Power Law based Local Search in Spider Monkey Optimization for Lower Order System Modeling

Ajay Sharma[a*], Harish Sharma[b], Annapurna Bhargava[c] and Nirmala Sharma[d]

[a,b,c,d]*Rajasthan Technical University, Kota, India*

(*Received 00 Month 20XX; final version received 00 Month 20XX*)

The nature-inspired algorithms (NIAs) have shown efficiency to solve many complex real-world optimization problems. The efficiency of NIAs is measured by their ability to find adequate results within a reasonable amount of time, rather than an ability to guarantee the optimal solution. This paper presents a solution for lower order system modeling using Spider monkey optimization (SMO) algorithm to obtain a better approximation for lower order systems and reflects almost original higher order system's characteristics. Further, a local search strategy, namely power law based local search is incorporated with SMO. The proposed strategy is named as power law based local search in SMO (PLSMO). The efficiency, accuracy, and reliability of the proposed algorithm is tested over 20 well known benchmark functions. Then, the PLSMO algorithm is applied to solve the lower order system modeling problem.

Keywords: Spider Monkey Optimization; Memetic Algorithm; Power Law based Local Search; Lower Order System Modeling; Integral Square Error; Impulse Response Energy

1. Introduction

Model order reduction is of great importance in power community from quite a long time. It is always important that the reduced model should reflect the same physical characteristics of original model such as stability, transient response, and steady state error, and so on while obtaining a reduced order. A lot of order reduction methods are there, each focusing on some properties of the system. The techniques available so far are problem specific and based on the original system stability retention and a nearby approximation response. A variety of model order reduction (MOR) techniques have been introduced, some of them are like Continued Fraction Expansion Method by (C. Chen & Shieh, 1968), Moment Matching Method by (Paynter & Takahashi, 1956), Pade Approximation Method by (Pade, 1892), Routh Approximation Method by (Hutton & Friedland, 1975), Routh Hurwitz Array Method by (Krishnamurthy & Seshadri, 1978), Stability Equation Method by (T. Chen, Chang, & Han, 1979), Differentiation Method by (P. O. Gutman & Molander, 1982), Truncation Method by (Gustafson, 1966), Dominant Pole Retention Method by (Davison, 1966), Factor Division Method by (T. N. Lucas, 1983), Least Square Method by (Shoji et al., 1985) etc. The above mentioned MOR techniques provide models which sometimes turn out to be non minimum phase. So for better results conventional techniques are used with some optimization techniques. The most popular techniques are error minimization of continuous time system. Various error minimization criterions are available such as Integral Square Error (ISE), Integral Time Square Error ($ITSE$), Integral of Absolute Error (IAE), and Integral Time Absolute Error ($ITAE$). The Integral Square Error (ISE) minimization technique is mostly applied. The integral of the squared error is minimized between the unit (either impulse or step) response

*Corresponding author. Email: ajay_2406@yahoo.com,

of the original higher order and new reduced lower order system. Some work already have been carried out using this technique in time domain by (Mishra & Wilson, 1980; Hickin & Sinha, 1976) and in frequency domain by (Vilbe & Calvez, 1990; Sinha & Pille, 1971; Sinha & Bereznai, 1971; Mukherjee & Mishra, 1987; Lamba, Gorez, & Bandyopadhyay, 1988; Hwang, 1984; Hickin & Sinha, 1978). (Luus, 1999) applied numerical optimization to minimize deviation between the frequency response of the higher order and lower order models. (Sanathanan Stanley et al., 1987) applied simplex method to obtain reduced models. (Hu, 1987) hybridized Pade approximation technique with the minimization of frequency deviations. (M. Quang & Pan., 1987) proposed a new strategy to utilizes both pole retention and frequency response matching. The deterministic methods have their own limitations of depending on initial guess, lack of robustness, time consuming, inability to solve real world problems, and many local optimal solution for nonlinear optimizing problems. The Swarm intelligence based stochastic algorithms have been shown their effectiveness to find solution of nonlinear, nonconvex or discrete optimization problems [(?, ?, ?, ?; Vesterstrøm & Thomsen, 2004)]. The algorithms that have emerged in recent years include ant colony optimization (ACO) [(?, ?)], particle swarm optimization (PSO) [(?, ?)], bacterial foraging optimization (BFO) [(Passino, 2002)], artificial bee colony (ABC) optimization algorithm [(Karaboga, 2005)] etc. Therefore, these meta-heuristics can be applied to overcome the problems associated with conventional techniques to solve the lower order system modeling problems [(Sivanandam & Deepa, 2007; Mukherjee & Mittal, 2005)]. The reported results of these strategies are motivating in terms of efficiency and accuracy.

In this paper, a new swarm intelligence based algorithm, namely "Spider Monkey Optimization" (SMO) [(Bansal, Sharma, Jadon, & Clerc, 2014)] is applied to solve the lower order system modeling problem. Here, we considered four models having different complexities and reduced the order by applying the SMO algorithm. Further, to improve the exploitation of search space capability of the SMO, a local search strategy, namely power law based local search [(Sharma, Bansal, & Arya, 2014)] is incorporated with SMO. The proposed strategy is named as power law local search based SMO (PLSMO). The proposed strategy is tested over 20 well known benchmark functions. Further, the considered higher model order reduction problems are solved using the proposed PLSMO. The reported results are compared with the SMO and other meta-heuristics and deterministic algorithms available in the literature.

Rest of the paper is organized as follows: Section 2, presents a brief review on local search strategies. Basic SMO is explained in section 3. Power law based local search strategy is explained and incorporated with SMO in section 4. In section 5, performance of the proposed strategy is analyzed. Application of PLSMO for lower order system modeling is explained in section 6. In section 7, results for various examples are compared. Finally, paper is concluded in section 8.

