Circuits, Systems, and Signal Processing (2022) 41:700–724
https://doi.org/10.1007/s00034-021-01816-4

A Novel Reduction Approach for Linear System Approximation

Nafees Ahamad[1] · Afzal Sikander[2] · Gagan Singh[1]

Received: 14 August 2020 / Revised: 31 July 2021 / Accepted: 2 August 2021 / Published online: 20 August 2021

Abstract

In this study, a novel system reduction approach is suggested for a linear time-continuous system. Motivated by various optimization techniques and reduction problems in system engineering, a new search algorithm, namely ant lion optimization (ALO), is being utilized for system approximation. This algorithm is based on the random walk of an ant lion for searching the food. Firstly, the approximated system is formulated by reducing the integral square error (ISE) between the higher-order and proposed approximated system using ALO. To validate the high efficiency and accuracy of the suggested approach, it is tested on four benchmark systems maximum up to 84th order including a time-delay system. It is revealed that approximated system characteristics, achieved by the suggested approach, are much closer to the characteristics of the higher system. Further, it is also revealed that the transient, steady-state, and frequency response characteristics of the higher-order system are preserved by the suggested approximated system. Additionally, the lowest value of ISE is observed with the proposed approximated system as compared to other approximated systems already available. Furthermore, the efficacy of the suggested approach is also investigated in terms of final convergence rate and CPU usage time by employing well-known the genetic algorithm (GA) and particle swarm optimization (PSO) to obtain approximated systems.

✉ Afzal Sikander
afzals@nitj.ac.in

Nafees Ahamad
nafees.ahamad@dituniversity.edu.in

Gagan Singh
gagan.singh@dituniversity.edu.in

[1] Department of Electrical and Electronics Communication Engineering, DIT University, Dehradun, India

[2] Department of Instrumentation and Control Engineering, Dr. B. R. Ambedkar National Institute of Technology Jalandhar, Punjab, India

Keywords Ant lion optimization · System approximation · Error minimization · Stability

1 Introduction

In system engineering, the order of the plant or designed controller could be very high which adds an additional burden on simulation and computation. If a higher-order system or controller can be approximated in terms of its order, it saves computation time and cost. System approximation methods may be classified into three headings: classical methods, using optimization algorithms, and mixed methods.

The classical methods are proposed by different researchers. They are based on simple techniques such as dominant pole preservation, dividing state matrix into two parts, retention of predominated Eigenvalues, continued fraction expansions, time moments matching, factor division method, clustering of poles and zeros, and Routh stability criterion. These classical methods are also mixed by many researchers to get the reduced-order system. The disadvantages of these methods are the higher cost of computation and sometimes the problem of stability. Recently, the Improved Balanced Realization technique [24] was proposed in which, the denominator was calculated based on balanced realization, whereas the numerator was calculated by the simple mathematical procedure. Some other classical methods based on second-order optimization are [9,10].

In the second method, optimization techniques are used for order reduction. They are based on the minimization of the error between the original and the reduced systems [13]. The error functions are taken as Integral of Absolute Error (IAE) [4], Integral of Time multiplied by Absolute Error (ITAE) [30], Integral of Square Error (ISE) [4], Integral of Time multiplied by Square Error (ITSE) [30], L1, L2, L-infinity norm [14], weighted error function [12], etc. Meta-heuristic algorithms inspired by nature are widely used in the optimization technique method [23]. In this category, the first one is the Genetic Algorithm (GA) [16] which works on the survival of fittest. Some other models are Pachycondyla apicalis metaheuristic (API) algorithm [21] based on foraging behavior of a population of Pachycondyla apicalis, Particle Swarm Optimization (PSO) [17] based on flocking behavior of birds, Harmony Search Algorithm (HSA) [18] based on musical improvisation, Cuckoo Search Algorithm (CSA) [3,27] based on a cuckoo bird which lays their own eggs in the nest of other host bird and Big Bang–Big Crunch [5] theory is based on the evolution of the Universe. These techniques are successfully applied in various fields such as: API in parameters identification of chaotic electrical system [19], Cuckoo Search in order reduction [27], PSO in fuzzy predictive adaptive controller design [6]. Further, these optimization techniques can also be mixed to get better results in any field of research [8].

In the third category, order reduction is done by using classical methods plus optimization algorithms. Nadi [2] proposed a mixed method for single-input single-output (SISO) and multi-input multi-output (MIMO) systems in which dominant poles were chosen from the original system, while the other parameters were selected by PSO. Big Bang–Big Crunch optimization was mixed with Routh approximation by Desai et al. [11], and recently, it was also mixed with Time moment matching [5]. In the above two methods, the numerator was calculated by Big Bang–Big Crunch optimization and the denominator was estimated by Routh approximation and Time moment matching, respectively. Two unified hybrid metaheuristic algorithms [15] for the discrete-time system were proposed by Ganguli in which a Gray Wolf optimization and Firefly algorithm were combined for order reduction. Hence, in spite of so many methods, still, new algorithms are in great demand.

The system approximation techniques provided till now are either problem-specific or offer large errors in the resultant reduced system. Moreover, they are time-consuming and computationally more expensive. This motivates the researchers to explore the new more effective, general, and computationally less time-consuming system approximation techniques. This article presents a new technique using ALO that tide over all the problems listed above. The ALO exceptionally reduces the error between the two systems because it has higher efficiency, high computational speed, and fast convergence. The competency of the proposed algorithm is shown by diminishing some reference systems, including a system of order 84, and the results thus obtained are better than the other techniques existing in the literature. Also, the performance of the proposed approximated system is not only analyzed in the time domain but also in the frequency domain. The final convergence rate and CPU usage time are less as compared to other optimization techniques such as genetic algorithm (GA) and particle swarm optimization (PSO). Therefore, this article contributes a more effective, computationally less expensive, and general technique of system approximation.

2 Problem Formulation

2.1 Definition of Order Reduction

Let us define a system transfer function of order n as

$$G_k(s) : u \rightarrow y_k \tag{1}$$

The objective of order reduction is to find out a new transfer function $R_r(s)$ of reduced order r which is defined as

$$R_r(s) : u \rightarrow y_r \quad \text{with} \quad r < k \tag{2}$$

provided input and output characteristics are the same for the above two systems, that is for the same input $u(t)$, output $y_k(t) \approx y_r(t)$. For the sake of definiteness, the type of systems considered in this paper is proper LTI systems. The proposed method is limited to transfer function matrices such that

- $G_k(s)$ are rational and stable, i.e., the poles lie on the left-half of the s-plane.
- In frequency domain, $G_k(jw)$ is nonzero for all ω including $\omega = \infty$.

