Soft Computing
https://doi.org/10.1007/s00500-023-09489-8

APPLICATION OF SOFT COMPUTING

Model order reduction for SISO and MIMO system using improved adaptive differential evolution algorithm

Aditya Prasad Padhy[1] · Sibarama Panigrahi[2] · Vinay Pratap Singh[3] · Prateek Pratyasha[4]

Accepted: 14 November 2023

Abstract

Mathematical models derived from the physical systems are usually complex and in the form of higher order differential equations. Such systems are difficult for analysis and controller synthesis. Therefore, it is desirable to develop an efficient algorithm for reducing such higher order systems to a lower order model by preserving all the significant characteristics of the original higher order system. For this, we have proposed an improved adaptive differential evolution algorithm (I-ADE), which is mixed with Routh approximation (RA) to determine the numerator and denominator coefficients of the corresponding lower order stable model (LOSM) by preserving the fundamental characteristics of a higher order stable system (HOSS). The superiority of the proposed method is illustrated by numerical test cases of single-input single-output (SISO) systems and multiple-input multiple-output (MIMO) systems. To evaluate the efficiency of proposed I-ADE algorithm, 23 benchmark functions are considered, and the results are statistically compared with nine promising meta-heuristic algorithms. The proposed I-ADE algorithm and the Routh approximation technique provide superior results in reducing the model order of SISO and MIMO systems. It achieves statistically the best rank among nine promising meta-heuristic algorithms employing the Friedman and Nemenyi Hypothesis test for optimizing 23 benchmark functions. This shows the proposed I-ADE algorithm's reliability, robustness and applicability for other optimization problems.

Keywords Model order reduction · Adaptive differential evolution based artificial neural network training · Routh approximation · SISO · MIMO

1 Introduction

In the last few decades, the model order reduction of higher order systems has been a significant research topic in the domain of control engineering. Mathematical modelling for complex physical systems is too expensive and tedious. Indeed, the mathematical modelling of intricate physical systems frequently yields higher-order differential equations or transfer functions. These more complex mathematical models can pose significant difficulties when it comes to both analyzing and constructing controller design. To overcome this scenario, model order reduction (MOR) has become the most promising technique to construct a corresponding lower order model that retains the significant characteristics of the higher order stable system (HOSS). A lower order stable model (LOSM) can enhance the comprehension of the original system, reduce computational complexities, and streamline the control design.

Various model order reduction techniques for linear continuous-time SISO systems have been documented in existing literature, encompassing both time-domain and frequency domain approaches (Fortuna et al. 2012; Schilders et al. 2008). Moreover, the model order reduction methods initially developed for SISO systems have also been extended to encompass MIMO systems. In the context of model order reduction for linear time-invariant (LTI)

✉ Sibarama Panigrahi
panigrahi.sibarama@gmail.com

[1] Department of Electrical and Electronics Engineering, ARKA JAIN University, Jamshedpur, Jharkhand, India

[2] Department of Computer Science and Engineering, National Institute of Technology, Rourkela, Odisha, India

[3] Department of Electrical Engineering, Malaviya National Institute of Technology, Jaipur, Rajasthan, India

[4] Department of Biomedical Engineering, National Institute of Technology, Raipur, Chhattisgarh, India

Published online: 16 January 2024

systems in the frequency domain, various techniques (Shamash 1974; Ashoor and Singh 1982) have been developed to match the time moments of HOSSs with LOSMs. One commonly used method is Padé approximation (Shamash 1974). Padé approximation is a straightforward approach that retains the initial time moments of the HOSS in the LOSM, while also matching their steady-state characteristics (Ashoor and Singh 1982). However, a significant limitation of Padé approximation is that it does not guarantee the stability of the lower order model, even when the higher order system is stable. This drawback raises concerns about the reliability of this technique. Researchers have introduced hybrid approaches to address the stability issue associated with Padé approximation (Hwang 1984; Lucas 1993; Mukherjee and Mittal 2005; Padhy et al. 2023). These hybrid techniques combine the characteristics of two different algorithms to generate more stable and accurate reduced-order models. Hwang (Hwang 1984) proposed a method that combines the Routh approximation (RA) technique with an integral square error (ISE) criterion to compute stable LOSMs. This approach minimizes the steady-state error between the original system and the reduced-order model. Lucas (Lucas 1993) presented a popular multipoint Padé approximation iterative technique focusing on generating optimal models by iteratively refining the approximation. This iterative process helps to improve the accuracy of the reduced-order model. Mukherjee and Mittal (Mukherjee and Mittal 2005) introduced a pole pattern technique for model reduction. This technique is designed to minimize the ISE between the higher order system and the lower order model, emphasizing the importance of capturing the system's dynamic behaviour accurately. Padhy et al. (Padhy et al. 2023) introduced a combined approach that utilizes both time moment (TM) matching and integrated stability equation methods to calculate lower order models. Furthermore, this technique aids in enhancing the stability performance of the lower order model. These hybrid approaches attempt to overcome the stability issues associated with Padé approximation and improve the accuracy of reduced-order models by integrating stability-retaining methods and optimal reduction techniques. Researchers continue to explore and develop various methodologies to effectively address the challenges of model reduction in the frequency domain.

Nowadays, soft computing techniques have fascinated researchers and industry to solve a wide variety of complex problems, including reducing large-scale systems. The recent emergence of nature-inspired swarm and evolutionary algorithms, namely, genetic algorithm (GA) (Goldberg 1989), particle swarm optimization (PSO) (Kennedy and Eberhart 1995), Moth-flame optimization algorithm (Mirjalili 2015), Sine cosine algorithm (Mirjalili 2016a; Padhy et al. 2021), Dragon fly algorithm (Mirjalili 2016b; Padhy 2022), artificial gorilla troops optimizer (Abdollahzadeh et al. 2021) have been used in the optimization problems. In the field of model order reduction, these algorithms are applied to minimize the objective/fitness function. The fitness function is often the integral square error (ISE), mean square error (MSE), or root mean square error (RMSE). Several authors (Vishwakarma and Prasad 2009; Desai and Prasad 2013a; Sikander and Prasad 2015a) proposed combined MOR methods in which denominator coefficients are computed using a mathematical technique, and the numerator polynomials are obtained by using swarm and evolutionary algorithms to match the system's transient response. A combined order reduction technique based on modified pole clustering and genetic algorithm is proposed in Vishwakarma and Prasad (2009). Desai and Prasad (Desai and Prasad 2013a) integrated the benefits of the Big Bang Big Crunch (BBBC) optimization technique and stability equation method for MOR. Another mixed technique for obtaining a lower order model based on the stability equation method and the nature-inspired PSO algorithm is presented in Sikander and Prasad 2015a. Furthermore, single nature-inspired optimization algorithms like PSO, GA, DF, SCA, etc., cannot provide global optimal solutions because of their limited feature and poor convergence characteristics. Therefore, now, researchers give attention towards hybrid optimization algorithms in the application of MOR. Evolutionary algorithms are frequently utilized to identify global maxima or minima of intricate fitness functions. In a referenced paper (Ganji et al. 2017), the author introduced a hybrid algorithm named PSO-DV. This algorithm integrates a selection mechanism into the Particle Swarm Optimization (PSO) approach and incorporates the differential vector operator from Differential Evolution (DE) to enhance the velocity update scheme within the PSO framework. Recently, a new optimization algorithm has been proposed by combining the benefits of the firefly algorithm (FA) (Yang and Slowik 2020) and Grey wolf optimization (GWO) (Mirjalili et al. 2014). In this hybrid algorithm, the solution vectors are initialised using the FA and the GWO algorithm is used to fine-tune the solution.

The main objective of this manuscript is to simplify the SISO and MIMO higher order linear continuous system by minimizing the integral square error of step response between HOSS and LOSM. The Routh approximation technique determines the denominator polynomial of the LOSM, while the numerator polynomial is determined using the proposed I-ADE algorithm. In summary, the key contributions of the paper are:

i. An improved adaptive DE algorithm (I-ADE) is proposed, which is reliable and robust.

ii. The denominator polynomial of the lower-order model is determined by the Routh approximation technique.

iii. The numerator polynomial of the lower-order model is determined by minimizing the step response ISE and using the proposed I-ADE algorithm.

iv. Numerical test cases of SISO systems and MIMO systems illustrate the superiority of the proposed method.

v. To evaluate the reliability, robustness and applicability of proposed I-ADE algorithm to different optimization problems, statistical comparative performance analysis is carried out in optimizing 23 benchmark functions using nine promising meta-heuristic algorithms.

This article has been structured into six sections. Section 1 contains the introduction and a comprehensive literature survey on model order reduction. The problem formulation of the order reduction has been given in Sect. 2. Differential evolution and algorithmic overview are discussed in Sect. 3. The proposed method of order reduction is illustrated in Sect. 4. Simulated results are provided in Sect. 5. Finally, Sect. 6 comprises the conclusion and the future scope of the work presented.

2 Problem formulation

The assessment of the proposed method's performance involves the utilization of both SISO and MIMO systems. Therefore, in this section, the SISO and MIMO systems are briefly mentioned.

2.1 Single-input single-output (SISO) system

Consider an h^{th} order large-scale dynamic HOSS transfer function described in the form of Eq. 1

$$G_h(s) = \frac{C_h(s)}{D_h(s)} = \frac{c_0 + c_1 s + c_2 s + \cdots + c_{h-1} s^{h-1}}{d_0 + d_1 s + d_2 s + \cdots + d_h s^h}, \tag{1}$$

where $c_0, c_1, c_2, \cdots, c_{h-1}$ and $d_0, d_1, \cdots, d_h$ are the coefficients of higher order stable system. The prime objective of the representation (given in Eq. 2) is to determine an order 'l' of the lower order model, which is less than the order 'h' of higher order system. Further, the lower order model must retain the significant characteristics of the higher order system in terms of its transient and steady-state parameters.

$$\hat{H}_l(s) = \frac{\hat{E}_l}{\hat{F}_l} = \frac{\hat{e}_0 + \hat{e}_1 s + \hat{e}_2 s^2 + \cdots + \hat{e}_{l-1} s^{l-1}}{\hat{f}_0 + \hat{f}_1 s + \hat{f}_2 s^2 + \cdots + \hat{f}_l s^l}, \tag{2}$$

where $\hat{e}_0, \hat{e}_1, \hat{e}_2 \cdots \hat{e}_{l-1}$ and $\hat{f}_0, \hat{f}_1, \hat{f}_2 \cdots \hat{f}_l$ are the unknown coefficients of lower order stable model.

2.2 Multi-input multi-output (MIMO) system

This paper extends the research to encompass systems with m inputs and n outputs, which are represented by the transfer matrix illustrated in Eq. 3. Such systems are commonly referred to as multi-input multi-output (MIMO) systems. These systems are frequently employed in the modelling of real time control structure of power plant model. Thus, the nth-order MIMO model is expressed as in Eq. 3.

$$G_h(s) = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \cdots & G_{1m}(s) \\ G_{21}(s) & G_{22}(s) & \cdots & G_{2m}(s) \\ \vdots & \vdots & \ddots & \vdots \\ G_{n1}(s) & G_{n2}(s) & \cdots & G_{nm}(s) \end{bmatrix}, \tag{3}$$

where $i = 1, 2, 3, \cdots, n; \quad j = 1, 2, 3, \cdots, m$

Now, individual transfer functions are expressed in corresponding numerator and denominator form as given by Eq. 4.

$$H_h(s) = \frac{1}{D_h(s)} \begin{bmatrix} C_{11}(s) & C_{12}(s) & \cdots & C_{1m}(s) \\ C_{21}(s) & C_{22}(s) & \cdots & C_{2m}(s) \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1}(s) & C_{j2}(s) & \cdots & C_{nm}(s) \end{bmatrix} \tag{4}$$

In general,$h_{ij}(s) = \frac{C_{ij}(s)}{D_h(s)}$,
where $i = 1, 2, 3, \cdots, n; \quad j = 1, 2, 3, \cdots, m$

The MIMO model of order $l (l < h)$ is represented by Eq. 5

$$\hat{H}_l(s) = \frac{1}{\hat{F}_l(s)} \begin{bmatrix} \hat{E}_{11}(s) & \hat{E}_{12}(s) & \cdots & \hat{E}_{1m}(s) \\ \hat{E}_{21}(s) & \hat{E}_{22}(s) & \cdots & \hat{E}_{2m}(s) \\ \vdots & \vdots & \ddots & \vdots \\ \hat{E}_{n1}(s) & \hat{E}_{j2}(s) & \cdots & \hat{E}_{nm}(s) \end{bmatrix}$$
$$\hat{H}_l(s) = \frac{\hat{E}_{ij}(s)}{\hat{F}_l(s)}, \tag{5}$$

where $\hat{E}_{ij}(s)$ and $\hat{F}_l(s)$ are the numerator and denominator polynomial of LOSM transfer functions, respectively. $i = 1, 2, 3, \cdots, n; \quad j = 1, 2, 3, \cdots, m$

3 Differential evolution and algorithmic overview

Storn and Price (Kenneth and Storn 1997) proposed an efficient evolutionary algorithm called differential evolution (DE) whose extended versions have shown promising results in the CEC competition series (Awad et al.

