Reduced Order Modeling of Linear MIMO Systems Using Particle Swarm Optimization

Dia Abu-Al-Nadi
Dept. Electrical Engineering
University of Jordan
Amman, Jordan
dnadi@ju.edu.jo

Othman MK Alsmadi
Dept. Electrical Engineering
University of Jordan
Amman, Jordan
Othman_mk@yahoo.com

Zaer S Abo-Hammour
Dept. Mechatronics Engineering
University of Jordan
Amman, Jordan
zaer_hr@yahoo.com

Abstract—**In this work, a model order reduction (MOR) technique for a linear multivariable system is proposed using the combined advantage of retaining the dominant poles and the error minimization using the particle swarm optimization. The state space matrices of the reduced order system are chosen such that the dominant eigenvalues of the full order system are unchanged. The other system parameters are chosen using the particle swarm optimization with objective function to minimize the mean squared errors between the outputs of the full order system and the outputs of the reduced order model when the inputs are unit step. The proposed algorithm has been applied successfully, a 10th order Multiple-Input_Multiple-Output (MIMO) linear model for a practical power system was reduced to a 4th order and an 8th order Single-Input-Single-Output (SISO) system was reduced to a 2nd order.**

Keywords-Model Order Reduction;MIMO Systems;Particle Swarm Optimization.

I. INTRODUCTION

Many physical systems are translated into mathematical model via higher order differential equations. It is usually recommended to reduce the order of this model while keeping the dominant behavior of the original system. This will help to better understanding of the physical system, reduce computational complexity, reduce hardware complexity and simplify the controller design.

Different techniques for order reduction of linear continuous MIMO system in time domain as well as in frequency domain are available in the literature [1-4]. For model order reduction, there are different scenarios that can be performed. One scenario obtains reduced models that are completely new and not related to the original models in terms of their critical frequencies of either SISO or MIMO systems. On the other hand, another scenario obtains reduced models that preserve the original system important properties, such as dominant frequencies of either SISO or MIMO systems. It is to be noted that the later scenario is more preferable, if possible, due to its meaningful physical interpretation in obtaining similar models and due to minimum changes in the original systems [5].

The MOR problem has been investigated in literature extensively. Willcox and Perarie [6] proposed an algorithm to reduce the model order using the proper orthogonal decomposition (POD) analysis of the primal and dual systems, low-rank, reduced-range approximations to the controllability and observability gramians. Fujimoto and Scherpen [7] proposed a singular perturbation type balanced realization and model reduction for discrete non-linear dynamical systems based on Hankel singular value analysis, which preserves the related controllability and observability properties. Heydari and Pedram [3] proposed a spectrally weighted balanced truncation technique for tightly coupled integrated circuit (IC) interconnects, when the interconnected circuit parameters change because of statistical variations in the manufacturing process. Rabiei and Pedram [8] proposed a method that uses the truncated balanced realization technique as well as the Schur decomposition to develop an efficient numerical method for the order reduction of linear time invariant (LTI) systems. Gugercin et al. [9] proposed an iterative rational Krylov approach (IRKA) for optimal H_2 model reduction. Their approach is concerned with SISO-type systems only and is based on minimizing the H_2-norm. This minimization leads to a non-convex problem that can get stuck at local minima and hence optimality will not be achieved [10]. Recently, Parmar et al. [4] proposed a reduction method with pole centroid retaining in the reduced model. Their method deals with SISO systems only. In addition, it works for real poles only. Genetic algorithm (GA)-based MOR, on the contrary, has received some of the researchers' attention as well. Recently, Panda et al. [11] employed a particle swarm optimization technique to obtain a reduced-order model of SISO large-scale linear systems. Their technique is based on integral square error (ISE). Vishwakarma and Prasad [12] proposed a mixed method for reducing the order of large-scale linear systems. They have synthesized the denominator of the reduced-order transfer function (TF) using modified pole clustering, whereas the coefficients of the numerator elements are computed using GA. Parmar et al. [13] presented a technique for MOR using GA for SISO linear time systems. They have focused on obtaining a reduced-order model that maintains stability and retains the steady-state value.

