MODEL-ORDER REDUCTION OF CLASS OF NONLINEAR SYSTEMS WITH APPLICATION TO ELECTRICAL CIRCUITS

BY

SAEED SALIM AHMED BA RAYYAN

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

DECEMBER 2016

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **SAEED S. BA RAYYAN** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN ELECTRICAL ENGINEERING**.

Thesis Committee

Dr. Salim Ibrir (Adviser)

Prof. Samir Al-Baiyat (Member)

Dr. Jamil Bakhashwain (Member)

Dr. Ali Al-Shaikhi
Department Chairman

Dr. Salam A. Zummo
Dean of Graduate Studies

12/2/17

Date

I lovingly dedicate this thesis to my parents, wife, sons, brothers, sisters and friends who supported me at each step of the way.

ACKNOWLEDGMENTS

All praise be to Allah for his limitless help and guidance. Pease and blessings of Allah be upon his prophet Muhammad (PBUH).

I wish to express my deep appreciation and thanks to Dr. Salim Ibrir, thesis adviser, for his guidance and professional advice without which the success of this work would not be possible.

I wish to thank my thesis committee Prof. Samir Al-Baiyat, and Dr. Jamil Bakhashwain for their helpful remarks and cooperation.

Finally, I would like to thank Kingdom of Saudi Arabia, King Fahd University of Petroleum and Minerals (KFUPM), Hadhramout Establishment for Human Development (HEHD), family, friends and all well-wishers for their support and encouragement, which serve as a driving force for this research and my study at large.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

MOR	:	Model Order Reduction.
PSO	:	Particle Swarm Optimization.
LMI	:	Linear Matrix Inequality.
GA	:	Genetic Algorithms.
HSV	:	Hankel Singular Values.
SVD	:	Singular Value Decomposition.

THESIS ABSTRACT

NAME: Saeed S. Ba Rayyan

TITLE OF STUDY: Model-Order Reduction of Class of Nonlinear Systems with Application to Electrical Circuits

MAJOR FIELD: Electrical Engineering

DATE OF DEGREE: Dec, 2016

The objective of this thesis is the development of model-order reduction (MOR) techniques for class of nonlinear systems. The reduced order system can take the role of the original system, if the difference between its output and the output of the original system for the same input stay within the acceptable error tolerance for a given period of time.

In particular, two MOR methods will be developed, both of the method based on the projection technique, where the states of the original system are projected using a projection matrix "V". In the developed approaches, the MOR procedures are applied directly to the original states of the nonlinear system. In first method, the projection matrix "V" is obtained from the transformation matrix of the linearized system.

The other development is constructing the projection matrix "V" using the Particle Swarm Optimization "PSO", where the fitness value is the difference between the reduced system output and the output of the complete (high order) system.

The main issue of the MOR, stability preservation, is studied. This property proof that the reduced system obtained using the developed approaches is stable. Moreover, the PSO-based MOR method is tested for two nonlinear electrical circuits. As observed from the results of the numerical tests, the developed approach provide a superior accuracy results comparing with three existing MOR algorithms.

ملخص الرسالة

الاسم الكامل: **سعيد سالم أحمد باريان**

عنوان الرسالة: **تصغير النماذج للأنظمة الغير خطية مع تطبيقها على الدوائر الكهربائية**

التخصص: **هندسة كهربائية**

تاريخ الدرجة العلمية: **ديسمبر 2016**

في هذه الأطروحة تمت دراسة موضوع تصغير الانظمة لفئة من النظم الغير خطية. من خلال دراستنا قمنا باستنباط طريقتين لتصغير الأنظمة الغير خطية بناء على إسقاط متحولات الحالة للنظام الأصلي للحصول على الأنظمة المصغرة. يتم ذلك عن طريق الحصول على مصفوفة الإسقاط من خلال التمثيل الخطي للأنظمة الغير خطية. ان تصغير الأنظمة الغير خطية في مجمل الدراسات السابقة يقوم على فكرة الاسقاط للنظام التقريبي لها وليس للنظام الأصلي. ومن أجل التغلب على هذه المشكلة، نقترح تطبيق الإسقاط مباشرة على النظام الغير خطي الأصلي. أما الاستنباط الآخر هو بناء مصفوفة الإسقاط باستخدام خوارزمية Particle Swarm Optimization، حيث تكون قيمة الملائمة fitness value هو الفرق بين مخرجات النموذج الأصلي والنموذج المصغر. القضية الرئيسية في تصغير النظم هو المحافظة على استقراريه الأنظمة، حيث تم اثبات ان الأنظمة المصغرة اللي يتم الحصول عليها باستخدام هذه النماذج مستقرة. في هذه الأطروحة، تم اختبار الطرق المقترحة لاثنين من الدوائر الكهربائية غير الخطية. كما لوحظ من نتائج الاختبارات انها ذات دقة فائقة بالمقارنة مع ثلاثة خوارزميات سابقة.

CHAPTER 1

INTRODUCTION

In this chapter, a brief introduction to the Model-Order Reduction "MOR" concept is presented. Also, some of the previous MOR techniques for linear and nonlinear systems are summarized and explained. In particular, four of the main linear MOR techniques are studied in details with application to a linear system. These techniques are: the modal truncation, balanced truncation, Schur method, and Hankel norm reduction.

1.1 Introduction

Model order reduction "MOR" is an important tool that is used to avoid computational complexity of large systems. In MOR, the reduced-order system takes the role of the large original system. However, the reduced system retains most properties of the original one. Therefore, the original system can be studied by simulating just the reduced one and thus make the design work much easier. MOR has become a significant tool in many areas e.g. circuit simulation and feedback

design. Whenever the order of the system is large, it becomes necessary to use MOR to avoid computational complexity of such problems. MOR can be used in control of large-scale dynamical systems, image processing and other engineering fields [1, 2]. Moreover, MOR is a very interesting and meaningful mathematical problem in its own right.

1.1.1 General Idea of the Model-Order Reduction Problem

Consider the linear system described by the following state-space form:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{1.1}$$

where, $x(t) \in \mathbb{R}^n$ is a vector of states, $u(t) \in \mathbb{R}^m$, $y(t) \in \mathbb{R}^p$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{p\times n}$. The order of the original system (1.1) is large and the model order reduction aims to find another system of the form:

$$\begin{aligned} \dot{x}_r(t) &= A_r x_r(t) + B_r u(t) \\ y_o(t) &= C_r x_r(t) \end{aligned} \tag{1.2}$$

where the state variables x_r is of dimension r, $A_r \in \mathbb{R}^{r\times r}$, $B_r \in \mathbb{R}^{r\times m}$, $C_r \in \mathbb{R}^{p\times r}$ and $r \ll n$. The reduced-order system has very close behavior with the large original system. In other words, if there exists an input $u(t)$ to both systems, then the reduced system (1.2) generates an output which is very close to the output

of the original system. Thus once we have the reduced system, we can study the interesting behavior and properties of the original system to a good precision by just studying or simulating the reduced one.

In many engineering situations, people have to deal with nonlinear systems. Suppose there exist a nonlinear system as follows:

$$\begin{aligned} \dot{x} &= f(x) + Bu(t) \\ y &= Cx \end{aligned} \tag{1.3}$$

MOR aims to find a reduced-order nonlinear system:

$$\begin{aligned} \dot{x}_r &= f_r(x_r) + B_r u \\ y_o &= C_r x_r \end{aligned} \tag{1.4}$$

such that its behavior is very close to the behavior of the original one, where $x \in \mathbb{R}^n$ and $x_r \in \mathbb{R}^r$ with $r << n$ and both $f(x)$ and $f_r(x_r)$ are nonlinear (vector) functions.

Nonlinear model order reduction is desired in areas that involve nonlinear systems such as image processing, and simulation of nonlinear circuits. However, there are challenges encountered in nonlinear model order reduction. These challenges include the lack of a guaranteed stability and lack of error bound for the obtained-reduced system [3]. Despite the difficulties that nonlinear model order reduction contains, lots of researches have introduced reduction techniques for nonlinear systems [4, 5, 6, 7, 8, 9]. Virous methods of model reduction for

nonlinear systems have been developed in the last decades. The goal of these techniques is to avoid the complexity of large nonlinear systems. One of these methods is called linearization MOR [10]. The idea of this approach is started by linearizing the large nonlinear system and obtain a linear system. Then, apply the reduction-order procedures to the linearized system. However, it does not give a good approximation [11]. Another method called the quadratic MOR [11]. In this method, the terms of the nonlinear systems are expanded using Taylor expansion. In this method, the nonlinear systems are approximated using the quadratic approximation, this done by deleting the parts of Taylor expansion which are greater than two degree, i.e. this method obtains a quadratic reduced system for the original nonlinear system. Also, it gives better accuracy than the linearization method.Reference [12] proposed the bilinearization moder-reduction technique. In this technique, the nonlinear system needed to be approximated using a bilinear system, and then the procedure of MOR is applied to the bilinear system [13, 14]. The accuracy of this technique is better than the quadratic model-reduction technique [15]. Variational analysis MOR presented in [16]. In this method, the original nonlinear system is transformed into several linear systems using variational equation theory [17], then apply the MOR on each one of these linear systems [18]. Trajectory piecewise linear method is another MOR method of nonlinear systems [19]. In this approach, the original nonlinear system needed to represented by the piecewise-linear systems, then the reduction techniques for linear systems are applied to theses systems [20]. Derivative matching

method is proposed in [21]. The idea of this method is to form an orthogonal projection matrix. The construction of this matrix depending on the derivatives of the state variables of the nonlinear system, and then use this matrix to project the original system. Proper orthogonal decomposition is another method of MOR [22]. This method uses least-squares approximation to approximate the original nonlinear system [23, 24]. Balanced truncation for nonlinear systems is another approach that developed by Scherpen [8, 9]. Recently, the approach of convex optimization has been introduced for MOR for nonlinear systems, and the low order systems are obtained by solving a set of linear matrix inequalities (LMIs) [4, 5, 25]. Nevertheless, reduction of nonlinear systems is still a hot research area and needs to be further investigated. In this study, two new approaches of "MOR" for a class of nonlinear systems have been developed. The main idea of the developed approaches is applying the projection matrix directly to the states of the original nonlinear systems to obtain the low order systems. The first algorithm constructing the projection matrix by linearizing the original nonlinear system using Jacobian method. Then, from the balanced transformation matrix, we can get the projection matrix. The second approach uses the Particle Swarm Optimization "PSO" to construct the projection matrix.

1.2 Literature Review

During the last decades, lots of research works have focused and investigated the concept of MOR due to its importance in many areas such as: control design,

circuit simulation, and image processing [1, 26, 27, 2]. The main idea of MOR concept is to use the reduced order system, that has less dimension, instead of the original system, that usually has a large dimension. However, the behavior of the original system is preserved. One of the most important use of MOR techniques is to develop a controller with a low dimension, instead of developing it with a high dimension. There are several reduction techniques that have been introduced during the last decades such as: balanced truncation technique [28, 29, 30, 31], Moment matching technique [1], projection-based techniques [32, 33, 34, 35, 36, 37, 38, 39], and optimal and convex-optimization techniques [25, 4, 5, 40, 41, 42, 43, 44].

The purpose of the model-reduction techniques is to produce a system with low order. However, the essential characteristics of the original system are preserved. For this purpose, it is necessary to define certain indices that guarantee bounded error approximation. The H_∞ norm of the difference between the output of the original system and the output of the reduced system is one of the most important measures of error approximation [42]. Some model order reduction techniques have shown good performance for a variety of dynamical systems such as Hankel norm approximation [45, 46, 47], and H_2-norm minimization-based techniques [48]. The LMI has been introduced as "MOR" techniques in [43, 44, 49, 50, 51, 52]. For more details on the use of model order reduction approaches using balanced truncation and Kalman's minimal realization techniques, the reader is referred to references [1, 53, 54, 45, 55, 56]. More details on model order truncation techniques are

available in [29]. Optimization techniques have been introduced to reduce the dimension of the systems, such as genetic algorithms "GA", and particle swarm optimization "PSO" [57, 58, 59, 60].