2. Recent Local Search Techniques

The motivation of incorporating local search strategies with basic algorithms comes from the necessity to balance exploration and exploitation capabilities of algorithms [(C. M. P. E. Neri Ferrante; Cotta, Vol. 379, 2012)]. This can be used as a tool to enhance the exploitation capability of the global search algorithm so that the already identified region may be exhaustively searched. The significant hybridizations of the local search strategies with the global search algorithms are briefly reviewed as follows:(F. Neri & Tirronen, 2009) incorporated scale factor local search with DE. (Caponio, Neri, & Tirronen, 2009) adapted superfit control in DE with three local search algorithms to improve exploitation capability. (Wang, Wang, & Yang, 2009) proposed an evolutionary algorithm which combines greedy crossover based hill climbing and steepest mutation based hill climbing techniques. (?, ?) proposed a competitive and cooperative co-evolutionary technique with multi-objective PSO algorithm design. (Tang, Mei, & Yao, 2009) included Merge-Split (MS) operator which is capable of searching using large step sizes, and thus has the potential to search the solution space more efficiently and is less likely to be trapped in local optima. (Repoussis, Tarantilis, & Ioannou, 2009) presented an arc-guided evolutionary algorithm to solve vehicle routing problem. (Richer, Goëffon, & Hao, 2009) proposed memetic algorithm (implemented in the software Hydra) based on integration of an effective local search operator with a specific topological tree crossover operator to present Phylogenic reconstruction memetic approach. (Gallo, Carballido, & Ponzoni, 2009) integrated a new hybrid approach with an evolutionary algorithm with local search for microarray biclustering is presented. During incorporation of two mechanisms: the first one avoids loss of good solutions through generations and overcomes the high degree of overlap in the final population; and the other one preserves an adequate level of genotypic diversity. (Nguyen, Ong, & Lim, 2009) proposed an evolution and individual learning based balance using theoretical upper bound derived while the search is in progress. (Ong, Lim, & Chen, 2010) presented memetic computing as an interesting strategy to solve complex optimization problems. (Mininno & Neri, 2010) incorporated local search as Memetic DE in noisy optimization. (Mezura-Montes & Velez-Koeppel, 2010) modified ABC with two local search strategies and works on different percentages of function evaluations. (X. Chen, Ong, Lim, & Tan, 2011) presented realization of memetic computing through memetic algorithm. (Kang, Li, Ma, & Li, 2011) proposed a Hooke Jeeves Artificial Bee Colony algorithm (HJABC). (Kang, Li, & Ma, 2011) hybridized ABC algorithm with Rosenbrock's rotational direction strategy to propose Rosenbrock's ABC (RABC). (Fister, Fister Jr, Brest, & Žumer, 2012) proposed Nelder-Mead algorithm (NMA) and random walk direction exploitation (RWDE) algorithm, hybridized with ABC. (Fister et al., 2012) proposed memetic firefly algorithm for combinatorial optimization. (Sharma, Bansal, & Arya, 2013) included opposition based lévy flight local search with ABC. (Sharma, Jadon, Bansal, & Arya, 2013) applied lévy flight local search strategy with DE. (Sharma et al., 2014) incorporated memetic power law with ABC algorithm. (Lastra, Molina, & Benítez, 2015) presented a new design of MA-SW-Chains to solve extremely high-dimensional problems related to GPU-based massively parallel architecture. (Sharma, Bansal, Arya, & Yang, 2016) integrated Lévy flight local search strategy with artificial bee colony algorithm (ABC). (Lee & Kim, 2015) incorporated memetic feature selection algorithm for multi-label classification procedures to refine the feature subsets found through GAs. (Bhuvana & Aravindan, 2015) integrated preferential local search using adaptive weights for multi-objective optimization problems with evolutionary algorithm. The results of incorporation of local search strategies are continuously motivating researchers to inculcate new local search methods with some other algorithms. Here, we incorporated a power law based local search strategy with SMO algorithm. The SMO algorithm is explained in the subsequent section followed by the proposed strategy.

3. Spider Monkey Optimization (SMO) algorithm

Spider Monkey Optimization (SMO) algorithm developed by (Bansal et al., 2014), is based on fission-fusion social behaviour of spider monkeys. Here individual form temporary small parties which belongs to a larger community.

The SMO algorithm consists of six phases namely Local Leader phase, Global Leader phase, Local Leader Learning phase, Global Leader Learning phase, Local Leader Decision phase and Global Leader Decision phase.

3.1. *Initialization of the Population*

Initially, SMO generates an initial equally distributed population of N spider monkeys where each monkey SM_i (i = 1, 2, ...,N) is a D-dimensional vector and SM_i represents the i^{th} Spider Monkey (SM) in population. Each SM belongs to probable solution to find optimum solution of the considered problem. Initialization of each SM_i takes place as follows:

$$SM_{ij} = SM_{minj} + U(0,1) \times (SM_{maxj} - SM_{minj}) \tag{1}$$

In j^{th} direction the upper and lower bounds of SM_i are SM_{minj} and SM_{maxj} while $U(0,1)$ is a random number uniformly distributed in the range [0, 1].

3.2. *Local Leader Phase (LLP)*

The information received from local leader and local group member is utilized for current position update of each spider monkey SM. Fitness value of this new position is calculated, if it is found to be better than old position, the SM moves to the better position. The equation for position update for i^{th} SM (member of k^{th} local group) in current phase is as follows:

$$SMnew_{ij} = SM_{ij} + U(0,1) \times (LL_{kj} - SM_{ij}) + U(-1,1) \times (SM_{rj} - SM_{ij}) \tag{2}$$

where SM_{ij} is the j^{th} dimension of the i^{th} SM, LL_{kj} represents the j^{th} dimension of the k^{th} local group leader position. SM_{rj} is the j^{th} dimension of the r^{th} SM which is chosen arbitrarily within k^{th} group such that $r \neq i$, U(0, 1) is a uniformly distributed random number between 0 and 1.

3.3. *Global Leader Phase (GLP)*

The knowledge received from Global Leader and Local group member is utilized to update position of all SMs as per following position update equation

$$SMnew_{ij} = SM_{ij} + U(0,1) \times (GL_j - SM_{ij}) + U(-1,1) \times (SM_{rj} - SM_{ij}) \tag{3}$$

Where GL_j is the j^{th} dimension of the global leader position and j is the randomly chosen index. The positions of spider monkeys (SM_i) are updated based on a variable $prob_i$ which are calculated using their fitness. Here, a better candidate is given more chances to make it better. The probability $prob_i$ is calculated as equation 4:

$$prob_i = 0.9 \times \frac{fitness_i}{max\_fitness} + 0.1, \tag{4}$$

Here $fitness_i$ is the fitness value of the i^{th} SM and $max\_fitness$ is the highest fitness in the group. The new generated position of SM's fitness is evaluated with old fitness and the better one is adopted.

3.4. *Global Leader Learning (GLL) phase*

Greedy selection is used to select the position of SM having best fitness in the population as the new position of Global Leader. If the Global Leader is not updating its position then Global Limit Count is incremented by 1.

3.5. *Local Leader Learning (LLL) phase*

Greedy selection is used to select the position of SM having best fitness in that group as new Local Leader position. If the Local Leader position is not updated than the Local Limit Count is incremented by 1.

3.6. *Local Leader Decision (LLD) phase*

All members of a minor group update their position either by random initialization or by using combined information from Global Leader and Local Leader through equation (5).

$$SMnew_{ij} = SM_{ij} + U(0,1) \times (GL_j - SM_{ij}) + U(0,1) \times (SM_{ij} - LL_{kj}); \tag{5}$$

It is clear that the updated dimension of this SM is attracted towards global leader and repels from the local leader.

3.7. *Global Leader Decision (GLD) Phase*

Global Leader divides the population into minor groups if it is not updated up to preset number of iteration called Global Leader Limit. The population is evenly distributed into two groups and then three groups and so on respectively till the maximum number of groups (MG) are formed, LLL process is initiated to choose the local leader in the newly formed groups. If maximum number of groups is formed and even then the global leader position is not updated then the global leader combines all the minor groups to form a single group.

4. Power Law based Local Search in SMO

In hybrid algorithms, the main algorithm explore the search space to find out most suitable search space region, while the surroundings are scrutinized by local search part. In this scrutinizing, the role of step size of the solutions is of great significance in exploiting the identified region. For the same purpose, a local search strategy, namely Power law based local search (PLLS) by (Sharma et al., 2014) is hybridized with SMO algorithm. The proposed hybridized strategy is named as power law local search based SMO (PLSMO) algorithm. In PLLS, step size is iteratively decreased to exploit the identified search region in the periphery of the best candidate solution. A parameter u is used to reduce step sizes based on power law function of iteration counter. The position update equation of an i^{th} individual is shown in equation (6):

$$x_{ij}(t+1) = x_{ij}(t) + \alpha\beta \text{sign}(rand[0,1] - \frac{1}{2}) \times u(t), \tag{6}$$

In this equation, the step size control parameter is α and the social learning component is $\beta = (x_{ij} - x_{kj})$, and x_{ij} is the best solution in the current population. The $\text{sign}(rand[0,1] - \frac{1}{2})$ essentially provides a random sign or direction, while u is a power law function based parameter of iteration counter (t) and computed using equation (7):

$$u(t) = t^{-\lambda},\ (1 < \lambda \leq 3), \tag{7}$$

The step length decreases with a decrease in u, as the iteration counter increases as per equation (7). The u thus calculated, helps to exploit the search space through iterations. The step size to update a candidate solution is $(\beta \times \alpha sign(rand[0,1] - \frac{1}{2}) \times u)$ a random walk process with a power-law distribution with decreasing step-length and having a heavy tail.

