Evolutionary Techniques for Model Order Reduction of Large Scale Linear Systems

S. Panda, J. S. Yadav, N. P. Patidar and C. Ardil

Abstract**—**Recently, genetic algorithms (GA) and particle swarm optimization (PSO) technique have attracted considerable attention among various modern heuristic optimization techniques. The GA has been popular in academia and the industry mainly because of its intuitiveness, ease of implementation, and the ability to effectively solve highly non-linear, mixed integer optimization problems that are typical of complex engineering systems. PSO technique is a relatively recent heuristic search method whose mechanics are inspired by the swarming or collaborative behavior of biological populations. In this paper both PSO and GA optimization are employed for finding stable reduced order models of single-input- single-output large-scale linear systems. Both the techniques guarantee stability of reduced order model if the original high order model is stable. PSO method is based on the minimization of the Integral Squared Error (ISE) between the transient responses of original higher order model and the reduced order model pertaining to a unit step input. Both the methods are illustrated through numerical example from literature and the results are compared with recently published conventional model reduction technique.

Keywords**—**Genetic Algorithm, Particle Swarm Optimization, Order Reduction, Stability, Transfer Function, Integral Squared Error.

I. INTRODUCTION

THE exact analysis of high order systems (HOS) is both tedious and costly as HOS are often too complicated to be used in real problems. Hence simplification procedures based on physical considerations or using mathematical approaches are generally employed to realize simple models for the original HOS. The problem of reducing a high order system to its lower order system is considered important in analysis, synthesis and simulation of practical systems. Bosley and Lees [1] and others have proposed a method of reduction based on the fitting of the time moments of the system and its reduced model, but these methods have a serious disadvantage that the reduced order model may be unstable even though the original high order system is stable.

To overcome the stability problem, Hutton and Friedland [2], Appiah [3] and Chen *et. al.* [4] gave different methods, called stability based reduction methods which make use of some stability criterion. Other approaches in this direction include the methods such as Shamash [5] and Gutman *et. al.* [6]. These methods do not make use of any stability criterion but always lead to the stable reduced order models for stable systems.

Some combined methods are also given for example Shamash [7], Chen *et. al.* [8] and Wan [9]. In these methods the denominator of the reduced order model is derived by some stability criterion method while the numerator of the reduced model is obtained by some other methods [6, 8, 10].

In recent years, one of the most promising research fields has been “Evolutionary Techniques”, an area utilizing analogies with nature or social systems. Evolutionary techniques are finding popularity within research community as design tools and problem solvers because of their versatility and ability to optimize in complex multimodal search spaces applied to non-differentiable objective functions. Recently, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) techniques appeared as a promising algorithm for handling the optimization problems. GA can be viewed as a general-purpose search method, an optimization method, or a learning mechanism, based loosely on Darwinian principles of biological evolution, reproduction and ‘‘the survival of the fittest’’ [11]. GA maintains a set of candidate solutions called population and repeatedly modifies them. At each step, the GA selects individuals from the current population to be parents and uses them to produce the children for the next generation. In general, the fittest individuals of any population tend to reproduce and survive to the next generation, thus improving successive generations. However, inferior individuals can, by chance, survive and also reproduce. GA is well suited to and has been extensively applied to solve complex design optimization problems because it can handle both discrete and continuous variables, non-linear objective and constrain functions without requiring gradient information [12–16].

PSO is inspired by the ability of flocks of birds, schools of fish, and herds of animals to adapt to their environment, find rich sources of food, and avoid predators by implementing an information sharing approach. PSO technique was invented in

S. Panda is working as Professor in the Department of Electrical and Electronics Engineering, National Institute of Science and Technology, Berhampur, Orissa, INDIA. (e-mail: panda_sidhartha@rediffmail.com).