In this literature review, some of model order reduction techniques are presented as follows:

1.2.1 State Truncations

Let's consider that we have the following system:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1.5}$$

In this representation, $x(t) \in \mathbb{R}^n$, where $x(t)$ is the state variable of the linear system. The input of the system is u, where $u(t) \in \mathbb{R}^m$. The output of the system is y, where $y(t) \in \mathbb{R}^p$.

Let's assume that x of the original system (usually with large dimension) composed of x_1 and x_2;

$$x = \begin{bmatrix} x_1 \\ \\ x_2 \end{bmatrix}$$

where $x_1 \in \mathbb{R}^r$, $x_2 \in \mathbb{R}^{n-r}$

The matrices of the original system (1.5) are partitioned as follows:

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, C = \begin{bmatrix} C_1 & C_2 \end{bmatrix}$$

here, the sub-matrix $A_{11} \in \mathbb{R}^{r \times r}$, the sub-matrix $B_1 \in \mathbb{R}^{r \times m}$, the matrix $C1 \in \mathbb{R}^{p \times r}$, and the submatrix $D \in \mathbb{R}^{p \times m}$. Note that D matrix does not affected by this partitioning [1].

1.2.2 Modal Truncation

Consider that we have a stable system (1.5), the transformation matrix of this system can be define as:

$$x = Tx_0 \tag{1.6}$$

here, the matrix T is called the transformation matrix, and this matrix is non-singular, where $T \in \mathbb{R}^{n \times n}$. This procedure is nothing but only to re-write the state variable of the original system in another basis which is different from the original one, so that, the behavior of the original system (1.5) has not affected by this transformation.

By substituting equation (1.6) in equation (1.5), we got the following:

$$\begin{aligned} x_0(t) &= T^{-1}ATx_0(t) + T^{-1}Bu(t) \\ y(t) &= CTx_0(t) + Du(t) \end{aligned} \tag{1.7}$$

In this system, the transformation matrix "T" is used to obtain the similarity transformation of the original matrix A, this transformation can be written as $T^{-1}AT$. This transformation has a main property which is the eigenvalues of the transformation matrix $T^{-1}AT$ remains the same as the eigenvalues of the main matrix A. So, A will lead to a special format by properly choosing the similarity transformation "T".

For the matrix A of the original system, we can re-write the polynomial $p(s) = det(sI - A)$ in different format as the following:

$$\begin{aligned} p(s) &= det(sI - A) \\ &= p_0 + p_1 s + ... + p_n s^n \\ &= (s - \lambda_1)(s - \lambda_2)(s - \lambda_3)...(s - \lambda_n) \end{aligned} \tag{1.8}$$

Here, the natural frequencies λ_i's of the original system for the canonical form modal are assumed to be different from each other. Then, for each λ_i there exists a different eigenvector t_i that has dimension of n such that $At_i = \lambda_i t_i$. These eigenvectors are stored in a single matrix called T, where $T \in \mathbb{R}^{n \times n}$:

The matrix T consist of a set of eigenvectors such as :

$$T = \begin{bmatrix} t_1 & t_2 & t_3 & ... & t_n \end{bmatrix}$$

where t_1 is the eigenvector for the natural frequency λ_1, t_2 is the eigenvector of the natural frequency λ_2, and so on.

After we obtain the transformation matrix T, we apply the similarity transformation $T^{-1}AT$.

Finally, the transformed matrix A will take the diagonal form ; i.e.

$$A_0 = T^{-1}AT = diag(\lambda_1, ..., \lambda_n)$$

The new form of the matrix A_0 is called the Jordan form. However, the modal canonical form is the transformation of the whole system.

Definition 1.1 ***Modal Canonical Form*** *The canonical form for the the linear system* (1.5) *can be written as the following:*

$$\begin{aligned} x_0(t) &= A_0x(t) + B_0u(t) \\ y(t) &= C_0x(t) + D_0u(t) \end{aligned} \tag{1.9}$$

here, $A_0 = T^{-1}AT$, $B_0 = T^{-1}B$, $C_0 = CT$, *and* $D_0 = D$.

Here, we partition the states x_0 into x_r and x_x as the following:

$$x_0 = \begin{bmatrix} x_r \\ \\ x_x \end{bmatrix}$$

where $x_r \in \mathbb{R}^r$, $r << n$, and $x_x \in \mathbb{R}^{n-r}$. The idea of the reduced system is to omit the states that have low frequencies "x_x" and save the states that have high frequencies "x_r", so that the obtained system (with low order) saved the main properties of the original one.

Definition 1.2 ***Modal Truncation****: From the canonical form of the system* (1.9)*, the following system:*

$$\begin{aligned} \dot{x_r}(t) &= A_r x_r(t) + B_r u(t) \\ y_o(t) &= C_r x_r(t) + D_r u(t) \end{aligned} \tag{1.10}$$

is called r-th order modal truncation of the original system [61].

Remark 1.1 *Model truncation technique has several features which distinguish it from other techniques. One of these features is that the eigenvalues based on this method can be easily adapted to a more general time invariant systems, also, this method preserves the stability, where the eigenvalues of the original system do not change. However, there are some disadvantages for this technique, such as a lack of computationally feasible error bound and the approximation accuracy is lower compared with other techniques of model order reduction.*

1.2.3 Balanced Truncation Methods

The second popular method of MOR is called the balanced truncation technique. This technique requires the original system to be in the balanced state space form before truncating the states. This form is an input-output representation of the form in the original system (1.5), where the controllability and the observability gramians are in diagonal form and equal to each other [62]. This method will be presented in this section, but first, we will define the balanced state representation of the system.

1.2.3.1 Balanced State Space Representation

This method starts by transforming the original system to the balanced state space representation, and then eliminate the less-affected states from the balanced representation. The stability of the reduced system is preserved using this technique. Suppose that we have a stable system (1.5), let's define two matrices, P and Q as the following: The controllability gramian "P" can be defined as follows;

$$P = \int_{t0}^{t} \mathrm{e}^{At} B B^T \mathrm{e}^{A^T t} dt \tag{1.11}$$

All eigenvalues of the matrix A are negative and are real numbers. This come from the assumption that the system (1.5) is stable, then the controllability gramian matrix (1.11) is well defined. Here, the matrix P is real, symmetric and has a dimension of $n \times n$.

Also, the observability gramian "Q" can be defined as follows:

$$Q = \int_{t0}^{t} \mathrm{e}^{A^T t} C^T C \mathrm{e}^{At} dt \tag{1.12}$$

The observability gramian matrix (1.12) is well defined. Here the matrix Q is real, symmetric and has a dimension of $n \times n$.

Theorem 1.1 *If there exist a stable linear system* (1.5)*, then there exist a solution*

for the lyapunov equation:

$$AP + PA^T + BB^T = 0 \tag{1.13}$$

where P is unique and positive definite.

Similarly, there exist a solution for the following lyapunov equation:

$$A^TQ + QA + C^TC = 0 \tag{1.14}$$

where Q is unique and positive definite.

Definition 1.3 *The linear system* (1.5) *is called balanced, if the matrices P and Q are diagonal and equal to each other. In other words, the representation of the system* (1.5) *is called balanced if:*

$$P = Q = \Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ \dots & \dots & \ddots & \vdots \\ 0 & 0 & 0 & \sigma_n \end{bmatrix}$$

where P is the controllability gramian, Q is the observability gramian, and σ_i's represent the Hankel singular values. Note that σ_i's are greater than zero and in descending order i.e. $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$ [63].

1.2.3.2 Existence of Balance State Space Representations

Consider that the basis of the state space system (1.5) has been changed, the gramians of this change will be as following:

The changes of the state parameters will be;

$$\begin{aligned} A_0 &= T^{-1}AT \\ B_0 &= T^{-1}B \\ C_0 &= CT \\ D_0 &= D \end{aligned} \tag{1.15}$$

The gramian of the controllability gramian of the transformed system (balanced form) can be written as:

$$P_0 = \int_{t0}^{t} \mathrm{e}^{A_0 t} B_0 {B_0}^T \mathrm{e}^{{A_0}^T t} dt \tag{1.16}$$

Substituting equation (1.15) in equation (1.16) yields;

$$P_0 = \int_{t0}^{t} \mathrm{e}^{(T^{-1}AT)t} (T^{-1}B)(T^{-1}B)^T \mathrm{e}^{(T^{-1}AT)^T t} dt \tag{1.17}$$

From linear algebra, using the following matrix properties;

$e^{(T^{-1}AT)} = T^{-1}e^{A}T$, $(T^{-1}B)^T = B^T(T^{-1})^T$ and $e^{(T^{-1}AT)^T} = T^T e^{A^T}(T^{-1})^T$

We get :

$$P_0 = \int_{t0}^{t} T^{-1} e^{At} T T^{-1} B B^T T^{-T} T^T e^{At} T^{-T} dt \tag{1.18}$$

Finally, this equation will take the following form:

$$P_0 = T^{-1} P (T^{-1})^T \tag{1.19}$$

Using the same procedure for the observability transformed gramian, the observability gramian was defined as ;

$$Q_0 = \int_{t0}^{t} e^{A_0^T t} C_0^T C_0 e^{A_0 t} dt \tag{1.20}$$

Substituting equation (1.15) in equation (4.2) yields;

$$Q_0 = \int_{t0}^{t} e^{(T^{-1}AT)^T t} T^T C^T C T e^{T^{-1}ATt} dt \tag{1.21}$$

Using linear algebra properties for matrices, we got the following :

$$Q_0 = \int_{t0}^{t} T^T e^{A^T} C^T C e^{At} T dt \tag{1.22}$$

Finally, this equation will take the following form:

$$\therefore Q_0 = T^T Q T \tag{1.23}$$

To conclude, the gramians of both the controllability (1.19) and the observability (1.23) depend on the basis of the original system (1.5). However, applying the state space transformation to the original system has no effect to the eigenvalues of PQ, i.e.

$$P_0 Q_0 = T^{-1} P (T^{-1})^T T^T Q = T^{-1} P Q T$$

1.2.3.3 Construction of Transformation Matrix

The following algorithm shows how to construct the transformation matrix "T", which is a non-singular matrix. Moreover, the transformation matrix "T" diagonalize the gramians P and Q.

INPUT: Consider there exist a linear system, the objective is to find another system with low order. However, the obtained system save the main properties of the original system. The state space parameters for the large system are (A,B,C,D).

- Step 1: The procedure started by solving the lyapunov equations and find the gramins P and Q for the original system using the equations (1.13) and (1.14), (this can be done in matlab using the command [lyap]).

- Step 2: Factorize the controllability gramian as $P = R^T R$ (this can be done in matlab using the command [chol]).

- Step 3: Construct the matrix RQR^T and factorize it as : $RQR^T = U^T \Sigma U$, where the matrix U is a unitary matrix, and the other matrix "Σ" is diagonal matrix, where its diagonal elements are σ_i, i.e. $\Sigma = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$, and $(\sigma_i = (\lambda_i(PQ))^{1/2})$.

- Step 4: Finally, we can defined the transformation matrix as follows:

$$T = R^T U \Sigma^{-1/2}$$

OUTPUT: The state parameters of the transformed system will take the following form: $A_0 = T^{-1}AT$, $B_0 = T^{-1}B$, $C_0 = CT$, $D_0 = D$.

From the previous development, it's clear that the balanced form of any linear system can be obtained by changing its basis. However, this system must be controllable and observable [2],[64].