The pseudo-code of the proposed $PLLS$ strategy with SMO is shown in algorithm 1. In $PLLS$, only the best individual of the current population updates itself in its neighborhood. In Algorithm

Algorithm 1 Power Law based Local Search Strategy with SMO:

```
Input optimization function Minf(x), α and λ;
The best solution x_best is selected in swarm;
Counter initialized at t = 1;
while (u > ϵ) do
   Evaluate u(t) = t^(−λ);
   New solution x_new is generated using u(t) and α by Algorithm 2.
   Evaluate f(x_new).
   if f(x_new) < f(x_best) then
      x_best = x_new;
   end if
   t = t + 1;
end while
```

1, ϵ determines the termination of local search.

Algorithm 2 Generation of new solution:

```
Input values best solution x_best, u, and α ;
for j = 1 to D do
   if U(0,1) > p_r then
      x_newj = x_bestj + (x_bestj − x_kj) × αsign(rand[0,1] − 1/2) × u;
   else
      x_newj = x_bestj;
   end if
end for
Return x_new
```

Here, p_r is perturbation rate (a number between 0 and 1) to control the amount of perturbation in the best solution, $U(0,1)$ is a uniformly distributed random number, D is the dimension of the problem and x_k is a randomly selected solution within population. See section 5.1 for details of these parameter settings.

The proposed $PLSMO$ consists of seven phases: Local Leader phase, Global Leader phase, Local Leader Learning phase, Global Leader Learning phase, Local Leader Decision phase, Global Leader Decision phase and PLLS phase. The pseudo-code of the $PLSMO$ algorithm is shown in algorithm 3.

5. Experimental results and analysis

In this section, the performance of PLSMO is evaluated and analysed. To evaluate the performance of $PLSMO$, 20 different global optimization problems (f_1 to f_{20}) are selected (listed in Table 1). These are continuous optimization problems and have different degrees of complexity and multimodality.

Algorithm 3 Power Law based Local Search in SMO ($PLSMO$):

Parameters initialized;
while Termination criteria **do**
  Step 1: Local Leader phase.
  Step 2: Global Leader phase.
  Step 3: Local Leader Learning phase.
  Step 4: Global Leader Learning phase.
  Step 5: Local Leader Decision phase.
  Step 6: Global Leader Decision phase.
  Step 7: Apply Power Law Local search strategy ($PLLS$) phase.
end while
Print best solution.

5.1. *Experimental setting*

To establish the efficiency of PLSMO algorithm, it is compared with SMO algorithm, since (Bansal et al., 2014) have shown that the SMO algorithm performs well on uni-model, multimodel, separable, and non-separable functions as compared to ABC, PSO, and DE algorithms. To test PLSMO and SMO over various considered test problems, following experimental settings are adopted [(Bansal et al., 2014)]:

- The Swarm size $N = 50$,
- $MG = 5$,
- $GlobalLeaderLimit$=50,
- $LocalLeaderLimit$=1500,
- $pr \in [0.1, 0.4]$, linearly increasing over iterations,

$$pr_{G+1} = pr_G + (0.4 - 0.1)/MIR \tag{8}$$

  where, G is the iteration counter, MIR is the maximum number of iterations.
- The maximum number of function evaluations (which is set to be 2.0×10^{04}) or the corresponding acceptable error (mentioned in Table 1) is stopping criteria,
- The number of simulations/run =100.

5.2. *Results Comparison*

In this section, a comparative study is carried out between SMO and it's proposed variant PLSMO by comparing them in terms of success rate (SR), average number of function evaluations (AFE), mean error (ME), and standard deviation (SD). The numerical results are presented in Table 2, 3, 4, and 5 respectively for SR, AFE, ME, and SD. In comparison, the preference is given to SR, AFE, and ME for all algorithms. The results reveal that $PLSMO$ performs better for functions f_3 - f_{11}, f_{13} - f_{16}, f_{19}, and f_{20}. For functions f_{17}, f_{18} both the algorithms perform almost same, while for functions f_1, f_2, and f_{12} SMO performs better than $PLSMO$. The result show that $PLSMO$ outperforms in most of the cases as compare to the basic SMO. As the functions f_3 - f_5, f_7 - f_{16}, f_{19}, and f_{20} are multimodal in nature and PLSMO performs better on these functions so we can say generally PLSMO performs better than SMO for multimodel functions.

Table 1: Test problems.D: Dimensions, C: Characteristic, U: Unimodal, M: Multimodal, S: Separable, N: Non-Separable, AE: Acceptable Error

