3} - b_4z^{-4}}\right] \tag{18}$$

In this case the total number of fitness function evaluations is set to 20,000. As seen in Fig. 8, the GA gets trapped into a local solution very early because the chromosomes become stagnant, converging to a suboptimal solution. However, the CSO and PSO do not stagnate and reach their corresponding minimum noise floor levels. In this case the CSO outperforms the PSO and GA with higher convergence speed and smaller RMSE value. The CSO takes 200 generations to reach its minimum MSE level whereas the PSO takes 300 generations to reach its corresponding value. The minimum MSE obtained using the CSO is smaller than that of the PSO. Table 11 shows that the CSO is able to estimate the parameters more accurately than those by GA and PSO based methods.

Tables 12 and 13 show that the CSO provides the best average results in terms of RMSE and MSD. Table 14 indicates that the CSO needs higher computation time than the GA and PSO. Hence

Table 13
Results of MSD metric for example-3 modeled using a 4th order IIR filter.

MSD	CSO	PSO	GA
Best	1.39082×10^{-5}	0.000112718	0.025196598
Worst	1.4134×10^{-5}	0.001337828	0.160397073
Average	$\mathbf{1.40367 \times 10^{-5}}$	0.000476185	0.084828165
Median	1.40524×10^{-5}	0.00024282	0.076786791
Std. dev.	9.67033×10^{-8}	0.000473742	0.038506348

Table 14
Computational time (in seconds) required by each algorithm for example-3 modeled using a 4th order IIR filter.

Compuational time	CSO	PSO	GA
Best	99.421875	32.265625	297.34375
Worst	99.828125	32.640625	298.140625
Average	99.6328125	**32.525**	297.7171875
Median	99.640625	32.5703125	297.6796875
Std. dev.	0.124183531	0.116275386	0.224194401

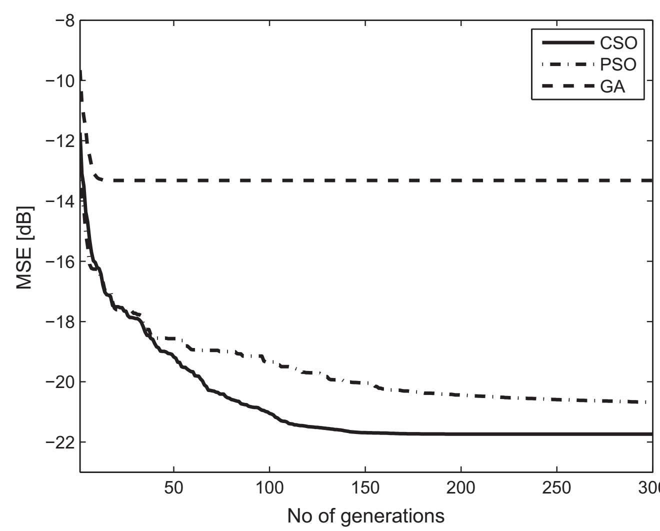

Fig. 9. Convergence characteristic for example-3 modeled using a 3rd order IIR filter.

Table 12
Results of RMSE metric for example-3 modeled using a 4th order IIR filter.

RMSE	CSO	PSO	GA
Best	5.94209×10^{-5}	6.11456×10^{-5}	0.007158637
Worst	5.9444×10^{-5}	0.000142515	0.044913437
Average	$\mathbf{5.94284 \times 10^{-5}}$	8.73251×10^{-5}	0.017414962
Median	5.94247×10^{-5}	7.72487×10^{-5}	0.0124739
Std. dev.	8.30206×10^{-9}	2.62681×10^{-5}	0.012255228

Table 15
Results of RMSE metric for example-3 modeled using a 3rd order IIR filter.

RMSE	CSO	PSO	GA
Best	0.006705056	0.006705062	0.019374825
Worst	0.006705056	0.015665928	0.092520249
Average	**0.006705056**	0.008548452	0.046595405
Median	0.006705056	0.006800867	0.04455507
Std. dev.	2.99901×10^{-11}	0.003747328	0.023286793

there is a trade off between the quality of solution and computation time.