In other words, MOR leads to optimization problem in the following manner. MOR defines the problem of finding the reduced-model $R_r(s)$ from the full-order plant $G_k(s)$ using optimization formulation such that

$$\text{Minimize ISE} = \int_0^\infty e(t)^2 \mathrm{d}t \tag{3}$$

where $e(t) = y_k(t) - y_r(t)$.

We have selected ISE criterion because it quickly eliminates large errors in both transient and steady-state in comparison with other integral error performance measures such as ISE, IAE, ITSE.

2.2 Single-Input Single-Output (SISO) Systems

Let the higher order system is given as follows-

$$G_k(s) = \frac{N_{k-1}(s)}{D_k(s)} = \frac{\sum_{i=0}^{k-1} b_i s^i}{\sum_{i=0}^{k} a_i s^i} \tag{4}$$

where a_i and b_i are constants, whereas for unity steady state output $a_0 = b_0$.

The objective is to obtain the reduced system of order $'r'$ $(r < k)$ such that it preserves all necessary characteristics of the higher order system and represented as follows-

$$R_r(s) = \frac{N_{r-1}(s)}{D_r(s)} = \frac{\sum_{i=0}^{r-1} d_i s^i}{\sum_{i=0}^{r} c_i s^i} \tag{5}$$

where c_i and d_i are unknown constants.

3 Ant Lion Optimization (ALO)

The Ant Lion or Antlion is a recently developed meta-heuristic optimization technique in 2015 by Seyedali [20] which basically models the trapping of ants by antlions. Antlions are also called doodlebugs sometimes. They all come under the Myrmeleontidae family and live in two phases, larva and adult. They have a very interesting hunt mechanism when they are in larvae form. A small coned-shaped trapped are made by antlions to catch ants. They sit under the pit and wait for prey to be trapped. Antlions eat the flesh of trapped prey and throw the leftovers outside the pit. This pit is again prepared for the next hunting. They make a big pit when they are more hungry, and this is the inspiration for the ALO algorithm. The method includes five main steps on the stochastic base of ants, building traps, ants trapped in traps, capturing ants in traps, and rebuilding traps. The mathematical model of ALO can be brief in the following five steps

Step 1: Random walk of ants It can be mathematically written as

$$X_i = \begin{bmatrix} 0,\ f_r(1),\ f_r(1) + f_r(2), \ldots, \sum_{j=1}^{N-1} f_r(j), \\ \sum_{j=1}^{N} f_r(j) \end{bmatrix} \tag{6}$$

where $i = 1, 2, \ldots din$ and din is the size of ant and antlion, N is the number of iterations and $f_r(j)$ is a stochastic function given as

$$f_r(j) = \begin{cases} 1 & \text{if } P_{\text{rand}} > 0.5 \\ -1 & \text{if } P_{\text{rand}} \leq 0.5 \end{cases} \tag{7}$$

where P_{rand} is the pseudo-random number generated in the interval of [0, 1]. Previous random walks can be mapped into the realistic search space in the position with the lower and upper limits using the following formula

$$Y_i = \left(\frac{X_i - a_i}{b_i - a_i}\right) \times (d_i - c_i) + c_i \tag{8}$$

where a_i and b_i are the min and max values of X_i ; c_i and d_i are the min and max antlion in ith dimensions. In Eq. (8), X_i is standardized in the domain [0, 1] using $\left(\frac{X_i - a_i}{b_i - a_i}\right)$ and then converted in the domain $[c_i \ d_i]$.

Step 2: Trapping ants in antlions' pit

If minimum and maximum changing limit are denoted by $c_{\min}^{'}$ and $d_{\max}^{'}$ at current iteration and the position of antlion selected on the bases of its fitness value using the Roulette wheel is denoted by P_{Antlion}, then ants movement can be mathematically written as

$$c_{\min} = c_{\min}{'} + P_{Antlion}, \qquad d_{\max} = d_{\max}{'} + P_{\text{Antlion}} \tag{9}$$

Step 3: Building traps

The probability of establishing a trap by antlions is proportional to their skill value and is selected by the Roulette wheel.

Step 4: Ants sliding

Once the ants are stuck, they will try to getaway and the process of sliding the ant will occur. The updated values of $c_{\min}^{'}$ and $d_{\max}^{'}$ are calculated by the following equations

$$c_{\min}^{'} = \frac{lb}{10^w \times (t/N\ )}, \qquad d_{\max}^{'} = \frac{ub}{10^w \times (t/N\ )} \tag{10}$$

where t represents the current iteration and the constant w is defined according to the latest iteration.

Step 5: Elitism

The location of each ant is updated according to the random walk and is selected by the wheel of the Roulette and the elite; mathematically, it can be written as

$$P_{\text{update}} = \frac{R_{\text{antlion}} + R_{\text{elite}}}{2} \tag{11}$$

where P_{update} denotes the updated location, R_{antlion} and R_{elite} are the random walks around the antlion and elite selected by the Roulette wheel.

Step 6: Catching Prey

When the ant reaches the bottom of the pit, antlion must take its place. When prey is captured, $Antlion = Ant$, provided $f(Ant) < f(Antlion)$.

ALO has less number of parameters to be initialized, better search performances, and fast convergence speed [20] as compared to GA and PSO. This algorithm is also tested for Multi-layer Perceptrons Training [31], and the classification rate results are the best (>90%) as compared to GA, PSO, and Ant Colony Optimization (ACO). A tabular comparison is shown below (Table 1), which depicts that ALO has less computational complexity and balanced exploration and exploitation.

4 Proposed Methodology

The theory of optimization is being employed here to calculate the reduced numerator and denominator by minimizing the predefined fitness function. Although the reduced system may be any type/order for simplification, the following reduced system is considered in this paper.

$$R_2(s) = \frac{d_0 + d_1 s}{c_0 + c_1 s + c_2 s^2} \tag{12}$$

where d_0, d_1, c_0, c_1 and c_2 are unknown coefficients of numerator and denominator, respectively, which are to be determined.

Therefore, the proposed algorithm is used to achieve the best values of the coefficients in Eq. (12), for reduce system, by minimizing the performance index described by Eq. (3). The algorithmic steps to find out numerator and denominator coefficients of the reduced system given by Eq. (5) are given in Algorithm 1.