J.S.Yadav is working as Assistant Professor in Electronics and Communication Engg. Department, MANIT Bhopal, India (e-mail: jsy1@rediffmail.com)

aculty N.P.patidar is working as a Senior Lecturer in Electrical Engineering Department, MANIT, Bhopal, India. (e-mail: nppatidar@yahoo.com)

C. Ardil is with National Academy of Aviation, AZ1045, Baku, Azerbaijan, Bina, 25th km, NAA (e-mail: cemalardil@gmail.com).

the mid 1990s while attempting to simulate the choreographed, graceful motion of swarms of birds as part of a sociocognitive study investigating the notion of collective intelligence in biological populations [17]. In PSO, a set of randomly generated solutions propagates in the design space towards the optimal solution over a number of iterations based on large amount of information about the design space that is assimilated and shared by all members of the swarm [12, 18-20]. Both GA and PSO are similar in the sense that these two techniques are population-based search methods and they search for the optimal solution by updating generations. Since the two approaches are supposed to find a solution to a given objective function but employ different strategies and computational effort, it is appropriate to compare their performance.

In this paper, two evolutionary methods for order reduction of large scale linear systems are presented. In both the methods, evolutionary optimization techniques are employed for the order reduction where both the numerator and denominator coefficients of ROM by minimizing an Integral Squared Error (ISE) criterion. The obtained results are compared with a recently published conventional method to show their superiority.

II. Statement of The Problem

The Let the n^{th} order system and its reduced model ($r < n$) be given by the transfer functions:

$$G(s) = \frac{\sum_{i=0}^{n-1} d_i s^i}{\sum_{j=0}^{n} e_j s^i} \tag{1}$$

$$R(s) = \frac{\sum_{i=0}^{r-1} a_i s^i}{\sum_{j=0}^{r} b_j s^i} \tag{2}$$

where a_i, b_j, d_i, e_j, are scalar constants.

The objective is to find a reduced r^{th} order reduced model $R(s)$ such that it retains the important properties of $G(s)$ for the same types of inputs.

III. Overview of Genetic Algorithm (GA)

Genetic algorithm (GA) has been used to solve difficult engineering problems that are complex and difficult to solve by conventional optimization methods. GA maintains and manipulates a population of solutions and implements a survival of the fittest strategy in their search for better solutions. The fittest individuals of any population tend to reproduce and survive to the next generation thus improving successive generations. The inferior individuals can also survive and reproduce.

Implementation of GA requires the determination of six fundamental issues: chromosome representation, selection function, the genetic operators, initialization, termination and evaluation function. Brief descriptions about these issues are provided in the following sections.

A. Chromosome representation

Chromosome representation scheme determines how the problem is structured in the GA and also determines the genetic operators that are used. Each individual or chromosome is made up of a sequence of genes. Various types of representations of an individual or chromosome are: binary digits, floating point numbers, integers, real values, matrices, etc. Generally natural representations are more efficient and produce better solutions. Real-coded representation is more efficient in terms of CPU time and offers higher precision with more consistent results.

B. Selection function

To produce successive generations, selection of individuals plays a very significant role in a genetic algorithm. The selection function determines which of the individuals will survive and move on to the next generation. A probabilistic selection is performed based upon the individual's fitness such that the superior individuals have more chance of being selected. There are several schemes for the selection process: roulette wheel selection and its extensions, scaling techniques, tournament, normal geometric, elitist models and ranking methods.

The selection approach assigns a probability of selection P_j to each individuals based on its fitness value. In the present study, normalized geometric selection function has been used. In normalized geometric ranking, the probability of selecting an individual P_i is defined as:

$$Pi = q^{'}(1-q)^{r-1} \tag{3}$$

$$q^{'} = \frac{q}{1-(1-q)^{P}} \tag{4}$$

where,

q = probability of selecting the best individual

r = rank of the individual (with best equals 1)

P = population size

C. Genetic operators

The basic search mechanism of the GA is provided by the genetic operators. There are two basic types of operators: crossover and mutation. These operators are used to produce new solutions based on existing solutions in the population. Crossover takes two individuals to be parents and produces two new individuals while mutation alters one individual to produce a single new solution. The following genetic operators are usually employed: simple crossover, arithmetic crossover and heuristic crossover as crossover operator and uniform mutation, non-uniform mutation, multi-non-uniform mutation, boundary mutation as mutation operator. Arithmetic crossover and non-uniform mutation are employed in the present study as genetic operators. Crossover generates a random number r from a uniform distribution from 1 to m and creates two new individuals by using equations:

$$x_i' = \begin{Bmatrix} x_i, \; if \;\; i < r \\ y_i \;\; otherwise \end{Bmatrix} \quad (5)$$

$$y_i' = \begin{Bmatrix} y_i, \; if \;\; i < r \\ x_i \;\; otherwise \end{Bmatrix} \quad (6)$$