In this work, the particle swarm optimization (PSO) is utilized for MOR of MIMO systems. The rest of the paper is organized as follows: Section II is the statement of the problem. In section III, the PSO algorithm is stated; section IV is designated for results and discussion, and finally, conclusions are presented in section V.

II. PROBLEM STATEMENT

MOR is investigated both for MIMO and SISO systems, for the MIMO systems, the state space representation was adopted while for the SISO systems, the transfer function model is used.

A. MOR for MIMO systems

Consider the following n^{th} order LTI system:

$$\dot{x}_f(t) = A_f x(t) + B_f u(t) \tag{1}$$

$$y_f(t) = C_f x(t) + D_f u(t) \tag{2}$$

where $x_f \in \Re^n$ is the state vector, $u \in \Re^p$, and $y_f \in \Re^m$ are the input and output vectors, respectively. The matrices A_f, D_f, C_f, and D_f are the full order system matrices with their appropriate dimensions. Let the eigenvalues of the above full order system be given as: $-\lambda_1 < -\lambda_2 < \cdots < -\lambda_n$.

On the other hand, consider the reduced order LTI system with order r:

$$\dot{x}_r(t) = A_r x_r(t) + B_r u(t) \tag{3}$$

$$y_r(t) = C_r x_r(t) + D_r u(t) \tag{4}$$

where $x_r \in \Re^r$ is the state vector of the reduced order system, $u \in \Re^p$, and $y_r \in \Re^m$ are the input and output vectors,respectively. The matrices A_r, B_r, C_r, and D_r are the reduced order system matrices with their appropriate dimensions. The eigenvalues of reduced order system are chosen to be the dominant eigenvalues of the full order system given as: $-\lambda_1 < -\lambda_2 < \cdots < -\lambda_r$.

The A_r matrix is chosen to be a diagonal matrix with the dominant eigenvalues are assigned as the diagonal elements. The elements of other matrices are chosen by the PSO.

B. MOR for SISO Systems

Consider an nth order SISO LTI system with the following transfer function:

$$G(s) = \frac{a_0 + a_1 s + a_2 s^2 + \cdots + a_{n-1} s^{n-1}}{b_0 + b_1 s + b_2 s^2 + \cdots + b_n s^n} \tag{5}$$

with the eigenvalues of the system to be $-\lambda_1 < -\lambda_2 < \cdots < -\lambda_n$.

Let the reduced order model be

$$G_r(s) = \frac{\alpha_0 + \alpha_1 s + \alpha_2 s^2 + \cdots + \alpha_{r-1} s^{r-1}}{\beta_0 + \beta_1 s + \beta_2 s^2 + \cdots + \beta_r s^r} \tag{6}$$

The coefficients of the denominator of $G_r(s)$ are chosen such that the eignenvalues of the low order system are the dominant roots of the full order system as follows: $-\lambda_1 < -\lambda_2 < \cdots < -\lambda_r$, while the coefficients of the numerator are chosen by the PSO algorithm.

III. THE PSO ALGORITHM

The PSO is a multiple-agent optimization algorithm developed by Kennedy and Eberhart [14] in 1995. The major advantage of the PSO over other stochastic optimization methods is its simplicity. The standard PSO is implemented by assuming a swarm of particles (called trial solutions). Each particle moves in the solution space by improving its position according to suitable updating equations. This is performed on the basis of information on each particle's previous best performance and the best previous performance of its neighbors (global best). The updating equations for the PSO are sequentially applied at each individual. Unlike other stochastic algorithms, the PSO is based upon the cooperation among the trial-solutions and not on their competition. In order to describe the steps of the PSO algorithm, we will define the given parameters and the necessary specifications. Hence, two parts can be classified:

A. Definitions and parameters setting:

- Set the full order system parameters.
- Set appropriate level step inputs to the system.
- Simulate the outputs, yf, of the full order system with a suitable sampling time.
- Choose a suitable order of the reduced order system based on the dominant eigenvalues.
- Set the PSO parameters:
- The size of the particle, P.
    - o The number of particles in the swarm, *M*.