5 Computational Experiments

This section discusses four different examples to show the superiority of the proposed MOR method. The first and second examples are fourth- and eighth-order SISO systems. The third example has a time delay which becomes a non-minimum system after Pade approximation of its time delay. The fourth example is of 84th order which has been reduced to a second-order system using the proposed algorithm. The results of all examples are compared with other up-to-date order reduction techniques available

Table 1 A comparison of ALO

S. no.	Parameters	ALO	GA	PSO
1	Complexity	O(n)	$O(n^2)$	O(n)
2	General feature	Random search population based	Random search Population based	Random search population based
3	Individual memory	None	None	Yes: pBest updating
4	Social operator	ALO_{Best}	Selection crossover	gBest
5	Exploitation & Exploration	Balance Exploration & Exploitation	Tunable	Higher w: Exploration Lower w: Exploitation

Algorithm 1: MOR by Proposed Algorithms

```
Initialize the No of ants and antlions ($c_0, c_1, \ldots c_{m_{r-1}}$ and $d_0, d_1, \ldots d_{n_r}$) randomly say $N$;
Calculate ISE using Equation (3) for ants and antlions;
Select elite antlion ( = antlion with minimum ISE);
while Number of iterations are less than specified do
    for Each Ant (1 to N) do
        Use Roulette Wheel and select an antlion out of N;
        Update c and d using Equation (10);
        Create normalized random walk with the help of Equations (6) and (8);
        Update antlion position using Equation (11);
    Estimate the fitness (ISE) of all ants using Equation(3);
    Replace an antlion with its corresponding ant if $(ISE)_{ant} < (ISE)_{antlion}$;
    Update Elite if $(ISE)_{antlion} < (ISE)_{elite}$;
Print the Elite ($c_0, c_1, \ldots c_{m_{r-1}}$ and $d_0, d_1, \ldots d_{n_r}$) with its $ISE$ value
```

Table 2 ALO parameters

Parameters	Values
Population (No. of Ants, N)	50
Maximum iterations count (t)	300/500
No. of Variables (dim)	Problem dependent
Random Number	[0,1]

in the literature as well as GA and PSO methods. Several trials are made, and the best solutions (least ISE value) are considered with ALO parameters shown in Table 2. All problems are solved with Intel(R) Core(TM) i3-5005U, 2.000 GHz processor, and 4.0 GB RAM computer.

Example 1 Consider a fourth-order benchmark problem ([5,28,29]) with the following transfer function

$$G(s) = \frac{s^3 + 7s^2 + 24s + 24}{s^4 + 10s^3 + 35s^2 + 50s + 24} \tag{13}$$

The second-order approximation of the above system with the proposed algorithm has the following transfer function

$$G_{r\text{Prop}}(s) = \frac{14.29s + 28.78}{18.66s^2 + 45.18s + 28.87} \tag{14}$$

This system is also reduced to second order by GA and PSO techniques and cost versus iterations curves are shown in Fig. 1. It is clear from this figure that the convergence rate of ALO is slow up to around 50 iterations, but afterward it becomes better and the obtained value of ISE is less. Figures 2 and 3 show the time and frequency domains comparisons of different systems. The proposed system is consistent in the time and frequency domain. Table 3 shows the comparison in terms of its ISE error criterion,

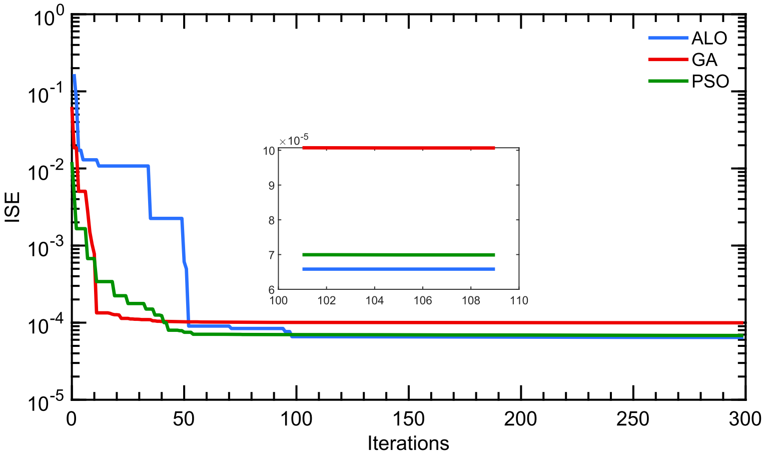


Fig. 1 Cost versus iterations of Example 1

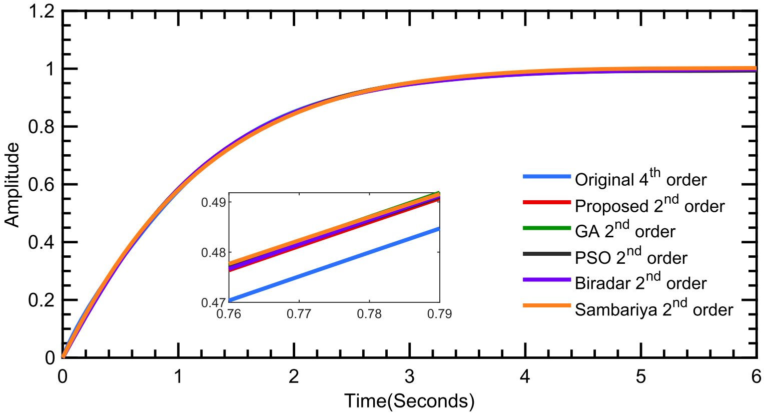
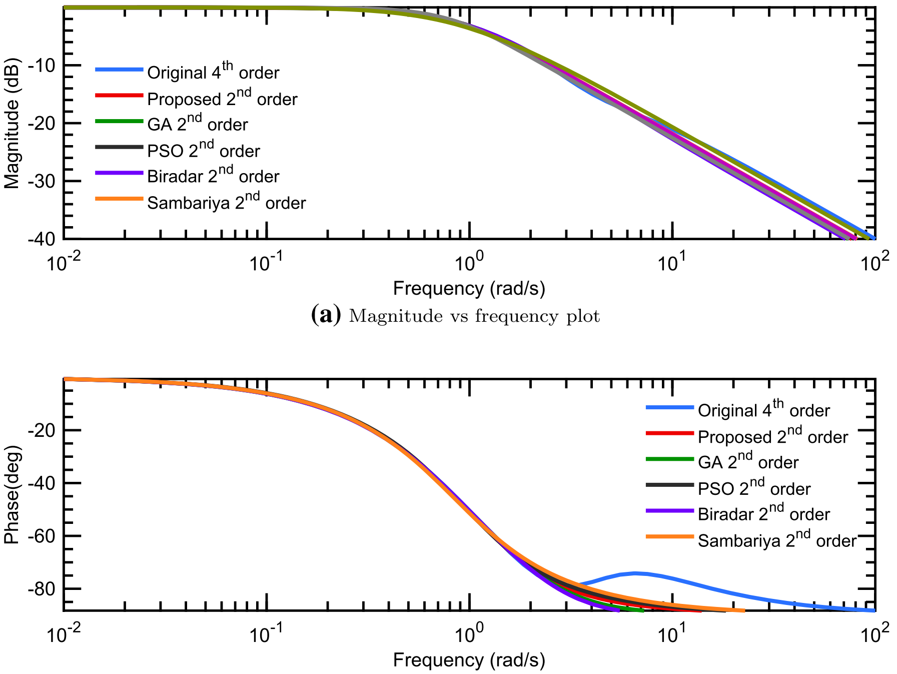


Fig. 2 Time domain comparison of Example 1

and CPU simulation time. The proposed system has the least error (6.39×10^{-05}) and simulation time. Rise time, Settling time, Overshoot, and Peak time are shown in Table 4. These values are comparable with GA and PSO, whereas better than the other methods of order reduction.