Arithmetic crossover produces two complimentary linear combinations of the parents, where r = U (0, 1):

$$\bar{X}' = r\,\bar{X} + (1-r)\,\bar{Y} \quad (7)$$

$$\bar{Y}' = r\,\bar{Y} + (1-r)\,\bar{X} \quad (8)$$

Non-uniform mutation randomly selects one variable j and sets it equal to an non-uniform random number.

$$x_i' = \begin{Bmatrix} x_i + (b_i - x_i)\, f(G) & if \;\; r_1 < 0.5, \\ x_i + (x_i + a_i)\, f(G) & if \;\; r_1 \geq 0.5, \\ x_i, & otherwise \end{Bmatrix} \quad (9)$$

where,

$$f(G) = (r_2(1 - \frac{G}{G_{\max}}))^b \quad (10)$$

r_1, r_2 = uniform random nos. between 0 to 1.

G = current generation.

$G_{\max}$ = maximum no. of generations.

b = shape parameter.

D. Initialization, termination and evaluation function

An initial population is needed to start the genetic algorithm procedure. The initial population can be randomly generated or can be taken from other methods.

The GA moves from generation to generation until a stopping criterion is met. The stopping criterion could be maximum number of generations, population convergence criteria, lack of improvement in the best solution over a specified number of generations or target value for the objective function.

Evaluation functions or objective functions of many forms can be used in a GA so that the function can map the population into a partially ordered set. The computational flowchart of the GA optimization process employed in the present study is given in Fig. 1.

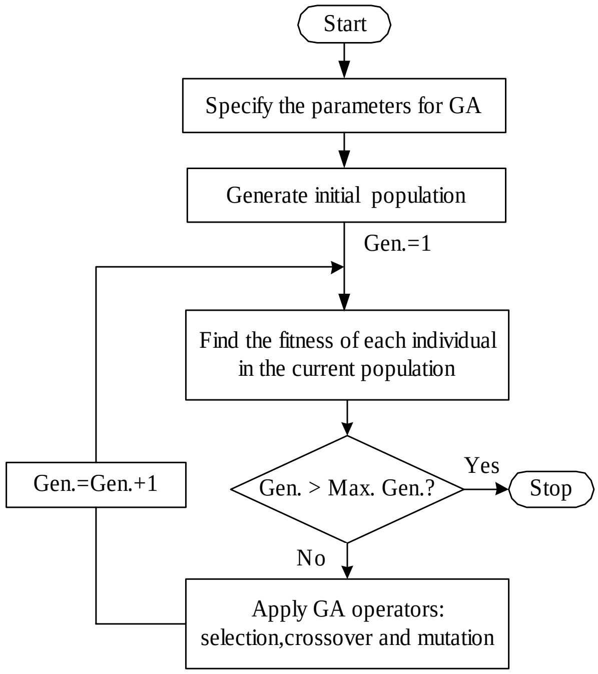

Fig. 1. Flowchart of genetic algorithm

IV. Particle Swarm Optimization Method

In conventional mathematical optimization techniques, problem formulation must satisfy mathematical restrictions with advanced computer algorithm requirement, and may suffer from numerical problems. Further, in a complex system consisting of number of controllers, the optimization of several controller parameters using the conventional optimization is very complicated process and sometimes gets struck at local minima resulting in sub-optimal controller parameters. In recent years, one of the most promising research field has been “Heuristics from Nature”, an area utilizing analogies with nature or social systems. Application of these heuristic optimization methods a) may find a global optimum, b) can produce a number of alternative solutions, c) no mathematical restrictions on the problem formulation, d) relatively easy to implement and e) numerically robust. Several modern heuristic tools have evolved in the last two decades that facilitates solving optimization problems that were previously difficult or impossible to solve. These tools include evolutionary computation, simulated annealing, tabu search, genetic algorithm, particle swarm optimization, etc. Among these heuristic techniques, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) techniques appeared as promising algorithms for handling the optimization problems. These techniques are finding popularity within research community as design tools and problem solvers because of their versatility and ability to optimize in complex multimodal search spaces applied to non-differentiable objective functions.

The PSO method is a member of wide category of swarm intelligence methods for solving the optimization problems. It is a population based search algorithm where each individual is referred to as particle and represents a candidate solution.