- o The counter of iteration (I = 1) and the maximum number of iterations, L_{max}.

- Definition of the solution space: A reasonable range for the parameters should be chosen. This requires specifications of the minimum and maximum values for each parameter.

- Definition of a fitness function: This step is the link between the optimization algorithm and the physical problem in hand. A good fitness function that is well representative of the parameters is crucial in the PSO algorithm. In this work, the fitness function is defined by the weighted-mean-squared error

$$WMSE = \frac{1}{N}\sum_{k=1}^{m}\sum_{i=1}^{N} w_k [y_f(k,i) - y_r(k,i)]^2 \tag{7}$$

where N is the number of samples, m is the number of outputs, w_k is a weight used to emphasize the k^{th} error, $y_f(k,i)$ is the i^{th} sample of the k^{th} output of full order system and $y_r(k,i)$ is the ith sample of the k^{th} output of reduced order system.

In this paper, the fitness function used in the PSO algorithm is the minimization of WMSE

$$fitness = \min(WMSE) \tag{8}$$

B. The main steps of the PSO algorithm.

Step 1- Initialization:

The PSO starts by randomly initializing the position matrix, X, the velocity matrix, V, and the personal best matrix, P, of each particle in the swarm such that

$$\mathbf{X} = \begin{bmatrix} \mathrm{x}_1 \\ \vdots \\ \mathrm{x}_i \\ \vdots \\ \mathrm{x}_M \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1P} \\ \vdots & \vdots & \ddots & \vdots \\ x_{i1} & x_{i2} & \cdots & x_{iP} \\ \vdots & \vdots & \ddots & \vdots \\ x_{M1} & x_{M2} & \cdots & x_{MP} \end{bmatrix} \tag{9}$$

and the velocity matrix is

$$\mathbf{V} = \begin{bmatrix} \mathrm{v}_1 \\ \vdots \\ \mathrm{v}_i \\ \vdots \\ \mathrm{v}_M \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1P} \\ \vdots & \vdots & \ddots & \vdots \\ v_{i1} & v_{i2} & \cdots & v_{iP} \\ \vdots & \vdots & \ddots & \vdots \\ v_{M1} & v_{M2} & \cdots & v_{MP} \end{bmatrix} \tag{10}$$

The personal best position can be defined by the matrix

$$\mathbf{P} = \begin{bmatrix} \mathrm{pbest}_1 \\ \vdots \\ \mathrm{pbest}_i \\ \vdots \\ \mathrm{pbest}_M \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1P} \\ \vdots & \vdots & \ddots & \vdots \\ p_{i1} & p_{i2} & \cdots & p_{iP} \\ \vdots & \vdots & \ddots & \vdots \\ p_{M1} & p_{M2} & \cdots & p_{MP} \end{bmatrix} \tag{11}$$

The global best solution gbest is the row of personal best matrix, **P**, with the best fitness function given as

$$gbest = \min(fitness(\mathrm{pbest}_\mathrm{i})) = \begin{bmatrix} g_1 & g_2 & \cdots & g_P \end{bmatrix} \tag{12}$$

In most cases, the initial position is the only location encountered by each particle at the start of the algorithm. Hence, it will be regarded as the particle's respective personal best.

Step 2- Particle Updating.