6.3.2. Case-2

The 4th order plant shown in (15) can be modeled using a 3rd order IIR filter. Hence the transfer function of the model is given by

$$H_S(z) = \left[\frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - b_1 z^{-1} - b_2 z^{-2} - b_3 z^{-3}}\right] \tag{19}$$

In this case the total number of fitness function evaluations is set to 15,000. The convergence characteristic shown in Fig. 9 indicates that the CSO and PSO have equal convergence speed up to 40 generations and thereafter the CSO converges faster than the PSO. The minimum MSE level achieved by the CSO is lower than that of the PSO which signifies a better solution. Table 15 shows that the average value of RMSE obtained using the CSO is smaller than that obtained using the PSO and GA.

6.4. Example 4

The transfer function of the plant is given by (Krusienski & Jenkins, 2004)

$$H_S(z) = \left[\frac{0.1084 + 0.5419z^{-1} + 1.0837z^{-2} + 1.0837z^{-3} + 0.5419z^{-4} + 0.1084z^{-5}}{1 + 0.9853z^{-1} + 0.9738z^{-2} + 0.3864z^{-3} + 0.1112z^{-4} + 0.0113z^{-5}}\right] \tag{20}$$

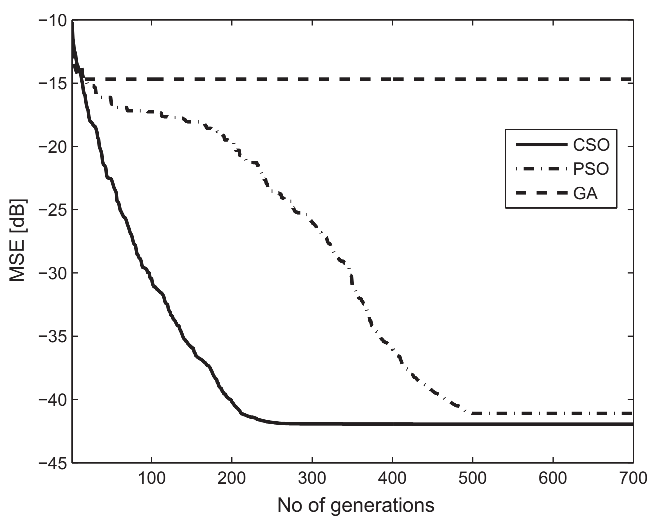

Fig. 10. Convergence characteristic for example-4 modeled using a 5th order IIR filter.

Table 16
Parameter estimation for example-4 modeled using a 5th order IIR filter.

Parameter	Actual value	Estimated value		
		GA	PSO	CSO
a_0	0.1084	0.5083	0.2484	0.1038
a_1	0.5419	0.7449	0.3789	0.5403
a_2	1.0837	1.0303	1.6960	1.0813
a_3	1.0837	1.0714	1.4109	1.0803
a_4	0.5419	0.7067	0.8467	0.5447
a_5	0.1084	0.3578	0.2684	0.1145
b_1	−0.9853	−0.6080	−1.0628	−0.9768
b_2	−0.9738	−0.9316	−0.7275	−0.9632
b_3	−0.3864	−0.3451	−0.4842	−0.3827
b_4	−0.1112	−0.3382	−0.3291	−0.1137
b_5	−0.0113	−0.1848	−0.2238	−0.0167

Table 17
Results of RMSE metric for example-4 modeled using a 5th order IIR filter.

RMSE	CSO	PSO	GA
Best	6.35514×10^{-5}	7.27396×10^{-5}	0.013335606
Worst	6.44926×10^{-5}	9.14957×10^{-5}	0.064170518
Average	$\mathbf{6.39373 \times 10^{-5}}$	7.7614×10^{-5}	0.033987521
Median	6.39725×10^{-5}	7.61366×10^{-5}	0.030856278
Std. dev.	2.90265×10^{-7}	5.73649×10^{-6}	0.014807147

Table 18
Results of MSD metric for example-4 modeled using a 5th order IIR filter.

MSD	CSO	PSO	GA
Best	0.000101089	0.000770198	0.029050486
Worst	0.002362064	0.007822291	0.159676077
Average	**0.000453387**	0.004608008	0.068120781
Median	0.00023241	0.004687136	0.06642881
Std. dev.	0.000680324	0.002776328	0.036147013

Table 19
Computational time (in seconds) required by each algorithm for example-4 modeled using a 5th order IIR filter

Compuational time	CSO	PSO	GA
Best	166.921875	54.453125	598.046875
Worst	168.28125	55	599.984375
Average	167.7640625	**54.840625**	598.8296875
Median	167.8671875	54.8828125	598.8125
Std. dev.	0.38727382	0.160835492	0.587062577