1.2.3.4 Model Reduction using Balanced Truncation Technique

This technique is popular and used to reduce the large linear systems. The idea of this technique is based on the connection between the gramians and the energy. The gramians "P and Q" play the main rule in this technique, if the corresponding states of the linear system needed to be moved, an amount of energy is put into this system. The singular values of "P" correspond to this amount. Moreover, the states of the original system generates an energy. The singular values of the observability gramian "Q" reffering to this energy. In balanced system, the values that give a measurement of the importance of the states is called "the Hankel singular values". These values are defined as the square roots of PQ and they give an indication to the importance of the states of the system. The most affected states in the system have the largest singular values. In model order reduction issues, the system can dispense the states that have a small singular values and the

new system (the reduced one) will be the best possible approximation system to the original system that has a full states [53]. For the reduction by truncation, ones the original system is transformed to the balanced form, the MOR by truncation will be applied to it as showing below:

$A_r = EA_0E^T$, $B_r = EB_0$, $C_r = C_0E^T$, and $D_r = D_0$

where $E = \begin{bmatrix} I_{(r,r)} & 0_{(r,n-r)} \end{bmatrix}$, r is the new order of the system "reduced-system order", and n is the old order of the system "original-system order", which is high and greater than the new order.

1.2.3.5 The Corresponding between Projection and Balanced Truncation

The balanced truncation explained earlier is the most popular technique of MOR. However, this technique is a projection method. In this technique, the gramians play the major role in the projection subspace. The corresponding between the projection method and balanced truncation method can be explained as follows:
INPUT: The large original and stable system (1.5) that needed to be reduced, with (A,B,C,D) as the state space parameters.

- Step 1: Assume there exist a transformation matrix "T", where $T \in \mathbb{R}^{n\times n}$. Then, we construct two matrices depending on T. Let's call these matrices V and U, where $V = T^{-1}E^T$ and $U^T = ET$. Here, $E \in \mathbb{R}^{r\times n}$, and $E = \begin{bmatrix} I_{(r,r)} & 0_{(r,n-r)} \end{bmatrix}$.
- Step 2: Let $x = Vx_r$;

where $x_r \in \mathbb{R}^r$, $V \in \mathbb{R}^{n \times r}$

- Step 3: Put the value of x that we got in step 2 into the original system (1.5)

$$\begin{aligned} \dot{x} &= V\dot{x_r} = Ax + Bu \\ y_o &= CVx_r \end{aligned} \tag{1.24}$$

Then,

$$\begin{aligned} V\dot{x_r} &= AVx_r + Bu \\ y_o &= CVx_r \end{aligned} \tag{1.25}$$

By multiplying the state equation from left side by U^T, we got:

$$U^T V\dot{x}_r = U^T AVx_r + U^T Bu \tag{1.26}$$

where $U^T \in \mathbb{R}^{r \times n}$

- Step 4: By multiplying equation (1.26) from left side by $(U^T V)^{-1}$, we got the following:

$$\begin{aligned} \dot{x}_r &= (U^T V)^{-1} U^T AVx_r + (U^T V)^{-1} U^T Bu \\ y_o &= CVx_r \end{aligned} \tag{1.27}$$

OUTPUT: The output of this development is the reduced system with order r:

$$\begin{aligned} \dot{x}_r &= A_r x_r + B_r u \\ y_o &= C_r x_r + D_r u \end{aligned} \tag{1.28}$$

where: $A_r = (U^T V)^{-1} U^T A V$, $B_r = (U^T V)^{-1} U^T B$, $C_r = CV$, and $D_r = D$.

From the previous algorithm, we conclude that the eigenvalues of the matrices"P and Q" are invariant under the state space transformation.

Let $\lambda_1, \cdots, \lambda_n$ are the eigenvalues of "PQ", and the square root of these eigenvalues denoted by σ_i, then:

$\sigma_i = \sqrt{(}\lambda_i) = \lambda_i^{1/2}(PQ)$.

where all λ_i's are real and positive for $i = 1, ..., n$.

Remark 1.2 *Balanced truncation technique is an important projection method, it is idea depends on the choice of the projection subspace based on the gramians P and Q. In this technique, the original system is converted to the balanced system via the state-space transformation "T",i.e $A_0, B_0, C_0, D_0 = (T^{-1}AT, T^{-1}B, CT, D)$. The transformation matrix has some advantages e.g. $P = TP_0T^T$, and $Q = T^{-T}Q_0T^{-1}$. Here, the states are arranged in descending order according to how controllable and observable the states are, and then the discardable states are truncated. The new system, obtained by using the balanced truncation technique, is characterized by a number of properties such as stability preservation and the difference between its output and the output of the complete (high order) system for the same input stay within the acceptable error and it can be calculated using the following equation [65]:*

$$||y - y_o||_\infty = 2 \sum_{i=r+1}^{n} \sigma_i$$

1.2.4 The Schur Method for Model Reduction

In balanced truncation method, there are some difficulties that can face when we build the transformation matrices (e.g. when these transforming matrices are ill-conditioning). Safonov and Chiang [66] had developed another method that can overcome these difficulties by suggested that, "if the original system transformed to an alternative system through the orthogonal matrices, then the reduced system can be truncated from the new form".

The idea of Schur method is just to replace the T and T^{-1} by the orthogonal matrices. These matrices are perfectly-conditioned and have the same properties of the transformation matrix "T" and its inverse "T^{-1}". The Schur method constructs the orthogonal matrices using the Real-Schur Form (RSF) of the gramians P and Q.

Schur algorithm for MOR can be expressed as follows [67]:

INPUT: This algorithm is applied to the system in equation (1.5) that has the state space parameters (A,B,C,D)

- Step 1: Solve the Lyapunov equations (1.13) and (1.14), and obtain the gramian matrices P and Q.

- Step 2: Find the eigenvalues of the gramians PQ, and determine the largest values of them. Determination of the number of the largest eigenvalues helping to determine the order of the reduced system "r". Then, find the orthonormal bases V_p and V_q, where V_p, $V_q \in \mathbb{R}^{n\times r}$, this can be done using

means of ordered Schur factorizations.

- Step 3: Find the singular value decomposition (SVD) of vectors that we found in step 2, $V_q^T V_p$, i.e. $U_q \Sigma U_p^T = SVD(V_q^T V_p)$

- Step 4: Compute the transforming matrices :

  $S_p = V_p U_p \Sigma^{-1/2}$, $S_q = V_q U_q \Sigma^{-1/2}$

OUTPUT: $A_r = S_q^T A S_p$, $B_r = S_q^T B$, $C_r = C S_p$, and $D_r = D$

In this method, the main properties of the original system is preserved. However it does not give the balance realization, these properties showing in the following theorem [68]:

Theorem 1.2 *Assume that we used the Schur method to reduce a high order system, then for the reduced system there exist a transfer function and it is denoted by G_r , where $G_r(s) = C_r(sI - A_r)^{-1}B_r$. It is noted that, G_r is equal to the transfer function of the reduced system that is obtained using balanced-truncation method. Moreover, the gramians for the new system using this algorithm can be written as:*

$P_r = S_q^T P S_q$, $Q_r = S_p^T Q S_P$

where P_r is controllability gramian of the reduced-system and Q_r is the observability gramian of the reduced system.

Remark 1.3 *Schur MOR was developed to overcome the computational difficulties that may confront us when we construct the transformation matrix due to possibility of ill-conditioning. It is noted that, the transfer function obtained using*

this approach is the same as the transfer function obtained through the balanced truncation method. However, the Schur method has a different computational problem[68].

1.2.5 Hankel Norm Model Reductions

This technique distinguishes from the others by its optimal approximation model. It is closely related to the balanced truncation technique that we discussed above. However, for the balanced truncation we do not known whether the truncated system (with order r) is the optimal approximation or not. In this method, the truncated system will be the optimal approximation [69].

The idea of this technique is derived from the arguments of the energy transmission. Assume there exists a linear system "G" (with high order n) that maps the input (u) to the output (y) as $y = Gu$. Then, for the reduced system we select G_r that maps u to the new output y_o, where G_r has order of r ($r << n$). The error, which is defined as the difference between reduced-system output "y_o" and the output of the complete (high order) system "y" for the same inputs, can be calculated as:

$$e = \sup_{u \in \mathcal{L}_2[0,\infty)} \frac{\int_0^\infty (y(t) - y_o(t))^T (y(t) - y_o(t))}{\int_0^\infty u(t)^T u(t)}$$

This technique aims to obtain a transfer function G_r for the reduced system with smaller degree than the transfer function of original system G, such that the difference between their outputs "e" is minimized. This method is presented in details in this section.

1.2.5.1 Hankel Operator

For the linear system, the Hankel operator is defined as the prediction operator that used to " map the past input to the future output" [1], with the assumption that the future input is zero.

Assume there exist a linear system with state space representation as in equation (1.5). If the input to this system is $u(t)$, where $u \in (-\infty, 0]$, then the output of this system is determined by the following equation:

$$y(t) = \int_{-\infty}^{0} Ce^{A(t-\tau)}Bu(\tau)d\tau, t > 0$$

If we put $v(t) = u(-t)$, then the output "y" will be $y(t) = (\Gamma_G v)(t)$, where $t > 0$. Here Γ_G is the Hankel operator and it defined as:

$$(\Gamma_G v)(t) = \int_{-\infty}^{0} Ce^{A(t+\tau)}Bv(\tau)d\tau$$

1.2.5.2 Hankel Norm

It can be defined as the induced norm $\mathcal{L}_2[0, \infty)$ of the Hankel operator of the system, we can write it as $||G||_H = ||\Gamma_G||$

From the definition above, we can write the Hankel norm as follows:

$$||G||_H^2 = \sup_{u \in \mathcal{L}_2[0,\infty)} \frac{\int_0^\infty y(t)^T y(t)\ dt}{\int_0^\infty u(t)^T u(t)\ dt}$$

or it can be written as the following:

$$\int_0^\infty y(t)^T y(t)\ dt = ||G||_H^2 \int_0^\infty u(t)^T u(t)\ dt$$

It's clear that, the output of the linear system is nothing but the squared Hankel norm of that system times its input. This method aims to obtain a less-order system with a transfer function G_r of degree r, such that $||G - G_r||$ is minimized and $r << n$ [65].

1.2.5.3 Hankel singular values and the Hankel norm

The maximum singular value of a linear system is the Hankel norm of that system. However, these singular values are nothing but the squared-eigenvalues of PQ and these values are in decrescent order, i.e

$$\sigma_1 \geq \sigma_2 \ldots \geq \sigma_n \geq 0$$

Theorem 1.3 *If we have a linear and stable system, then:*

- *Denoting to the Hankel operator by H, the singular values of H is nothing but the squared eigenvalues of the gramians "PQ".*
- *The square root of the largest eigenvalue of PQ is called the Hankel norm of that system.*

1.2.5.4 Computing the Optimal Hankel Norm Approximation

In balance truncation section, we explained how to obtain the balanced representation of the original system in order to truncate it and obtain the reduced-system with order r ($r << n$). However, this technique did not obtain the optimal approximation. In other words, the reduced system obtained using the balance truncation has not this criterion in case that the original system with order n has a higher order than the reduced system of order r. However, the Hankel norm reduction technique involve a criterion like this.

Assume that we have a transfer function of a stable system G. This approach aims to find a reduced-order system with transfer function G_r, such that $||G - G_r||_H$ is minimized. Here, $||G - G_r||_H$ is the error of the system [45, 70]. The algorithm below shows that how to find the reduced system state parameters (A_r, B_r, C_r, and D_r) using the Hankel model reduction:
INPUT A large-order system G which is stable, controllable, and observable with states (A, B, C, and D).

- Step 1: This algorithm started by computing the singular values "SVD" for the large-order system and put them in a way such that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$.

- Step 2: Use the transformation matrix to transform the system to the bal-

anced form:

$$P = Q = \begin{bmatrix} g_1 & 0 \\ 0 & g_2 \end{bmatrix}$$

where g_1 are the singular values of the system ordered as $diag(\sigma_1, \ldots, \sigma_r, \sigma_{r+k}, \ldots \sigma_n)$ and $g_2 = \sigma_{r+1} I_k$

- Step 3: According to the partitioning of the gramians, the original-system states partitioned into the following:

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, B = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}, C = \begin{bmatrix} c_1 & c_2 \end{bmatrix}$$

where $a_{11} \in \mathbb{R}^{(n_{min}-k)\times(n_{min}-k)}$, $b_1 \in \mathbb{R}^{(n_{min}-k)\times m}$, $c_1 \in \mathbb{R}^{P\times(n_{min}-k)}$

define $\Gamma = \Sigma_1^2 - \sigma_{r+1}^2 I$

- Step 4: Determine a matrix U such that this matrix should be a unitary matrix, it satisfies that $B_2 + C_2^T U = 0$.