Test Problem	Objective function	Search Range	Optimum Value	D	AE	C
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	[-5.12 5.12]	$f(\vec{0}) = 0$	30	$1.0E-05$	S,U
De Jong f4	$f_2(x) = \sum_{i=1}^{D} i.(x_i)^4$	[-5.12 5.12]	$f(\vec{0}) = 0$	30	$1.0E-05$	S,M
Ackley	$f_3(x) = -20 + e + exp(-\frac{0.2}{D}\sqrt{\sum_{i=1}^{D} x_i{}^3})$ $-exp(\frac{1}{D}\sum_{i=1}^{D} \cos\,(2\pi x_i)x_i)$	[-1 1]	$f(\vec{0}) = 0$	30	$1.0E-05$	M,N
Zakharov	$f_4(x) = \sum_{i=1}^{D} x_i{}^2 + (\sum_{i=1}^{D} \frac{ix_i}{2})^2 + (\sum_{i=1}^{D} \frac{ix_1}{2})^4$	[-5.12 5.12]	$f(\vec{0}) = 0$	30	$1.0E-02$	M,N
Inverted cosine wave	$f_5(x) = -\sum_{i=1}^{D-1}\left(\exp\left(\frac{-(x_i^2+x_{i+1}^2+0.5x_ix_{i+1})}{8}\right)\times \mathrm{I}\right)$	[-5 5]	$f(\vec{0}) = -D+1$	10	$1.0E-05$	M,N
Neumaier 3 Problem (NF3)	$f_6(x) = \sum_{i=1}^{D} (x_i - 1)^2 - \sum_{i=2}^{D} x_i x_{i-1}$	$[-D^2\ D^2]$	$f_{min} = -\frac{(D(D+4)(D-1))}{6}$	10	$1.0E-01$	U,N
Beale function	$f_7(x) = [1.5 - x_1(1-x_2)]^2 + [2.25 - x_1(1-x_2^2)]^2 + [2.625 - x_1(1-x_2^3)]^2$	[-4.5 4.5]	$f(3, 0.5) = 0$	2	$1.0E-05$	N,M
Colville function	$f_8(x) = 100[x_2 - x_1^2]^2 + (1-x_1)^2 + 90(x_4 - x_3^2)^2 + (1-x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4-1)^2] + 19.8(x_2-1)(x_4-1)$	[-10 10]	$f(\vec{1}) = 0$	4	$1.0E-05$	N.M
Kowalik	$f_9(x) = \sum_{i=1}^{11}[a_i - \frac{x_1(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}]^2$	[-5 5]	$f(0.192833, 0.190836, 0.123117, 0.135766) = 0.000307486$	4	$1.0E-05$	M,N
2D Tripod function	$f_{10}(x) = p(x_2)(1+p(x_1)) + \|(x_1 + 50p(x_2)(1-2p(x_1)))\| + \|(x_2 + 50(1 - 2p(x_2)))\|$	[-100 100]	$f(0, -50) = 0$	2	$1.0E-04$	N,M
Shifted Rosenbrock	$f_{11}(x) = \sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2) + f_{bias},\ z = x - o + 1,$ $x = [x_1, x_2, \ldots.x_D],\ o = [o_1, o_2, \ldots o_D]$	[-100 100]	$f(o) = f_{bias} = 390$	10	$1.0E-01$	S,M
Shifted Sphere	$f_{12}(x) = \sum_{i=1}^{D} z_i^2 + f_{bias},\ z = x - o\ ,x = [x_1, x_2, \ldots.x_D],\ o = [o_1, o_2, \ldots o_D]$	[-100 100]	$f(o) = f_{bias} = -450$	10	$1.0E-05$	S,M
Six-hump camel back	$f_{13}(x) = (4 - 2.1x_1^2 + x_1^4/3)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	[-5 5]	$f(-0.0898, 0.7126) = -1.0316$	2	$1.0E-05$	N,M
Easom's function	$f_{14}(x) = -\cos x_1 \cos x_2 e^{((-(x_1-\pi)^2-(x_2-\pi)^2))}$	[-10 10]	$f(\pi, \pi) = -1$	2	$1.0E-13$	S,M
Dekkers and Aarts	$f_{15}(x) = 10^5 x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^{-5}(x_1^2 + x_2^2)^4$	[-20 20]	$f(0, 15) = f(0, -15) = -24777$	2	$5.0E-01$	N,M
Hosaki Problem	$f_{16} = (1 - 8x_1 + 7x_1^2 - 7/3x_1^3 + 1/4x_1^4)x_2^2 \exp(-x_2)$, subject to $0 \leq x_1 \leq 5, 0 \leq x_2 \leq 6$	[0,5] [0,6]	-2.3458	2	$1.0E-06$	N,M
McCormick	$f_{17}(x) = \sin(x_1 + x_2) + (x_1 - x_2)^2 - \frac{3}{2}x_1 + \frac{5}{2}x_2 + 1$	$-1.5 \leq x_1 \leq 4, -3 \leq x_2 \leq 3$	$f(-0.547, -1.547) = -1.9133$	30	$1.0E-04$	N,M
Meyer and Roth Problem	$f_{18}(x) = \sum_{i=1}^{5}(\frac{x_1x_3t_i}{1+x_1t_i+x_2v_i} - y_i)^2$	[-10 10]	$f(3.13, 15.16, 0.78) = 0.4E-04$	3	$1.0E-03$	U,N
Shubert	$f_{19}(x) = -\sum_{i=1}^{5} i\cos((i+1)x_1+1)\sum_{i=1}^{5} i\cos((i+1)x_2+1)$	[-10 10]	$f(7.0835, 4.8580) = -186.7309$	2	$1.0E-05$	S,M
Sinusoidal	$f_{20}(x) = -[A\prod_{i=1}^{D}\sin(x_i - z) + \prod_{i=1}^{D} sin(B(x_i - z))],\ A = 2.5, B = 5, z = 30$	[0 180]	$f(90 \vec{+} z) = -(A+1)$	10	$1.0E-02$	N,M

Table 2: Comparison based on Success Rate (SR), TP: Test Problems

TP	PLSMO	SMO	TP	PLSMO	SMO
f_1	100	100	f_{11}	**63**	55
f_2	100	100	f_{12}	100	100
f_3	100	100	f_{13}	**44**	41
f_4	100	100	f_{14}	100	100
f_5	**100**	98	f_{15}	100	100
f_6	**100**	88	f_{16}	**12**	4
f_7	100	100	f_{17}	100	100
f_8	100	100	f_{18}	100	100
f_9	97	97	f_{19}	100	100
f_{10}	100	100	f_{20}	**100**	68

Table 3: Comparison based on Average number of Function Evolutions (AFE), TP: Test Problems

TP	PLSMO	SMO	TP	PLSMO	SMO
f_1	13174.79	12642.3	f_{11}	**143075.57**	153265.51
f_2	11198.07	10725.66	f_{12}	6140.89	5963.76
f_3	**31217.67**	32438.7	f_{13}	**112397.94**	123292.66
f_4	**121187.1**	142800.47	f_{14}	**11986.99**	12351.24
f_5	**62503.35**	84627.81	f_{15}	1229.54	1234.53
f_6	**109047.36**	171308.42	f_{16}	**176103.19**	200076.76
f_7	**1315.74**	1512.72	f_{17}	724.83	724.68
f_8	**17720.53**	50540.99	f_{18}	1927.24	1950.32
f_9	**36332.34**	44965.16	f_{19}	**3661.39**	4365.9
f_{10}	**14405.7**	15435	f_{20}	**27244.63**	155242.05

Boxplot analysis have been carried out for both PLSMO and SMO algorithms to evaluate consolidated performances. In boxplot analysis tool empirical distribution of data is presented graphically [(Williamson, Parker, & Kendrick, 1989)]. The boxplots for $PLSMO$ and SMO are shown in Figure 1. It is clear from the boxplots of $PLSMO$ and SMO that $PLSMO$ is better than SMO algorithm as interquartile range and median are comparatively low.

Further, it is clear from figure 1 that the distributions are skewed (the quartiles have different distances to the means). This means, a non-parametric test "the Mann-Whitney U test" is used to analyze the significant differences of the results. It is a test that does not make assumptions about the underlying distribution. The Mann-Whitney U rank sum test [(Mann & Whitney, 1947)] is performed at 5% level of significance and respective results are shown in Table 7. In this table, '+' sign indicates that the PLSMO is better than the SMO algorithm, '=' sign indicates a similarity and '-' sign indicates that the algorithm is not better or the difference is quite small. The more number of '+' sign of Table 7, shows the superiority of PLSMO over SMO.

The convergence speed of PLSMO is also compared with SMO through acceleration rate (AR) [(Rahnamayan, Tizhoosh, & Salama, 2008)], which is calculated by equation 9. It is clear from equation 9 that the PLSMO converges fast as compared to SMO for $AR > 1$. The reported results are shown in table 6. It is clear from table 6 that the PLSMO converges faster than the SMO algorithm in most of the cases for the considered test problems.

Table 4: Comparison based on Mean Error (ME), TP: Test Problems

TP	PLSMO	SMO	TP	PLSMO	SMO
f_1	**8.96E-06**	8.87E-06	f_{11}	**7.17E-01**	1.36E+00
f_2	**8.68E-06**	8.49E-06	f_{12}	**7.63E-06**	7.26E-06
f_3	**9.40E-06**	9.26E-06	f_{13}	**1.83E-05**	1.91E-05
f_4	**9.65E-03**	9.15E-03	f_{14}	**4.85E-14**	4.52E-14
f_5	**8.42E-06**	5.31E-03	f_{15}	4.90E-01	4.90E-01
f_6	**9.54E-06**	1.29E-05	f_{16}	1.02E-05	1.11E-05
f_7	5.36E-06	5.35E-06	f_{17}	8.79E-05	8.76E-05
f_8	**8.66E-04**	7.97E-04	f_{18}	1.95E-03	1.95E-03
f_9	**1.12E-04**	1.00E-04	f_{19}	5.31E-06	4.79E-06
f_{10}	6.69E-05	6.52E-05	f_{20}	**8.37E-03**	1.03E-02