6.4.1. Case-1

This 5th order plant can be modeled using a 5th order IIR filter. Hence the transfer function of the model is given by

$$H_S(z) = \left[\frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4} + a_5 z^{-5}}{1 - b_1 z^{-1} - b_2 z^{-2} - b_3 z^{-3} - b_4 z^{-4} - b_5 z^{-5}}\right] \tag{21}$$

In this case the total number of fitness function evaluations is set to 35,000. Fig. 10 shows that the CSO has very high convergence speed in comparison to the PSO and GA. The CSO takes 250 iterations to converge to its minimum MSE value whereas the PSO takes

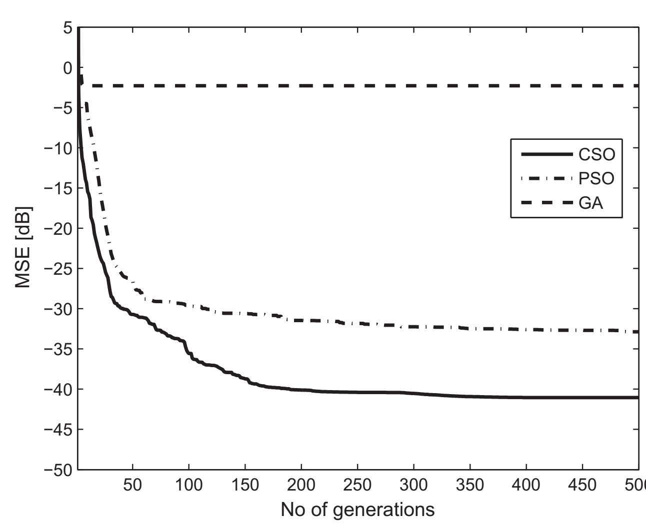

Fig. 11. Convergence characteristic for example-4 modeled using a 4th order IIR filter.

Table 20
Performance measures for example-4 modeled using a 4th order IIR filter.

RMSE	CSO	PSO	GA
Best	6.9475×10^{-5}	6.93727×10^{-5}	0.084596041
Worst	0.000145159	0.003837955	290488.5142
Average	$\mathbf{7.85538 \times 10^{-5}}$	0.000516557	32386.62672
Median	7.03466×10^{-5}	9.71482×10^{-5}	0.583570895
Std. dev.	2.49878×10^{-5}	0.00124584	96788.75683

500 iterations to reach its corresponding minimum MSE value. The CSO and PSO perform well with respect to minimum MSE level. Although the minimum MSE value obtained using the CSO is smaller than that of the PSO but the difference is negligible. Table 16 shows that the estimated parameters using the CSO matches very well with the actual parameters. Tables 17 and 18 show that the average value of RMSE and MSD obtained using the PSO and CSO are very close to each other. The computational time required for the CSO, as shown in Table 19, is higher than that of the PSO.

6.4.2. Case-2

The 5th order plant shown in (15) can be modeled using a 4th order IIR filter. Hence the transfer function of the model is given by

$$H_S(z) = \left[\frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}{1 - b_1 z^{-1} - b_2 z^{-2} - b_3 z^{-3} - b_4 z^{-4}}\right] \tag{22}$$

In this case the total number of fitness function evaluations is set to 25,000. Fig. 11 indicates that the convergence speed of CSO and PSO are almost equal but the minimum MSE level obtained using CSO is much below than that of PSO. Table 20 shows that the average value of RMSE in CSO is much smaller than that of PSO and GA which signifies the good quality of solution obtained using CSO.

7. Sensitivity analysis

This section demonstrates three experiments carried out for studying the effect of different parameters on the performance of CSO algorithm.

7.1. Experiment 1: effect of variation in population size

The number of cats as well as the number of iterations are varied for maintaining the total number of fitness function evaluations. Different simulations using 5, 25, 50 and 100 cats are carried out. The results are discussed below in detail.

7.1.1. Example 1

Table 21 shows that a population size of 50 cats provided the best average results with respect to RMSE and MSD.

7.1.2. Example 2

Table 22 shows that the average values of RMSE and MSD obtained in all cases are almost equal but the use of 5 cats provided a smaller standard deviation with respect to both the metrics.

7.1.3. Example 3

Table 23 shows a mixed set of results. The best average result with respect to RMSE is obtained using a population of 100 cats. In comparison to this, the use of 50 cats provided a competitive value of RMSE with a smaller standard deviation. A population size of 50 cats provided the best average result with respect to MSD.

7.1.4. Example 4

A population size of 50 cats provided best average results with respect to both the metrics as shown in Table 24.