Example 2 Consider an eighth-order benchmark problem ([1,5,28]) with transfer function given by the following equation

$$G(s) = \frac{18s^7+514s^6+5982s^5+36380s^4+122664s^3 + 222088s^2+185760s^1+40320}{s^8+36s^7+546s^6+4536s^5+22449s^4+67284s^3+118124s^2+109584s^1+40320} \tag{15}$$



(a) Magnitude vs frequency plot

(b) Phase vs frequency plot

Fig. 3 Frequency domain comparison of Example 1

Table 3 ISE and CPU time comparison for different systems of Example 1

Systems	ISE	CPU time (s)
Proposed (14)	6.39×10^{-05}	175.45
GA	9.60×10^{-05}	197.31
PSO	6.64×10^{-05}	195.87
[5]	1.15×10^{-04}	–
[25]	2.23×10^{-04}	–
[11]	2.55×10^{-04}	–
[22]	1.60×10^{-03}	–
[23]	1.70×10^{-03}	–

The above system is reduced to second order by the proposed technique which results in following transfer function

$$G_{r\text{Prop}}(s) = \frac{71.23s + 21.68}{4.19s^2 + 28.99s + 21.83} \tag{16}$$

The cost versus iterations plots are shown in Fig. 4. Although the convergence rates are similar in this example, the final ISE value is minimum in the case of ALO. The step response and Bode plots are shown in Figs. 5 and 6 for the original and reduced

Table 4 Transient response comparison for different systems of Example 1

Systems	Rise time (s)	Settling time (s)	%Overshoot	Peak	Peak time (s)
Original	2.2603	3.9308	0	0.9990	6.8847
Proposed (14)	2.2294	3.7342	0	0.9964	6.4294
GA	2.2713	3.9438	0	0.9999	6.8778
PSO	2.1981	3.5442	0.0879	0.9935	6.1793
[5]	2.2680	3.9580	0	1.0004	9.9194
[25]	2.2767	3.6319	0.2496	1.0022	6.0624
[11]	2.2785	3.6199	0.2738	1.0027	5.9728
[22]	2.1885	3.2197	1.3008	1.0122	4.9915
[23]	2.5289	4.5536	0	0.9990	8.1718

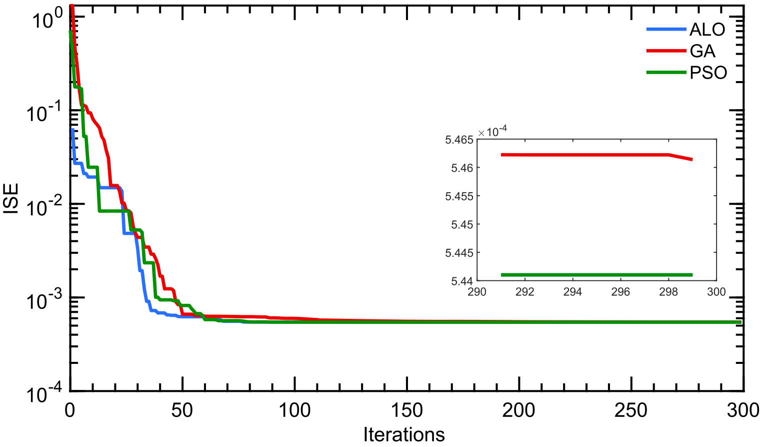


Fig. 4 Cost versus iterations of Example 2

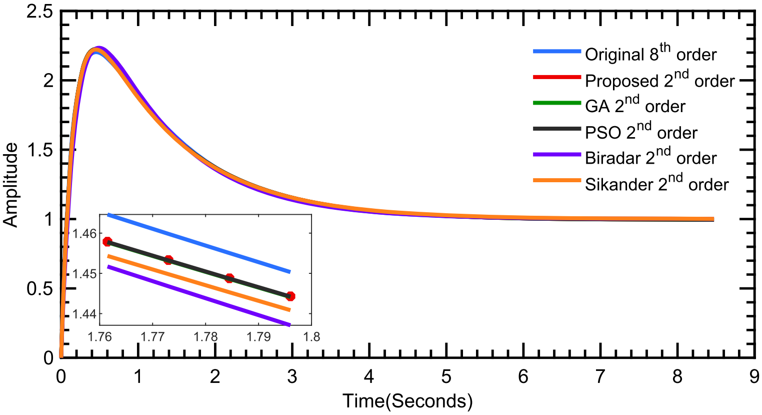
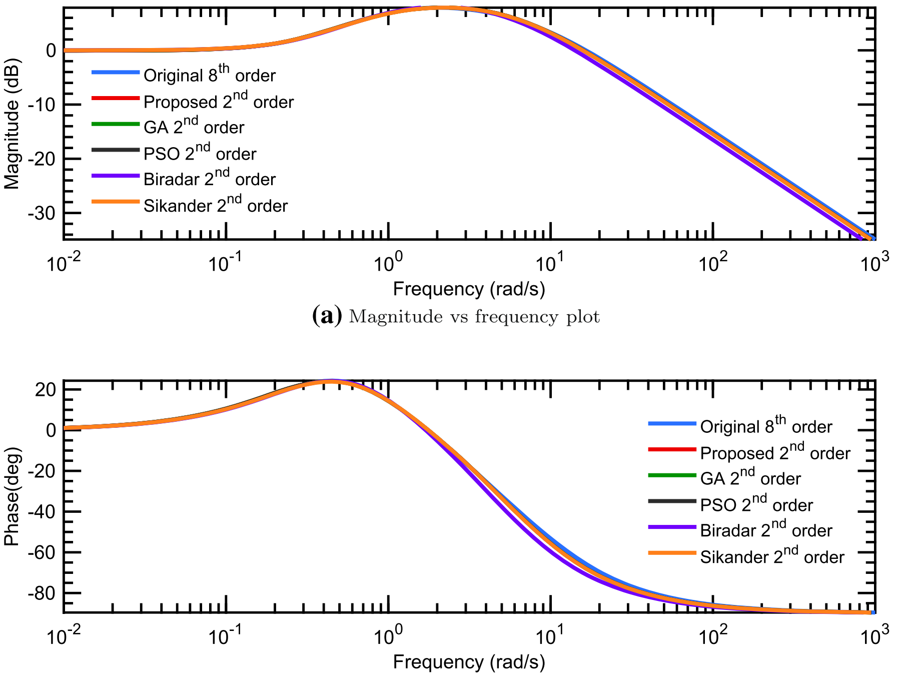


Fig. 5 Time domain comparison of Example 2

systems. It is clear that the proposed system matches in time as well as frequency domain over the entire time and frequency range. The proposed method has a step response and Bode plot more closed to the original system and comparable with GA and PSO. Table 5 shows the comparison in terms of ISE and CPU time. The obtained simulation time is the least as compared to GA and PSO. Table 6 shows Rise time, Settling time, Overshoot, and Peak time. These values are more closely matched with the original system as compared to Biradar, Sikander, Desai, and Abdullah methods.