- Step 5: The state space dimension of the system $\hat{G}$ with order $\hat{n}$, where $\hat{n} = n - r$, can be defined as: $\hat{A} = \Gamma^{-1}(\sigma_{r+1}^2 a_{11}^T + g_1 a_{11} g_1 - \sigma_{r+1} c_1^T U b_1^T)$

  $\hat{B} = \Gamma^{-1}(g_1 b_1 + \sigma_{r+1} c_1^T U)$

  $\hat{C} = c_1 g_1 + \sigma_{r+1} U b_1^T$

  $\hat{D} = D - \sigma_{r+1} U$

- Step 6 Finally, the subsystem $\hat{\Sigma}$ is determined by selecting its state space as showing below:

$$\hat{A} = \begin{bmatrix} \hat{A}_{-ve} & 0 \\ 0 & \hat{A}_{+ve} \end{bmatrix}, \hat{B} = \begin{bmatrix} \hat{B}_{-ve} \\ \hat{B}_{+ve} \end{bmatrix}, \hat{C} = \begin{bmatrix} \hat{C}_{-ve} & \hat{C}_{+ve} \end{bmatrix}$$

where $\hat{A}_{-ve}$ is stable with dimension of $\leq r$, while $\hat{A}_{+ve}$ is not stable.

OUTPUT: set

$A_r = \hat{A}_{-ve}$

$B_r = \hat{B}_{-ve}$

$C_r = \hat{C}_{-ve}$

$D_r = \hat{D}_{-ve}$

The reduced system that defined as

$\frac{d\zeta}{dt} = A_r\zeta(t) + B_r u(t)$

$y(t) = C_r\zeta(t) + D_r u(t)$

represents the reduced system that is obtained using Hankel norm approximation, and the error of this model reduction technique defined as $||G - G_r||_H = \sigma_{r+1}$.

1.2.6 H_2-Model Reduction

The H_2 model reduction problem is defined as: finding a reduced-stable system G_r with order of r (where $r \ll n$) such that the H_2-norm of the error $E(s)$ is small, where $E(s) = ||G(s) - G_r(s)||$, $G(s) = C(sI - A)^{-1}B$, and $G_r(s) = C_r(sI - A_r)^{-1}B_r$,

Let $E(s)$ be the error with the realization triple (A_e, B_e, C_e).

The H_2-norm is defined as the trace of the matrix.

$$S = ||E(s)||_{H_2}^2 = \frac{1}{2\pi}\int_{-\infty}^{+\infty} E(jw)^T E(jw)dw$$

write the above equation in state space realization form [71]:

$$S = \int_0^{+\infty} trace((C_e e^{A_e t}B_e)^T(C_e e^{A_e t}B_e))dt = \int_0^{+\infty} trace((C_e e^{A_e t}B_e)(C_e e^{A_e t}B_e)^T)dt$$

Take:

$$S = \int_0^{+\infty} trace((C_e e^{A_e t}B_e)^T(C_e e^{A_e t}B_e))dt$$

Then,

$$S = trace(B_e^T(\int_0^{+\infty} e^{A_e^T t}C^T C e^{A_e t}dt)B_e)$$

$$\therefore S = trace(B_e^T Q_e B_e)$$

where Q_e is the observability gramian and $Q_e = \int_0^{+\infty} e^{A_e^T t}C_e^T C_e e^{A_e t}$

If we take,

$$S = \int_0^{+\infty} trace((C_e e^{A_e t}B_e)(C_e e^{A_e t}B_e)^T)dt$$

Then,

$$S = trace(C_e(\int_0^{+\infty} e^{A_e t}B_e B_e^T e^{A_e^T t}dt)C_e^T)$$

$$\therefore S = trace(C_e P_e C_e^T)$$

where P_e is the controllability gramian and $P_e = \int_0^{+\infty} e^{A_e t} B_e B_e^T e^{A_e^T t} dt$

where, $A_e = \begin{bmatrix} A & 0 \\ 0 & A_r \end{bmatrix}$,$B_e = \begin{bmatrix} B \\ B_r \end{bmatrix}$,$C_e = \begin{bmatrix} C & -C_r \end{bmatrix}$,

$$P_e = \begin{bmatrix} P & X \\ X^T & P_r \end{bmatrix}, Q_e = \begin{bmatrix} Q & Y \\ Y^T & Q_r \end{bmatrix}.$$

The lyapunov equations will be in the form:

$$\begin{bmatrix} A^T & 0 \\ 0 & A_r^T \end{bmatrix} \times \begin{bmatrix} Q & Y \\ Y^T & Q_r \end{bmatrix} + \begin{bmatrix} Q & Y \\ Y^T & Q_r \end{bmatrix} \times \begin{bmatrix} A & 0 \\ 0 & A_r \end{bmatrix} + \begin{bmatrix} C^T \\ -C_r^T \end{bmatrix} \times \begin{bmatrix} C & -C_r \end{bmatrix} = 0$$

and

$$\begin{bmatrix} A & 0 \\ 0 & A_r \end{bmatrix} \times \begin{bmatrix} P & X \\ X^T & P_r \end{bmatrix} + \begin{bmatrix} P & X \\ X^T & P_r \end{bmatrix} \times \begin{bmatrix} A^T & 0 \\ 0 & A_r^T \end{bmatrix} + \begin{bmatrix} B \\ B_r \end{bmatrix} \times \begin{bmatrix} B^T & B_r^T \end{bmatrix} = 0$$

To minimize the H_2-norm,S, we must minimize the error function, i.e. minimize S:

$$S = trace(\begin{bmatrix} B^T & B_r^T \end{bmatrix} \begin{bmatrix} Q & Y \\ Y^T & Q_r \end{bmatrix} \begin{bmatrix} B \\ B_r \end{bmatrix})$$

In the above expression, (Q, Y, and Q_r) are depending on A, Ar, C,and C_r. B and Q are constant. Or equivalent: $S = trace(\begin{bmatrix} C & -C_r \end{bmatrix} \begin{bmatrix} P & X \\ X^T & P_r \end{bmatrix} \begin{bmatrix} C^T \\ -C_r^T \end{bmatrix})$

here, (P, X,and P_r) depend on A, A_r, B, and B_r. C and P are constant. There

are many researches on H_2 to find the low-order system with a transfer function G_r such that the difference between its output and the output of the complete (high order) system is minimized. One of these methods used LMIs [25, 51, 65], also genetic algorithms "GA" and particle swarm optimization"PSO" used to find the reduced system [59, 60]

1.2.7 H_∞-Model Reduction

The H_∞-model reduction aims to find the reduced-order system of order r ($r \ll n$) such that $||G(s) - Gr(s)||_\infty$ is small ,where $G(s) = C(sI - A)^{-1}B$, and $G_r(s) = C_r(sI - A_r)^{-1}B_r$. This technique has received many considerable attention. In [42], H_∞-model reduction problem is converted to Hankel norm model reduction problem. The problem of H_∞ was solved using LMIs approach [65, 43], also this problem was solved using genetic algorithms and particle swarm optimization [60].

1.2.8 Numerical Example:

In this section, a stable linear system is used as a numerical example. This example has been used in many researches [72]. The order of this example is 4, and it reduced to the second order. Here, the step signal is used as an input to

both systems. The states of the original system as follows:

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 & 0 & -150 \\ 1 & 0 & 0 & -145 \\ 0 & 1 & 0 & -113 \\ 0 & 0 & 1 & -19 \end{bmatrix} x + \begin{bmatrix} 4 \\ 1 \\ 0 \\ 0 \end{bmatrix} u(t) \tag{1.29}$$

$$y(t) = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} x(t)$$

This system is reduced to the 2^{nd} order due to the good separation between the second Hankel singular value and the third one. Four of the previous approaches used to reduce the original system (1.29) as follows:

1.2.8.1 Modal Truncation:

The first algorithm that we tested is the modal truncation technique. The following system represent the 2^{nd} order system obtained using the modal order reduction method:

$$\dot{x}_r(t) = \begin{bmatrix} -1 & -5.337 \\ -1.11 * 10^{-16} & -3 \end{bmatrix} x_r + \begin{bmatrix} -4.642 \\ 1.047 \end{bmatrix} u(t)$$

$$y_o(t) = \begin{bmatrix} -0.005603 & -0.01915 \end{bmatrix} x_r(t)$$

Figure (1.1) compares the response of the system (1.29) that has the fourth order to the reduced one with the second order. Here, the reduced system obtained by using the modal truncation method. In this example, the H_∞ norm of error is

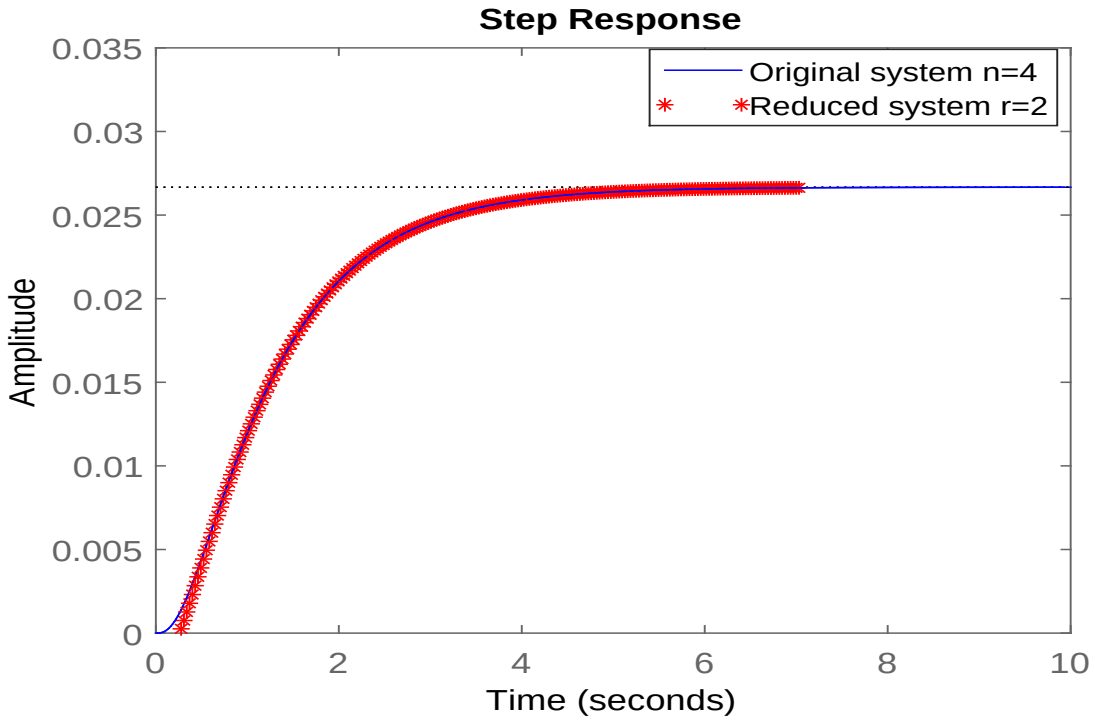

Figure 1.1: System outputs y and y_o using modal truncation

0.0032.