2016, 2017; Akhmedova et al. 2018). The recent review papers (Bilal et al. 2020; Das et al. 2016) present the algorithmic developments of DE algorithm. It is a stochastic search algorithm initiating chromosome set, commonly known as decision vectors or population) where each chromosome has of a set of genes, often called as decision variables or individuals. In this technique, chromosomes of a generation follows three major steps such as mutation, crossover and optimal selection iteratively to feed the population forwardly till an optimal solution is obtained.

The approach towards DE algorithm is mentioned in the stepwise manner:

1. Parameter initialization for problem statement, technique to be used and decision vector.
2. Computation of the fitness value for each decision variable.
3. This step comprises of three sub-steps.
a. Apply the mutation operator to produce the mutant vector.
b. Crossover between the target vector and mutant vector to generate trial vector.
c. Selection of the fittest between target vector and trial vector

Sub-step a to c to be followed to produce chromosomes for following generation.

4. If the stopping criteria has achieved, then continue with step-5 otherwise back to step-3.
5. Determine the most suitable fitness value as the optimal solution.

Consider the set of decision vector of nth generation is $pop_n = \left(P_n^1, P_n^2, \cdots, P_n^i, \cdots P_n^{psz}\right)$, with $P_n^i = \left(G_n^{j1}, G_n^{j2}, \cdots G_n^{jk} \cdots G_n^{jl}\right)$ for $j = 1, 2, 3 \cdots psz$, $l =$ each chromosome length,$G_n^{jk} =$ kth gene of the jth variable in nth generation. Thereafter, mutation for jth chromosome generates a mutant vector $MV_n^j = \left(MV_n^{j1}, M_n^{j2}, \cdots M_n^{jk} \cdots M_n^{jl}\right)$, which is implemented in the crossover operator with the chosen chromosome $P_n^i = \left(G_n^{j1}, G_n^{i2}, \cdots G_n^{jk} \cdots G_n^{jl}\right)$ to generate a trial vector $TV_n^j = TV_n^{j1}, TV_n^{j2}, \cdots TV_n^{jk} \cdots TV_n^{jl}$.

The DE algorithm shows changes in characteristics through the mutation and crossover operators. Different mutation operators are proposed in Price et al. (2006), Corne et al. (1999) to produce mutant vector. Different DE mutation schemes include DE/best/1 (Eq. 6), DE/rand/1 (Eq. 7), DE/rand/2 (Eq. 8), DE/best/2 (Eq. 9) and DE/target-to-best/1 (Eq. 10). Once the mutant vector is generated using any of the mutation scheme, crossover operators such as binomial (Eq. 11) or exponential (Eq. 12) is applied to obtain a trial vector. Thereafter, selection process (Eq. 13) is implemented to identify fittest chromosomes between trial vector and target vector for choosing the fittest individual among the chromosomes for the following generations.

$$\mathbf{DE/best/1} : MV_n^j = p_n^{best} + S^f\left(p_n^{rnd1} - p_n^{rnd2}\right) \tag{6}$$

$$\mathbf{DE/rand/1} : MV_n^j = p_n^{rnd1} + S^f\left(p_n^{rnd2} - p_n^{rnd3}\right) \tag{7}$$

$$\mathbf{DE/rand/2} : MV_n^j = p_n^{rnd1} + E\left(p_n^{rnd2} - p_n^{rnd3}\right) + S^f\left(p_n^{rnd4} - p_n^{rnd5}\right) \tag{8}$$

$$\mathbf{DE/best/2} : MV_n^j = p_n^{best} + S^f\left(p_n^{rnd1} - p_n^{rnd2}\right) + S^f\left(p_n^{rnd3} - p_n^{rnd4}\right) \tag{9}$$

$$\mathbf{DE/target - to - best/1} : MV_n^j = p_n^j + S^f\left(p_n^{best} - p_n^{rnd1}\right) + S^f\left(p_n^{best} - p_n^{rnd2}\right) \tag{10}$$

$$TV_n^{ji} = \begin{cases} MV_n^{ji} & if\ rand(0,1) < Cp \\ G_n^{ji} & else \end{cases} \quad \text{for } i = 1,\ 2, \ldots l$$

$$TV_n^{ji} = \begin{cases} MV_n^{ji} & h \le i \le (h+t) \\ G_n^{ji} & else \end{cases} \quad \text{for } i = 1,\ 2, \ldots l$$

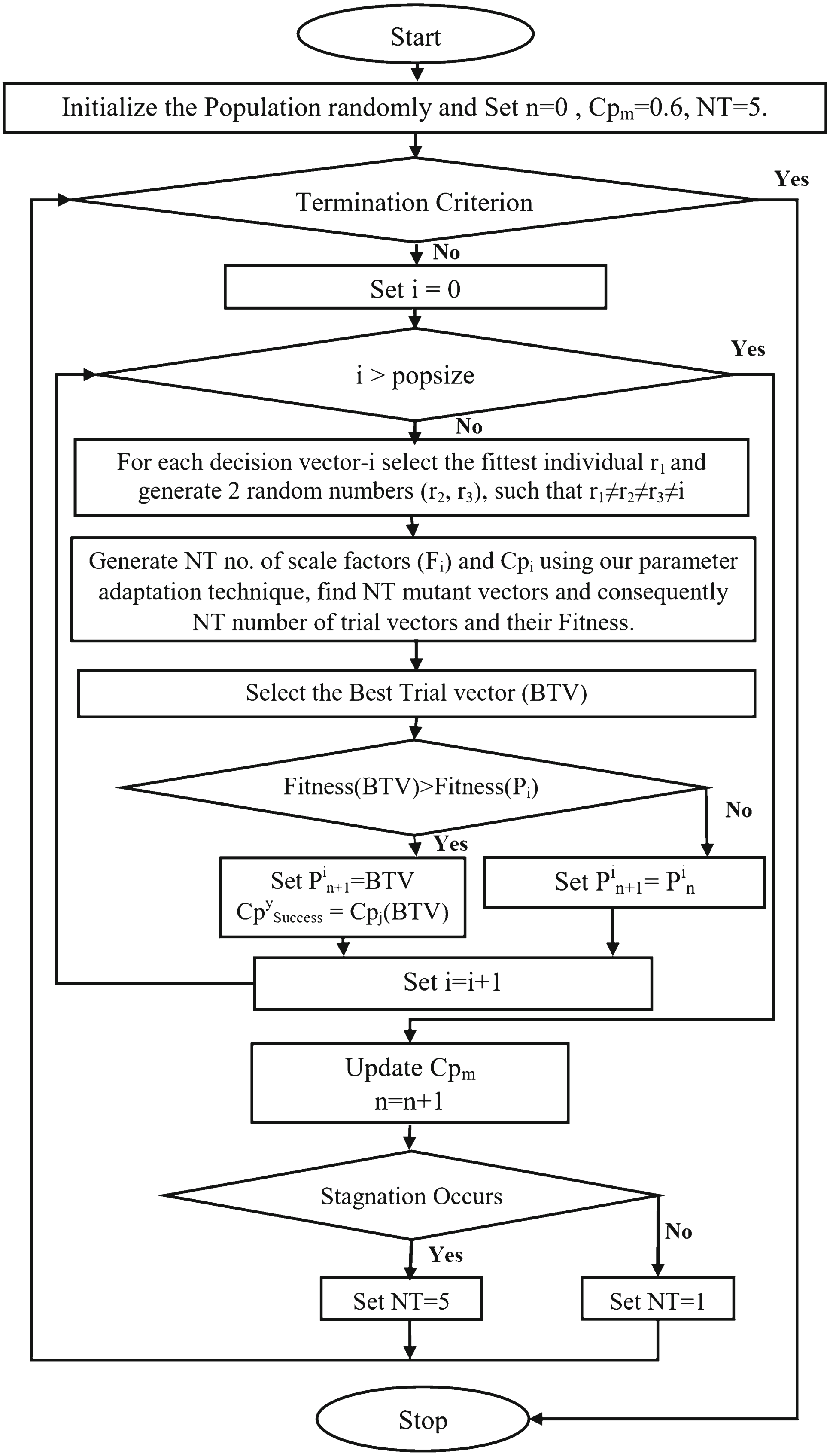
$$P_{n+1}^j = \begin{cases} TV_n^j & if\ TV_n^j > P_n^j \\ P_n^j & else \end{cases} \tag{11}$$

$$TV_n^{ji} = \begin{cases} MV_n^{ji} & h \le i \le (h+t) \\ G_n^{ji} & else \end{cases} \tag{12}$$

$$P_{n+1}^j = \begin{cases} TV_n^j & if\ TV_n^j > P_n^j \\ P_n^j & else \end{cases} \tag{13}$$

The abbreviations used in this paper are denoted as $DE/c/d$, where DE is differential evolution, c is meant for choosing the base vector in mutation, d denotes the number of difference vectors used in mutation; MV_n^j meant for jth mutant vector in nth generation;p_n^{best} is the most suitable vector among the decision vector set in nth generation;p_n^{rnd} is a randomly selected decision vector;p_n^j is the target vector; the crossover probability is denoted as Cp; S^f is termed to be the scale factor;TV_n^{ji} represents the ith gene of jth chromosome in nth generation; rand (0,1) ranges between 0 and 1 chosen randomly. It should be noted that all the selected decision vectors for mutation must be distinct to each other.

Fig. 1 Flowchart of proposed I-ADE Algorithm


Start
Initialize the Population randomly and Set n=0 , Cp$_m$=0.6, NT=5.
Termination Criterion
Yes
No
Set i = 0
i > popsize
Yes
No
For each decision vector-i select the fittest individual r$_1$ and generate 2 random numbers (r$_2$, r$_3$), such that r$_1$≠r$_2$≠r$_3$≠i
Generate NT no. of scale factors (F$_i$) and Cp$_i$ using our parameter adaptation technique, find NT mutant vectors and consequently NT number of trial vectors and their Fitness.
Select the Best Trial vector (BTV)
Fitness(BTV)>Fitness(P$_i$)
No
Yes
Set P$^i_{n+1}$=BTV Cp$^y_{Success}$ = Cp$_j$(BTV)
Set P$^i_{n+1}$= P^i_n
Set i=i+1
Update Cp$_m$ n=n+1
Stagnation Occurs
No
Yes
Set NT=5
Set NT=1
Stop

4 Methodology

This research aims to introduce a novel mixed-order reduction method applicable to continuous-time fixed coefficient systems. The proposed approach leverages the Routh Approximation method to calculate the denominator coefficients of the lower order model and employs the I-ADE optimization algorithm to determine the optimal numerator coefficients for the lower order model. The primary objective of this study is to minimize the Integral of Squared Error (ISE) between the step responses of the original higher order system and the lower order model. In this article, SISO and MIMO test systems from existing literature are selected as benchmarks to evaluate the effectiveness of the proposed technique. To gauge its performance, the proposed method is compared against established reduction methods in the literature. The paper provides step responses and Bode diagrams for both the original higher order system and its corresponding lower order models. A comprehensive comparative analysis is conducted using various error metrics and time-domain specifications. The simulation results and the comparative study collectively demonstrate that the proposed approach yields an acceptably accurate lower order model for the original higher order system.