Table 5: Comparison based on Standard Deviation (SD), TP: Test Problems

TP	PLSMO	SMO	TP	PLSMO	SMO
f_1	**8.15E-07**	8.37E-07	f_{11}	**1.99E+00**	3.55E+00
f_2	**9.49E-07**	1.20E-06	f_{12}	**1.47E-06**	1.86E-06
f_3	**4.93E-07**	9.32E-07	f_{13}	1.50E-05	1.46E-05
f_4	**4.95E-04**	9.68E-04	f_{14}	3.12E-14	2.94E-14
f_5	**1.58E-06**	5.22E-02	f_{15}	5.88E-03	5.18E-03
f_6	**8.60E-07**	1.10E-05	f_{16}	3.75E-06	2.24E-06
f_7	2.81E-06	2.85E-06	f_{17}	7.07E-06	6.53E-06
f_8	**1.54E-04**	1.92E-04	f_{18}	2.90E-06	2.66E-06
f_9	**1.16E-04**	8.53E-05	f_{19}	5.94E-06	5.50E-06
f_{10}	2.47E-05	2.50E-05	f_{20}	**1.54E-03**	6.32E-03

Table 6: Acceleration Rate (AR) for PLSMO Vs SMO algorithms

TP	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
PLSMO Vs SMO	0.959583	0.957813	1.039113	1.178347	1.353972	1.570954	1.14971	2.852115	1.237607	1.071451
TP	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}
PLSMO Vs SMO	1.071221	0.971156	1.09693	1.030387	1.004058	1.136134	0.999793	1.011976	1.192416	5.698079

$$AR = \frac{AFE_{SMO}}{AFE_{PLSMO}}, \tag{9}$$

6. Application of $PLSMO$ in Lower Order system Modeling

Lower order system modeling i.e. model order reduction (MOR) is a part of electrical engineering specially control system and power system. The real world control system and power system models are of very high order. By using model order reduction techniques these original higher

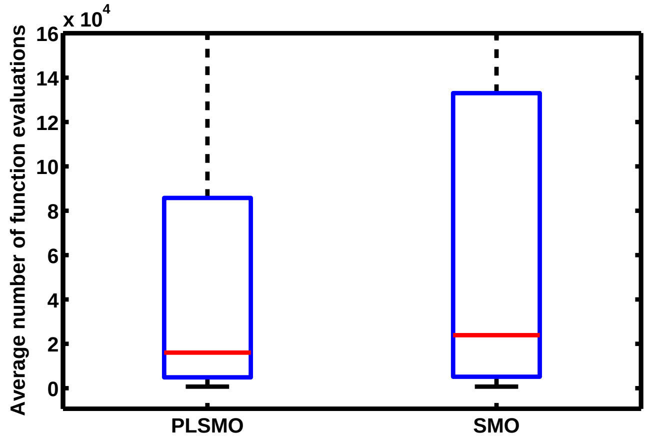

Figure 1: Boxplots graphical representation for average number of function evaluations

Table 7: Comparison of algorithms based on the Mann-Whitney U rank sum test at a $\alpha = 0.05$ significance level and mean number of function evaluations, TP: Test Problem.

TP	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
PLSMO Vs SMO	-	-	+	+	+	+	+	+	+	+
TP	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}
PLSMO Vs SMO	+	-	+	+	=	+	=	=	+	+

order models can be reduced to quite lower order, non complex models with keeping the similar properties of original systems. Although many conventional approaches by (C. Chen & Shieh, 1968; Krishnamurthy & Seshadri, 1978; T. Chen et al., 1979; Gustafson, 1966; Davison, 1966; T. N. Lucas, 1983) of MOR guarantee about the stability factor of new reduced order model but some time the model may turn to a non minimum phase. The detail description of the MOR problem is given as follows:

For an n^{th} order single input single output linear time invariant system

$$G(s) = \frac{N(s)}{D(s)} = \frac{\sum_{i=0}^{n-1} a_i s^i}{\sum_{i=0}^{n} b_i s^i} \tag{10}$$

Here a_i and b_i are known constants.
The aim is to get a reduced r^{th} order model in transfer function form R(s), where $r < n$ as equation. The reduced order model should retain almost similar characteristics of original model.

$$R(s) = \frac{N_r(s)}{D_r(s)} = \frac{\sum_{i=0}^{r-1} a'_i s^i}{\sum_{i=0}^{r} b'_i s^i} \tag{11}$$

Here a_i' and b_i' are unknown constants.
Mathematically, the Integral Square Error of step responses of the original and the reduced system can be expressed by error index J as (Gopal, 2002),

$$J = \int_0^\infty [y(t) - y_r(t)]^2 \, dt. \tag{12}$$

Here $y(t)$ is the unit step response of the original system and $y_r(t)$ is the unit step response of the reduced system. J is error index which is a function of a_i' and b_i'. The aim is to determine a_i' and b_i' of reduced order model so J is minimized.

In this paper, to solve the MOR problem, a modified objective function is used which is based on both ISE and IRE [(Bansal & Sharma, 2012)]. The considered objective function is shown in equation 13.

$$objective\_value = |ISE| + \frac{|IRE_R - IRE_O|}{IRE_R + IRE_O} \tag{13}$$

Here, the IRE for the original and the reduced order models are given by:

$$IRE = \int_0^\infty g(t)^2 \, dt. \tag{14}$$

where, $g(t)$ is the impulse response of the system.
Therefore, both, ISE and IRE, are used to construct the objective function for minimizing the ISE and difference between IRE of high order model and reduced order model.

here, ISE is integral square error of difference between responses given by equation (12), IRE_O and IRE_R are the impulse response energy response for original and reduced order model. It is advantageous to use such system because it minimizes ISE as well as IRE of both higher and reduced order model.

7. Results for Various Examples of Lower Order System Modeling

In this section, the PLSMO and SMO algorithms are applied to solve the complex lower order system modeling problems. Here, total four MOR problems are considered for getting the simulation results as shown in Table 7. The best solution obtained out of 100 runs is reported as the global optimal solution. The reported solutions are in the form of step and impulse responses. The results available in literature are compared with that of $PLSMO$, SMO, and other stochastic as well as deterministic methods.
Tables 9 - 12 present the original and the reduced systems for examples 1 - 4 respectively. In these tables results obtained by $PLSMO$ and SMO are compared with that of the basic DE, Pade approximation method, Routh approximation method and other earlier reported results. The result shown in these tables reveals that for example 1, ISE and IRE obtained using $PLSMO$ and SMO algorithms are almost same as original system values. For example 2, 3, and 4, $PLSMO$ provides least value of ISE and almost similar IRE as compared to original systems.

The unit step responses and impulse responses of the original and the reduced systems using $PLSMO$, SMO, DE, Pade Approximation, and Routh Approximation for examples 1, 2, 3, and 4 are shown in Figure 2 respectively. It is clear from Figure 2 that the step response and impulse response of $PLSMO$ is more near to the original systems as compared to the other considered strategies. Therefore, it may be stated that the PLSMO may be considered to solve the lower order system modeling problems.

8. Conclusion

In this paper, to enhance the exploitation capability of spider monkey optimization (SMO) algorithm, a power law based local search strategy is incorporated with it. To analyze the performance of the proposed strategy, namely PLSMO, it is tested over 20 well known benchmark functions. The competitiveness of the proposed PLSMO is established through various tests like Mann-Whitney U rank sum test, Boxplots, Acceleration Rate etc. Further, to see the robustness of the proposed strategy, it is applied to solve the lower order system modeling problems in which the complex higher order models of power system are reduced in almost similar lower order, non complex systems having similar properties of the original systems. The proposed PLSMO are applied on four higher order complex systems to get the respective lower order systems. The reported results are compared with the SMO and other deterministic, and meta-heuristics presented in the literature. The simulation results depict that the PLSMO algorithm may be considered as a competitive method in exploring the parameters during system modeling problems.