For each iteration, the particles will be moved into the solution space. The algorithm will act on each particle such that each particle will move in a direction to improve its fitness function. The following steps summarize the action encountered on each particle in the swarm:

a) *Update the Particle's velocity.* The particle's velocity will be updated according to three vector elements: the first is the relative location to its corresponding pbesti; the second is its relative location to gbest; and the third vector is a scaled factor of the old velocity. For each particle, the velocity update is

$$\mathbf{v}_\mathbf{i}^{\mathbf{t+1}} = w^{t+1}\mathbf{v}_\mathbf{i}^\mathbf{t} + c_1\eta_1(\mathbf{pbest}_\mathbf{i}^\mathbf{t} - \mathbf{x}_\mathbf{i}^\mathbf{t}) + c_2\eta_2(\mathbf{gbest} - \mathbf{x}_\mathbf{i}^\mathbf{t}) \tag{13}$$

The superscript t+1 and t refer to the time index of the next and the current iterations. η_1 and η_2 are two uniformly random numbers in the interval [0,1]. A good choice for c1 and c2 are both 2.0. The parameter wt is a number called the inertial weight which is a scaling factor of the previous velocity of the particles. It has been demonstrated that PSO algorithms converges faster if w is chosen to be linearly damped with iterations [14]. A good choice to start with is w1=0.9 at the first iteration and linearly decreases to wLmax=0.4 with the last iteration.

b) Movement Updating of the particles:

Once the velocity of each particle is determined, the position will be updated

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \Delta t\, \boldsymbol{v}_i^t \tag{14}$$

For simplicity, Δt is chosen to be unity.

c) Evaluate the fitness function for the new position and compare it with the fitness function of the pbest, if *fitness*($\boldsymbol{x}_i$) < *fitness* (***pbest****ᵢ*) then ***pbest****ᵢ* = $\boldsymbol{x}_i$

d) Compare the fitness function of the new position with the fitness function of *gbest*

if *fitness*($\boldsymbol{x}_i$) < *fitness* (***gbest***) then ***gbest*** = $\boldsymbol{x}_i$

e) Repeat (a), (b), (c), and (d) for the whole M particles.

Step 3- Check if maximum iteration reached or a specified termination criteria is satisfied. Then, the solution is gbest. Otherwise, update w and go to the next iteration.

IV. RESULTS AND DISCUSSION

To demonstrate the proposed method of the PSO model reduction, we will consider two dynamical examples. The first one is 2-input 2-output, 10th order power system represented with its state space full order system [15]. The second example is a single input single output 8th order transfer function [16].

Example 1

Consider the following 2-input, 2-output, 10th order power system with the following state space model:

$$A_f = \begin{bmatrix} -0.5517 & 0 & -0.3091 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1695 \\ -0.0410 & 0 & -0.0350 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 314.1593 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 9.5540 & 0 & -0.8660 & -20 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0.0421 & -0.0328 \\ -0.1962 & 10.8696 & -0.1672 & 0 & 0 & -10.8696 & 0 & 0 & 0 & 0 \\ -0.9386 & 51.9849 & -0.7999 & 0 & 0 & -41.1153 & -10.8696 & 0 & 0 & 0 \\ -0.9386 & 51.9849 & -0.7999 & 0 & 0 & -41.1153 & -10.86.96 & -0.1 & 0 & 0 \\ 0 & 0 & 0 & -1000 & -1000 & 0 & 0 & 1000 & -20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1.0526 & -0.8211 \end{bmatrix}$$

$$B_f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1000 & 0 \\ 0 & 0.0926 & 0 & 0 & 0 & 0.4428 & 2.1179 & 2.1179 & 0 & 0 \end{bmatrix}^T$$

$$C_f = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.4777 & 0 & -0.0433 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$D_f = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The eigenvalues of the full order system are:

{-18.9311 ± 2.0250i, -12.1968, -9.6484, -0.2394 ± 3.2350i

-0.8972 ± 1.3560i, -2.1313, -0.1001}

The following eigenvalues were chosen to be in the reduced order model

{-0.2394 ± 3.2350i, -0.8972 ± 1.3560i}

Hence, the reduced order state space model will be

$$A_r = \begin{bmatrix} -0.2394 & -3.2350 & 0 & 0 \\ 3.2350 & -0.2394 & 0 & 0 \\ 0 & 0 & -0.8972 & -1.3560 \\ 0 & 0 & 1.3560 & -0.8972 \end{bmatrix}$$

$$B_r = \begin{bmatrix} x_1 & x_5 \\ x_2 & x_6 \\ x_3 & x_7 \\ x_4 & x_8 \end{bmatrix} \qquad C_r = \begin{bmatrix} x_9 & x_{10} & x_{11} & x_{12} \\ x_{13} & x_{14} & x_{15} & x_{16} \end{bmatrix} \quad D_r = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

The coefficients of the B_r and the C_r matrices are evaluated using the PSO algorithm as follows:

$$B_r = \begin{bmatrix} -2.4213 & -1.4544 \\ -3.5196 & 1.0311 \\ 2.5894 & -2.1142 \\ -.9881 & 0.7407 \end{bmatrix}$$ and

$$C_r = \begin{bmatrix} 1.3483 & -2.5905 & 2.4427 & -2.3762 \\ -0.1990 & -0.1990 & 0.1311 & 0.4203 \end{bmatrix}$$

Simulating both models (the full and the reduced) to a step input are shown in Figure 1.

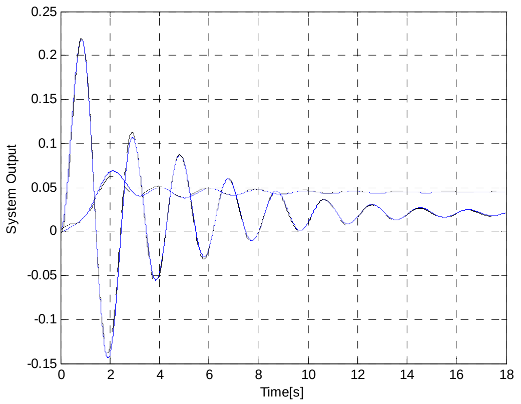

Fig.1: The two outputs of the system for both the full and the low order models caused by a step input for Example 1.(The full order outputs; solid, the reduced order outputs; dashed)

Example 2

Consider the 8th order transfer function

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

The eigenvalues of the full order system are { -1, -2, -3, -4, -5, -6, -7, -8}. The reduced order model is designed to have the following transfer function (TF):

$$G_r(s) = \frac{x_1 s + x_2}{s^2 + x_3 s + x_4}$$

Coefficients of the numerator and the denominator are evaluated using the PSO. Hence,

the reduced order system was obtained as

$$G_r(s) = \frac{17.0989s + 5.0742}{s^2 + 6.9722s + 5.1514}$$

with poles {-6.1321 and -0.8401}. Again, simulating the full and the reduced order models to a step input produced the output shown in Figure 2.

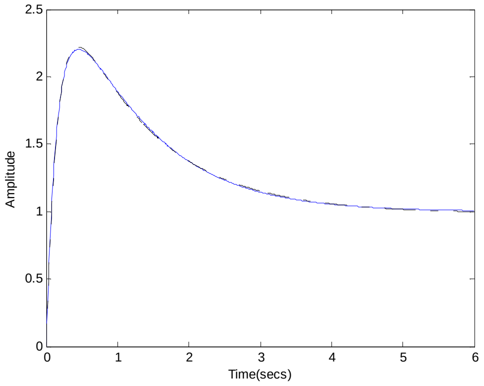

Fig.2: The output of the system for both the full and the reduced order models caused by a step input for Example 2. (The full order output; solid, the reduced order output; dashed)

V. Conclusion and Future Work

Model order reduction has been implemented by using the PSO algorithm. The behavior of the original system was preserved in the WMSE sense. The dominant poles of the full order system were kept unchanged in the reduced order model. The retaining of the dominant poles will guarantee that the overall behavior of the reduced order system will be almost the same as the original system. Based on the results obtained by the illustrative examples, it is concluded that the proposed method achieved satisfactory results. As for future work, implementation of this technique on actual physical systems will take place. Systems with very large dimensions will be considered to explore the powerfulness of the method. Also, comparison of the PSO algorithm with other evolutionary optimization techniques will be investigated.