Example 3 Consider a seventh-order delayed system ([5]) with following transfer function

$$G(s) = \frac{(4000s + 50000)e^{-0.3s}}{s^7 + 69s^6 + 1764s^5 + 20280s^4 + 102500s^3 + 221375s^2 + 187500s + 50000} \tag{17}$$



(a) Magnitude vs frequency plot

(b) Phase vs frequency plot

Fig. 6 Frequency domain comparison of Example 2

Table 5 ISE and CPU time comparison for different systems of Example 2

Systems	ISE	CPU time (s)
Proposed (16)	5.4410×10^{-04}	136.79
GA	5.4418×10^{-04}	181.38
PSO	5.4414×10^{-04}	170.21
[5]	3.9169×10^{-03}	–
[28]	7.2544×10^{-04}	–
[11]	1.9252×10^{-00}	–

In above system, time delay of 0.3 sec can be approximated to third order by Pade approximation, so system of Eq. (17) converts to tenth order of the form $G(s) = N(s)/D(s)$, where $N(s)$ and $D(s)$ are given below

$$\begin{aligned} N(s) &= -4000s^4 + 110000s^3 - 666700s^2 - 15560000s + 222200000 \\ D(s) &= s^{10} + 109s^9 + 5191s^8 + 141300s^7 + 2396000s^6 + 25680000s^5 \\ &\quad + 167500000s^4 + 610500000s^3 \\ &\quad + 1111000000s^2 + 866700000s + 222200000 \end{aligned} \tag{18}$$

Table 6 Transient response comparison for different systems of Example 2

Systems	Rise time (s)	Settling time (s)	%Overshoot	Peak	Peak time (s)
Original	5.6930×10^{-2}	4.8201	1.2035×10^{2}	2.2035	4.4900×10^{-1}
Proposed(16)	5.8790×10^{-2}	5.1779	1.2396×10^{2}	2.2248	4.5584×10^{-1}
GA	5.8816×10^{-2}	5.1749	1.2394×10^{2}	2.2248	4.5610×10^{-1}
PSO	5.8797×10^{-2}	5.1779	1.2394×10^{2}	2.2246	4.5584×10^{-1}
[1]	5.9100×10^{-2}	5.0544	1.2381×10^{2}	2.2469	4.4650×10^{-1}
[5]	6.6748×10^{-2}	4.7990	1.2331×10^{2}	2.2342	4.9388×10^{-1}
[28]	5.9629×10^{-2}	5.1045	1.2214×10^{2}	2.2219	4.4378×10^{-1}
[11]	5.2429×10^{-1}	8.7784	6.1535×10^{1}	1.6153	2.2757

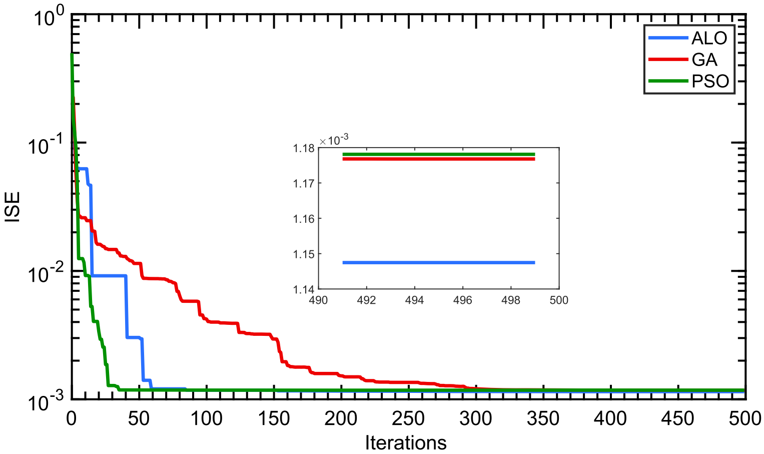


Fig. 7 Time domain comparison of Example 3

Table 7 ISE comparison for different systems of Example 3

Systems	ISE	CPU time (s)
Proposed (19)	1.1550×10^{-3}	136.93
GA	1.1763×10^{-3}	197.42
PSO	1.1781×10^{-3}	153.95
[5]	2.9282×10^{-3}	-
[11]	5.9292×10^{-2}	-

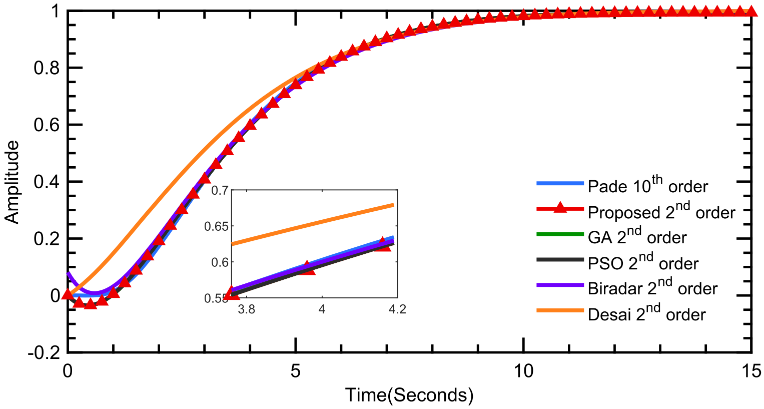
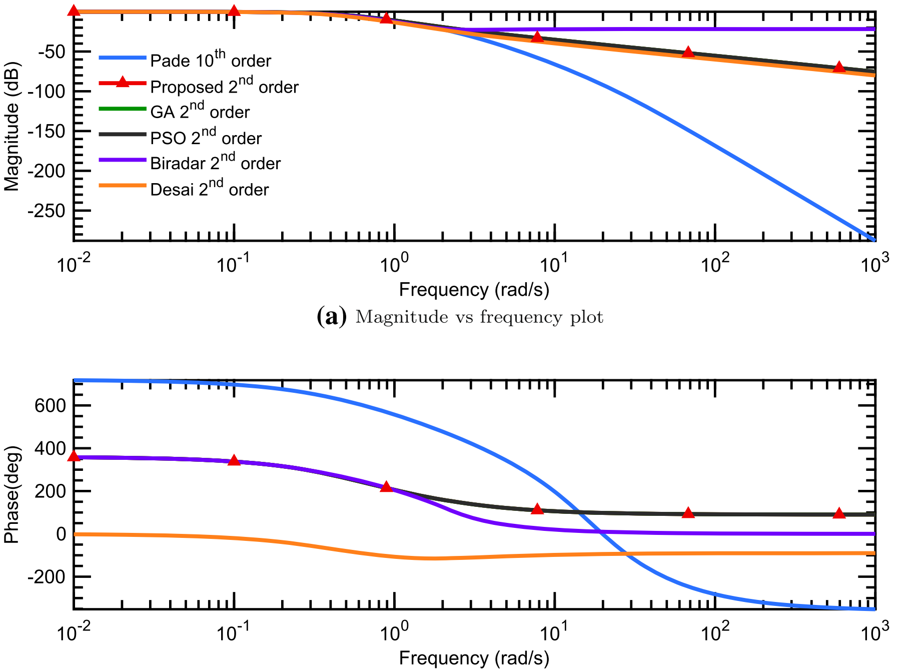