Each particle in PSO flies through the search space with an adaptable velocity that is dynamically modified according to its own flying experience and also to the flying experience of the other particles. In PSO each particles strive to improve themselves by imitating traits from their successful peers. Further, each particle has a memory and hence it is capable of remembering the best position in the search space ever visited by it. The position corresponding to the best fitness is known as *pbest* and the overall best out of all the particles in the population is called *gbest* [12].

The modified velocity and position of each particle can be calculated using the current velocity and the distances from the $pbest_{j,g}$ to $gbest_g$ as shown in the following formulas [12,17-20]:

$$v_{j,g}^{(t+1)} = w * v_{j,g}^{(t)} + c_1 * r_1(\ ) * (pbest_{j,g} - x_{j,g}^{(t)}) + c_2 * r_2(\ ) * (gbest_g - x_{j,g}^{(t)}) \tag{11}$$

$$x_{j,g}^{(t+1)} = x_{j,g}^{(t)} + v_{j,g}^{(t+1)} \tag{12}$$

With $j = 1,2,...,n$ and $g = 1,2,...,m$

Where,

n = number of particles in the swarm

m = number of components for the vectors v_j and x_j

t = number of iterations (generations)

$v_{j,g}^{(t)}$ = the *g*-th component of the velocity of particle *j* at iteration t , $v_g^{\min} \le v_{j,g}^{(t)} \le v_g^{\max}$;

w = inertia weight factor

c_1, c_2 = cognitive and social acceleration factors respectively

r_1, r_2 = random numbers uniformly distributed in the range (0, 1)

$x_{j,g}^{(t)}$ = the *g*-th component of the position of particle *j* at iteration *t*

$pbest_j$ = *pbest* of particle *j*

$gbest$ = *gbest* of the group

The *j*-th particle in the swarm is represented by a *d*-dimensional vector $x_j = (x_{j,1}, x_{j,2}, \ldots\ldots, x_{j,d})$ and its rate of position change (velocity) is denoted by another *d*-dimensional vector $v_j = (v_{j,1}, v_{j,2}, \ldots\ldots, v_{j,d})$. The best previous position of the *j*-th particle is represented as $pbest_j = (pbest_{j,1}, pbest_{j,2}, \ldots\ldots, pbest_{j,d})$. The index of best particle among all of the particles in the swarm is represented by the $gbest_g$. In PSO, each particle moves in the search space with a velocity according to its own previous best solution and its group's previous best solution. The velocity update in a PSO consists of three parts; namely momentum, cognitive and social parts. The balance among these parts determines the performance of a PSO algorithm. The parameters c_1 and c_2 determine the relative pull of *pbest* and *gbest* and the parameters r_1 and r_2 help in stochastically varying these pulls. In the above equations, superscripts denote the iteration number. Fig.2. shows the velocity and position updates of a particle for a two-dimensional parameter space. The computational flow chart of PSO algorithm employed in the present study for the model reduction is shown in Fig. 3.

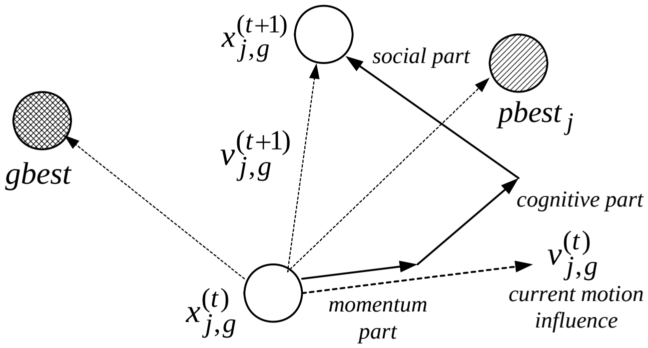

Fig. 2. Description of velocity and position updates in particle swarm optimization for a two dimensional parameter space