References

Aguirre, L. (1992). The least squares padé method for model reduction. *International journal of systems science*, *23*(10), 1559–1570.

Bansal, J. C., & Sharma, H. (2012). Cognitive learning in differential evolution and its application to model order reduction problem for single-input single-output systems. *Memetic Computing*, *4*(3), 209–229.

Bansal, J. C., Sharma, H., Jadon, S. S., & Clerc, M. (2014). Spider monkey optimization algorithm for numerical optimization. *Memetic Computing*, *6*(1), 31–47.

Bhuvana, J., & Aravindan, C. (2015). Memetic algorithm with preferential local search using adaptive weights for multi-objective optimization problems. *Soft Computing*, *20*(4), 1365–1388. Retrieved from `http://dx.doi.org/10.1007/s00500-015-1593-9` doi:

Caponio, A., Neri, F., & Tirronen, V. (2009). Super-fit control adaptation in memetic differential evolution frameworks. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, *13*(8), 811–831.

Chen, C., & Shieh, L. (1968). A novel approach to linear model simplification. *International Journal of Control*, *8*(6), 561–570.

Chen, T., Chang, C., & Han, K. (1979). Reduction of transfer functions by the stability-equation method. *Journal of the Franklin Institute*, *308*(4), 389–404.

Chen, X., Ong, Y., Lim, M., & Tan, K. (2011). A multi-facet survey on memetic computation. *IEEE Transactions on Evolutionary Computation*, *15*(5), 591–607.

Davison, J. E. (1966). A method for simplifying linear dynamic systems. *IEEE Trans. Automatic Control*, *AC-11;1*, 93-101.

Eydgahi, A., Shore, E., Anne, P., Habibi, J., & Moshiri, B. (2003). A matlab toolbox for teaching model order reduction techniques. In *International conference on engineering education.*

Fister, I., Fister Jr, I., Brest, J., & Žumer, V. (2012). Memetic artificial bee colony algorithm for large-scale global optimization. *Arxiv preprint arXiv:1206.1074*.

Gallo, C., Carballido, J., & Ponzoni, I. (2009). Bihea: A hybrid evolutionary approach for microarray biclustering. *Advances in Bioinformatics and Computational Biology*, 36–47.

Gopal, M. (2002). *Control systems: principles and design*. Tata McGraw-Hill Education.

Gustafson, R. D. (1966). A paper and pencil control system design. *Trans ASME J. Basic Engg.*, 329-336.

Hickin, J., & Sinha, N. (1978). Canonical forms for aggregated models. *International Journal of Control*, *27*(3), 473–485.

Hickin, J., & Sinha, N. K. (1976). Reduction of linear system by canonical forms. *Electronics Letters*, *12;21*, 551-553.

Hu, X. (1987). Ff-padé method of model reduction in frequency domain. *Automatic Control, IEEE Transactions on*, *32*(3), 243–246.

Hutton, M., & Friedland, B. (1975). Routh approximations for reducing order of linear, time-invariant systems. *Automatic Control, IEEE Transactions on*, *20*(3), 329–337.

Hwang, C. (1984). Mixed method of routh and ise criterion approaches for reduced-order modeling of continuous-time systems. *Journal of dynamic systems, measurement, and control*, *106*(4), 353–356.

Kang, F., Li, J., & Ma, Z. (2011). Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions. *Information Sciences*, *181(16)*, 3508–3531.

Kang, F., Li, J., Ma, Z., & Li, H. (2011). Artificial bee colony algorithm with local search for numerical optimization. *Journal of Software*, *6*(3), 490–497.

Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. *Techn. Rep. TR06, Erciyes Univ. Press, Erciyes.*

Krishnamurthy, V., & Seshadri, V. (1978). Model reduction using the routh stability criterion. *Automatic Control, IEEE Transactions on*, *23*(4), 729–731.

Lamba, S., Gorez, R., & Bandyopadhyay, B. (1988). New reduction technique by step error minimization for multivariable systems. *International journal of systems science*, *19*(6), 999–1009.

Lastra, M., Molina, D., & Benítez, J. M. (2015). A high performance memetic algorithm for extremely high-dimensional problems. *Information Sciences*, *293*, 35–58.

Lee, J., & Kim, D.-W. (2015). Memetic feature selection algorithm for multi-label classification. *Information Sciences*, *293*, 80–96.

Lucas, T. (1986). Continued-fraction expansion about two or more points: a flexible approach to linear system reduction. *Journal of the Franklin Institute*, *321*(1), 49–60.

Lucas, T. N. (1983). Factor division: a useful algorithm in model reduction. *IEE Proceedings Control Theory and Applications. 130 (6).*

Luus, R. (1999). Optimal reduction of linear systems. *Journal of the Franklin Institute*, *336*(3), 523–532.

Mann, H., & Whitney, D. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, *18*(1), 50–60.

Marshall, S. (1983). Comments on viability of methods for generating stable reduced order models. *Automatic Control, IEEE Transactions on*, *28*(5), 630–631.

Mezura-Montes, E., & Velez-Koeppel, R. (2010). Elitist artificial bee colony for constrained real-parameter optimization. In *2010 congress on evolutionary computation (cec2010), ieee service center, barcelona, spain* (pp. 2068–2075).

Mininno, E., & Neri, F. (2010). A memetic differential evolution approach in noisy optimization. *Memetic Computing*, *2*(2), 111–135.

Mishra, R., & Wilson, D. (1980). A new algorithm for optimal reduction of multivariable systems. *International Journal of Control*, *31*(3), 443–466.

Mittal, A. K., Prasad, R., & Sharma., S. P. (2004). Reduction of linear dynamic systems using an error minimization technique. *Journal of Institution of Engineers India, IE(I) Journal*, *EL, 84*, 201-206.

M. Quang, C. M. L., & Pan., C. W. (1987). Model reduction by power decomposition and frequency response matching. *IEEE Trans. Auto. Control*, *AC-32*, 59-62.

Mukherjee, S., & Mishra, R. (1987). Order reduction of linear systems using an error minimization technique. *Journal of the Franklin Institute*, *323*(1), 23–32.

Mukherjee, S., & Mittal, R. (2005). Order reduction of linear discrete systems using a genetic algorithm. *Applied Mathematical Modelling*, *29*(6), 565–578.

Mukherjee, S., Mittal, R., et al. (2005). Model order reduction using response-matching technique. *Journal of the Franklin Institute*, *342*(5), 503–519.

Neri, C. M. P. E., Ferrante; Cotta. (Vol. 379, 2012). *Handbook of memetic algorithms* (S. S. in Computational Intelligence., Ed.). Springer, Studies in Computational Intelligence,.

Neri, F., & Tirronen, V. (2009). Scale factor local search in differential evolution. *Memetic Computing*, *1*(2), 153–171.

Nguyen, Q., Ong, Y., & Lim, M. (2009). A probabilistic memetic framework. *IEEE Transactions on Evolutionary Computation*, *13*(3), 604–623.

Ong, Y., Lim, M., & Chen, X. (2010). Memetic computationpast, present & future [research frontier]. *Computational Intelligence Magazine, IEEE*, *5*(2), 24–31.

Pade, H. (1892). Sur la representation approaches dune function par des fraction rationelles. *Annales Scientifiques de IEcole Normale Supieure*, *9*, 1-93.

Pal, J. (1979). Stable reduced-order pade approximants using the routh-hurwitz array. *Electronics letters*, *15*(8), 225–226.