Fig. 8 Time domain comparison of Example 3

This tenth order system is reduced to a second order with following transfer function

$$G_{r\text{Prop}}(s) = \frac{-0.1394s + 0.2491}{0.8484s^2 + 0.8339s + 0.2512} \tag{19}$$

The convergence rate of ALO is between GA and PSO as shown in Fig. 7, whereas time taken to simulate (CPU time) and ISE is the least, as shown in Table 7. The step



(a) Magnitude vs frequency plot

(b) Phase vs frequency plot

Fig. 9 Frequency domain comparison of Example 3

response and Bode plot are shown in Figs. 8 and 9, respectively. It can be noted from the Bode plot that all methods are not reliable for higher frequencies, whereas they are performing well for a lower range of frequencies. A comparison in terms of ISE is drawn in Table 7 which shows that ALO, GA, and PSO are performing well. The step response specifications are shown in Table 8 which shows that genetic algorithm and particle swarm optimization techniques are comparable with ALO.

Example 4 Consider problem of 84 order [7] represented as follows

$$A = \begin{bmatrix} A_1 & 0 & 0 & 0 & 0 & 0 & 0 & a_{78} & 0 & 0 \\ 0 & A_2 & 0 & 0 & 0 & 0 & 0 & 0 & \ddots 0 & \\ 0 & 0 & A_3 & 0 & 0 & 0 & 0 & 0 & 0 & a_{154} \\ 0 & 0 & 0 & A_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & A_7 & 0 & 0 & 0 \\ a_1 & 0 & 0 & 0 & 0 & 0 & 0 & A_8 & 0 & 0 \\ 0 & \ddots 0 & 0 & 0 & 0 & 0 & 0 & \ddots & 0 & \\ 0 & 0 & a_{77} & 0 & 0 & 0 & 0 & 0 & 0 & A_{42} \end{bmatrix}$$

Table 8 Transient response comparison for different systems of Example 3

Systems	Rise time (s)	Settling time (s)	Overshoot	Undershoot	Peak	Peak time (s)
`Original`	5.3179	10.227	0	2.0010×10^{-4}	9.9944×10^{-1}	17.375
`Proposed` (19)	5.2210	9.1364	0.1395	3.3676×10^{0}	9.9267×10^{-1}	13.868
`GA`	5.2300	9.2041	0.1063	3.5388×10^{0}	9.9308×10^{-1}	14.212
`PSO`	5.2480	9.2435	0.0975	3.5054×10^{0}	9.9340×10^{-1}	14.368
[5]	5.4572	10.3190	0	0	9.9909×10^{-1}	15.8870
[11]	6.2731	10.2160	0.0088	0	1.0001×10^{-1}	17.1420