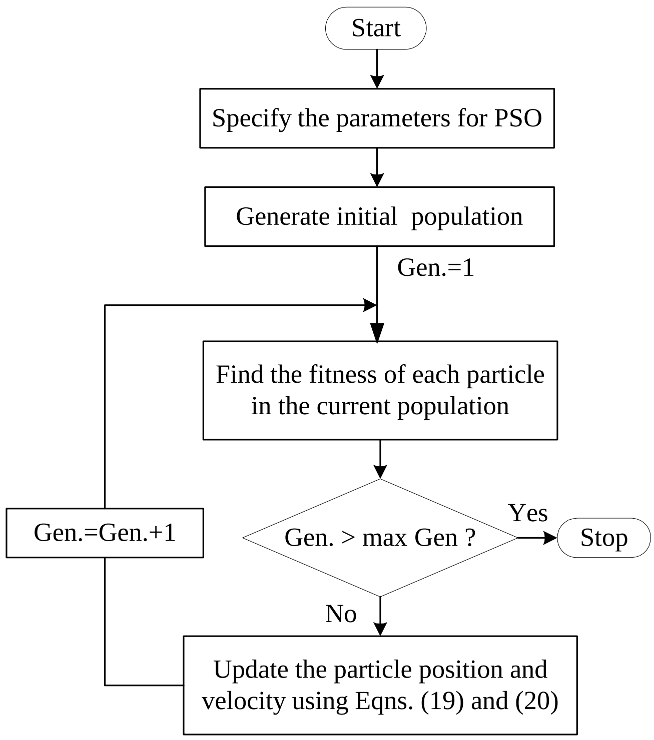

Fig. 3. Flowchart of PSO for order reduction

V. Numerical Examples

Let us consider the system described by the transfer function described by transfer function [21, 22]:

$$G(s)=\frac{s^3+7s^2+24s+24}{(s+1)(s+2)(s+3)(s+4)} \qquad (13)$$

For which a second order reduced model $R_2(s)$ is desired.

A. Application of PSO and GA

While applying PSO and GA, a number of parameters are required to be specified. An appropriate choice of the parameters affects the speed of convergence of the algorithm.

Implementation of PSO, several parameters are required to be specified, such as c_1 and c_2 (cognitive and social acceleration factors, respectively), initial inertia weights, swarm size, and stopping criteria. These parameters should be selected carefully for efficient performance of PSO. The constants c_1 and c_2 represent the weighting of the stochastic acceleration terms that pull each particle toward *pbest* and *gbest* positions. Low values allow particles to roam far from the target regions before being tugged back. On the other hand, high values result in abrupt movement toward, or past, target regions. Hence, the acceleration constants were often set to be 2.0 according to past experiences. Suitable selection of inertia weight, w, provides a balance between global and local explorations, thus requiring less iteration on average to find a sufficiently optimal solution. As originally developed, w often decreases linearly from about 0.9 to 0.4 during a run [17, 18]. One more important point that more or less affects the optimal solution is the range for unknowns. For the very first execution of the program, wider solution space can be given, and after getting the solution, one can shorten the solution space nearer to the values obtained in the previous iterations.

Implementation of GA normal geometric selection is employed which is a ranking selection function based on the normalized geometric distribution. Arithmetic crossover takes two parents and performs an interpolation along the line formed by the two parents. Non-uniform mutation changes one of the parameters of the parent based on a non-uniform probability distribution. This Gaussian distribution starts wide, and narrows to a point distribution as the current generation approaches the maximum generation.

The objective function J is defined as an integral squared error of difference between the responses given by the expression:

$$J=\int_0^{t_\infty}[y(t)-y_r(t)]^2\,dt \qquad (14)$$

Where

$y(t)$ and $y_r(t)$ are the unit step responses of original and reduced order systems.

B. Results

The reduced 2nd order model employing PSO technique is obtained as follows:

$$R_2(s)=\frac{2.9319s+7.8849}{3.8849s^2+11.4839s+7.8849} \qquad (15)$$

The reduced 2nd order model employing GA technique is obtained as follows:

$$R_2(s)=\frac{5.2054s+8.989}{6.6076^2+14.8941s+8.989} \qquad (16)$$

The convergence of objective function with the number of generations for both PSO and GA is shown in Fig. 4. The unit step responses of original and reduced systems by both the methods are shown in Figs. 5 and 6 for PSO and GA method respectively. For comparison, the unit step response of a recently published ROM obtained by conventional Routh Approximation method [21] is also shown in Figs. 5 and 6.