1.2.8.2 Balanced Truncation Method

The second algorithm that we have discussed before is the balanced truncation method, it used to reduce the system (1.29) to the second order:

The following system represent the second order result using the balanced trun-

cation method

$$\dot{x}_r(t) = \begin{bmatrix} -0.04378 & -1.1685 \\ -1.1685 & -3.1353 \end{bmatrix} x_r + \begin{bmatrix} -0.1181 \\ 0.1307 \end{bmatrix} u(t)$$

$$y_o(t) = \begin{bmatrix} -0.1181 & -0.1307 \end{bmatrix} x_r(t)$$

Figure (1.2) compares the response of the system (1.29) that has the fourth order to the reduced one with the second order. Here, the reduced system obtained by using the balanced truncation method. In this example, the H_∞ norm of error is

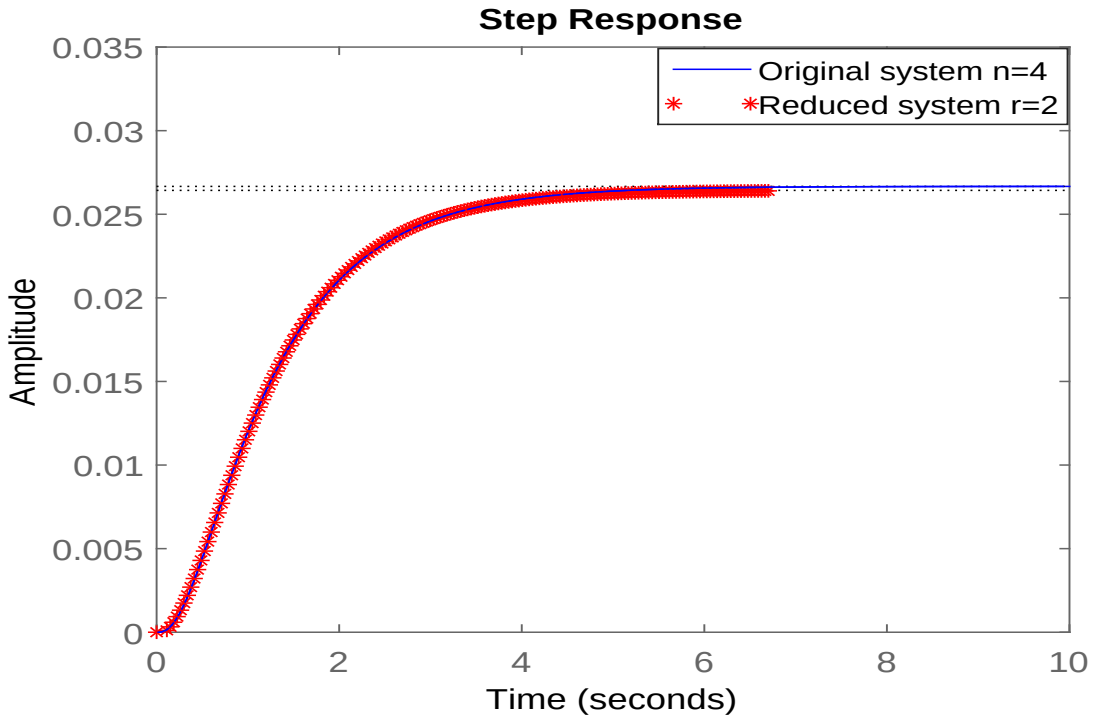

Figure 1.2: System outputs y and y_o using balanced truncation

$2.4662 * 10^{-4}$.

1.2.8.3 Schur Method

The Schur method is used to reduce the system (1.29) to the second order:

The following system represents the reduced system that obtained using the Schur method

$$\dot{x}_r(t) = \begin{bmatrix} -2.554 & 13.37 \\ -0.01013 & -1.019 \end{bmatrix} x_r + \begin{bmatrix} -3.245 \\ -0.9127 \end{bmatrix} u(t)$$

$$y_o(t) = \begin{bmatrix} -0.01094 & 0.04232 \end{bmatrix} x_r(t)$$

Figure (1.3) compares the response of the system (1.29) with order four to the reduced one with the second order. Here, the reduced system obtained using the Schur method. In this example, the H_∞ norm of error is $6.4629 * 10^{-4}$.

1.2.8.4 Hankel Norm Reductions

Finally, Hankel norm reduction is used to reduce the system (1.29) to the second order:

The following system represent the reduced system that obtained using the Hankel norm reductions

$$\dot{x}_r(t) = \begin{bmatrix} -2.801 & 2.428 \\ 0 & -1.052 \end{bmatrix} x_r + \begin{bmatrix} -0.06776 \\ 0.1665 \end{bmatrix} u(t)$$

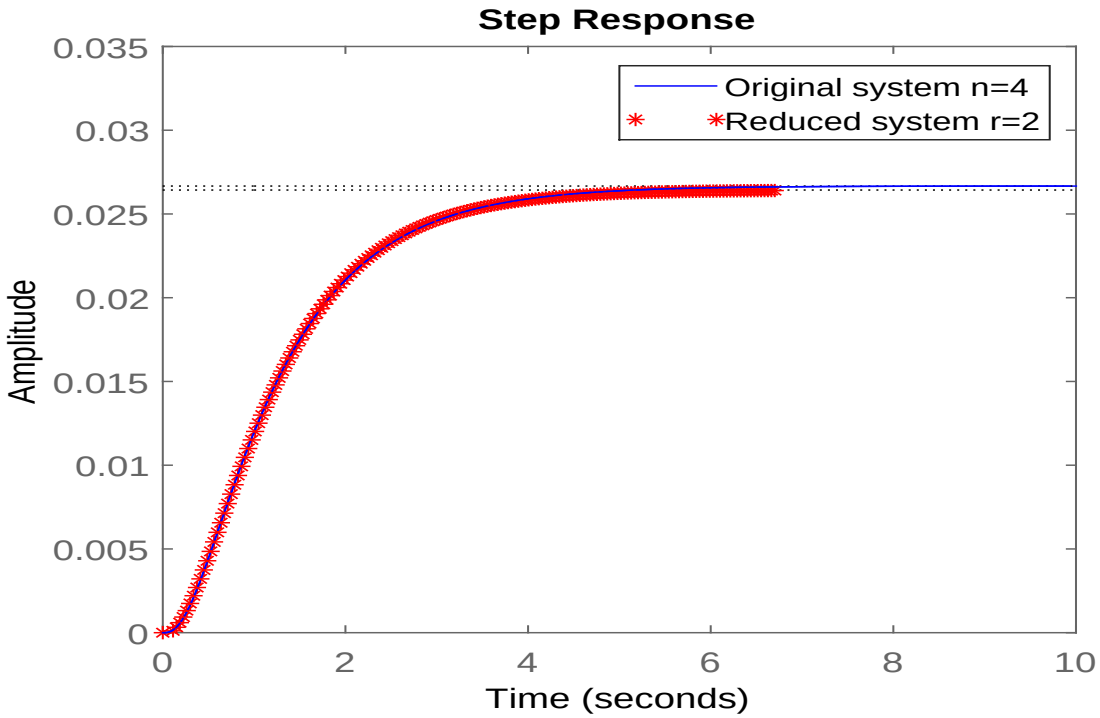

Figure 1.3: System outputs y and y_o using Schur method

$$y_o(t) = \begin{bmatrix} 0.1755 & 0.041597 \end{bmatrix} x_r(t)$$

Figure (1.4) compares the response of the system (1.29) with order four to the reduced one with the second order. Here, the reduced system obtained using the Hankel norm reductions. In this example, the H_∞ norm of error is $2.5438 * 10^{-4}$. The Hankel norm reduction gives the smallest H_∞ norm of error, while the modal truncation gives the largest H_∞ norm of error. The balanced truncation and the Schur MOR have almost the same H_∞ norm of error.

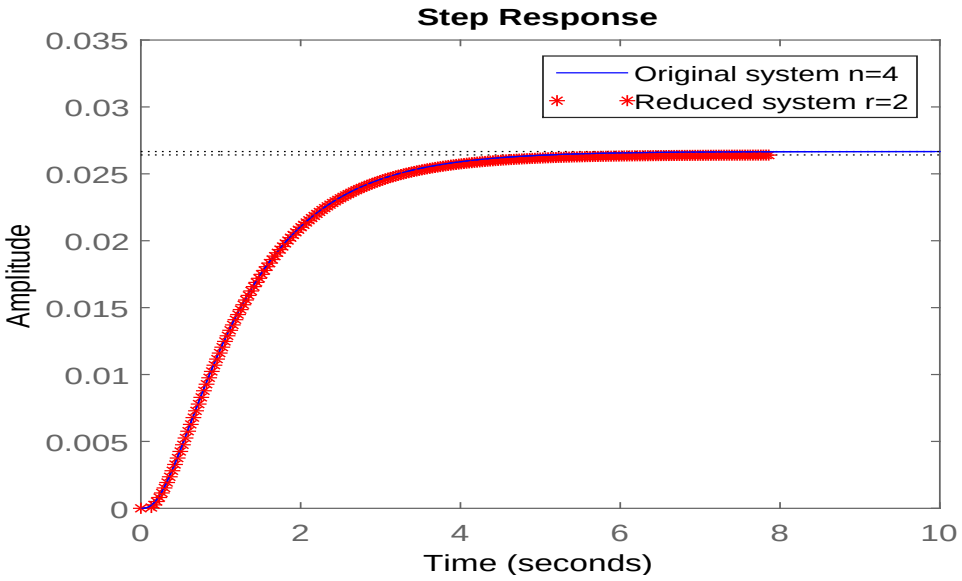

Figure 1.4: System outputs y and y_o using Hankel norm reductions

1.3 Thesis Objectives

This study aims to develop a new algorithms for MOR for nonlinear systems. In addition, the stability analysis of the proposed approaches is studied. Also, two nonlinear electrical circuits are used to illustrate the developed techniques. According to the above objectives, the scope of this thesis could be listed as the follows:

- Study some of the popular techniques of MOR for linear and nonlinear systems.
- Develop an algorithm of MOR for a class of nonlinear systems. In this approach, the linearization method is applied to the large nonlinear system to obtain the projection matrix from the linearized system. This matrix is

used to project the states of the large nonlinear system to obtain a reduced-order system.

- Solve the problem of MOR for the developed approach using particle swarm optimization to construct the projection matrix, no need to linearize the original nonlinear system.
- Stability analysis of the developed algorithms.
- Study the effectiveness of the developed algorithms through simulation, by applying these algorithms to two nonlinear electrical circuits.

1.4 Thesis Organization

The documentation for this research is broken down into four chapters; Chapter one provides the reader with an introduction and literature review. It also gives the reader a summary of how the research was carried out. In Chapter two, an algorithm of MOR for a class of nonlinear systems is developed. This algorithm is designed to project the original states of the nonlinear system so as to obtain the reduced system with less order using a projection matrix. This matrix obtained by linearized the original nonlinear system. In Chapter three, the projection matrix is constructed using the particle swarm optimization which in turn projects the states of the large nonlinear system. Furthermore, these chapters, Chapter two and Chapter three, discussed the stability preservation of the reduced systems. Finally, in Chapter Four, the performance of the developed MOR approaches is

analyzed. Moreover, the efficiency of the developed algorithms is compared with three of existing studies using results of two of nonlinear electrical circuits.

CHAPTER 2

PROJECTION BASED METHOD FOR NONLINEAR SYSTEMS

2.1 Introduction

In this chapter, a new method of MOR for a class of nonlinear systems is developed. This technique is mainly based on the projection technique, where the projection matrix is obtained from the transformation matrix of the linearized system. Then, to obtain the reduced-order system, the states of the original nonlinear system have to be projected through the projection matrix. Firstly, the projection-based procedure for linear systems is summarized. Then, the developed approach of MOR for nonlinear systems is presented. Finally, the efficiency of the developed approach is tested and investigated using nonlinear electrical

circuit.

2.2 Projection Method for Linear Systems

To understand the meaning of the MOR using the projection technique, this technique is used to reduce the order of a stable linear system as follows:

Consider the linear system described by the following form

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{2.1}$$

where, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $C \in \mathbb{R}^{p\times n}$. The symbols m and p are denoted to the number of the inputs and outputs of the linear system, respectively. The MOR that based on projection method is aiming to approximate $x(t)$ by its projection $x_r(t)$. It can achieve that by selecting a matrix V, such that $x = Vx_r$, where $V \in \mathbb{R}^{n\times r}$. Here, the symbols n and r are denoted to the order of the original system and the reduced system, respectively.