4.1 Proposed improved adaptive differential evolution algorithm (I-ADE)

In this paper, an improved adaptive differential evolution algorithm (I-ADE) is proposed. Figure 1 presents the flowchart of the proposed I-ADE algorithm. The algorithm operates in three stages: initialization, iteration and termination. In the initialization stage, the algorithm starts with a *psz* number of randomly generated decision vectors from an upper bound and lower bound with each decision vector denoting a candidate solution of the problem. Each decision vector has a set of variables with each variable denoting an independent variable of the problem. The generation counter *n,* mean of crossover probability Cp_m and number of trial vectors upon stagnation is initialized. In the iteration stage, mutation, crossover and selection operations are repeatedly applied to move the population of decision vectors to next generation until a termination criterion is satisfied. In the proposed algorithm, DE/best/1 mutation scheme is used with a modification in generating multiple mutant vectors upon stagnation. In this algorithm, it is assumed that the population stagnates if the rate of improvement of best solution is less than 1%. The novelty of the proposed algorithm lies in its parameter adaption which is presented in Algorithm 1.

When stagnation occurs, *NT* scale factors (S^f) are employed to generate *NT* number of mutant vectors. Then, the target vector and mutant vectors undergo a crossover operation to generate multiple trial vectors. Finally, the best trial vector is used for selection operation between the trial vector and target vector. The generation of multiple trial vectors around the existing solution helps in getting out of stagnation. This is because: a) If the difference vector $P_n^{r1} - P_n^{r2}$ becomes small, or the vector converges to a tiny domain, then the S^f having larger values are used to generate a larger difference vector which ultimately assists in avoiding the suboptimal peaks causing stagnation, b) If the difference vector $P_n^{r1} - P_n^{r2}$ is large, the decision vector may jump the near-optimal solutions. But the use of S^f with smaller value may overcome this problem. This is because when a larger difference vector is multiplied with a smaller scale factor (less than one), the difference vector large value reduces significantly which will assist in making local search around the existing solution. Thereby avoids the chances of overshooting the near-optimal solution due to addition of larger difference vector value. Thus, the multiple S^f from a uniform distribution ranging (0–2) will be beneficial for both local search and global search.

When stagnation does not occur, the initial generation scale factor values are larger while the later generation scale factor values are smaller. Thus, initially, it makes exploration of the search space while in later generations, it does exploitation of the solution by making local search around the existing solutions. Thereby, a trade between exploration and exploitation is implicitly maintained.

The crossover probability (*Cp*) does not depend on any stagnation criterion. Additionally, the *Cp* is generated from a large Gaussian distribution with a fixed value of standard deviation and mean of 0.1 and Cp_m, respectively. The adaptation of Cp_m is predominantly depends on the preserving the better crossover probabilities to obtain the most suitable trial vectors. The implication of arithmetic mean during adjusting Cp_m avoids the bias of *Cp* towards smaller values (as in Eq. 14). Additionally, Cp_m uses $Cp_{success}$ along with a factor of randomness. Thereby, it not only makes use of better crossover probabilities but also adds an element of randomness into it. In this research, Gaussian distribution is used on Cp_m which improvises the chances to obtain the fittest trial vectors with the help of Cp_m values within a unity range. The value of Cp_m is initially set to 0.5, thereby gradually updated in every iteration (using Eq. 14).

$$\begin{aligned} W_{cr}^{f} &= 0.7 + 0.3 \times rand(0,1) \\ Cp_m &= W_{cr}^{f} * Cp_m + \left(1 - W_{cr}^{f}\right) \times mean(Cp_{success}), \end{aligned} \tag{14}$$

where W_{cr}^{f} is the weight factor in the range of (0.7, 1).$Cp_{success}$ is the set of *Cp* generating better trial vectors.

Table 1 Beta table

	$q_0^0 = q_0$ $q_0^1 = q_1$	$q_2^0 = q_2$ $q_2^1 = q_3$	$q_4^0 = q_4$ $q_4^1 = q_5$	$q_6^6 = q_6$ $\cdots$	$\cdots$
$\beta_1 = q_0^0/q_0^1$	$q_0^2 = q_0^2 - \beta_1 q_2^1$	$q_2^2 = q_4^0 - \beta_1 q_4^1$	$q_4^2 = q_6^0 - \beta_1 q_6^1$	$\cdots$	
$\beta_2 = q_0^1/q_0^2$	$q_0^3 = q_2^1 - \beta_2 q_2^2$	$q_2^3 = q_4^1 - \beta_2 q_4^2$	$\cdots$		
$\beta_3 = q_0^2/q_0^3$	$q_0^4 = q_2^2 - \beta_3 q_2^3$	$q_2^4 = q_4^2 - \beta_3 q_4^3$	$\cdots$		
$\beta_4 = q_0^3/q_0^4$	$q_0^5 = q_2^3 - \beta_4 q_2^4$	$\cdots$	$\cdots$		
$\beta_5 = q_0^4/q_0^5$	$q_0^6 = q_2^4 - \beta_5 q_2^5$	$\cdots$			
$\beta_6 = q_0^5/q_0^6$	$\cdots$	$\cdots$			

Once the best trial vector is obtained, it is used in selection to move the candidate solution with better fitness to next generation. Once the termination criteria are satisfied, the decision vector of final generation with best fitness value is used as the solution of the problem. The proposed algorithm is an improved and faster version of ADE-ANNT (Panigrahi and Behera 2020) algorithm. Hence, it is named as I-ADE.

Algorithm 1 Pseudo-code of parameter adaptation

```
If Stagnation
        Generate $NT(>1)$ scale factors $S^f = \left(S_1^f, S_2^f, S_3^f, \cdots S_{NT}^f\right)$
        $S^f = rand(1, NT) * Max\_S_m^f$
Else
        Generate $v(=1)$ scale factor $\left(S_1^f\right) S^f = \left(S_1^f\right)$
        $S_1^f = a - epoch * ((a) / Max\_iteration)$
End of if
Crossover probability generation
$Cp = Gaussianrnd(Cp_m, 0.1)\ with\ Cp \in [0;1]$
```

Here rand (1, NT) represents NT numbers from a uniform distribution space in a span of (0, 1), Gaussianrnd (Cp_m, 0.1) is generated from a Gaussian distribution having standard deviation and mean of 0.1 and Cp_m, respectively. In this algorithm, when there is less than 1% change in the best solution between two generations then only multiple trial vectors are generated. Otherwise, only one trial vector will be generated and the computational efficiency of the algorithm is same to other algorithms. Additionally, one can employ parallel computing to compute the multiple trial vectors parallel which will again diminish the questions on computational complexity and applicability of the proposed algorithm in practice. Of course, the number of trial vectors and choice of stagnation criteria maintains a trade between exploration and exploitation which need to be carefully chosen based on the problem in hand.

4.2 Application of Routh approximation for computation of denominator polynomials of lower-order model

Routh approximation (RA) is initially highlighted by Hutton et al. (1975) to reduce the order of HOSS to its corresponding LOSM in the frequency domain. The technique is quite simple to use, thereby used for preserving the stable approximants in LOSM, only if the HOSS is termed to be stable. This technique is suitable to evaluate denominator coefficients of the LOSM depending on the β-table (as in Table 1).

Step-1: Compute the reciprocal transformation of HOSS using Eq. 15.

$$\overline{Q}_h(s) = \frac{1}{s} Q_h\left(\frac{1}{s}\right) \tag{15}$$

Step-2: Considering the coefficients of $\overline{Q}(s)$, determine the values of $\beta_1, \beta_2, \cdots \beta_n$, by using Beta table.

Step-3: Obtain lth-order denominator polynomial $\hat{Q}_l$ considering the Beta coefficients of $\overline{Q}_h(s)$.

For second order, $\hat{Q}_2(s) = \beta_1 \beta_2 s^2 + \beta_2 s + 1$.

For third order, $\hat{Q}_3(s) = \beta_1 \beta_2 \beta_3 s^3 + \beta_2 \beta_3 s^2 + (\beta_1 + \beta_3) s + 1$.

Table 2 Parameter configuration of different optimization algorithms

Algorithm	Parameters	Values
I-ADE	F	(0,1)
	Gaussianrnd	$(Cp_m, 0.1)$
	Mean (Cp_m)	0.6
	Standard deviation (Stndv)	0.1
ADE-ANNT	v	$(0, 1)$
	Cauchyrnd	$(0, 1)$
	Weight factor W_{cr}^{f}	$(0.8, 1)$
	Scale parameter	0.1
	Location parameter F_m	0.5
	Gaussianrnd	$(Cp_m, 0.1)$
	Mean (Cp_m)	0.6
	Standard deviation (Stndv)	0.1
DE	rnd	(0,1)
	Scaling factor	0.5
	Crossover factor	0.8
PSO	Inertia weight	$1 \rightarrow 0.7$
	Random velocity rule weight (lower Limit)	0
	Random velocity rule weight (higher limit)	2
GWO	a	$2 \rightarrow 0$(decreases linearly)
	A	$2 \times a \times rnd_1 - a$
	C	$2 \times rnd_2$
ALO	rnd	(0.1)
WOA	a	$2 \rightarrow 0$(Decreases linearly)
	a_2	$-1 \rightarrow -2$
	A	$2 \times a \times rnd_1 - a$
	C	$2 \times rnd_2$
SSA	C_1	$2 \times exp(-4/T^2)$
	C_2	rnd
	C_3	rnd
SCA	a	2
	r_1	$a \rightarrow 0$(decreases linearly
	r_2	$2 \times \Pi \times rnd$
	r_3	rnd
	r_4	rnd
GOA	$C_{\max}$	1
	$C_{\min}$	0.00004
MFO	a	$-1 \rightarrow -2$(linearly decreases)
	b	1
DF	r	$\frac{(ub-lb)}{4} + (ub - lb) \times \left(\frac{t}{T}\right)^2$
	w	$0.9 - \frac{t}{T} \times (0.9 - 0.4)$
	Distraction weight (dt_c)	$0.1 \times \frac{t}{T} \times \left(\frac{0.1-0}{2}\right)$
	Separation weight (sw)	$2 \times rnd \times (dt\_c)$
	Alignment weight (a_w)	$2 \times rnd \times (dt\_c)$
	Cohesion weight (c_w)	$2 \times rnd \times (dt\_c)$
	Food attraction weight (f_w)	$2 \times rnd$

Table 3 Parameter configuration of the Unimodal benchmark Functions

Sl. No.	Function	Dimension	Range	$f_{\min}$
1	$f_1(u) = \sum_{j=1}^{n} u^2$	30	[− 100, 100]	0
2	$f_2(u) = \sum_{j=1}^{n} \lvert u_j \rvert + \prod_{j=1}^{n} \lvert u_j \rvert$	30	[− 10,10]	0
3	$f_3(u) = \sum_{j=1}^{n} \left(\sum_{i=1}^{j} u_j \right)^2$	30	[− 100, 100]	0
4	$f_4(u) = max_j \{ \lvert u_j \rvert, 1 \le j \le n \}$	30	[− 100, 100]	0
5	$f_5(u) = \sum_{j=1}^{n-1} \left[100 \left(u_{j+1} - u_j^2 \right)^2 - (u_j - 1)^2 \right]$	30	[− 30, 30]	0
6	$f_6(u) = \sum_{j=1}^{n} (u_j + 0.5)^2$	30	[− 100, 100]	0
7	$f_7(u) = \sum_{j=1}^{n} j u_j^4 + rand[0, 1]$	30	[− 1.28, 1.28]	0

Equation 16 represents the generalized form of denominator polynomial.

$$\overline{Q}_l = \beta_l s \overline{Q}_{l-1}(s) + \overline{Q}_{l-2}(s) \tag{16}$$

4.3 Application of proposed I-ADE algorithm for computation of numerator polynomials of lower order model

The proposed I-ADE algorithm is used to compute the numerator polynomials of lower order model. For this each numerator polynomial is considered as a decision variable and all numerator polynomials form a decision vector. Then, the mutation, crossover and selection scheme of I-ADE algorithm employing the proposed parameter adaption scheme is repeatedly applied until the termination criteria are satisfied. Note that the objective function is a minimization one and is used to minimize the integral square error (ISE) between the step responses of original higher order system and the reduced model. Once the I-ADE algorithm terminates, the decision vector of the final generation with the best fitness (lowest ISE) is selected as the values of numerator polynomials of lower order model.