Passino, K. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *Control Systems Magazine, IEEE*, *22*(3), 52–67.

Paynter, H. M., & Takahashi, Y. (1956). A new method of evaluation dynamic response of counter flow and parallel flow heat exchangers. *Trans. ASME J. Dynam. Syst. Meas. Control*, *27*, 749-753.

P. O. Gutman, C. F. M., & Molander, P. (1982). Contributions to the model reduction problem. *IEEE Trans. Automatic Control*, *AC-27; 2*, 454-455.

Prasad, R., & Pal, J. (1991). Stable reduction of linear systems by continued fractions. *Institution of Engineers India part EL Electrical Enginggering Division*, *72*, 113–113.

Rahnamayan, S., Tizhoosh, H., & Salama, M. (2008). Opposition-based differential evolution. *Evolutionary Computation, IEEE Transactions on*, *12*(1), 64–79.

Repoussis, P., Tarantilis, C., & Ioannou, G. (2009). Arc-guided evolutionary algorithm for the vehicle routing problem with time windows. *Evolutionary Computation, IEEE Transactions on*, *13*(3), 624–647.

Richer, J.-M., Goëffon, A., & Hao, J.-K. (2009). A memetic algorithm for phylogenetic reconstruction with maximum parsimony. In *Evolutionary computation, machine learning and data mining in bioinformatics* (pp. 164–175). Springer.

Sanathanan Stanley, B., et al. (1987). A comprehensive methodology for siso system order reduction. *Journal of the Franklin Institute*, *323*(1), 1–21.

Shamash, Y. (1975). Linear system reduction using pade approximation to allow retention of dominant modes. *International Journal of Control*, *21*(2), 257–272.

Sharma, H., Bansal, J. C., & Arya, K. (2013). Opposition based lévy flight artificial bee colony. *Memetic Computing*, *5*(3), 213–227.

Sharma, H., Bansal, J. C., & Arya, K. (2014). Power law-based local search in artificial bee colony. *International Journal of Artificial Intelligence and Soft Computing*, *4*(2), 164–194.

Sharma, H., Bansal, J. C., Arya, K. V., & Yang, X.-S. (2016). Lvy flight artificial bee colony algorithm. *International Journal of Systems Science*, *47*(11), 2652-2670. Retrieved from `http://dx.doi.org/10.1080/00207721.2015.1010748` doi:

Sharma, H., Jadon, S. S., Bansal, J. C., & Arya, K. (2013). Lèvy flight based local search in differential evolution. In *Swarm, evolutionary, and memetic computing* (pp. 248–259). Springer.

Shoji, K., et al. (1985). Model reduction for a class of linear dynamic systems. *Journal of the Franklin Institute*, *319*(6), 549–558.

Sinha, N., & Bereznai, G. (1971). Optimum approximation of high-order systems by low-order models. *International Journal of Control*, *14*(5), 951–959.

Sinha, N., & Pille, W. (1971). A new method for reduction of dynamic systems. *International Journal of Control*, *14*(1), 111–118.

Sivanandam, S., & Deepa, S. (2007). A genetic algorithm and particle swarm optimization approach for lower order modelling of linear time invariant discrete systems. In *Conference on computational intelligence and multimedia applications, 2007. international conference on* (Vol. 1, pp. 443–447).

Tang, K., Mei, Y., & Yao, X. (2009). Memetic algorithm with extended neighborhood search for capacitated arc routing problems. *IEEE Transactions on Evolutionary Computation*, *13*(5), 1151–1166.

Vesterstrøm, J., & Thomsen, R. (2004). A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems. In *Evolutionary computation, 2004. cec2004. congress on* (Vol. 2, pp. 1980–1987).

Vilbe, P., & Calvez, L. (1990). On order reduction of linear systems using an error minimization technique. *Journal of the Franklin Institute*, *327*(3), 513–514.

Wang, H., Wang, D., & Yang, S. (2009). A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems. *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, *13*(8), 763–780.

Williamson, D. F., Parker, R. A., & Kendrick, J. S. (1989). The box plot: a simple visual method to interpret data. *Annals of internal medicine*, *110*(11), 916–921.

Table 8: List of MOR Problem Examples

S. No.	**Source**	**Original Model**
1	(Shamash, 1975)	$G_1(s) = \frac{18s^7+514s^6+5982s^5+36380s^4+122664s^3+222088s^2+185760s+40320}{s^8+36s^7+546s^6+4536s^5+22449s^4+67284s^3+118124s^2+109584s+40320}$
2	(T. Lucas, 1986)	$G_2(s) = \frac{8169.13s^3+50664.97s^2+9984.32s+500}{100s^4+10520s^3+52101s^2+10105s+500}$
3	(Aguirre, 1992)	$G_4(s) = \frac{4.269s^3+5.10s^2+3.9672s+0.9567}{4.3992s^4+9.0635s^3+8.021s^2+5.362s+1}$
4	(Eydgahi, Shore, Anne, Habibi, & Moshiri, 2003)	$G_5(s) = \frac{s^4+35s^3+291s^2+1093s+1700}{s^9+9s^8+66s^7+294s^6+1029s^5+2541s^4+4684s^3+5856s^2+4629s+1700}$

Table 9: Comparison of the Methods for example 1

Method of order reduction	**Reduced Models;** $R_1(s)$	**ISE**	**IRE**
Original [(Eydgahi et al., 2003)]	$G_1(s)$	-	21.740
PLSMO	$\frac{17.3224s+5.3708}{s^2+7.0251s+5.3708}$	0.8×10^{-3}	21.740
SMO	$\frac{17.3484s+5.3995}{s^2+7.0495s+5.3995}$	0.8×10^{-3}	21.731
DE [(Bansal & Sharma, 2012)]	$\frac{20s+5.6158}{s^2+9.2566s+5.6158}$	0.3729×10^{-1}	21.908
Pade Approximation [(Bansal & Sharma, 2012)]	$\frac{15.1s+4.821}{s^2+5.993s+4.821}$	1.6177	19.426
Routh Approximation [(Bansal & Sharma, 2012)]	$\frac{1.99s+0.4318}{s^2+1.174s+0.4318}$	1.9313	1.8705
(P. O. Gutman & Molander, 1982)	$\frac{4[133747200s+203212800]}{85049280s^2+552303360s+812851200}$	8.8160	4.3426
(Hutton & Friedland, 1975)	$\frac{1.98955s+0.43184}{s^2+1.17368s+0.43184}$	18.3848	1.9868
(Krishnamurthy & Seshadri, 1978)	$\frac{155658.6152s+40320}{65520s^2+75600s+40320}$	17.5345	2.8871
(Mittal, Prasad, & Sharma., 2004)	$\frac{7.0908s+1.9906}{s^2+3s+2}$	6.9159	9.7906
(Mukherjee & Mishra, 1987)	$\frac{7.0903s+1.9907}{s^2+3s+2}$	6.9165	9.7893
(Mukherjee, Mittal, et al., 2005)	$\frac{11.3909s+4.4357}{s^2+4.2122s+4.4357}$	2.1629	18.1060
(Pal, 1979)	$\frac{151776.576s+40320}{65520s^2+75600s+40320}$	17.6566	2.7581
(Prasad & Pal, 1991)	$\frac{17.98561s+500}{s^2+13.24571s+500}$	18.4299	34.1223
(Shamash, 1975)	$\frac{6.7786s+2}{s^2+3s+2}$	7.3183	8.9823