Table 21
Results of experiment 1 for example-1.

Cats	5	25	50	100
Residual mean square error				
Best	6.36409×10^{-5}	6.36384×10^{-5}	6.36395×10^{-5}	6.3656×10^{-5}
Worst	9.41598×10^{-5}	9.19842×10^{-5}	6.46289×10^{-5}	0.000107301
Average	7.10532×10^{-5}	6.51574×10^{-5}	$\mathbf{6.38494 \times 10^{-5}}$	7.02987×10^{-5}
Median	6.55783×10^{-5}	6.38655×10^{-5}	6.38062×10^{-5}	6.49575×10^{-5}
Std. dev.	9.16574×10^{-6}	5.14602×10^{-6}	2.89055×10^{-7}	1.03726×10^{-5}
Mean square deviation				
Best	2.2038×10^{-6}	1.61063×10^{-6}	1.1998×10^{-6}	3.52587×10^{-6}
Worst	0.000987183	0.000427491	2.81183×10^{-5}	0.000877768
Average	2.10133×10^{-4}	4.25725×10^{-5}	$\mathbf{1.30846 \times 10^{-5}}$	1.5666×10^{-4}
Median	7.7913×10^{-5}	2.16451×10^{-5}	1.37276E−05	4.42795×10^{-5}
Std. dev.	0.000268908	7.86491×10^{-5}	9.94175×10^{-6}	0.000218603

Table 22
Results of experiment 1 for example-2.

Cats	5	25	50	100
Residual mean square error				
Best	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}
Worst	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}	6.35202×10^{-5}
Average	$\mathbf{6.35201 \times 10^{-5}}$	6.35201×10^{-5}	6.35201×10^{-5}	6.35202×10^{-5}
Median	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}	6.35202×10^{-5}
Std. dev.	2.39044×10^{-20}	3.54041×10^{-19}	6.30775×10^{-18}	9.75838×10^{-12}
Mean square deviation				
Best	1.22363×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}	1.22114×10^{-5}
Worst	1.22363×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}	1.23027×10^{-5}
Average	$\mathbf{1.22363 \times 10^{-5}}$	1.22363×10^{-5}	1.22363×10^{-5}	1.22353×10^{-5}
Median	1.22363×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}	1.22295×10^{-5}
Std. dev.	8.4146×10^{-13}	5.52693×10^{-12}	1.55061×10^{-11}	2.58051×10^{-8}

Table 23
Results of experiment 1 for example-3.

Cats	5	25	50	100
Residual mean square error				
Best	5.99052×10^{-5}	6.36054×10^{-5}	5.94209×10^{-5}	5.36052×10^{-5}
Worst	6.3735×10^{-5}	6.50873×10^{-5}	5.9444×10^{-5}	6.31824×10^{-5}
Average	6.06195×10^{-5}	6.38207×10^{-5}	5.94284×10^{-5}	$\mathbf{5.46555 \times 10^{-5}}$
Median	6.36087×10^{-5}	6.36315×10^{-5}	5.94247×10^{-5}	6.3653×10^{-5}
Std. dev.	3.01096×10^{-8}	4.16343×10^{-7}	8.30206×10^{-9}	2.78358×10^{-6}
Mean square deviation				
Best	8.0026×10^{-6}	6.46947×10^{-6}	1.39082×10^{-5}	5.43683×10^{-6}
Worst	2.09556×10^{-5}	6.74267×10^{-5}	1.4134×10^{-5}	0.000280473
Average	1.45471×10^{-5}	1.85609×10^{-5}	$\mathbf{1.40367 \times 10^{-5}}$	4.3053×10^{-5}
Median	1.4225×10^{-5}	1.35916×10^{-5}	1.40524×10^{-5}	1.44927×10^{-5}
Std. dev.	2.80984×10^{-6}	1.68531×10^{-5}	9.67033×10^{-8}	7.77498×10^{-5}

Table 24
Results of experiment 1 for example-4.