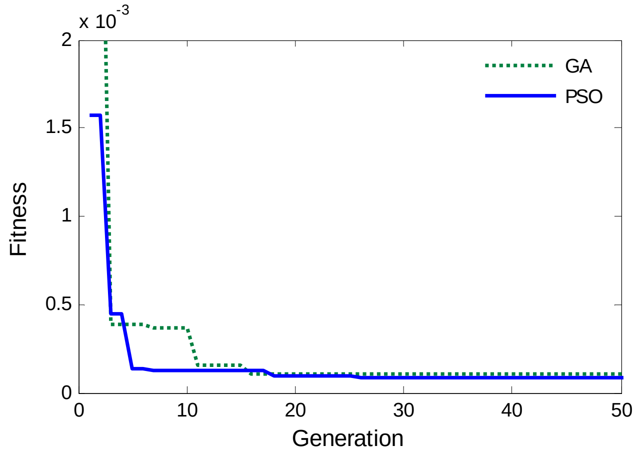

Fig. 4. Flowchart of PSO for order reduction

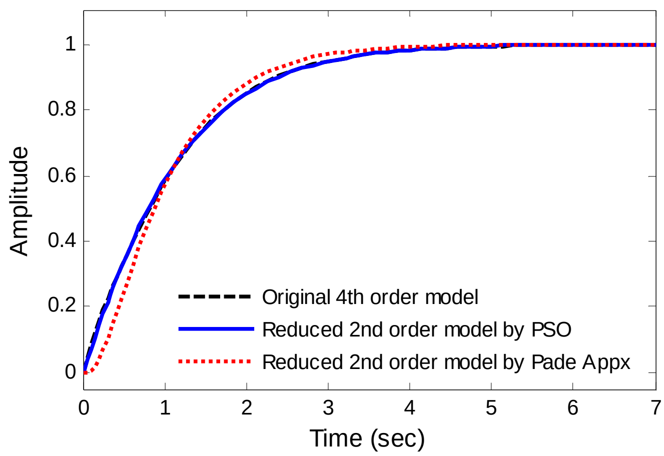

Fig. 5. Step Responses of original model and ROM by PSO

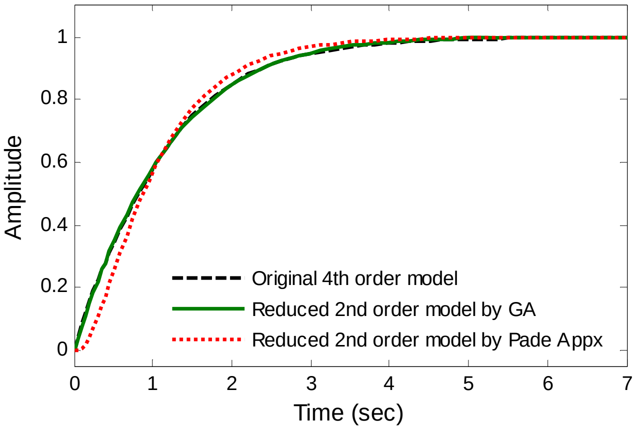

Fig. 6. Step Responses of original model and ROM by GA

It can be seen that the steady state responses of both the proposed reduced order models are exactly matching with that of the original model. Also, compared to conventional method of reduced models, the transient response of evolutionary reduced model by PSO and GA is very close to that of original model.

VI. Comparison of Methods

The performance comparison of both the proposed algorithm for order reduction techniques is given in Table I. The comparison is made by computing the error index known as integral square error ISE [23, 24] in between the transient parts of the original and reduced order model, is calculated to measure the goodness/quality of the [i.e. the smaller the ISE, the closer is $R(s)$ to $G(s)$, which is given by:

$$ISE = \int_0^{t_\infty} [y(t) - y_r(t)]^2 \, dt \qquad (17)$$

Where $y(t)$ and $y_r(t)$ are the unit step responses of original and reduced order systems for a second- order reduced respectively. This error index is calculated for various reduced order models which are obtained by us and compared with the other well known order reduction methods available in the literature.

VI. Conclusion

In this paper, two evolutionary methods for reducing a high order large scale linear system into a lower order system have been proposed. Particle swarm optimization and genetic algorithm methods based evolutionary optimization techniques are employed for the order reduction where both the numerator and denominator coefficients of reduced order model are obtained by minimizing an Integral Squared Error (ISE) criterion. The obtained results are compared with a recently published conventional method and other existing well known methods of model order reduction to show their superiority. It is clear from results presented in Table 1 that both the proposed methods give minimum ISE error compared to any other order reduction technique.