5 Simulated results and discussion

5.1 Experimental evaluation of proposed I-ADE algorithm for optimizing benchmark functions

The proposed I-ADE algorithm's performance is evaluated using benchmark functions, both unimodal and multimodal, as detailed in Mirjalili's work (El-Kenawy et al. 2022). This includes information on the dimensionality (Dim), ranges, and optimal solutions (fmin) for these benchmark functions. To assess the effectiveness of the proposed I-ADE algorithm, it is compared with several well-known meta-heuristic algorithms. Specifically, these algorithms are the original SCA, PSO, ALO, DE, MFO, and DF. Since the proposed I-ADE method is a modification of the DE (Differential Evolution) algorithm, the original DE algorithm is chosen as a comparison benchmark. Additionally, PSO, ALO, and DE have recently been applied in the context of model order reduction, making them relevant for comparison. Two other popular meta-heuristic algorithms, MFO and DF, are also included as comparative methods. The control parameters for SCA, PSO, ALO, and MFO remain the same. However, for the DE algorithm, a specific strategy called DE/best/1/bin is employed, with control parameter values F set to 0.5 and Cr set to 0.6. All algorithms, including the proposed I-ADE, are run independently 20 times, each with a swarm size of 20 search agents (also referred to as individuals or chromosomes) and a maximum of 400 iterations. For the convenience of interested readers, the algorithm parameters are summarized in Table 2.

In this field, it is a common practice to evaluate the performance of algorithms using a set of well-established mathematical functions that have known global optima. Computational experiments are typically conducted on a set of 23 benchmark functions, which can be categorized into three groups: unimodal functions, multimodal functions with varying numbers of local optima, and fixed-dimension multimodal functions. The details of these three categories of functions are mentioned below.

- Benchmark functions 1–7 are unimodal functions (as in Table 3). These functions are primarily used to assess how well algorithms can exploit the search space and

Table 4 Parameter configuration of the flexible dimension multimodal benchmark functions

Sl. No.	Function	Dimension	Range	$f_{\min}$
8	$f_8(u) = \sum_{j=1}^{n} -u\,sin\left(\sqrt{u_j}\right)$	30	[− 100, 100]	0
9	$f_9(u) = \sum_{j=1}^{n} \left[u_i^2 - 10cos\left(2\Pi u_j\right) + 10\right]$	30	[− 5.12, 5.12]	0
10	$f_{10}(u) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{j=1}^{n} u_j^2}\right) - exp\left(\frac{1}{n}\sum_{j=1}^{n} cos\left(2\Pi u_j\right)\right) + 20 + e$	30	[− 32,32]	0
11	$f_{11}(u) = \frac{1}{4000}\sum_{j=1}^{n}\left[u_j^2 - \prod_{j=1}^{n} cos\left(\frac{u_j}{\sqrt{j}}\right) + 1\right]$	30	[− 600, 600]	0
12	$f_{12}(u) = \frac{\Pi}{n}\left\{10sin(\Pi y_1) + \sum_{j=1}^{n-1}\left(y_j - 1\right)^2\left[1 + 10sin^2\left(\Pi y_{j+1}\right)\right] + (y_n + 1)^2\right\}$ $+\sum_{j=1}^{n} t\left(u_j, 10, 100, 4\right)$ $y_j = 1 + \frac{u_j + 1}{4}$ $t(u_i, a, k, m) = \begin{cases} k\left(u_j - a\right)^m & u_j > a \\ 0 & -a < u_j < a \\ k\left(-u_j - a\right)^m & u_j < -a \end{cases}$	30	[− 50, 50]	0
13	$f_{13}(u) = 0.1\left\{\begin{array}{c} sin^2(3\Pi) + \sum_{j=1}^{n}\left(u_j - 1\right)^2\left[1 + sin^2(3\Pi u_i + 1)\right] \\ +(u_n - 1)^2\left[1 + sin^2(2\Pi u_n)\right] + \sum_{j=1}^{n} t\left(u_j, 5, 100, 4\right) \end{array}\right\}$	30	[− 50, 50]	0

find the global optimum. Unimodal functions have a single global optimum.

- Benchmark functions 8–13 are flexible dimension multimodal functions (as in Table 4). These functions are characterized by having multiple local optima, and the number of local optima tends to increase with the number of dimensions. These functions are utilized to evaluate the ability of algorithms to explore the search space effectively and locate multiple local optima.
- Benchmark functions 14–23 are fixed-dimension multimodal benchmark functions (as in Table 5). They are designed to test the exploration capabilities of algorithms in situations where the dimensionality of the optimization problem is fixed, and multiple local optima exist.

In summary, these benchmark functions serve as a standardized way to test and compare the performance of algorithms in terms of their ability to exploit global optima, explore the search space efficiently, and handle optimization problems with various characteristics and dimensions. The global optimum values of the benchmark functions are also given in Tables 3, 4 and 5 to give an idea to the readers about the performances of the proposed algorithm.

Table 6 provides a summary of the mean and standard deviation (StDev) results obtained from the suggested I-ADE algorithm as well as the compared algorithms across the benchmark functions (f_1 to f_{23}). These results reveal that the I-ADE algorithm has achieved impressive performance, even obtaining zero values for both mean and standard deviation in some cases. Additionally, it outperforms the compared algorithms in various instances. Table 7, on the other hand, presents the outcomes of the Wilcoxon signed-rank test (WSRT) conducted on the 23 sample functions (f_1 to f_{23}). Since meta-heuristic algorithms exhibit stochastic behaviour, it is essential to employ statistical tests to make conclusive assessments. The WSRT, conducted at a 95% significance level, is used to determine whether a method is inferior ($-$), superior ($+$), or equivalent ($\approx$) to the proposed I-ADE algorithm. These statistical tests aid in providing a rigorous and reliable evaluation of the algorithms, allowing for comparisons and identifying the strengths and weaknesses of each method. These results are crucial in making informed

Table 5 Parameter configuration of the fixed dimension multimodal benchmark functions

Sl. No.	Function	Dimension	Range	$f_{\min}$
14	$f_{14}(u) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} \left(u_i - a_{ij}\right)^6}\right)^{-1}$	2	[− 50, 50]	1
15	$f_{15}(u) = \sum_{j=1}^{11} \left[a_j - \frac{u_j\left(b_j + b_j u_2\right)}{b_j^2 + b_j u_3 + u_4}\right]^2$	4	[− 5, 5]	0.00030
16	$f_{16}(u) = 4u_1^2 + 2.1u_1^4 + \frac{1}{3}u_1^6 + u_1 u_2 - 4u_2^2 + 4u_2^4$	2	[− 5, 5]	− 1.0316
17	$f_{17}(u) = \left(u_2 - \frac{5.1}{4\pi^2}u_1^2 + \frac{5}{\pi}u_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos u_1 + 10$	2	[− 5, 5]	0.398
18	$f_{18}(u) = \left[1 + (u_1 + u_2 + 1)^2\left(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right)\right]$ $\times\left[30 + (2u_1 - 3u_2)^2 \times \left(18 - 32u_1 + 12u_1^2 + 48u_2 - 36u_1u_2 + 27u_2^2\right)\right]$	2	[− 2, 2]	3
19	$f_{19}(u) = -\sum_{j=1}^{4} C_j exp\left(-\sum_{i=1}^{3} a_{ji}\left(u_j - P_{ji}\right)^2\right)$	3	[0, 1]	− 3.86
20	$f_{20}(u) = -\sum_{j=1}^{4} C_j exp\left(-\sum_{i=1}^{6} a_{ji}\left(u_j - P_{ji}\right)^2\right)$	6	[0, 1]	− 3.32
21	$f_{21}(u) = -\sum_{j=1}^{5}\left[\sum_{i=1}^{4}\left(u_i - a_{ji}\right)^2 + c_j\right]^{-1}$	4	[0, 10]	− 10.1532
22	$f_{22}(u) = -\sum_{j=1}^{7}\left[\sum_{i=1}^{4}\left(x_i - a_{ji}\right)^2 + c_j\right]^{-1}$	4	[0, 10]	− 10.4028
23	$f_{23}(u) = -\sum_{j=1}^{10}\left[\sum_{i=1}^{4}\left(x_i - a_{ji}\right)^2 + c_j\right]^{-1}$	4	[0, 10]	− 10.5363

decisions about the algorithm's suitability for optimization tasks.

Table 8 presents the frequency of statistical significance comparative algorithms than proposed I-ADE algorithm than other in optimizing 23 benchmark functions. The ADE-ANNT algorithm provides inferior, superior and equivalent results than proposed I-ADE algorithm in 10, 1 and 12 benchmark functions, respectively. Table 8 reveals that the proposed I-ADE algorithm has demonstrated statistically superior performance when compared to several other algorithms, including the original ADE, DE, GWO, WOA, PSO, MFO, ALO, DF, and SCA. This statistical analysis provides strong evidence of the effectiveness of the I-ADE algorithm across various benchmark functions. To rank the meta-heuristic algorithms comprehensively, taking into account all 23 functions, a Friedman and Nemenyi hypothesis test is conducted at a 95% confidence level, and the results are depicted in Fig. 2. It is evident from Fig. 2 that the proposed I-ADE algorithm attains the lowest mean rank (51.34), which is significantly lower than the mean ranks of other algorithms by at least a critical distance of 2.8. ADE-ANNT acquires the second rank, SCA acquires the worst rank, and so on. The GWO and PSO algorithms are statistically equivalent to each other, since the difference in mean rank is less than the critical distance 2.8.