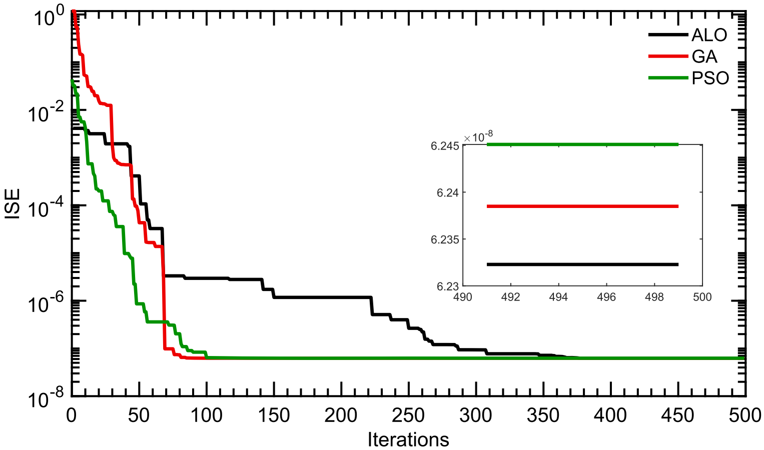


Fig. 10 Time domain comparison of Example 4

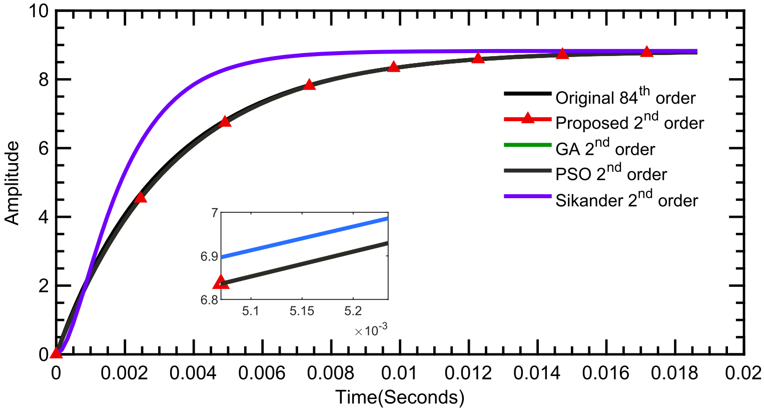


Fig. 11 Time domain comparison of Example 4

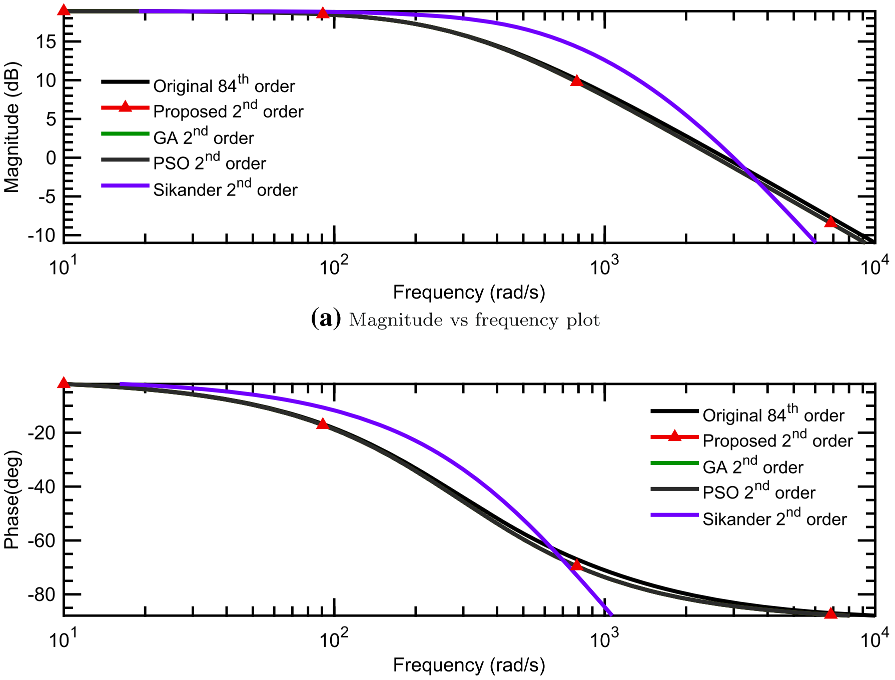
where $A_i = \begin{bmatrix} -734 & 171 \\ -9 & -734 \end{bmatrix}; \quad i = 1, 2 \ldots 42$
and $a_j = 196; \quad j = 1, 2 \ldots 154$

$B = [B_{i,1}]_{84\times 1}; \quad i = 1, 2 \ldots 84$

The values of $[B_i]$ are given in Table 11 in Appendix. $C = [C_{1,j}]_{1\times 84}; \quad j = 1, 2 \ldots 84 = B^T$

The lower order system obtained from the proposed method is shown below

$$G_{r\text{Prop}}(s) = \frac{606.47s + 818.5}{0.2339s^2 + 69.04s + 92.75} \tag{20}$$



(a) Magnitude vs frequency plot

(b) Phase vs frequency plot

Fig. 12 Frequency domain comparison of Example 4

Table 9 ISE comparison for different systems of Example 4

Systems	ISE	CPU time (s)
Proposed (20)	6.2323×10^{-8}	131.92
GA	6.2539×10^{-8}	230.11
PSO	6.2451×10^{-8}	145.16
[26]	2.1023×10^{-3}	-

The cost versus iterations graphs are shown in Fig. 10 for different optimization techniques. The initial convergent rate of ALO is comparable with GA and PSO which becomes inferior later on, but the final obtained value of ISE is the least. This higher order system has also been reduced to second order by Afzal [26] with following transfer function

$$G_{r\text{Sikander}}(s) = \frac{371.3s + 919.3}{0.1432s^2 + 42.43s + 104.2} \tag{21}$$

The step response of the original and reduced systems is shown by Fig. 11 and Bode plot by Fig. 12. It is evident that the response in time and frequency domains is very well approximated by the proposed ALO method. This approach provides good approximation at lower as well as at higher frequencies as shown by the Bode plot.

Table 10 Transient response comparison for different systems of Example 4

Systems	Rise time (s)	Settling time (s)	Overshoot	Peak	Peak time (s)
`Original`	7.4464×10^{-3}	1.3412×10^{-2}	0	8.8164	2.4372×10^{-2}
`Proposed` (20)	7.4758×10^{-3}	1.3313×10^{-2}	0	8.8240	3.5883×10^{-2}
`GA`	7.4752×10^{-3}	1.3311×10^{-2}	0	8.8240	3.5880×10^{-2}
`PSO`	7.4758×10^{-3}	1.3313×10^{-2}	0	8.8240	3.5883×10^{-2}
[26]	3.6748×10^{-3}	6.6207×10^{-3}	0	8.8187	1.1303×10^{-2}

For this particular problem, ALO, GA, and PSO all perform well in terms of ISE ($order 10^{-8}$) compared to [26] ($ISE = 2.1023 \times 10^{-3}$) as shown in Table 9. The obtained ISE is slightly less as compared to GA and PSO, and the CPU usage time is the least by the proposed method. The comparison in terms of Rise and Settling times, Over Shoot, Peak and Peak time for different systems is done in Table 10. These values are very closely matched with the original system by the reduced second-order system using ALO. The proposed algorithm also performed fairly well even for a very high order system.

6 Conclusion

A novel reduction approach for linear time-invariant (LTI) system approximation is advised in this paper. The presented approach is based on the concept of optimization, and therefore, the fast and more accurate Ant Lion Optimization (ALO) technique has been employed. The unknown parameters of the approximated system are calculated by minimization of integral square error (ISE) between the original and the approximated system. The efficacy of the proposed reduction approach is proved by taking different types of computational experiments. It reveals that the proposed reduction approach exhibits the best results for a SISO as well as a real system of partial differential equations of 84th order. The complexity of the 84th order system is reduced to very low (second order). An example of the time-delay system has been considered which also shows the supremacy of this technique. The proposed reduction approach is compared with the latest available methods, and the comparative results confirm the efficacy of the proposed approach in terms of ISE and CPU usage time. The following conclusions are drawn about ALO from the order reduction point of view.

- In general, the error ISE is reduced by ALO to the lowest value as compared to GA and PSO techniques.
- Initial convergent rate of ALO is comparable to GA and PSO techniques.
- The ALO takes minimum CPU time which results in fast order diminution.
- The proposed technique provides better results as compared to classical methods available in the literature.
- The proposed technique is fast and accurate as compared to existing optimization techniques of MOR.

Further, looking into the analysis of this paper, ALO can also be applied to a discrete-time system as well as an interval system in the future and it may also be applied to real-world problems such as Power system, Single machines infinite bus (SMIB), Two-wheeled mobile robot (TWMR), etc. Alternatively, it can be further modified to get a faster convergent rate.

Funding This study was not funded by anyone.

Availability of Data and Material This manuscript has no associated data.

Declarations

Conflict of interest The authors declare no conflict of interest.

Ethical Approval This article does not contain any studies with human participants performed by any of the authors.

Appendix A

See Table 11.