TABLE I: COMPARISON OF METHODS

Method	Reduced model	ISE
Proposed evolutionary method: PSO	$\dfrac{2.9319s + 7.8849}{3.8849s^2 + 11.4839s + 7.8849}$	8.2316×10^{-5}
Proposed conventional method: GA	$\dfrac{5.2054s + 8.989}{6.6076^2 + 14.8941s + 8.989}$	8.6581×10^{-5}
Routh Approx. [21]	$\dfrac{4.4713 - 0.189762s}{4.4713 + 4.76187s + s^2}$	0.008
Parthasarathy et al. [25]	$\dfrac{0.6997 + s}{0.6997 + 1.45771s + s^2}$	0.0303
Shieh and Wei [26]	$\dfrac{2.3014 + s}{2.3014 + 5.7946s + s^2}$	0.1454
Prasad and Pal [27]	$\dfrac{34.2465 + s}{34.2465 + 239.8082s + s^2}$	1.6885
J Pal [28]	$\dfrac{24 + 16.0008s}{24 + 42s + 30s^2}$	0.0118

References

[1] M.J.Bosley and F.P.Lees, “A survey of simple transfer function derivations from high order state variable models”, *Automatica*, Vol. 8, pp. 765-775, !978.

[2] M.F.Hutton and B. Fried land, “Routh approximations for reducing order of linear time- invariant systems”, *IEEE Trans. Auto. Control*, Vol. 20, pp 329-337, 1975.

[3] R.K.Appiah, “Linear model reduction using Hurwitz polynomial approximation”, *Int. J. Control*, Vol. 28, no. 3, pp 477-488, 1978.

[4] T. C. Chen, C.Y.Chang and K.W.Han, “Reduction of transfer functions by the stability equation method”, *Journal of Franklin Institute*, Vol. 308, pp 389-404, 1979.

[5] Y.Shamash, “Truncation method of reduction: a viable alternative”, *Electronics Letters*, Vol. 17, pp 97-99, 1981.

[6] P.O.Gutman, C.F.Mannerfelt and P.Molander, “Contributions to the model reduction problem”, *IEEE Trans. Auto. Control*, Vol. 27, pp 454-455, 1982.

[7] Y. Shamash, “Model reduction using the Routh stability criterion and the Pade approximation technique”, *Int. J. Control*, Vol. 21, pp 475-484, 1975.

[8] T.C.Chen, C.Y.Chang and K.W.Han, “Model Reduction using the stability-equation method and the Pade approximation method”, *Journal of Franklin Institute*, Vol. 309, pp 473-490, 1980.

[9] Bai-Wu Wan, “Linear model reduction using Mihailov criterion and Pade approximation technique”, *Int. J. Control*, Vol. 33, pp 1073-1089, 1981.

[10] V.Singh, D.Chandra and H.Kar, “Improved Routh-Pade Approximants: A Computer-Aided Approach”, *IEEE Trans. Auto. Control*, Vol. 49. No. 2, pp292-296, 2004.

[11] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989.

[12] S.Panda and N.P.Padhy, "Comparison of Particle Swarm Optimization and Genetic Algorithm for FACTS-based Controller Design", *Applied Soft Computing.* Vol. 8, Issue 4, pp. 1418-1427, 2008.

[13] S.Panda and N.P.Padhy, "Application of Genetic Algorithm for PSS and FACTS based Controller Design", ***International Journal of Computational Methods***, Vol. 5, Issue 4, pp. 607-620, 2008.

[14] S.Panda and R.N.Patel, "Transient Stability Improvement by Optimally Located STATCOMs Employing Genetic Algorithm" *International Journal of Energy Technology and Policy,* Vol. 5, No. 4, pp. 404-421, 2007.

[15] S.Panda and R.N.Patel, "Damping Power System Oscillations by Genetically Optimized PSS and TCSC Controller" *International Journal of Energy Technology and Policy*, Inderscience, Vol. 5, No. 4, pp. 457-474, 2007.

[16] S.Panda and R.N.Patel, "Optimal Location of Shunt FACTS Controllers for Transient Stability Improvement Employing Genetic Algorithm", *Electric Power Components and Systems*, Taylor and Francis, Vol. 35, No. 2, pp. 189-203, 2007.