Cats	5	25	50	100
Residual mean square error				
Best	6.96613×10^{-5}	6.96688×10^{-5}	6.35514×10^{-5}	6.98101×10^{-5}
Worst	6.97925×10^{-5}	7.0435×10^{-5}	6.44926×10^{-5}	7.26599×10^{-5}
Average	6.97074×10^{-5}	6.98354×10^{-5}	$\mathbf{6.39373 \times 10^{-5}}$	7.04079×10^{-5}
Median	6.97032×10^{-5}	6.97366×10^{-5}	6.39725×10^{-5}	7.02092×10^{-5}
Std. dev.	3.73038×10^{-8}	2.32799×10^{-7}	2.90265×10^{-7}	8.24356×10^{-7}
Mean square deviation				
Best	0.000253106	2.88×10^{-5}	0.000101089	2.68×10^{-5}
Worst	0.002787177	0.00277419	0.002362064	0.006054059
Average	0.0010258	0.001068403	**0.000453387**	0.001383773
Median	0.000735059	0.000718567	0.00023241	0.000153473
Std. dev.	0.000801124	0.001038054	0.000680324	0.002058661

Table 25
Results of experiment 2 for example-1.

MR	0.1	0.3	0.5	0.9
Residual mean square error				
Best	6.3651×10^{-5}	6.36431×10^{-5}	6.36803×10^{-5}	6.36395×10^{-5}
Worst	8.0201×10^{-5}	7.0273×10^{-5}	9.37721×10^{-5}	6.46289×10^{-5}
Average	6.9774×10^{-5}	6.60214×10^{-5}	6.85631×10^{-5}	$\mathbf{6.38494 \times 10^{-5}}$
Median	6.80219×10^{-5}	6.57149×10^{-5}	6.61504×10^{-5}	6.38062×10^{-5}
Std. dev.	6.12892×10^{-6}	2.21061×10^{-6}	9.08643×10^{-6}	2.89055×10^{-7}
Mean square deviation				
Best	9.40111×10^{-6}	3.02759×10^{-6}	2.72709×10^{-6}	1.1998×10^{-6}
Worst	0.000555309	0.000193278	0.000585652	2.81183×10^{-5}
Average	0.000156773	4.39593×10^{-5}	0.000119789	$\mathbf{1.30846 \times 10^{-5}}$
Median	5.34636×10^{-5}	1.72529×10^{-5}	6.51756×10^{-5}	1.37276×10^{-5}
Std. dev.	0.00021121	5.92296×10^{-5}	0.000172713	9.94175×10^{-6}

Table 26
Results of experiment 2 for example-2.

MR	0.1	0.3	0.5	0.9
Residual mean square error				
Best	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}
Worst	6.35202×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}
Average	6.35202×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}	$\mathbf{6.35201 \times 10^{-5}}$
Median	6.35202×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}
Std. dev.	8.70381×10^{-12}	6.77635×10^{-15}	1.24334×10^{-18}	9.02847×10^{-19}
Mean square deviation				
Best	1.21855×10^{-5}	1.22353×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}
Worst	1.22983×10^{-5}	1.22378×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}
Average	1.22441×10^{-5}	1.22363×10^{-5}	$\mathbf{1.22363 \times 10^{-5}}$	1.22363×10^{-5}
Median	1.22453×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}
Std. dev.	3.22245×10^{-8}	5.96481×10^{-10}	4.25312×10^{-12}	6.55057×10^{-12}

Table 27
Results of experiment 2 for example-3.

MR	0.1	0.3	0.5	0.9
Residual mean square error				
Best	6.53176×10^{-5}	6.36199×10^{-5}	6.16084×10^{-5}	5.94209×10^{-5}
Worst	7.77491×10^{-5}	6.41173×10^{-5}	6.16109×10^{-5}	5.9444×10^{-5}
Average	7.07399×10^{-5}	6.37839×10^{-5}	6.16096×10^{-5}	$\mathbf{5.94284 \times 10^{-5}}$
Median	6.99465×10^{-5}	6.36993×10^{-5}	6.16095×10^{-5}	5.94247×10^{-5}
Std. dev.	5.46176×10^{-6}	2.30961×10^{-7}	1.1516×10^{-9}	8.30206×10^{-9}
Mean square deviation				
Best	2.74923×10^{-5}	7.18913×10^{-6}	1.27049×10^{-5}	1.39082×10^{-5}
Worst	0.000315084	3.01003×10^{-5}	1.59965×10^{-5}	1.4134×10^{-5}
Average	0.000173796	1.71228×10^{-5}	$\mathbf{1.40199 \times 10^{-5}}$	1.40367×10^{-5}
Median	0.000176305	1.56009×10^{-5}	1.41533×10^{-5}	1.40524×10^{-5}
Std. dev.	0.000123787	9.84699×10^{-6}	1.73763×10^{-6}	9.67033×10^{-8}

7.1.5. Observation

In most of the cases, the use of 50 cats performed best with respect to all the metrics. Hence a population size of 50 cats is a reasonable choice for any optimization problem.