5.2 Performance evaluation of proposed method for optimizing the parameters of numerator and denominator polynomial of SISO system

In order to evaluate the performance of proposed method for model order reduction of SISO system, the fourth-order system (Vishwakarma and Prasad 2008) is considered which is expressed in Eq. 17. Equation 18 presents the reciprocal of denominator polynomial of HOSS. Then Routh approximation is used and the computed Beta table for denominator polynomial of the fourth-order SISO system is presented in Table 9.

$$G_h(s) = \frac{C_h(s)}{D_h(s)} = \frac{s^3 + 7s^2 + 24s + 24}{(s+1)(s+2)(s+3)(s+4)} \tag{17}$$

$$\tilde{D_h}(s) = s^h D_h\left(\frac{1}{s}\right) = 1 + 10s + 35s^2 + 50s^3 + 24s^4 \tag{18}$$

The second-order denominator polynomial is obtained by using Table 9 and is presented in Eq. 19.

$$\tilde{F_l}(s) = 1 + \beta_1 s + \beta_1\beta_2 s^2 = 1 + 1.656s + 0.7946s^2 \tag{19}$$

Table 6 Mean and standard deviation (StDev) of the suggested and compared algorithms on the benchmark Unimodal and multimodal functions

		I-ADE	ADE	DE	PSO	ALO	MFO	GWO	WOA	SCA	DF
f_1	Mean	9.06325×10^{-42}	1.35197×10^{-36}	1.16717×10^{-24}	6.21016×10^{-18}	3.13311×10^{-08}	4.8999×10^{-10}	2.48694×10^{-45}	2.77324×10^{-62}	1.23052×10^{-09}	5.24256854
	StDev	2.5141×10^{-41}	3.9587×10^{-36}	5.21975×10^{-24}	2.71977×10^{-17}	3.62548×10^{-08}	7.2166×10^{-10}	5.67213×10^{-45}	9.31702×10^{-62}	1.62901×10^{-09}	4.970987828
f_2	Mean	2.53016×10^{-23}	2.05416×10^{-21}	5.1245×10^{-19}	9.52013×10^{-13}	0.895916389	0.50000046	2.61258×10^{-26}	4.38685×10^{-43}	7.71627×10^{-08}	1.365764606
	StDev	6.08898×10^{-23}	1.7808×10^{-21}	2.29077×10^{-18}	2.11153×10^{-12}	1.378059917	2.236067873	4.04464×10^{-26}	7.17625×10^{-43}	9.17001×10^{-08}	1.029221885
f_3	Mean	2.64234×10^{-10}	1.48178×10^{-11}	8.793721×10^{-11}	0.000496182	0.351540736	750.4445685	1.07951×10^{-19}	248.1686894	1.664894558	162.6721456
	StDev	5.81314×10^{-10}	3.68478×10^{-11}	3.6572×10^{-10}	0.001029299	0.458707492	1831.592025	2.59799×10^{-19}	518.6546689	4.204515001	184.9021001
f_4	Mean	2.88452×10^{-13}	5.3341×10^{-10}	0.121828863	0.000728418	0.014178538	3.381642703	1.81782×10^{-14}	1.927008048	0.005071799	3.678582682
	StDev	1.20934×10^{-10}	1.53387×10^{-09}	0.314048332	0.001060164	0.022941205	6.063992616	3.0809×10^{-14}	2.809490471	0.008698713	2.171114232
f_5	Mean	5.304417428	1.453230154	7.715190499	5.555401418	104.1949874	13,728.13022	6.577336406	7.172473069	7.497975158	794.0626271
	StDev	2.202799751	1.955088409	11.21283197	2.633066566	191.0962031	32,881.61329	0.488224239	0.402008138	0.335356891	1044.63709
f_6	Mean	0	0	3.03501×10^{-25}	3.64434×10^{-10}	5.54855×10^{-08}	5.54116×10^{-10}	0.025140414		0.51899099	6.077423206
	StDev	0	0	1.3573×10^{-24}	1.1599910^{-18}	6.35496×10^{-08}	7.38719×10^{-10}	0.112404801	0.001888459	0.212559761	8.551957673
f_7	Mean	0.000512483	0.001494187	0.00372814	0.002990713	0.036514478	0.01036624	0.000603465	0.003223674	0.00441534	0.034303937
	StDev	0.000247422	0.000539375	0.002342327	0.001274593	0.021655122	0.007263619	0.000396856	0.003336385	0.004377442	0.019980126
f_8	Mean	− 3659.527194	− 4024.015204	− 3696.761786	− 2491.020339	− 2240.387349	− 3249.383116	− 2586.961473	− 3286.662605	− 2109.707925	− 2769.59734
	StDev	202.8486512	135.3135637	210.2426585	287.0159014	348.6007451	341.6182045	364.2712826	632.6195665	152.4474068	335.0156494
f_9	Mean	0.800136186	2.387901233	5.621516154	13.28268905	20.34685434	29.91617495	1.267079901	1.56743703	2.029185656	24.60123589
	StDev	0.947762364	1.527802305	2.928039773	5.980860387	11.1705259	15.48397197	2.386670695	7.009791499	5.152302536	10.93146608
f_{10}	Mean	0.057757425	4.44089×10^{-15}	0.222991031	5.20682×10^{-11}	0.164775495	0.968413221	1.15463×10^{-14}	5.32907×10^{-14}	1.003656546	3.349475319
	StDev	0.258299058	0	0.711997938	1.29017×10^{-10}	0.506644645	4.330839548	3.04963×10^{-14}	2.79385×10^{-15}	4.461102403	1.170878074
f_{11}	Mean	0.019171483	0.054742247	0.069745841	0.097966644	0.187362181	0.162129597	0.013921282	0.042589472	0.086906935	0.673092708
	StDev	0.029057319	0.032523143	0.063104413	0.048692131	0.094385621	0.072083877	0.017282567	0.068640453	0.110177309	0.460537012
f_{12}	Mean	0.015550353	0.031100706	0.033872055	1.33724×10^{-20}	3.024766196	0.186594643	0.002950425	0.112184532	0.107606981	1.903995372
	StDev	0.069543293	0.095725954	0.095580389	5.21381×10^{-20}	2.844908184	0.234460989	0.009588988	0.446508563	0.052776012	2.303474444
f_{13}	Mean	1.34978×10^{-32}	2.32419×10^{-32}	0.016810818	0.001098737	0.004852354	0.002197474	0.009073394	0.060232504	0.360143509	1.672482374
	StDev	2.80801×10^{-48}	4.24304×10^{-32}	0.070089106	0.00338184	0.007322338	0.00450912	0.028040658	0.062391297	0.083486025	3.249566358
f_{14}	Mean	0.998003838	0.998003838	1.047705191	4.291498924	2.284734143	3.071162232	5.4418184	2.767769149	1.396548783	1.19661161
	StDev	0	0	0.222271208	3.298178744	1.570970994	2.248994491	4.945072724	2.932966088	0.813379713	0.519308077
f_{15}	Mean	0.001739912	0.001469242	0.003385407	0.000623482	0.002282652	0.0009596	0.003433579	0.00100061	0.00092733	0.004598114
	StDev	0.004576179	0.004464701	0.007320857	0.000439637	0.004355519	0.00038453	0.007297581	0.000621977	0.000317631	0.012533426
f_{16}	Mean	− 1.031628453	− 1.031628453	− 1.031628453	− 1.031628453	− 1.031628453	− 1.031628453	− 1.031628416	− 1.031628452	− 1.031545784	− 1.031628453
	StDev	2.27813×10^{-16}	2.22045×10^{-16}	1.34776×10^{-16}	2.10033×10^{-16}	1.17842×10^{-13}	2.27813×10^{-16}	2.62305×10^{-16}	1.64724×10^{-06}	0.000106662	2.85619×10^{-10}
f_{17}	Mean	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.397887358	0.397890308	0.397895449	0.400651768	0.397887658
	StDev	0	0	0	0	2.41864×10^{-13}	0	4.05736×10^{-06}	1.10813×10^{-05}	0.002388078	1.33599×10^{-05}
f_{18}	Mean	3	3	3	3	3	3	3.000099829	3.000143099	3.000077443	3
	StDev	3.05643×10^{-16}	7.82563×10^{-16}	1.05387×10^{-16}	$1.71087\ 10^{-15}$	7.30591×10^{-13}	1.23524×10^{-13}	0.000149553	0.000352943	0.000118673	7.29309×10^{-09}
f_{19}	Mean	− 3.862782148	− 3.862782148	− 3.862782148	− 3.862782148	− 3.862782148	− 3.862782148	− 3.861997663	− 3.843854051	− 3.853720581	− 3.862692359
	StDev	2.27813×10^{-15}	2.27813×10^{-16}	2.27813×10^{-16}	2.21577×10^{-15}	7.42755×10^{-12}	2.27813×10^{-13}	0.002041211	0.040085011	0.00190779	0.000193813
f_{20}	Mean	− 3.274437923	− 3.244714643	− 3.274437923	− 3.262548611	− 3.285447935	− 3.25884211	− 3.270575461	− 3.134974943	− 2.877629988	− 3.257005438
	StDev	0.059758618	0.058181596	0.059758618	0.060990884	0.057298296	0.066323321	0.073677857	0.259406909	0.414243044	0.085479438
f_{21}	Mean	− 8.777407973	− 4.648094851	− 4.877470275	− 6.267615736	− 5.975444905	− 6.765681612	− 9.519991448	− 7.316564846	− 2.33937816	− 7.101324725
	StDev	2.864247329	2.98185516	2.912031531	3.392546434	2.619801424	3.567280792	1.981131135	2.56793894	1.656952951	2.889206127
f_{22}	Mean	− 8.874442075	− 6.953998676	− 7.561411658	− 8.540203532	− 5.423834292	− 8.371676773	− 10.40046959	− 6.93372788	− 3.224120987	− 8.545373392

Table 6 (continued)

		I-ADE	ADE	DE	PSO	ALO	MFO	GWO	WOA	SCA	DF
	StDev	3.136413881	3.322468837	3.608284939	3.316094129	3.126778749	3.213401168	0.001132934	3.301924921	1.45535133	2.588104468
f_{23}	Mean	− 9.795626877	− 7.403499526	− 3.835426803	− 7.877671731	− 5.439395074	− 6.876958981	− 10.53348779	− 8.366419454	− 3.928792249	− 6.343665342
	StDev	2.291585859	3.94811422	− 4.211740301	3.758561711	3.531618657	3.493389909	0.001538415	3.014419469	1.207042893	3.319765607

For computing the denominator polynomial of the desired reduced LOSM, the reciprocal transformation of Eq. 19 is taken and it is presented in Eq. 20.

$$\hat{F_l}(s) = s^2{+}1.6555s + 0.7946 \tag{20}$$

After computation of denominator polynomial mentioned in Eq. 20, the numerator polynomial of the LOSM is computed using I-ADE by setting population size to 50, the maximum number of iterations to 400 and minimizing the ISE between HOSS and LOSM. The numerator polynomial is presented in Eq. 21.

$$\hat{E_l}(s) = 0.8242s + 0.7889. \tag{21}$$

Thus, the obtained second-order transfer function stable model using the proposed technique is presented in Eq. 22.

$$\hat{H_l}(s) = \frac{\hat{E_l}}{\hat{F_l}} = \frac{0.8242s + 0.7889}{s^2{+}1.6555s + 0.7946}. \tag{22}$$

Figures 3 and 4 demonstrate that the error gap between the step and frequency responses of the HOSS and the LOSM obtained through the proposed method is remarkably small. This observation indicates that the LOSM effectively captures the essential characteristics of the original system. Equation 22 represents a second-order steady-state model for a fourth-order system, and it is worth noting that this reduced model encapsulates all the significant attributes of the original system. Furthermore, Table 10 provides evidence that the reduced-order model obtained through the proposed method maintains the stability of the original system while achieving impressive results in various error metrics, including integral square error (ISE), integral absolute error (IAE), integral time square error (ITSE), mean square error (MSE), and root mean square error (RMSE). These error indices demonstrate that the proposed LOSM performs better than other well-known reduction techniques found in the literature. As a result, it can be concluded that the simplified LOSM derived from the proposed method is the closest match to the original system's behaviour. This makes it highly useful for dynamic simulations of large-scale systems, as it reduces computational complexity, simulation time, and memory requirements while maintaining accuracy and stability.