Table 10: Comparison of the Methods for example 2

Method of order reduction	**Reduced Models;** $R_2(s)$	**ISE**	**IRE**
Original [(Eydgahi et al., 2003)]	$G_2(s)$	-	34.069
PLSMO	$\frac{103.3s+867.9}{s^2+169.4s+867.9}$	0.36×10^{-2}	34.0699
SMO	$\frac{107.1579043s+940.3090254}{s^2+182.3271464s+940.3090254}$	0.4202969×10^{-2}	34.068226
DE [(Bansal & Sharma, 2012)]	$\frac{220.8190s+35011.744}{s^2+1229.4502s+35011.744}$	0.4437568×10^{-2}	34.069218
(?, ?)	$\frac{93.7562s+1}{s^2+100.10s+10}$	0.8964×10^{-2}	43.957
Pade Approximation [(Bansal & Sharma, 2012)]	$\frac{23.18s+2.36}{s^2+23.75s+2.36}$	0.46005×10^{-2}	11.362
Routh Approximation [(Bansal & Sharma, 2012)]	$\frac{0.1936s+0.009694}{s^2+0.1959s+0.009694}$	2.3808	0.12041
(P. O. Gutman & Molander, 1982)	$\frac{0.19163s+0.00959}{s^2+0.19395s+0.00959}$	2.4056	0.11939
(T. Chen et al., 1979)	$\frac{0.38201s+0.05758}{s^2+0.58185s+0.05758}$	1.2934	0.17488
(Marshall, 1983)	$\frac{83.3333s+499.9998}{s^2+105s+500}$	0.193×10^{-2}	35.450

Table 11: Comparison of the Methods for example 3

Method of order reduction	**Reduced Models;** $R_3(s)$	**ISE**	**IRE**
Original [(Eydgahi et al., 2003)]	$G_3(s)$	-	0.54536
PLSMO	$\frac{0.8105s+2.7989}{s^2+2.9508s+2.7989}$	0.338×10^{-1}	0.5454
SMO	$\frac{0.8357s+3.311}{s^2+3.418s+3.311}$	0.339×10^{-1}	0.5454
DE [(Bansal & Sharma, 2012)]	$\frac{1.0755s+9.567}{s^2+9.4527s+10}$	0.364×10^{-1}	0.54535
(?, ?)	$\frac{4.0056s+0.9567}{8.021s^2+5.362s+1}$	0.22372	0.27187
Pade Approximation [(Bansal & Sharma, 2012)]	$\frac{1.869s+0.5585}{s^2+2.663s+0.5838}$	∞	0.75619
Routh Approximation [(Bansal & Sharma, 2012)]	$\frac{0.6267s+0.1511}{s^2+0.847s+0.158}$	∞	0.31715

Table 12: Comparison of the Methods for example 4

Method of order reduction	**Reduced Models;** $R_4(s)$	**ISE**	**IRE**
Original [(Eydgahi et al., 2003)]	$G_4(s)$	-	0.47021
PLSMO	$\frac{-0.6364s+1.054}{s^2+1.553s+1.054}$	0.204×10^{-1}	0.4698
SMO	$\frac{-0.603s+1.166}{s^2+2.031s+1.166}$	0.269×10^{-1}	0.4516
DE [(Bansal & Sharma, 2012)]	$\frac{-0.8068s+1.3083}{s^2+2.0221s+1.3083}$	0.302×10^{-1}	0.4845
Pade Approximation [(Bansal & Sharma, 2012)]	$\frac{-0.8153s+1.392}{s^2+2.081s+1.392}$	0.330×10^{-1}	0.49414
Routh Approximation [(Bansal & Sharma, 2012)]	$\frac{0.2643s+0.411}{s^2+1.119s+0.411}$	0.131	0.21486

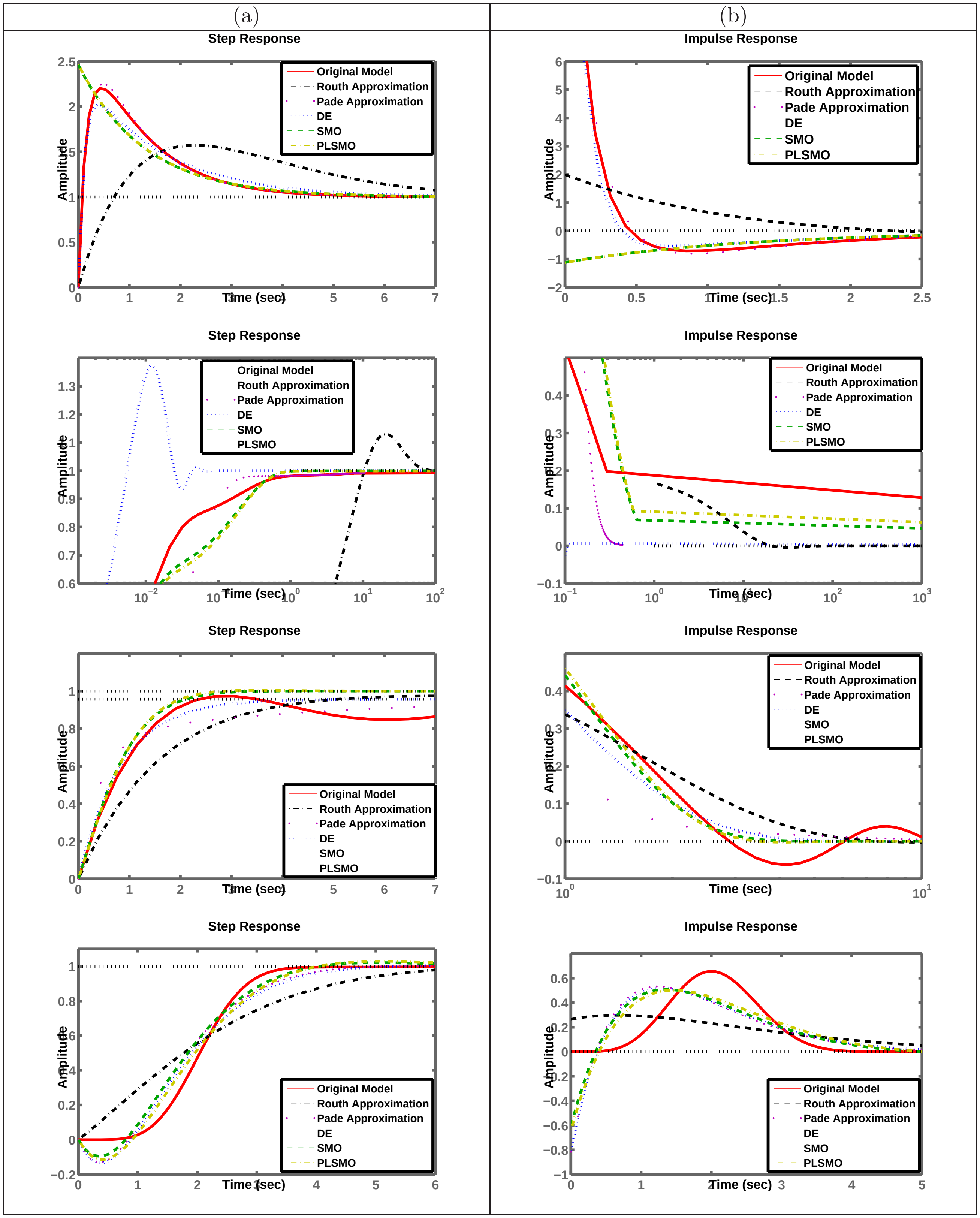

Figure 2: Responses comparison for example 1,2,3, and 4 respectively for (a) step response, (b) impulse response