7.2. Experiment 2: effect of variation in mixture ratio

In this experiment the value of MR is varied while all other parameters are kept constant. Different simulation exercises with MR as 0.1, 0.3, 0.5 and 0.9 are carried out.

7.2.1. Example 1

It can be seen in Table 25 that the best average results with respect to both the metrics are obtained using a value of 0.9 for MR.

7.2.2. Example 2

In this case, Table 26 shows that the variation of MR has very negligible effect on the performance of CSO. Different runs using different values of MR provided almost equal results with respect to both the metrics. With respect to RMSE, a value of 0.9 provided the smallest standard deviation whereas a value of 0.5 provided the smallest standard deviation with respect to MSD.

7.2.3. Example 3

In Table 27, we can see that the best average results with respect to RMSE is obtained using a value of 0.9 for MR. A value of 0.5 provided best average result with respect to MSD. With respect to MSD, a value of 0.9 for MR provided a competitive result with a smaller standard deviation.

Table 28
Results of experiment 2 for example-4.

MR	0.1	0.3	0.5	0.9
Residual mean square error				
Best	6.36733×10^{-5}	6.3651×10^{-5}	6.35301×10^{-5}	6.35514×10^{-5}
Worst	7.06647×10^{-5}	6.61591×10^{-5}	6.52024×10^{-5}	6.44926×10^{-5}
Average	6.5573×10^{-5}	6.44164×10^{-5}	6.40004×10^{-5}	$\mathbf{6.39373 \times 10^{-5}}$
Median	6.49925×10^{-5}	6.42009×10^{-5}	6.3629×10^{-5}	6.39725×10^{-5}
Std. dev.	2.01494×10^{-6}	7.76056×10^{-7}	6.30246×10^{-7}	2.90265×10^{-7}
Mean square deviation				
Best	4.08042×10^{-5}	0.000123839	4.30447×10^{-5}	0.000101089
Worst	0.003455811	0.004577738	0.003715041	0.002362064
Average	0.001448685	0.001455086	0.001318934	**0.000453387**
Median	0.001715977	0.001216132	0.001000565	0.00023241
Std. dev.	0.001256399	0.001336209	0.001299345	0.000680324

Table 29
Results of experiment 3 for example-1.

SMP	5	10	20	30
Residual mean square error				
Best	6.36395×10^{-5}	6.36385×10^{-5}	6.36425×10^{-5}	6.36393×10^{-5}
Worst	6.46289×10^{-5}	6.46893×10^{-5}	6.48547×10^{-5}	6.50807×10^{-5}
Average	$\mathbf{6.38494 \times 10^{-5}}$	6.38778×10^{-5}	6.40105×10^{-5}	6.38737×10^{-5}
Median	6.38062×10^{-5}	6.37235×10^{-5}	6.39176×10^{-5}	6.37221×10^{-5}
Std. dev.	2.89055×10^{-7}	3.4596×10^{-7}	3.79297×10^{-7}	4.3788×10^{-7}
Mean square deviation				
Best	1.1998×10^{-6}	4.205×10^{-6}	2.00798×10^{-6}	4.84527×10^{-6}
Worst	2.81183×10^{-5}	0.001265042	7.87663×10^{-5}	9.10502×10^{-5}
Average	$\mathbf{1.30846 \times 10^{-5}}$	0.000169101	2.2042×10^{-5}	2.38445×10^{-5}
Median	1.37276×10^{-5}	2.23996×10^{-5}	8.76381×10^{-6}	1.16025×10^{-5}
Std. dev.	9.94175×10^{-6}	0.000391597	2.72042×10^{-5}	2.65372×10^{-5}

Table 30
Results of experiment 3 for example-2.

SMP	5	10	20	30
Residual mean square error				
Best	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}
Worst	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}
Average	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}	$\mathbf{6.35201 \times 10^{-5}}$
Median	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}	6.35201×10^{-5}
Std. dev.	3.38572×10^{-17}	8.84094×10^{-20}	1.91662×10^{-20}	1.62881×10^{-20}
Mean square deviation				
Best	1.22363×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}
Worst	1.22365×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}
Average	1.22363×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}	$\mathbf{1.22363 \times 10^{-5}}$
Median	1.22363×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}	1.22363×10^{-5}
Std. dev.	4.52321×10^{-11}	3.88677×10^{-12}	2.21931×10^{-12}	9.7322×10^{-13}

Table 31
Results of experiment 3 for example-3.