5.3 Performance evaluation of proposed method for optimizing the parameters of numerator and denominator polynomial of MIMO system

Consider a sixth order MIMO system (Padhy et al. 2021; Bistritz and Shaked 1984) having two inputs two outputs (TITO) described by Eq. 23

Table 7 Wilcoxon signed-rank test results on unimodal and multimodal functions indicating the inferior (−), superior (+), or equivalent (≈) algorithm in comparison to the proposed I-ADE algorithm

	I-ADE	DE	PSO	ALO	MFO	GWO	WOA	SCA	DF
F1	−	≈	−	−	−	−	+	−	−
F2	−	≈	−	−	−	+	+	−	−
F3	+	+	−	−	−	−	−	−	−
F4	−	−	−	−	−	≈	−	−	−
F5	−	≈	≈	−	−	−	−	−	−
F6	≈	≈	−	−	−	−	−	−	−
F7	−	−	−	−	−	≈	−	−	−
F8	−	≈	−	−	−	−	−	−	−
F9	−	−	−	−	−	−	+	≈	−
F10	≈	≈	−	−	−	−	≈	−	−
F11	−	−	−	−	−	≈	≈	−	−
F12	≈	≈	−	−	−	−	−	−	−
F13	≈	−	−	−	−	−	−	−	−
F14	≈	≈	−	−	−	−	−	−	≈
F15	≈	≈	≈	−	≈	≈	−	−	−
F16	≈	≈	≈	≈	≈	−	−	−	≈
F17	≈	≈	≈	≈	≈	−	−	−	≈
F18	≈	≈	≈	≈	≈	−	−	−	≈
F19	≈	≈	≈	≈	≈	−	−	−	−
F20	≈	≈	≈	≈	≈	≈	−	−	≈
F21	−	−	−	−	≈	≈	≈	−	≈
F22	≈	≈	≈	−	≈	≈	−	−	≈
F23	≈	−	≈	−	≈	−	−	−	−

Table 8 Frequency summary of statistical significance of comparative meta-heuristic algorithms than proposed I-ADE algorithm

	Inferior (−)	Superior (+),	Equivalent (≈)
ADE-ANNT	10	1	12
DE	7	1	15
PSO	14	0	9
ALO	18	0	5
MFO	14	0	9
GWO	15	01	7
WOA	17	03	03
SCA	22	00	01
DF	16	00	07

$$H_h(s) = \frac{C_h(s)}{D_h(s)} = \frac{1}{D_h(s)}\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} \dfrac{2(s+5)}{(s+1)(s+10)} & \dfrac{(s+4)}{(s+2)(s+5)} \\ \dfrac{(s+10)}{(s+1)(s+20)} & \dfrac{(s+6)}{(s+2)(s+3)} \end{bmatrix}. \tag{23}$$

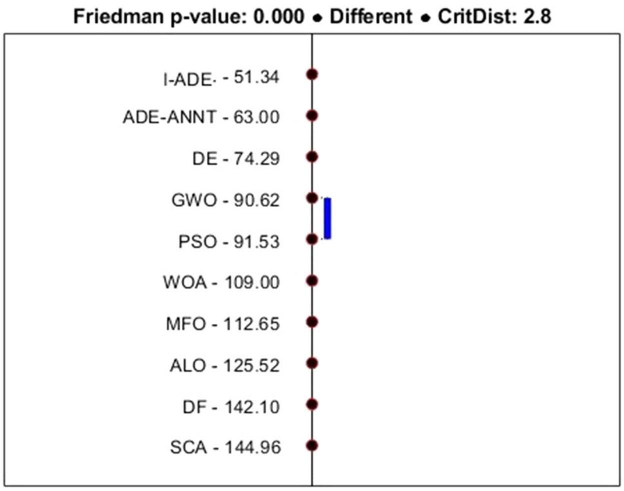


Fig. 2 Mean rank of meta-heuristic algorithms using 23 benchmark functions with P value = 0.0 and critical distance = 2.8

The denominator and numerator polynomial of the MIMO system is represented by Eqs. 24 and 25, respectively.

Table 9 Computed beta table for denominator polynomial of fourth-order SISO system

	24	35	1
	50	10	
β_1 = 0.49	30.23	1	
β_2 = 1.656	8.346	0	

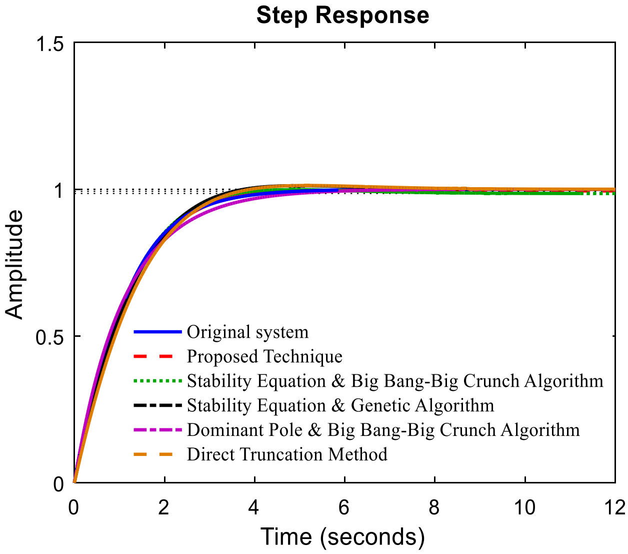


Fig. 3 Step response of HOSS and LOSMs for SISO system

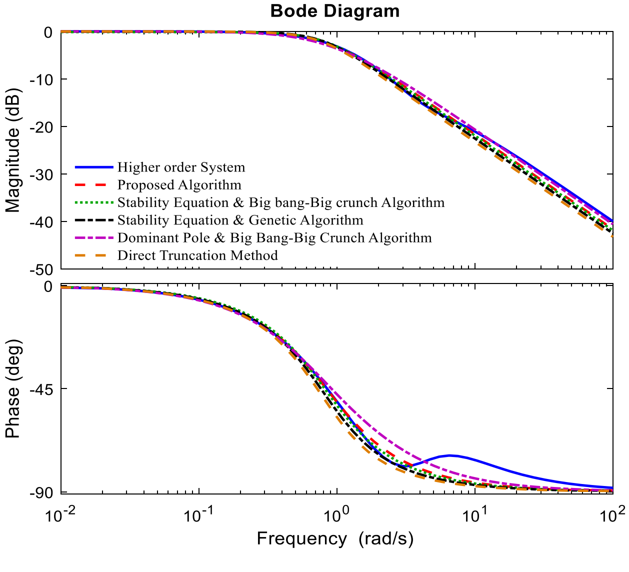


Fig. 4 Frequency response of HOSS and LOSMs for SISO system

$$D_h(s) = s^6 + 41s^5 + 571s^4 + 3491s^3 + 10060s^2 + 13100s + 6000 \tag{24}$$

$$\begin{aligned} C_{11}(s) &= 2s^5 + 70s^4 + 762s^3 + 3610s^2 + 7700s + 6000, \\ C_{12}(s) &= s^5 + 38s^4 + 459s^3 + 2182s^2 + 4160s + 2400, \\ C_{21}(s) &= s^5 + 30s^4 + 331s^3 + 1650s^2 + 3700s + 3000, \\ C_{22}(s) &= s^5 + 42s^4 + 601s^3 + 360s^2 + 9100s + 6000, \end{aligned} \tag{25}$$

By using beta table, the lower-order denominator polynomial is represented by Eq. 26.

$$\hat{F}_l(s) = s^2 + 1.548s + 0.709. \tag{26}$$

The numerator coefficients of the LOSMs are computed using I-ADE algorithm by minimizing the integral square error between higher-order system and lower-order model which are represented in Eq. 27.

$$\begin{aligned} \hat{E}_{11}(s) &= 0.834s + 0.707 \\ \hat{E}_{12}(s) &= 0.453s + 0.281 \\ \hat{E}_{21}(s) &= 0.444s + 0.354 \\ \hat{E}_{22}(s) &= 1.325s + 0.703 \end{aligned} \tag{27}$$

For transfer function $H_{11}(s)$.

The transfer matrix of second-order lower order model obtained by using proposed method which are represented by Eq. 28.

$$\begin{aligned} \hat{H}_{11}(s) &= \frac{\hat{E}_{11}(s)}{\hat{f}_l(s)} = \frac{0.838s + 0.707}{s^2 + 1.548s + 0.709} \\ \hat{H}_{12}(s) &= \frac{\hat{E}_{12}(s)}{\hat{f}_l(s)} = \frac{0.453s + 0.281}{s^2 + 1.548s + 0.709} \\ \hat{H}_{21}(s) &= \frac{\hat{E}_{21}(s)}{\hat{f}_l(s)} = \frac{0.444s + 0.354}{s^2 + 1.548s + 0.709} \\ \hat{H}_{22}(s) &= \frac{\hat{E}_{21}(s)}{\hat{f}_l(s)} = \frac{1.325s + 0.703}{s^2 + 1.548s + 0.709} \end{aligned} \tag{28}$$

For transfer function $H_{12}(s)$.

For transfer function $H_{21}(s)$.

For transfer function $H_{22}(s)$.

For MIMO system, the proposed LOSM is compared with the lower order models obtained by the particle swarm optimization and SE (Sikander and Prasad 2015a), BBBC optimization algorithm and RA (Desai and Prasad 2013b), cuckoo search optimization and SE (Narwal and Prasad

Table 10 Comparison of various model order reduction techniques based on performance error indices for SISO system

MOR techniques	LOSM	ISE	IAE	ITSE	MSE	RMSE
Proposed method	$\frac{0.824s+0.788}{s^2+1.655s+0.794}$	8.8230×10^{-5}	0.0171	1.7272×10^{-4}	1.287×10^{-05}	0.0036
Stability equation (SE) and Big Bang-Big crunch (BBBC) algorithm (Desai and Prasad 2013a)	$\frac{0.799s+0.675}{s^2+1.469s+0.685}$	3.448×10^{-4}	0.0364	0.0012	4.957×10^{-5}	0.0070
SE and genetic algorithm (GA) (Parmar et al. 2007a)	$\frac{0.7442s+0.699}{s^2+1.458s+0.6997}$	0.0015	0.0832	0.0055	2.213×10^{-4}	0.0149
Dominant pole retention and BBBC algorithm (Philip and Pal 2010)	$\frac{0.9314s+1.6091}{s^2+2.7562s+1.6092}$	0.0015	0.0819	0.0037	2.115×10^{-4}	0.0145
Truncation method (Smamash 1981)	$\frac{24s+24}{35s^2+50s+24}$	0.0030	0.1170	0.0069	4.421×10^{-4}	0.0210