References

[1] M.G. Safonov and R.Y. Chiang, A Schur Method for Balanced-Truncation Model Reduction, IEEE Trans on Automatic Control 34 (1989), pp. 729 –733.

[2] E. Fridman, Robust Sampled-Data H_∞ Control of Linear Singularly Perturbed Systems, IEEE Transactions on Automatic Control 51 (2006), pp. 470-475.

[3] P. Heydari and M. Pedram, Model-Order Reduction Using Variational Balanced Truncation with Spectral Shaping, IEEE Transactions on Circuits and Systems I 53 (2006), pp. 879-891.

[4] G. Parmar, S. Mukherjee, and R. Prasad, System reduction using factor division algorithm and eigen spectrum analysis, Applied Mathematical Modeling 31 (2007), pp. 2542–2552.

[5] T. Reis and T. Stykel, Balanced truncation model reduction of second-order systems, Mathematical and Computer Modeling of Dynamical Systems 14 (2008,), pp. 391-406.

[6] K. Willcox and J. Perarie, Balanced Model Reduction via the proper orthogonal decomposition, AIAA. J. ,40 , 2002, pp. 2323-2330.

[7] K. Fujimoto and J.M.A. Scharpen, Balancing and Model Reduction for Discrete-Time Nonlinear System based on Hankel Singular Value Analysis, Proceedings of the MTNS 2004, Leuven, Belgium, July 2004, pp. 343-347.

[8] P. Rabiei and M. Pedram, Model-Order Reduction of Large Circuitsusing Balanced Truncation, Proceedings of the IEEE ASP-DAC, Wanchai, Hong Kong, Jan. 1999, pp. 237-240.

[9] S. Gugercin, C. Athanasios, A.C. Antoulas, and C. Beattie, A Rational Krylov Iteration for Optimal H2 Model Reduction, Proceedings of the 17th Int. Symposium on Mathematical Theory on Networks and Systems, Kyoto, Japan, 2006, pp. 1665-1667.

[10] G. Obinata and B.D.O. Anderson, Model Reduction for Control System Design, Springer-Verlag, London, 2001.

[11] S. Panda, J.S. Yadav, N.P. Padidar, and C. Ardil, Evolutionary Techniques for Model Order Reduction of Large Scale Linear Systems, Int. J. Applied Sci. Eng. Technol.5 (2009) ,pp.22-28.

[12] G. Parmar M.K. Pandey, and V. Kumar, System Order Reduction Using GA for Unit Impulse Input and a Comparative Study Using ISE and IRE, International Conf. on Advances in Computing, Communications and Control, Mumbai, India , Jan. 2009, pp.23-24.

[13] C.B. Vishwakarma and R. Prasad, MIMO System Reduction Using Modified Pole Clustering and Genetic Algorithm, Modeling and Simulation in Engineering 2009(ID 540895) 2009, pp. 1-5.

[14] J. Kennedy and R.C. Eberhart, " Particle Swarm Optimization", in Proc. IEEE Conf. Neural Networks IV, Piscataway, NJ , 1995.

[15] G. Parmar, S. Mukherjee, and R. Prasad, Reduced order Modeling of linear MIMO Systems Using Genetic Algorithm, Int. J. simul. Model 6 (2007) 3, pp. 173-184.

[16] S. Mukherjee, M. Satakshi, and R.C. Mittal, Model order reduction using response matching technique, J. Franklin Inst. 342 (2005), pp. 503–519.