Table 11 Coefficients of $[B_{i,1}]$

i	$[B_{i,1}]$	i	$[B_{i,1}]$	i	$[B_{i,1}]$	i	$[B_{i,1}]$	i	$[B_{i,1}]$	i	$[B_{i,1}]$
1	7.624824	15	3.413381	29	3.468259	43	8.562598	57	2.056365	71	9.036562
2	2.354606	16	5.595597	30	3.058394	44	5.995426	58	2.93394	72	6.950758
3	7.570281	17	3.976737	31	9.01239	45	4.822466	59	2.561231	73	7.567942
4	2.321555	18	9.456895	32	1.065961	46	0.120089	60	3.126108	74	2.666435
5	4.608591	19	8.76504	33	2.963187	47	2.787891	61	5.486768	75	6.112229
6	0.353796	20	9.884494	34	8.515574	48	5.741295	62	8.308178	76	1.722831
7	5.242211	21	3.913699	35	7.294136	49	8.2792	63	2.974988	77	1.518785
8	3.992345	22	9.201152	36	6.427816	50	7.045744	64	8.77164	78	6.839423
9	8.994307	23	4.575071	37	2.681877	51	3.418584	65	7.548852	79	2.179521
10	1.71435	24	6.616429	38	7.281299	52	4.020264	66	3.25237	80	7.636844
11	0.247185	25	5.214498	39	8.92205	53	5.389039	67	2.581167	81	0.669466
12	5.921826	26	9.829012	40	5.099291	54	1.897445	68	7.341292	82	2.401817
13	5.702193	27	8.996692	41	5.530897	55	4.346914	69	4.374441	83	3.542152
14	5.824618	28	6.302044	42	8.808669	56	4.086833	70	3.124567	84	6.076599

References

1. H.N. Abdullah, A hybrid bacterial foraging and modified particle swarm optimization for model order reduction. Int. J. Electr. Comput. Eng. **9**(2), 1100–1109 (2019)
2. D. Abu-Al-Nadi, O.M. Alsmadi, Z.S. Abo-Hammour, Reduced order modeling of linear mimo systems using particle swarm optimization, in *The seventh international conference on Autonomic and Autonomous Systems* (ICAS 2011), pp. 3–7 (2011)
3. N. Ahamad, A. Sikander, G. Singh, Substructure preservation based approach for discrete time system approximation. Microsyst. Technol. **25**(2), 641–649 (2019)
4. A. Bhaumik, Y. Kumar, S. Srivastava, S.M. Islam, Performance studies of a separately excited dc motor speed control fed by a buck converter using optimized piλdμ controller, in *2016 International Conference on Circuit, Power and Computing Technologies* (ICCPCT), pp. 1–6. IEEE (2016)
5. S. Biradar, Y.V. Hote, S. Saxena, Reduced-order modeling of linear time invariant systems using big bang big crunch optimization and time moment matching method. Appl. Math. Model. **40**(15–16), 7225–7244 (2016)
6. S. Bououden, M. Chadli, F. Allouani, S. Filali, A new approach for fuzzy predictive adaptive controller design using particle swarm optimization algorithm. Int. J. Innovat. Comput. Inf. Control **9**(9), 3741–3758 (2013)
7. Y. Chahlaoui, P. Van Dooren, Benchmark examples for model reduction of linear time-invariant dynamical systems, in *Dimension Reduction of Large-Scale Systems*, pp. 379–392. Springer Berlin Heidelberg (2005)
8. I. Dassios, D. Baleanu, Optimal solutions for singular linear systems of caputo fractional differential equations. Math. Methods Appl. Sci. (2018)
9. I. Dassios, K. Fountoulakis, J. Gondzio, A preconditioner for a primal-dual newton conjugate gradient method for compressed sensing problems. SIAM J. Sci. Comput. **37**(6), A2783–A2812 (2015)
10. I.K. Dassios, Analytic loss minimization: theoretical framework of a second order optimization method. Symmetry **11**(2), 136 (2019)
11. S. Desai, R. Prasad, A novel order diminution of LTI systems using big bang big crunch optimization and Routh approximation. Appl. Math. Model. **37**(16–17), 8016–8028 (2013)
12. E. Eitelberg, Model reduction by minimizing the weighted equation error. Int. J. Control **34**(6), 1113–1123 (1981)
13. E. Eitelberg, Comments on model reduction by minimizing the equation error. IEEE Trans. Autom. Control **27**(4), 1000–1002 (1982)
14. R. El-Attar, M. Vidyasagar, Order reduction by l-1-and l-infinity-norm minimization. IEEE Trans. Autom. Control **23**(4), 731–734 (1978)
15. S. Ganguli, G. Kaur, P. Sarkar, A hybrid intelligent technique for model order reduction in the delta domain: a unified approach. Soft Comput. 1–14 (2018)
16. D.E. Golberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addion Wesley, Reading, 1989), p. 36
17. J. Kennedy, Particle swarm optimization. Encyclopedia of machine learning, pp. 760–766 (2010)
18. K.S. Lee, Z.W. Geem, A new structural optimization method based on the harmony search algorithm. Comput. Struct. **82**(9–10), 781–798 (2004)
19. F. Maamri, S. Bououden, M. Chadli, I. Boulkaibet, The Pachycondyla Apicalis metaheuristic algorithm for parameters identification of chaotic electrical system. Int. J. Parallel Emergent Distrib. Syst. **33**(5), 490–502 (2018)
20. S. Mirjalili, The ant lion optimizer. Adv. Eng. Softw. **83**, 80–98 (2015)
21. N. Monmarché, G. Venturini, M. Slimane, On how Pachycondyla Apicalis ants suggest a new search algorithm. Futur. Gener. Comput. Syst. **16**(8), 937–946 (2000)
22. G. Parmar, R. Prasad, S. Mukherjee, Order reduction of linear dynamic systems using stability equation method and ga. Int. J. Comput. Inf. Syst. Sci. Eng. **1**(1), 26–32 (2007)
23. B. Philip, J. Pal, An evolutionary computation based approach for reduced order modelling of linear systems, in *2010 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1–8. IEEE (2010)
24. A.K. Prajapati, R. Prasad, Order reduction in linear dynamical systems by using improved balanced realization technique. Circuits Syst. Signal Process. **38**, 1–15 (2019)
25. D. Sambariya, O. Sharma, Model order reduction using Routh approximation and cuckoo search algorithm. J. Autom. Control **4**(1), 1–9 (2016)

26. A. Sikander, Reduced Order Modelling for Linear Systems and Controller Design. Ph.D. thesis, Indian Institute of Technology Roorkee, Roorkee, India (2016)
27. A. Sikander, R. Prasad, A novel order reduction method using cuckoo search algorithm. IETE J. Res. **61**(2), 83–90 (2015)
28. A. Sikander, R. Prasad, Linear time-invariant system reduction using a mixed methods approach. Appl. Math. Model. **39**(16), 4848–4858 (2015)
29. C.B. Vishwakarma, R. Prasad, Clustering method for reducing order of linear system using Pade approximation. IETE J. Res. **54**(5), 326–330 (2008)
30. A. White, C. Kelly, Optimization of a control algorithm using a simulation package. Microprocess. Microsyst. **18**(2), 89–94 (1994)
31. W. Yamany, A. Tharwat, M.F. Hassanin, T. Gaber, A.E. Hassanien, T.H. Kim, A new multi-layer perceptrons trainer based on ant lion optimization algorithm, in *2015 Fourth International Conference on Information Science and Industrial Applications (ISI)*, pp. 40–45. IEEE (2015)