Let

$$x = Vx_r \tag{2.2}$$

where $x_r \in \mathbb{R}^r$ and $r \ll n$.

Substituting equation (2.2) into equation (2.1) yields:

$$\begin{aligned} V\dot{x_r} &= AVx_r + Bu \\ y_o &= CVx_r \end{aligned} \tag{2.3}$$

By choosing a matrix $U \in \mathbb{R}^{r\times n}$, such that $UV = I_r$ and multiply equation (2.3) from the left side by U, we can write:

$$\begin{aligned} \dot{x_r} &= UAVx_r + UBu \\ y_o &= CVx_r \end{aligned} \tag{2.4}$$

Then, the reduced system of order r will be written in the following form:

$$\begin{aligned} \dot{x_r} &= A_r x_r + B_r u \\ y_o &= C_r x_r \end{aligned} \tag{2.5}$$

where $A_r \in \mathbb{R}^{r\times r}$, $B_r \in \mathbb{R}^{r\times m}$, $C_r \in \mathbb{R}^{P\times r}$, $A_r = UAV, B_r = UB,$ and $C_r = CV$. In order to use the reduced system, with order r, instead of using the original system, with order n, the output of the obtained system "y_o" have to be closed to the output of the large system "y". In this technique, the projection matrix "V" is obtained from the corresponding between the projection and balanced truncation, where $V = T^{-1}E^T$, T is the state transformation matrix, and $E = \begin{bmatrix} I_{r,r} & 0_{r,n-r} \end{bmatrix}$. Here, we gave a brief summery of how to obtain the transformation matrix "T", and the projection matrices "V". Assume that we have a linear system (2.1),

then the first step is to find the transformation matrix "T". From this matrix we can obtain the projection matrix "V". This can be explained in the following algorithm:

- Step 1: The procedures of this algorithm started by solving the lyapunov equations and find the gramins P and Q for the original system using the equations (1.13) and (1.14), it can be solved easily in Matlab using the command [lyap].

- Step 2: Factorize the controllability gramian as $P = R^T R$ (this can be done in Matlab using the command [chol]).

- Step 3: Construct the matrix RQR^T and factorize it as : $RQR^T = U^T \Sigma U$, where U is a unitary matrix, and Σ is a diagonal matrix, where its diagonal elements are σ_i, i.e. $\Sigma = \text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_n)$, and $(\sigma_i = (\lambda_i(PQ))^{1/2})$.

- Step 4: Defined the transformation matrix "T" as follows:

$$T = R^T U \Sigma^{-1/2}$$

- Step 5: Define the projection matrices V and U as follows:

  $V = T^{-1}E^T$ and $U = (V^T P V)^{-1} V^T P$, such that $UV = I_r$ and $E = \begin{bmatrix} I_{r,r} & 0_{r,n-r} \end{bmatrix}$

2.2.1 Numerical Example

In this subsection, a fourth-order stable linear system [72] has been used as an example for the model-order reduction using the projection method. This system reduced to the second order. To make the comparison easy, the step signal is used as an input to both systems, the complete and the reduced systems. The original system representation is as the following:

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 & 0 & -150 \\ 1 & 0 & 0 & -145 \\ 0 & 1 & 0 & -113 \\ 0 & 0 & 1 & -19 \end{bmatrix} x + \begin{bmatrix} 4 \\ 1 \\ 0 \\ 0 \end{bmatrix} u(t) \tag{2.6}$$

$$y(t) = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} x(t)$$

Due to the gap between the second and the third Hankel singular values of the original system, the proposed method reduced the order of the original system to the second order. The reduced-order system can be represented as follows:

$$\dot{x}_r(t) = \begin{bmatrix} -0.4378 & -1.1685 \\ 1.1685 & -3.1353 \end{bmatrix} x_r + \begin{bmatrix} -0.1181 \\ 0.1307 \end{bmatrix} u(t)$$

$$y_o(t) = \begin{bmatrix} -0.1181 & -0.1307 \end{bmatrix} x_r(t)$$

Figure (2.1) compares between the response of the system (2.6) with order four to the reduced one with the second order, the reduced system obtained by using the projection method for linear systems.

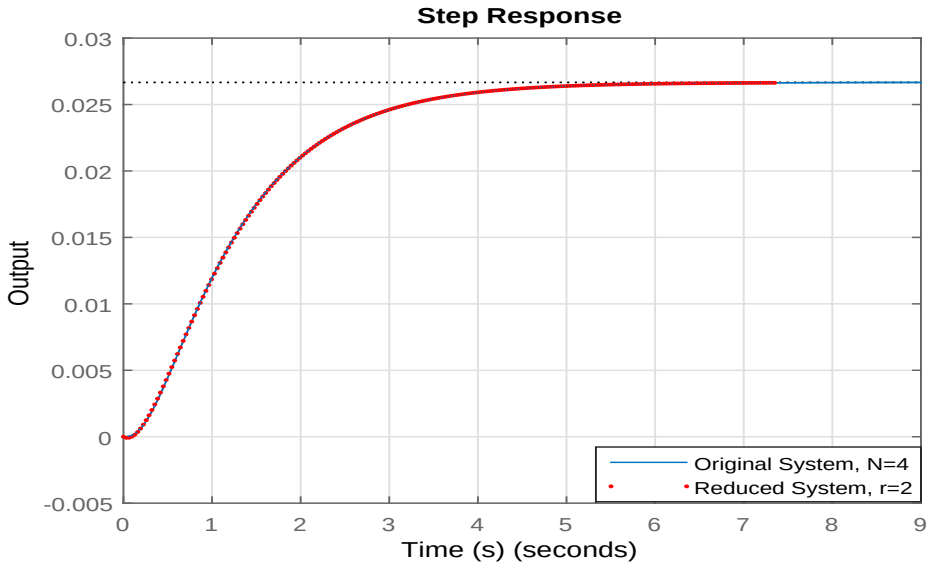

Figure 2.1: System outputs y and y_o using projection based method

It is clear from figure (2.1) that the output of the reduced system of order 2 is closed to the output of the original system of order 4. In this example, the step signal used as an input to the both systems.

2.3 Projection-Based Method For Nonlinear Systems

The developed algorithm of MOR for a class of nonlinear systems is presented in this section. The main idea of this approach is projecting the original states of the nonlinear system to obtain the reduced system with less order using a projection matrix. This matrix obtained by linearizing the original nonlinear system, the projection reduction for linear system has been explained in Section 2.2. The states of the original system is reduced using the projection technique via the projection matrix "V". The reduction procedures of this technique are applied directly to the original nonlinear system (not to the approximated one), and this advantage reduced the error that is introduced to the system during the approximation.

2.3.1 Algorithm Procedures

Consider the nonlinear system described by the following form

$$\begin{aligned} \dot{x} &= f(x) + Bu \\ y &= Cx \end{aligned} \tag{2.7}$$

In this system, x represents the states of the system and $x \in \mathbb{R}^n$, while y represents the output of the system. In equation (2.7), $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{P\times n}$, and $f(x)$ is a nonlinear function.

Projection-based algorithm that we developed to reduce the order of a class of nonlinear systems can be explained as follows:

Step 1: This algorithm started by approximating the states of the original system, $x(t)$, using the projection matrix V, that is obtained from the linearized system, where $x = Vx_r$, here $x_r \in \mathbb{R}^r$ and $V \in \mathbb{R}^{n\times r}$.

Step 2: Re-define the states of the nonlinear system such that the first one equal to the first row of the obtained matrix in step one, i.e. $x_1 = x(1,:)$, and do the same for the second state, i.e. $x_2 = x(2,:)$, up to the last state, i.e. $x_n = x(n,:)$.

Step 3: Replace x_1, x_2, ... x_n in the original nonlinear system (2.7), by the new values that was obtained from step 2. The new form of the system will be:

$$\begin{aligned} V\dot{x}_r &= f_r(Vx_r) + Bu \\ y_o &= CVx_r \end{aligned} \tag{2.8}$$

Step 4: Define a new matrix U, where $U \in \mathbb{R}^{r\times n}$ such that $UV = I_r$.

Step 5: Finally, multiply both sides of equation (2.8) from the left side by U to obtain the following:

$$\begin{aligned} UV\dot{x}_r &= Uf_r(Vx_r) + UBu \\ y_o &= CVx_r \end{aligned} \tag{2.9}$$

$\therefore$ The reduced nonlinear system will be written as the following:

$$\begin{aligned} \dot{x}_r &= U f_r(V x_r) + B_r u \\ y_o &= C_r x_r \end{aligned} \tag{2.10}$$

where $B_r = UB$, $C_r = CV$, and f_r is the reduced nonlinear function.

The main issue in this algorithm is how to select the element of the projection matrix "V" such that the error is small, i.e. the behavior of both systems, original and reduced, is closed to each other.

In the projection-based algorithm, the projection matrix "V" is constructed from the corresponding between the projection and balanced truncation, where $V = T^{-1}E^T$. Here, T is the state transformation matrix, and $E = \begin{bmatrix} I_{r,r} & 0_{r,n-r} \end{bmatrix}$.

In this technique, the construction of the projection matrix "V" is depending on the linearization of the original system and the procedures can be written as follows:

step 1: Linearize the original nonlinear system (2.7) using Jacobian linearization method and from this step we obtain the linear system of the following form:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{2.11}$$

here, the vector x has n-dimension, A is a constant matrix with dimension of $n \times n$. B and C are n-dimensional vectors where $B \in \mathbb{R}^{n \times m}$ and $C \in \mathbb{R}^{p \times n}$.

step 2: The second step is to construct the transformation matrix "T", and from

this matrix we can get the projection matrices V and U, these matrices will be used to project the states of the original nonlinear system. Construction these matrices, T, V, and U, depends on the linearized system (2.11). We have explained how to get these matrices for linear systems in Section 2.2

2.4 Stability Analysis

The stability analysis is one of the most important topics in control, through which we can come up with a summary of the behavior of the system without the need to compute its solution trajectories. The first person who studied the stability was Lagrange in 1788, but this study was restricted to conservative mechanical systems that can be described by Lagrangian equation of motion. Then came the Russian mathematician A. M. Lyapunov in 1892 [73]. He provided the basic definition of stability that still used up to the present day.

Theorem 2.1 *Assume that $x = 0$ is an equilibrium point for the following system:*

$$\dot{x} = f(x) \tag{2.12}$$

and the domain $D \subset \mathbb{R}^n$ is containing $x = 0$. Let $V = D \to \mathbb{R}$ be a continuously differentiable function, such that

$$V(0) = 0 \text{ and } V(x) > 0 \text{ in the domain } D - \{0\} \tag{2.13}$$

$$\dot{V}(x) \leq 0 \text{ in } D \tag{2.14}$$

Then, $x = 0$ is stable. Moreover, if:

$$\dot{V}(x) < 0 \text{ in } D - \{0\} \tag{2.15}$$

then the origin " $x = 0$ " is asymptotically stable. This theorem is called "Lyapunov theorem", the proof of this theorem is available in Khalil book [73].

Proposition 2.1 *If the large system is stable, i.e. it has a quadratic Lyapunov function "V" that satisfied the stability conditions, then the reduced system obtained using the proposed algorithm is stable. In other words, there exists quadratic Lyapunov function "V_r" that satisfied the stability conditions.*

Mathematically, Consider we have a nonlinear system of order n as the following:

$$\begin{aligned} \dot{x} &= f(x) + Bu \\ y &= Cx \end{aligned} \tag{2.16}$$

If there exist a Lyapunov function $V(x) = x^T Px$, such that $\dot{V}(x) = -Q(x)$, $\forall x \neq 0$, i.e.

$$\begin{aligned} \dot{V}(x) &= \dot{x}^T Px + x^T P\dot{x} \\ &= f^T(x)Px + x^T Pf(x) \\ &= -Q(x) \end{aligned} \tag{2.17}$$

Then, for any right-projection matrix V, there exists a left-projection matrix U such that the obtained-reduced system is stable, i.e. the reduced-order system have a Lyapunov function V_r, such that $\dot{V}_r(x_r) = -Q(Vx_r)$

Proof

Assume that there exists a matrix V, such that:

$$x = Vx_r \tag{2.18}$$

By substituting equation (2.18) into equation (2.16), we got the following:

$$\begin{aligned} V\dot{x}_r &= f(Vx_r) + Bu \\ y_o &= CVx_r \end{aligned} \tag{2.19}$$

Now, by choosing a matrix U, where

$$U = (V^T PV)^{-1} V^T P \tag{2.20}$$

here, $V \in \mathbb{R}^{n \times r}$, $U \in \mathbb{R}^{r \times n}$ and $UV = I_r$.