SMP	5	10	20	30
Residual mean square error				
Best	5.94209×10^{-5}	5.93155×10^{-5}	6.13153×10^{-5}	6.13152×10^{-5}
Worst	5.9444×10^{-5}	6.20655×10^{-5}	6.20511×10^{-5}	6.14863×10^{-5}
Average	$\mathbf{5.94284 \times 10^{-5}}$	5.99241×10^{-5}	6.14062×10^{-5}	6.13591×10^{-5}
Median	5.94247×10^{-5}	5.98348×10^{-5}	6.13242×10^{-5}	6.13445×10^{-5}
Std. dev.	8.30206×10^{-9}	2.30027×10^{-7}	2.28177×10^{-7}	5.15137×10^{-8}
Mean square deviation				
Best	1.39082×10^{-5}	1.35881×10^{-5}	8.47189×10^{-5}	1.37895×10^{-5}
Worst	1.4134×10^{-5}	6.13267×10^{-5}	2.98029×10^{-5}	3.39069×10^{-5}
Average	$\mathbf{1.40367 \times 10^{-5}}$	2.63553×10^{-5}	2.00045×10^{-5}	2.24482×10^{-5}
Median	1.40524×10^{-5}	2.06886×10^{-5}	2.00624×10^{-5}	2.22473×10^{-5}
Std. dev.	9.67033×10^{-8}	1.36336×10^{-5}	5.6689×10^{-6}	5.92794×10^{-6}

7.2.4. Example 4

As we can see in Table 28, a value of 0.9 for MR performed best with respect to both the metrics.

7.2.5. Observation

It is seen that in most cases, a value of 0.9 for MR provided best average results for different performance metrics. Hence a value of 0.9 for MR is adopted for our simulation exercise.

7.3. Experiment 3: effect of variation in SMP

In this experiment the value of SMP is varied while all other parameters are kept unchanged. Different values of SMP, i.e. 5, 10, 20 and 30, are used to study its effect on the performance of CSO.

7.3.1. Example 1

We can see in Table 29 that a value of 5 for SMP provided the best average results with respect to both the metrics.

7.3.2. Example 2

Table 30 shows that the variation in SMP has negligible effect on the performance of CSO. The difference between the results obtained using different values of SMP is negligible. A value of 30 provided the smallest standard deviations with respect to MSD and RMSE, respectively.

7.3.3. Example 3

In Table 31, we can see that the best average results with respect to both the metrics are obtained using a value of 5 for SMP.

Table 32
Results of experiment 3 for example-4.

SMP	5	10	20	30
Residual mean square error				
Best	6.35514×10^{-5}	6.56914×10^{-5}	6.56548×10^{-5}	6.56567×10^{-5}
Worst	6.44926×10^{-5}	6.71497×10^{-5}	6.62515×10^{-5}	6.65219×10^{-5}
Average	$\mathbf{6.39373 \times 10^{-5}}$	6.59151×10^{-5}	6.57813×10^{-5}	6.58152×10^{-5}
Median	6.39725×10^{-5}	6.57561×10^{-5}	6.57228×10^{-5}	6.57278×10^{-5}
Std. dev.	2.90265×10^{-7}	4.39164×10^{-7}	1.74347×10^{-7}	2.58969×10^{-7}
Mean square deviation				
Best	0.000101089	2.96707×10^{-5}	7.20645×10^{-5}	0.000196228
Worst	0.002362064	0.002579901	0.002119599	0.003154167
Average	**0.000453387**	0.000648277	0.000494018	0.001690847
Median	0.00023241	0.000238218	0.000262781	0.002165361
Std. dev.	0.000680324	0.000801589	0.00061448	0.00129892

7.3.4. Example 4

In this case, as we can see in Table 32, a value of 5 performed best with respect to both the metrics.

7.3.5. Observation

It is seen that a value of 5 for SMP provided best average results for most cases. Hence a value of 5 for SMP will be a reasonable choice for any problem.

8. Conclusion

Although the IIR system identification is an active area of research, major difficulties like local and diverged solutions and instability still exist in practice. In this paper the CSO, a recently developed evolutionary optimization tool, is introduced for IIR system identification. The new algorithm is outlined and has been applied to identification of few bench mark IIR plants. The performance assessment of the CSO, in identifying an unknown system with a reduced order model, is also carried out. Comparison of simulation results with those obtained by the GA and PSO based approach clearly exhibits superior identification performance of the proposed method in terms of the convergence speed, the RMSE and the MSD. However, the computational time of the proposed algorithm is higher than that of the PSO but lesser than its GA counterpart. Thus in general the CSO is a potential candidate for identification of IIR plant compared to other evolutionary computing based approaches.