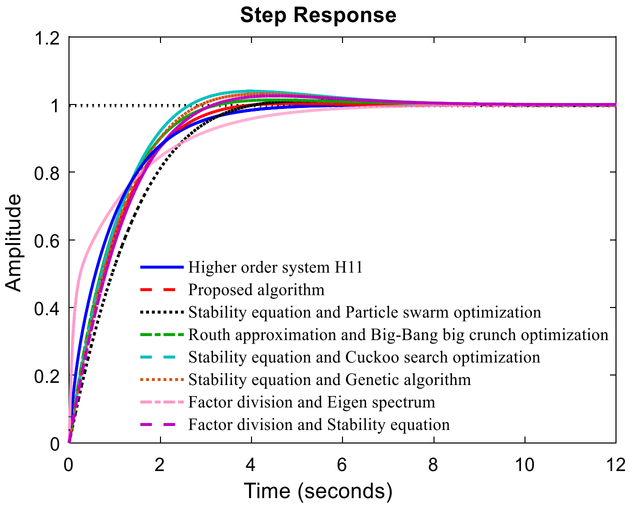


Fig. 5 Step response of HOSS $H_{11}(s)$ and LOSMs

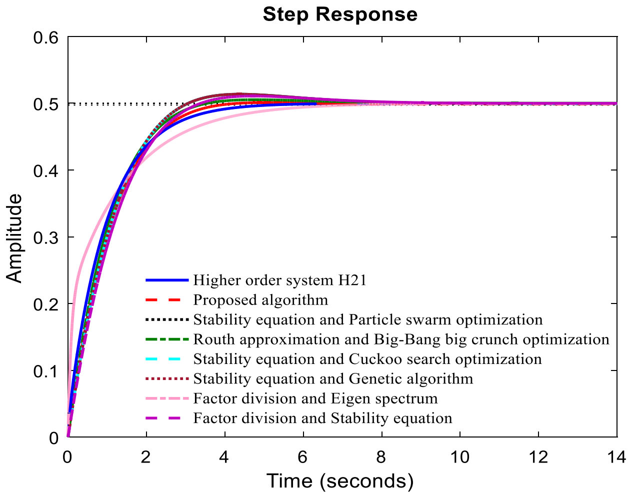


Fig. 7 Step response of HOSS $H_{21}(s)$ and LOSMs

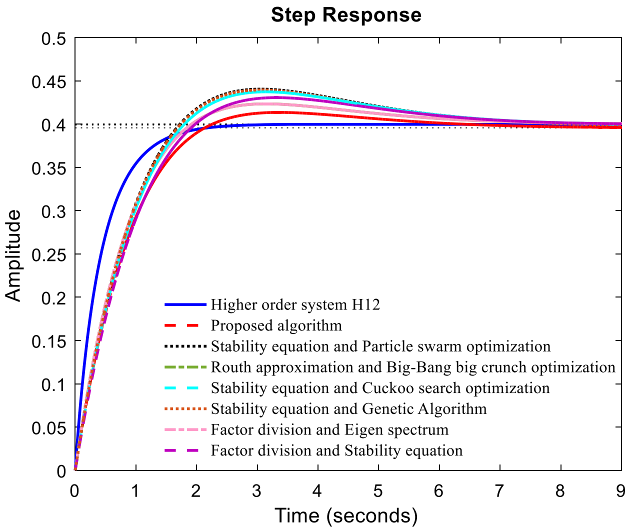


Fig. 6 Step reposes of HOSS $H_{12}(s)$ and LOSMs

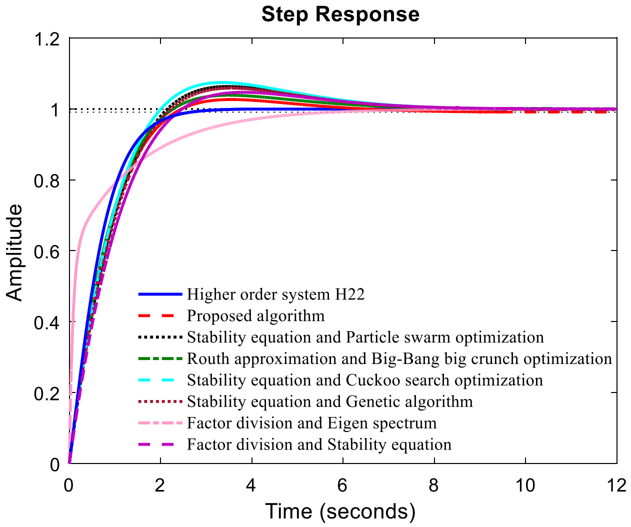


Fig. 8 Step responses of HOSS $H_{22}(s)$ and LOSMs

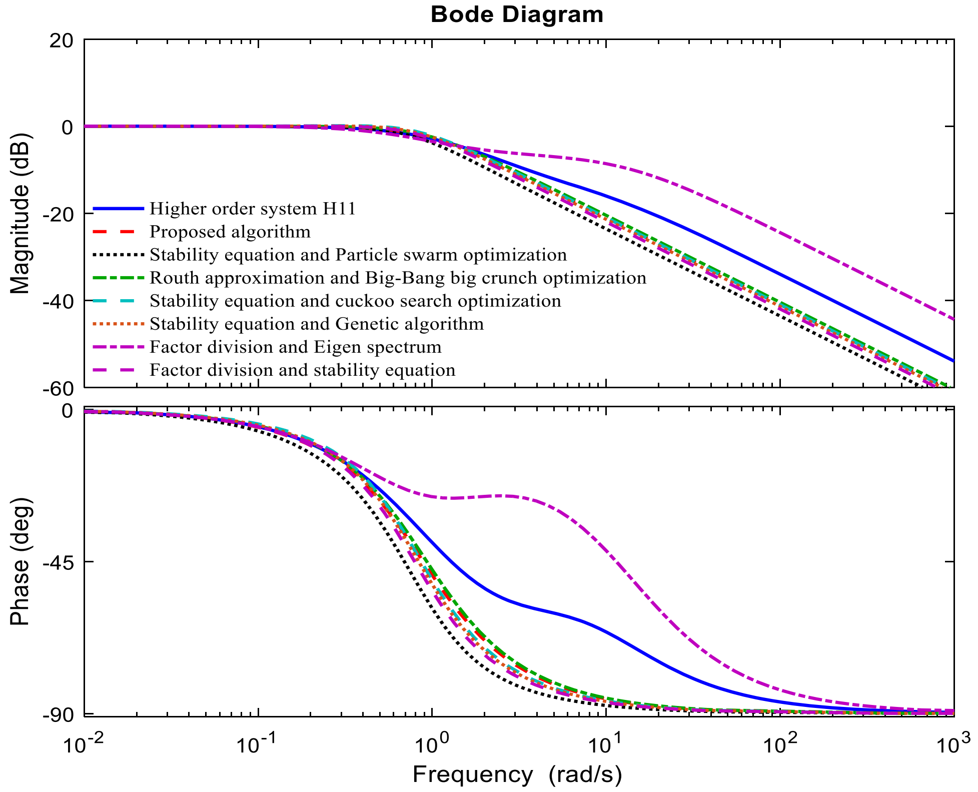


Fig. 9 Frequency response of HOSS $H_{11}(s)$ and LOSMs

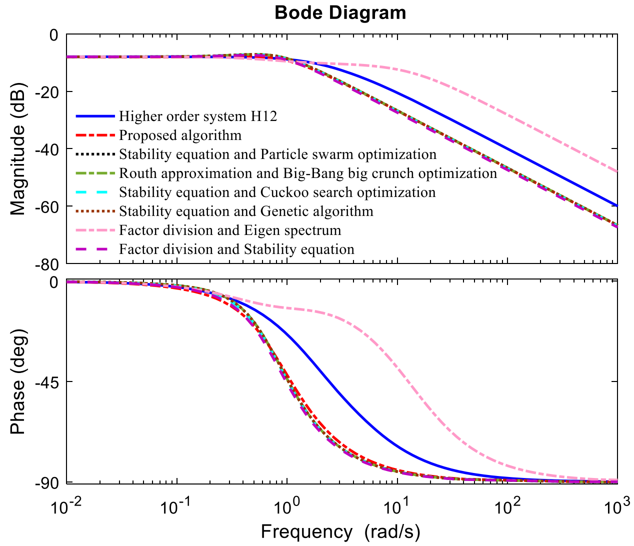


Fig. 10 Frequency responses of HOSS $H_{12}(s)$ and LOSMs

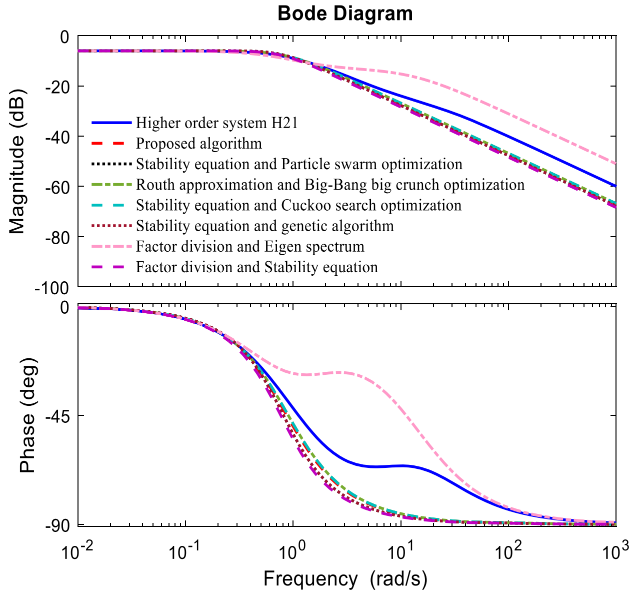


Fig. 11 Frequency responses of HOSS $H_{21}(s)$ and LOSMs

2016), GA and SE method (Parmar et al. 2007a), factor division (FD) and Eigen spectrum analysis (ESA) (Parmar et al. 2007b), FD and SE method (Sikander and Prasad 2015b) to assess the efficacy of the proposed model order reduction technique. Figures 5, 6, 7, and 8 illustrate the step responses of the LOSM alongside its corresponding HOSS. Additionally, Figs. 9, 10, 11, and 12 clearly demonstrate that the frequency responses of the LOSM by the proposed method closely resemble those of the HOSS. To further evaluate the performance of the LOSM, various performance indices, such as ISE, IAE, ITSE, MSE, and RMSE, are employed. These indices are used for comparing the LOSM with models obtained through existing

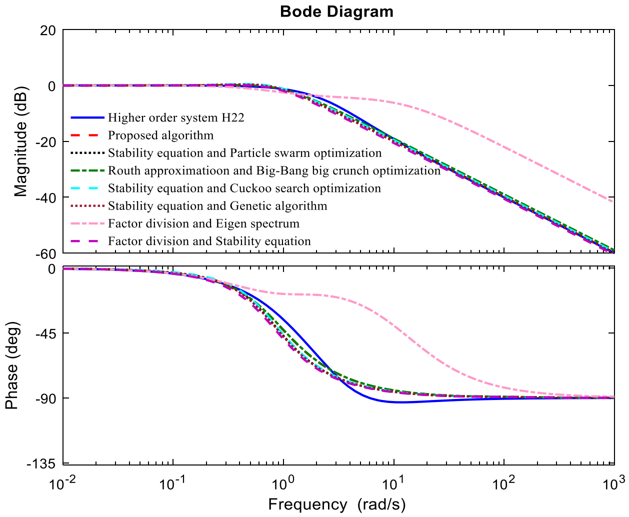


Fig. 12 Frequency response of HOSS $H_{22}(s)$ and LOMs

methods, as outlined in Tables 11, 12, 13, and 14. The optimal values for each of the test systems are obtained using the proposed method which are highlighted in bold face for easy identification.

6 Conclusion

A method based on the proposed improved adaptive differential evolution algorithm (I-ADE) and Routh approximation is implicated in the model order reduction of higher order SISO and MIMO systems. In the proposed method, the denominator polynomial is calculated using the Routh approximation technique and the numerator polynomial is obtained by employing the proposed I-ADE algorithm. The key factor of the proposed model order reduction method is obtaining an LOSM for an HOSS with minimum error

Table 11 Comparison of various model order reduction techniques based on performance error indices for $H_{11}(s)$ (Best values are presented in Bold face)

Model order reduction techniques	Lower-order model	ISE	IAE	ITSE	MSE	RMSE
Proposed technique	$\frac{0.889s+0.704}{s^2+1.548s+0.709}$	**6.5789 × 10^{-04}**	**0.0722**	**0.0035**	**4.9933 × 10^{-05}**	**0.0071**
Particle swarm optimization (PSO) and SE method (Sikander and Prasad 2015a)	$\frac{0.659s+0.616}{s^2+1.349s+0.618}$	0.0099	0.1983	0.0219	7.0978 × 10^{-04}	0.0266
BBBC optimization algorithm and (RA) method (Desai and Prasad 2013b)	$\frac{0.947s+0.709}{s^2+1.548s+0.709}$	0.0027	0.1264	0.0097	1.9544 × 10^{-04}	0.0140
Cuckoo Search Optimization and SE method (Narwal and Prasad 2016)	$\frac{0.893s+0.618}{s^2+1.349s+0.618}$	0.0107	0.2490	0.0379	7.6777 × 10^{-04}	0.0277
GA and SE method (Parmar et al. 2007a)	$\frac{0.850s+0.617}{s^2+1.349s+0.618}$	0.0061	0.1848	0.0239	4.3390 × 10^{-04}	0.0208
Factor division (FD) and Eigen spectrum analysis (ESA) (Parmar et al. 2007b)	$\frac{6.042s+8.470}{s^2+13.666s+8.470}$	0.0037	0.1431	0.0107	2.6301 × 10^{-04}	0.0162
FD and SE method (Sikander and Prasad 2015b)	$\frac{0.793s+0.618}{s^2+1.349s+0.618}$	0.0038	0.1336	0.0168	2.7276 × 10^{-04}	0.0165

Table 12 Comparison of various model order reduction techniques based on performance error indices for $H_{12}(s)$ (Best values are presented in Bold face)