After that, the equation (2.19) is multiplied by the matrix U from left side

$$\begin{aligned} UV\dot{x}_r &= Uf(Vx_r) + UBu \\ \dot{x}_r &= Uf(Vx_r) + UBu \end{aligned} \tag{2.21}$$

Assume that the function $V_r(x_r)$ is the Lyapunov function for the reduced-order system, where: $V_r(x_r) = x_r^T V^T PVx_r$.

Then, the derivation of the function $V_r(x_r)$ is:

$$\begin{aligned}\dot{V}_r(x_r) &= \dot{x}_r^T V^T P V x_r + x_r^T V^T P V \dot{x}_r \\ &= f^T(Vx_r)U^T V^T P V x_r + x_r^T V^T P V U f(Vx_r)\end{aligned} \tag{2.22}$$

Substitute equation (2.20) into equation (2.22), we get the following:

$$\begin{aligned}\dot{V}_r(x_r) = f^T(Vx_r)((V^T PV)^{-1}V^T P)^T V^T P V x_r \\ + x_r^T V^T P V((V^T PV)^{-1}V^T P) f(Vx_r)\end{aligned} \tag{2.23}$$

Equation (2.23) is equivalent to:

$$\begin{aligned}\dot{V}_r(x_r) = f^T(Vx_r)P^T V(V^T PV)^{-T}(V^T P^T V)^T x_r \\ + x_r^T(V^T PV)(V^T PV)^{-1}V^T P f(Vx_r)\end{aligned} \tag{2.24}$$

Since the matrix P is symmetric, that means $P = P^T$, then

$$(V^T PV)^{-T}(V^T P^T V)^T = I_r \tag{2.25}$$

Using the property (2.25), we can re-write equation (2.24) as the following:

$$\dot{V}_r(x_r) = f^T(Vx_r)PVx_r + x_r^T V^T P f(Vx_r) \tag{2.26}$$

Using the assumption in equation (2.18), we can re-write equation (2.26) as the

following:

$$\dot{V}_r(x_r) = f^T(x)Px + x^T Pf(x) \tag{2.27}$$

Based on equation (2.17), the left side of equation (2.27) is equivalent to the derivation of Vx_r, which can be represented mathematically as :

$$\dot{V}_r(x_r) = \dot{V}(Vx_r) \tag{2.28}$$

Then, for the reduced-order system, the derivative of its lyapunov function is:

$$\dot{V}_r(x_r) = -Q(Vx_r) \tag{2.29}$$

By using the lasalle theorem [73], the reduced-order system using the proposed technique is stable.

2.5 Numerical Result and Simulation

2.5.1 Nonlinear Circuit Example 1

To verify the developed approach, a nonlinear circuit is reduced from order 6 to order 3 using this method. Figure (2.2) shows the structure of the nonlinear circuit that used previously by Chen in 1999 [11]. The nonlinearity lies in the resistors, where it depends on the relationship between the resistors and the voltage applied on them. Here, for each resistor, we labeled two ends by a and b to specify the orientation due to probability of asymmetric, non-linear resistors. To make the

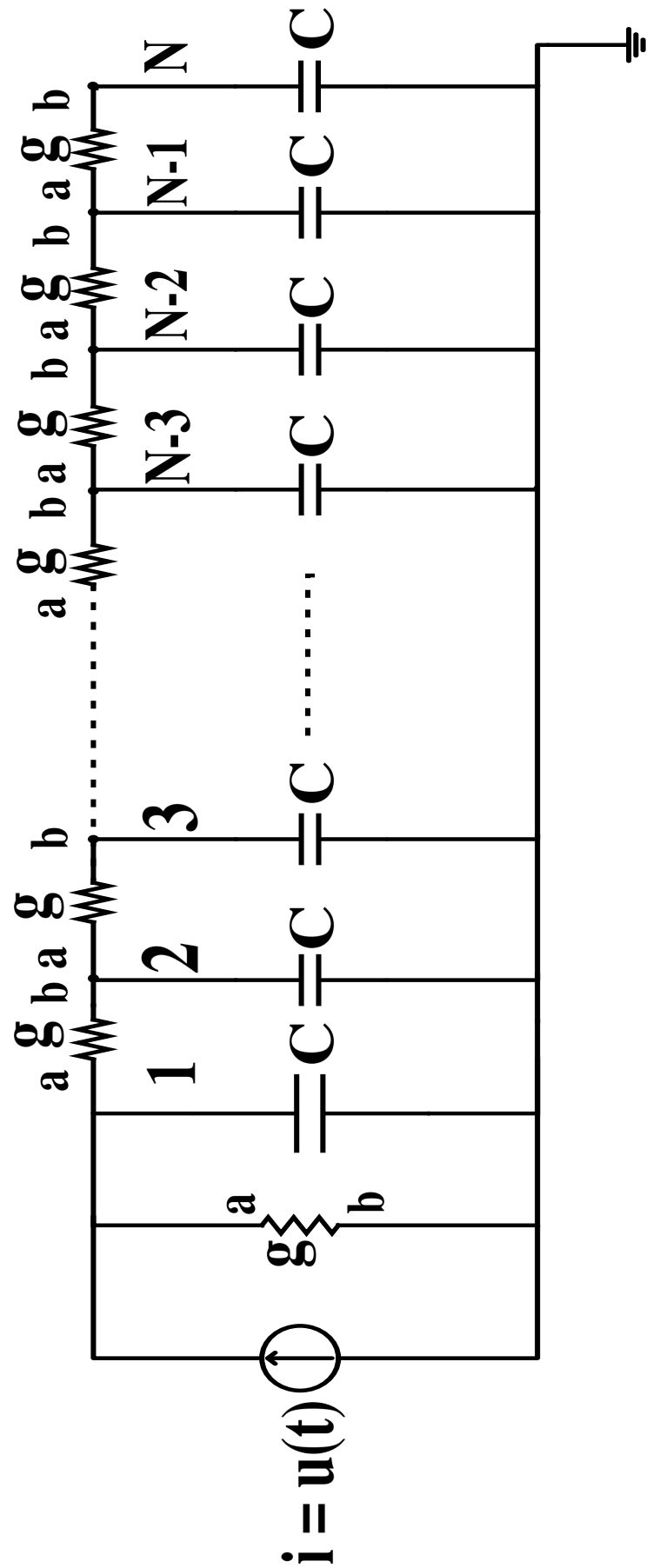

Figure 2.2: Nonlinear circuit example 1

electrical circuit simple, it was assumed that all resistors have the same value. Moreover, the depend profile, which is called the current-voltage (I-V) profile, is given by the function g. The current, that follows from a to b, is given by $I = g(x)$. In this example, the input is taken to be applied to the first node, where $i = u(t)$. In addition, the system output is the potential at the first node with state variables being the potential at node from N_1 to N_n.

The current at each node of this circuit and the voltage equations can be written as follows:

$$\begin{aligned} u(t) &= C\frac{dx_1}{dt} + g(x_1 - x_2) + g(x_1) \\ g(x_1 - x_2) &= C\frac{dx_2}{dt} + g(x_2 - x_3) \\ &\vdots \\ g(x_{n-1} - x_n) &= C\frac{dx_n}{dt} \end{aligned}$$

The equations above are equivalent to the standard form that can be written as follows:

$$C\tfrac{dx}{dt} = \begin{pmatrix} -g(x_1) - g(x_1 - x_2) \\ g(x_1 - x_2) - g(x_2 - x_3) \\ \vdots \\ g(x_{n-2} - x_{n-1}) - g(x_{n-1} - x_n) \\ g(x_{n-1} - x_n) \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} u(t) \qquad (2.30)$$

Here, $g(x)$ is the nonlinear function in this circuit, where $g(x) = e^{40x} + x - 1$

To make this circuit simple, we assume that $C = 1$ for all nodes, we can re-write

equation (2.30) as follows:

$$\frac{dx}{dt} = \begin{pmatrix} -g(x_1) - g(x_1 - x_2) \\ g(x_1 - x_2) - g(x_2 - x_3) \\ \vdots \\ g(x_{n-2} - x_{n-1}) - g(x_{n-1} - x_n) \\ g(x_{n-1} - x_n) \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} u(t) \tag{2.31}$$

Now, the current source $i = u(t)$ is selected to be applied to the system above. One of the most commonly used sources to study the behavior of the systems is a step source. This is because it is easy to represent and observe how the system behave [11]. By comparing the behavior of the reduced system to the original one, the accuracy of our developed approach can be assessed.

To make it simple, we assume that $n = 6$, then the equations of this circuit can be written as follows:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_6 \end{pmatrix} = \begin{pmatrix} -g(x_1) - g(x_1 - x_2) \\ g(x_1 - x_2) - g(x_2 - x_3) \\ \vdots \\ g(x_4 - x_5) - g(x_5 - x_6) \\ g(x_5 - x_6) \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} u \tag{2.32}$$

where $g(x) = e^{40x} + x - 1$

It's clear from figure (2.3) that the behavior of the obtained nonlinear system of order 3 is closed to the behavior of the original nonlinear system of order 6. The projection-based method is used to obtain the reduced-order system, and the step signal used as an input to both systems.

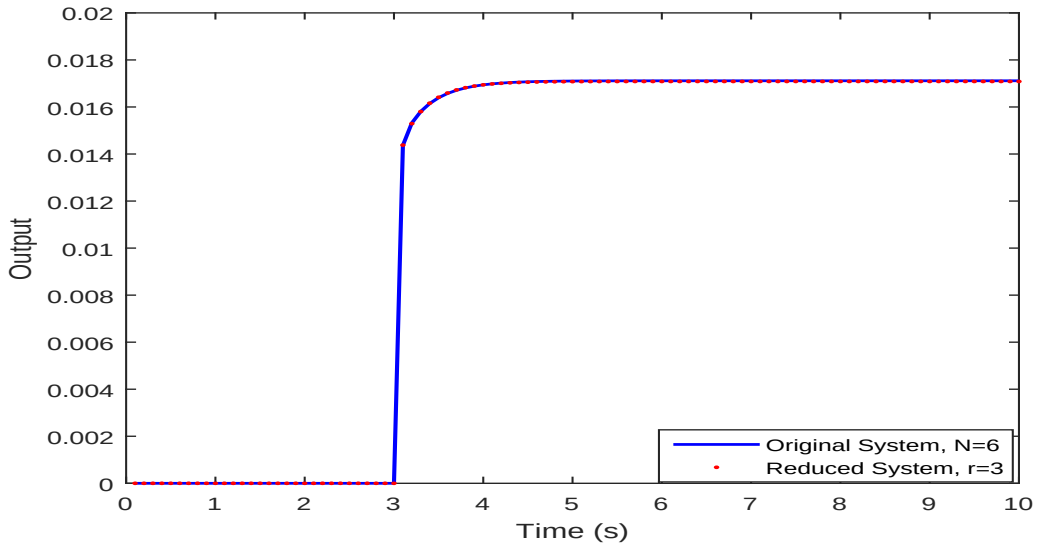

Figure 2.3: Comparison of the original system response vs. the reduced system

2.6 Conclusion

To conclude, the developed method of MOR for a class of nonlinear systems gives a reduced model that its behavior is closed to the behavior of the original system. The main advantage of this approach is using the projection technique to project the original states of the nonlinear system, no need to approximate the original system. This advantage reduced the error that is introduced to the system during approximation. In this method, the starting point is linearizing the original system. Then, the projection matrix "V" is constructed form the corresponding

between the balanced truncation and the projection technique. After obtaining the projection matrix "V", it will be used to project the states of the original system. Moreover, the stability analysis of the reduced system is studied, so that the reduced system obtained using the developed approach is proved to be stable.