[17] J. Kennedy and R.C.Eberhart, "Particle swarm optimization", *IEEE Int.Conf. on Neural Networks*, IV, 1942-1948, Piscataway, NJ, 1995.

[18] S.Panda, N.P.Padhy, R.N.Patel, "Power System Stability Improvement by PSO Optimized SSSC-based Damping Controller", *Electric Power Components & Systems*, Taylor and Francis, Vol. 36, No. 5, pp. 468-490, 2008.

[19] S.Panda and N.P.Padhy, "Optimal location and controller design of STATCOM using particle swarm optimization", *Journal of the Franklin Institute*, Elsevier, Vol.345, pp. 166-181, 2008.

[20] S.Panda, N.P.Padhy and R.N.Patel, "Robust Coordinated Design of PSS and TCSC using PSO Technique for Power System Stability Enhancement", *Journal of Electrical Systems,* Vol. 3, No. 2, pp. 109-123, 2007.

[21] C. B. Vishwakarma and R.Prasad, "Clustering Method for Reducing Order of Linear System using Pade Approximation", IETE Journal of Research, Vol. 54, Issue 5, pp. 326-330, 2008.

[22] S.Mukherjee, and R.N.Mishra, Order reduction of linear systems using an error minimization technique, *Journal of Franklin Inst*. Vol. 323, No. 1, pp. 23-32, 1987.

[23] S.Panda, S.K.Tomar, R.Prasad, C.Ardil, "Reduction of Linear Time-Invariant Systems Using Routh-Approximation and PSO", *International Journal of Applied Mathematics and Computer Sciences*, Vol. 5, No. 2, pp. 82-89, 2009.

[24] S.Panda, S.K.Tomar, R.Prasad, C.Ardil, "Model Reduction of Linear Systems by Conventional and Evolutionary Techniques", *International Journal of Computational and Mathematical Sciences*, Vol. 3, No. 1, pp. 28-34, 2009.

[25] R.Parthasarathy, and K. N. Jayasimha, System reduction using stability equation method and modified Cauer continued fraction, *Proc. IEEE* Vol. 70, No. 10, pp. 1234-1236, Oct. 1982.

[26] L.S. hieh, and Y.J.Wei, A mixed method for multivariable system reduction, *IEEE Trans. Autom. Control*, Vol. AC-20, pp. 429-432, 1975.

[27] R.Prasad, and J.Pal, Stable reduction of linear systems by continued fractions, *J. Inst. Eng. India, IE (I) J.EL*, Vol. 72, pp. 113-116, Oct. 1991.

[28] J. Pal, Stable reduced –order Pade approximants using the Routh-Hurwitz array, *Electronics letters*, Vol. 15, No. 8, pp. 225-226, April 1979.

Sidhartha Panda is a Professor at National Institute of Science and Technology, Berhampur, Orissa, India. He received the Ph.D. degree from Indian Institute of Technology, Roorkee, India in 2008, M.E. degree in Power Systems Engineering in 2001 and B.E. degree in Electrical Engineering in 1991. Earlier he worked as Associate Professor in KIIT University, Bhubaneswar, India & VITAM College of Engineering, Andhra Pradesh, India and Lecturer in the Department of Electrical Engineering, SMIT, Orissa, India.His areas of research include power system transient stability, power system dynamic stability, FACTS, optimisation techniques, distributed generation and wind energy.

Jigyendra Sen Yadav is working as Assistant Professor in Electronics and Communication Engineering Department, MANIT, Bhopal, India. He received the B.Tech.degree from GEC Jabalpur in Electronics and Communication Engineering in 1995 and M.Tech. degree from MANIT, Bhopal in Digital Communication in 2002. Presently he is persuing Ph.D. His area of interest includes Optimization techniques, Model Order Reduction, Digital Communication, Signal and System and Control System.

Narayana Prasad Patidar obtained his PhD from Indian Institute of Technology, Roorkee, in 2008. Presently, he is working as a Senior Lecturer in Electrical Engineering Department, MANIT, Bhopal, India. His research interests are voltage stability, security analysis, power system stability and intelligent techniques.

C. Ardil is with National Academy of Aviation, AZ1045, Baku, Azerbaijan, Bina, 25th km, NAA