Model order reduction techniques	Lower-order model	ISE	IAE	ITSE	MSE	RMSE
Proposed technique	$\frac{0.459s+0.280}{s^2+1.548s+0.709}$	**3.8515 × 10^{-04}**	**0.0531**	**0.0019**	**2.8863 × 10^{-05}**	**0.0054**
PSO and SE method (Sikander and Prasad 2015a)	$\frac{0.463s+0.247}{s^2+1.349s+0.618}$	0.0036	0.1392	0.0124	2.5542 × 10^{-04}	0.0160
BBBC optimization and RA method (Desai and Prasad 2013b)	$\frac{0.489s+0.283}{s^2+1.548s+0.709}$	0.0011	0.0751	0.0040	7.5086 × 10^{-05}	0.0087
Cuckoo search optimization and SE method (Narwal and Prasad 2016)	$\frac{0.451s+0.247}{s^2+1.349s+0.618}$	0.0029	0.1251	0.0107	2.0814 × 10^{-04}	0.0144
GA and SE method (Parmar et al. 2007a)	$\frac{0.461s+0.246}{s^2+1.349s+0.618}$	0.0033	0.1345	0.0116	2.3796 × 10^{-04}	0.0154
FD and ESA (Parmar et al. 2007b)	$\frac{3.941s+3.3883}{s^2+13.6666s+8.4707}$	0.0011	0.0639	0.0024	7.5952 × 10^{-05}	0.0087
FD and SE method (Sikander and Prasad 2015b)	$\frac{0.4272s+0.2471}{s^2+1.3495s+0.6181}$	0.0018	0.0926	0.0074	1.2905 × 10^{-04}	0.0114

Table 13 Comparison of various model order reduction techniques based on performance error indices for $H_{21}(s)$(Best values are presented in Bold face)

Model order reduction techniques	Lower-order model	ISE	IAE	ITSE	MSE	RMSE
Proposed technique	$\frac{0.434s+0.352}{s^2+1.548s+0.709}$	$\mathbf{1.4089 \times 10^{-04}}$	**0.0336**	$\mathbf{7.4489 \times 10^{-04}}$	$\mathbf{1.0716 \times 10^{-05}}$	**0.0033**
PSO and SE method (Sikander and Prasad 2015a)	$\frac{0.4092s+0.3086}{s^2+1.3495s+0.6181}$	0.0012	0.0808	0.0049	8.5690×10^{-05}	0.0093
BBBC algorithm and RA method (Desai and Prasad 2013b)	$\frac{0.455s+0.3546}{s^2+1.5482s+0.7091}$	4.5576×10^{-04}	0.0521	0.0017	3.2555×10^{-05}	0.0057
Cuckoo Search Optimization and SE method (Narwal and Prasad 2016)	$\frac{0.4314s+0.3091}{s^2+1.3492s+0.6181}$	0.0022	0.1145	0.0081	1.59561×10^{-04}	0.0126
GA and SE method (Parmar et al. 2007a)	$\frac{0.40933s+0.30860}{s^2+1.34952s+0.6181}$	0.0012	0.0810	0.0050	8.6094×10^{-05}	0.0093
FD and ESA (Parmar et al. 2007b)	$\frac{2.8097s+4.2354}{s^2+13.6666s+8.4707}$	0.0010	0.0751	0.0030	7.2185×10^{-05}	0.0085
FD and SE method (Sikander and Prasad 2015b)	$\frac{0.3795s+0.3090}{s^2+1.34952s+0.6181}$	8.2416×10^{-04}	0.0691	0.0035	5.8870×10^{-05}	0.0077

Table 14 Comparison of various model order reduction techniques based on performance error indices for $H_{22}(s)$ (Best values are presented in Bold face)

Model order reduction techniques	Lower-order model	ISE	IAE	ITSE	MSE	RMSE
Proposed technique	$\frac{1.103s+0.702}{s^2+1.548s+0.709}$	$\mathbf{1.7505 \times 10^{-03}}$	**0.1147**	**0.0089**	$\mathbf{1.7505 \times 10^{-04}}$	**0.0115**
PSO and SE method (Sikander and Prasad 2015a)	$\frac{1.0187s+0.6169}{s^2+1.3495s+0.6181}$	0.0083	0.1978	0.0342	5.9038×10^{-04}	0.0243
BBBC algorithm and RA method (Desai and Prasad 2013b)	$\frac{1.126s+0.7091}{s^2+1.5482s+0.7091}$	0.0031	0.1163	0.0129	2.1841×10^{-04}	0.0148
Cuckoo Search algorithm and SE method (Narwal and Prasad 2016)	$\frac{1.0579s+0.6181}{s^2+1.3492s+0.6181}$	0.0124	0.2575	0.0480	8.8915×10^{-04}	0.0298
GA and SE method (Parmar et al. 2007a)	$\frac{0.99766s+0.617112}{s^2+1.34952s+0.6181}$	0.0072	0.1713	0.0305	5.1238×10^{-04}	0.0226
FD and ESA (Parmar et al. 2007b)	$\frac{8.0195s+8.4707}{s^2+13.6666s+8.4707}$	0.0133	0.2321	0.0307	9.5307×10^{-04}	0.0309
FD and stability equation SE method (Sikander and Prasad 2015b)	$\frac{0.9338s+0.6181}{s^2+1.34952s+0.6181}$	0.0065	0.1935	0.0253	4.6369×10^{-04}	0.0215

bounds and retaining the significant characteristics of the HOSS. Additionally, the performance of the model order reduction method is evaluated by employing diverse test case studies for SISO and MIMO systems. The frequency and step responses show the transient and steady-state behaviour of HOSS and LOSM. The results obtained using the proposed method are better than those obtained using different model order reduction methods of the published literature pertaining to stability checks by employing step and frequency responses. Performance indices results indicate that the proposed method for LOSM is effective, efficient, reliable and feasible. Hence, it can be concluded that the proposed approximation method is stable and superior to the other alternative methods existing in the literature. Additionally, the proposed I-ADE algorithm is robust and reliable, which is statistically confirmed from the simulation results for optimizing 23 benchmark functions. The proposed I-ADE algorithm is so adaptive that it can be applied to a wide range of optimization problems with the least intervention. In the future, one can extend the proposed scheme for simplifying discrete, fractional-order SISO and MIMO systems.

Funding The authors have not disclosed any funding.

Data availability Data can be available on request to the corresponding author.

Declarations

Conflict of interest The authors declare no conflict of interest in the manuscript.

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

References

Abdollahzadeh B et al (2021) Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Int J Intell Syst 36:5887–5958

Akhmedova S, Stanovov V, Semenkin E (2018) LSHADE Algorithm with a Rank-based Selective Pressure Strategy for the Circular Antenna Array Design Problem. In ICINCO (1). 2018

Ashoor N, Singh V (1982) A note on low-order modeling. IEEE Trans Autom Control 27:1124–1126

Awad NH et al. (2016) An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems. In 2016 IEEE congress on evolutionary computation (CEC). IEEE

Awad NH, Ali MZ, Suganthan PN (2017) Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC2017 benchmark problems. in 2017 IEEE Congress on Evolutionary Computation (CEC). IEEE

Bilal, Pant M, Zaheer H, Garcia-Hernandez L, Abraham A (2020) Differential evolution: a review of more than two decades of research. Eng Appl Artif Intell 90:103479

Bistritz Y, Shaked U (1984) Minimal Pade model reduction for multivariable systems. J Dyn Syst Meas Control. https://doi.org/10.1115/1.3140688

Corne D et al (1999) New ideas in optimization. McGraw-Hill Ltd., UK

Das S, Mullick SS, Suganthan PN (2016) Recent advances in differential evolution–an updated survey. Swarm Evol Comput 27:1–30

Desai S, Prasad R (2013a) A new approach to order reduction using stability equation and big bang big crunch optimization. Syst Sci Control Eng an Open Access J 1:20–27

Desai S, Prasad R (2013b) A novel order diminution of LTI systems using Big Bang Big Crunch optimization and Routh Approximation. Appl Math Model 37:8016–8028

El-Kenawy E-SM et al (2022) Novel meta-heuristic algorithm for feature selection, unconstrained functions and engineering problems. IEEE Access 10:40536–40555

Fortuna L, Nunnari G, Gallo A (2012) Model order reduction techniques with applications in electrical engineering. Springer Science and Business Media

Ganji V et al (2017) A novel model order reduction technique for linear continuous-time systems using PSO-DV algorithm. J Control Autom Electr Syst 28:68–77

Goldberg DE (1989) Genetic algorithms in search optimization and machine learning. Addison-Wesley Reading. https://doi.org/10.1007/s10589-009-9261-6

Hutton M, Friedland B (1975) Routh approximations for reducing order of linear, time-invariant systems. IEEE Trans Autom Control 20:329–337

Hwang C (1984) Mixed method of Routh and ISE criterion approaches for reduced-order modeling of continuous-time systems. J Dyn Syst Meas Control. https://doi.org/10.1115/1.3140697

Kennedy J, Eberhart R (1995) Particle swarm optimization. In IEEE International Conference on Neural Networks, vol. 4, IEEE, pp. 1942–1948. https://doi.org/10.1109/ICNN.1995.488968

Kenneth SR, Storn RM (1997) Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J Global Optim 11(4):341–359

Lucas TN (1993) Optimal model reduction by multipoint Padé approximation. J Franklin Inst 330:79–93

Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249

Mirjalili S (2016a) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133

Mirjalili S (2016b) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27:1053–1073

Mirjalili S et al (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61

Mukherjee S, Mittal R (2005) Model order reduction using response-matching technique. J Franklin Inst 342:503–519

Narwal A, Prasad BR (2016) A novel order reduction approach for LTI systems using cuckoo search optimization and stability equation. IETE J Res 62:154–163

Padhy AP (2022) Model order approximation of linear time invariant system using dragonfly algorithm. J Algebr Stat 13:2890–2897

Padhy AP et al (2021) Model order reduction of continuous time multi-input multi-output system using sine cosine algorithm. Congress Intell Syst Proceed CIS 2(2022):503–513

Padhy AP, Singh V, Singh VP (2023) Stable approximation of S and MIMO linear dynamic systems. Sādhanā. https://doi.org/10.1007/s12046-023-02151-x

Panigrahi S, Behera H (2020) Time series forecasting using differential evolution-based ANN modelling scheme. Arab J Sci Eng 45:11129–11146

Parmar G, Prasad R, Mukherjee S (2007a) Order reduction of linear dynamic systems using stability equation method and GA. Int J Comput Inform Syst Sci Eng 1:26–32

Parmar G, Mukherjee S, Prasad R (2007b) System reduction using factor division algorithm and eigen spectrum analysis. Appl Math Model 31:2542–2552

Philip B, Pal J (2010) "An evolutionary computation based approach for reduced order modelling of linear systems," in 2010 IEEE International Conference on Computational Intelligence and Computing Research, 2010, pp. 1–8

Price K, Storn RM, Lampinen JA (2006) Differential evolution: a practical approach to global optimization. Springer Science and Business Media

Schilders WH et al (2008) Model order reduction: theory, research aspects and applications, vol 13. Springer

Shamash Y (1974) Stable reduced-order models using Padé-type approximations. IEEE Trans Autom Control 19:615–616

Sikander A, Prasad R (2015a) Soft computing approach for model order reduction of linear time invariant systems. Circuits Syst Signal Process 34:3471–3487

Sikander A, Prasad R (2015b) Linear time-invariant system reduction using a mixed methods approach. Appl Math Model 39:4848–4858

Smamash Y (1981) Truncation method of reduction: a viable alternative. Electron Lett 17:97–99

Vishwakarma C, Prasad R (2008) Clustering method for reducing order of linear system using Pade approximation. IETE J Res 54:326–330

Vishwakarma C, Prasad R (2009) MIMO system reduction using modified pole clustering and genetic algorithm. Model Simul Eng 2009:1–5

Yang X-S, Slowik A (2020) Firefly algorithm. Swarm intelligence algorithms. CRC Press, pp 163–174

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.