Acknowledgement

The work was supported by the Department of Science and Technology, Govt. of India under Grant No. SR/S3/EECE/065/2008.

References

Abe, M., & Kawamata, M. (1998a). Evolutionary digital filtering for iir adaptive digital filters based on the cloning and mating reproduction. *IEICE Transactions on Fundamentals*, 398–406.

Abe, M., & Kawamata, M. (1998b). A single dsp implementation of evolutionary digital filters. In *Proceedings of the IEEE international workshop on intelligent signal processing and communication systems* (pp. 253–257).

Abe, M., Kawamata, M., & Higuchi, T. (1996). Convergence behavior of evolutionary digital filters on a multiple-peak surface. In *Proceedings of the IEEE international symposium on circuits and systems* (Vol. 2, pp. 185–188).

Astriim, K. J., & Eykhoff, P. (1971). System identification – a survey. *Automatica, 7*, 123–162.

Chu, S.-C., & Tsai, P.-W. (2007). Computational intelligence based on the behavior of cats. *International Journal of Innovative Computing, Information and Control, 3*, 163–173.

David, R. A. (1981). A modified cascade structure for iir adaptive algorithms. In *Proceedings of the l5th asilomar conference: Circuits, systems, computers* (pp. 175–179).

Friedlander, B. (1982). System identification techniques for adaptive signal processing. *IEEE Transactions on Acoustics Speech and Signal Processing, 30*, 240–246.

Johnson, C. R. Jr., (1984). Adaptive iir filtering: Current results and open issues. *IEEE Transactions on Information Theory, 30*, 237–250.

Karaboga, N. (2009). A new design method based on artificial bee colony algorithm for digital iir filters. *Journal of the Franklin Institute, 346*, 328–348.

Krusienski, D. J., & Jenkins, W. K. (2004). Particle swarm optimization for adaptive iir filter structures. In *Proceedings of the IEEE congress on evolutionary computation* (pp. 965–970).

Ljung, L. (1987). *System identification: Theory for the user*. NJ, Prentice-Hall: Englewood Cliffs.

Luitel, B., & Venayagamoorthy, G. K. (2010). Particle swarm optimization with quantum infusion for system identification. *Engineering Applications of Artificial Intelligence, 23*, 635–649.

Majhi, B., & Panda, G. (2010). Development of efficient identification scheme for nonlinear dynamic systems using swarm intelligence techniques. *Expert Systems with Applications, 37*, 556–566.

Nayeri, M. (1988). A weaker sufficient condition for the unimodality of error surfaces associated with exactly matching adaptive iir filters. In *Proceedings of the 22nd asilomar conference: Signals, systems, computers, Pacific Grove, California* (pp. 35–38).

Ng, S., Leung, S., Chung, C., Luk, A., & Lau, W. (1996). The genetic search approach: A new learning algorithm for adaptive iir filtering. *IEEE Transactions on Signal Processing, 13*, 38–46.

Regalia, P. A. (1992). Stable and efficient lattice algorithms for adaptive iir filtering. *IEEE Transactions on Signal Processing, 40*, 375–388.

Shynk, J. J. (1989a). Adaptive iir filtering. *IEEE Transactions on Acoustics Speech and Signal Processing*, 4–21.

Shynk, J. J. (1989b). Adaptive iir filtering using parallel-form realizations. *IEEE Transactions on Acoustics Speech and Signal Processing, 37*, 519–533.

Siiderstrom, T., Ljung, L., & Gustavsson, I. (1978). A theoretical analysis of recursive identification methods. *Automatica, 14*, 231–244.

Soderstrom, T. (1975). On the uniqueness of maximum likelihood identification. *Automatica, 11*, 193–197.

Soderstrom, T., & Stoica, P. (1982). Some properties of the output error method. *Automatica, 18*, 93–99.

White, S. A. (1975). An adaptive recursive digital filter. In *Proceedings of the 9th asilomar conference: Circuits, systems, computers* (pp. 21–25).

Widrow, B., & Strearns, S. D. (1985). *Adaptive signal processing*. NJ, Prentice-Hall: Englewood Cliffs.