CHAPTER 3

PSO-BASED ALGORITHM FOR NONLINEAR SYSTEMS

3.1 Introduction

PSO stands for Particle Swarm Optimization technique that uses particles to explore the search space of the problem. These particles are moving towards the optimal solution of the given problem. PSO introduced by Kennedy and Eberhart in 1995 [74]. The idea of this technique comes from an observation of some swarming habits in our live such as swarming of blocks of birds and schools of fishes. Particle swarm optimization is a population-based search algorithm. This algorithm has some advantages over the other optimization techniques, e.g. its concept is simple and easily programmable. Moreover, it can converge speedily to the optimal solution within the search space of the problem. When we take the behavior of flocking birds as an example, we note that when the birds are

searching for food in a region where there is only one place of food in it, all birds do not know where the food is, but they know how far away it is in each iteration. Therefore, the flocks of birds tend to follow the bird that is closest to the food up to the desired area. In PSO, each bird in the search space of the problem represents a single solution "Particle". All particles have positions and velocities and with each iteration the particles evaluate the fitness function and get the fitness value. The best fitness value is saved and the particles move in the search space following the current optimum particles.

The procedures of the PSO are shown below [75]:

1. PSO algorithm starts by initializing the population of the particles (particles at time t) with a random values for their positions and velocities, these values must be within the search space of the problem.

2. Evaluate the fitness function of the given problem for each particle in the search space.

3. In each iteration, the PSO algorithm compares the particles fitness values with the saved value of Pbest. In this algorithm, the best fitness value is called "Pbest". If the current fitness value of the particles is better than the saved Pbest, then the value of Pbest is updated to the current fitness value, i.e. Pbest=current fitness value. Moreover, the PSO saved the position of the new Pbest.

4. Set the best fitness value achieved so far as the global best (gbest), i.e. gbest= the best value of Pbest, and save its position.

5. Update the velocity for each particle as stated in the following:

$$v_j(t) = w(t)v_j(t-1) + c_1 r_1(x_j^*(t-1) - x_j(t-1)) + c_2 r_2(x_j^{**}(t-1) - x_j(t-1)) \tag{3.1}$$

where

t	Iteration number.
j	the particle index.
v	the velocity of the j^{th} particle.
x	the position of the j^{th} particle.
c_1 & c_2	positive-acceleration constant, usually $c_1 = c_2 = 2$.
w	inertial weight.
r_1 & r_2	random numbers between 0 and 1.
x_j^*	best position for j^th particle (pbest)
x_j^{**}	global best, best of pbest.

6. Update the position for each particle as stated in the following:

$$x_j(t) = v_j(t) + x_j(t-1) \tag{3.2}$$

Also, here the inertial weight will be updated as stated in the following:

$$w(t) = \alpha w(t-1) \tag{3.3}$$

7. Loop to the step 2, if one of the criteria is satisfied, then stop. These criteria might be one of the following:

   - Achieve the desired fitness value.
   - Reach the maximum number of iteration, this value is determined by the programmer.

The whole procedures of the particle swarm optimization "PSO" can be summarized in the flowchart given in figure 3.1.

3.2 A Motivating Example

As a motivation example, the particle swarm optimization algorithm is used to obtain a reduced linear system. Consider the linear system has the following form:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{3.4}$$

where $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $C \in \mathbb{R}^{P\times n}$. The objective is to use the PSO to find the reduced linear system of the form:

$$\begin{aligned} \dot{x}_r &= A_r x_r + B_r u \\ y_o &= C_r x \end{aligned} \tag{3.5}$$

where the order of this system is r, such that $r \ll n$, and $x_r \in \mathbb{R}^r$, $A_r \in \mathbb{R}^{r\times r}$, $B_r \in \mathbb{R}^{r\times m}$, $C_r \in \mathbb{R}^{P\times r}$. However, the behavior, or the output "y_o", of the

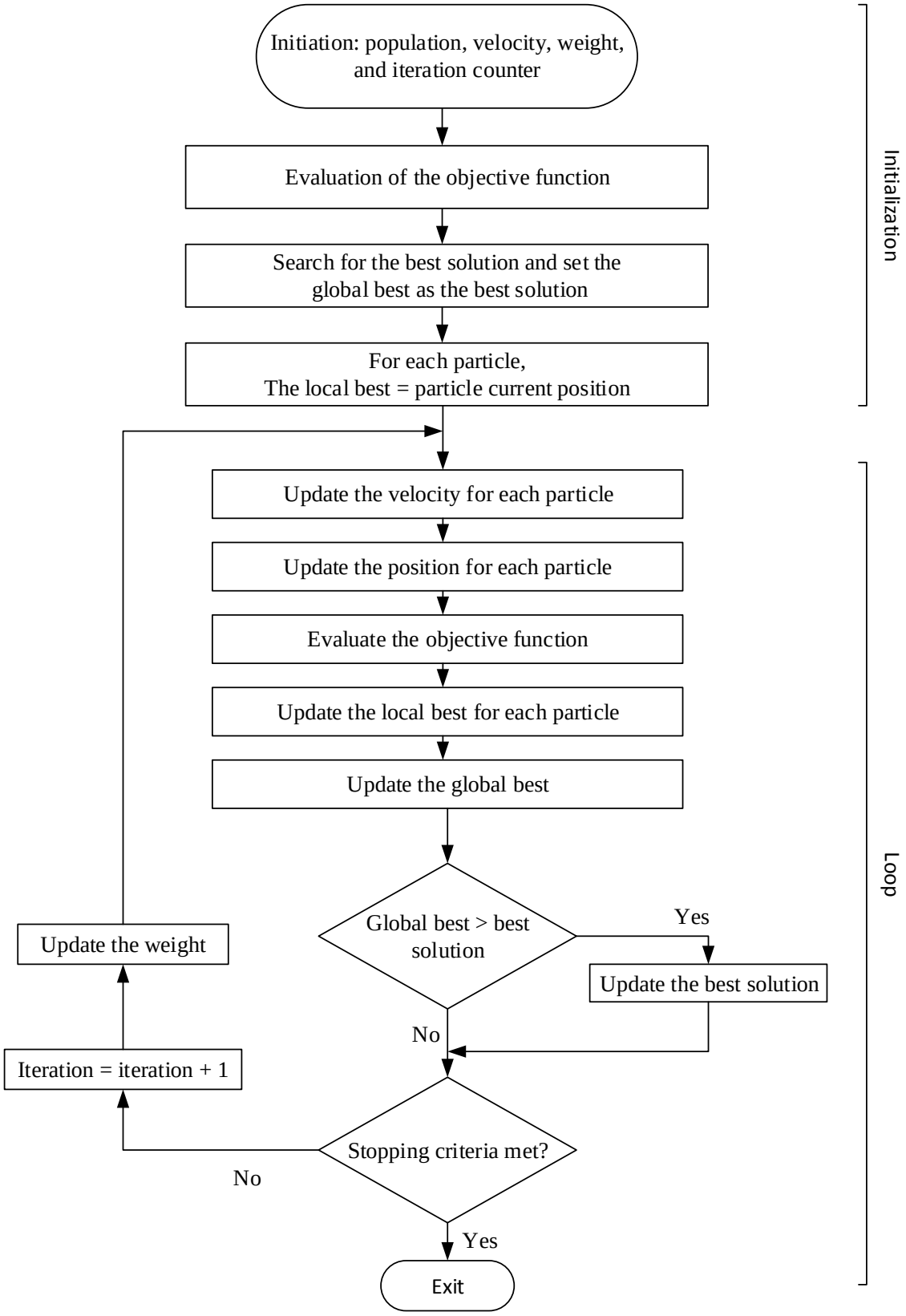

Figure 3.1: Particle Swarm Optimization Flowchart

obtained system should be closed to the behavior "y" of the original system. To achieve this objective, we need to set some definitions and parameters as listed below:

- Set all parameters of the state space of the original system.
- Set the suitable order of the reduced system and consider that we have the original system (3.4), the singular values of this system could be defined as $\sigma = \lambda^{1/2}(PQ)$, where Q and P are the controllability and the observability gramians, respectively. If r is the order of the reduced system, then $\sigma_{r+1} \leq ||y - y_o||_\infty$, where σ_{r+1} is the $(r+1)$ Hankel singular value "HSV" of the original system (3.4). In other words, the H_∞ of the reduced system will not be less than the highest HSV that has been dropped.
- Set the parameters of the PSO as the following:
  - Set the number of the swarm "N".
  - Set the size of the particles which needed in this problem "P".
  - Set the iteration counter as $t = 1$
  - Define the search space of the problem. This could be determined by setting the maximum and minimum values for each parameter in the swarm.
  - Define the fitness function. This function is the most important point in PSO, it links PSO with the problem that needed to be optimized. In our problem, the fitness function is the H_∞-norm of the error E, where

E could be defined as the maximum error between the outputs of the original and the obtained systems. In other words, $fitness = E = ||y - y_o||_\infty$. Then, the objective is getting the minimum value of the error ($fitness = min(E)$).

The main steps for using the particle swarm optimization algorithm to obtain the reduced linear system can be summarized as follows:

1 Select random values within the range specified for the positions $X(0)$ and velocities $V(0)$;

$$X(0) = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_i \\ \vdots \\ \vec{x}_N \end{bmatrix}$$

where $\vec{x}_i = \begin{bmatrix} x_{i1} & \dots & x_{ip} \end{bmatrix}$

V is the velocity matrix,

$$V(0) = \begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_i \\ \vdots \\ \vec{v}_N \end{bmatrix}$$

where $\vec{v}_i = \begin{bmatrix} v_{i1} & \dots & v_{ip} \end{bmatrix}$

2 Vectorize the rows of the matrix X to produce vectors $\vec{x}_1$, $\vec{x}_2$ ... $\vec{x}_N$. Choose the first vector "$\vec{x}_1$" and construct the reduced system "sysr", then evaluate the objective function "$E = \| \ sys - sysr \ \|_\infty$", where "sys" is the original system. Save the value of E and do it for $\vec{x}_2$, and compare between the value of the objective function "E", continue up to $\vec{x}_N$. Choose the lowest value of E to be the local and global best for initial values.

P_{best} is the individual best,

$$P_{best} = \begin{bmatrix} \vec{p}_{best,1} \\ \vec{p}_{best,2} \\ \vdots \\ \vec{p}_{best,i} \\ \vdots \\ \vec{p}_{best,N} \end{bmatrix}$$

where $\vec{p}_{best,i} = \begin{bmatrix} p_{best,i1} & \dots & p_{best,ip} \end{bmatrix}$

The global best is defined as best of the personal best, i.e. $g_{best} = \begin{bmatrix} g_1 & g_2 & \dots & g_P \end{bmatrix}$

3 Set the iteration number as $t = 1$.

4 Update the velocity according to equation (3.1).

5 Update the position of each particle according to equation (3.2).

6 Compute the fitness value of the new position.

7 Find p_{best} of each particle according to:

$$p_{best,i}^{t+1} = \begin{cases} p_{best}^{t} & if f_i^{t+1} > p_{best,i}^{t} \\ x_i^{t+1} & if f_i^{t+1} \leq p_{best,i}^{t} \end{cases}$$

where f is the fitness value.

8 Find the global best according to: $g_{best} = min(P_{best,i}^{t})$

9 Stopping criterion:

check if the desired fitness value satisfied or the maximum number of iteration is met. If one of these criteria is achieved, then stop and the solution will be the g_{best}. The position that satisfies this value will be used to construct the reduced system. Otherwise, update the inertial weight, t=t+1, and go to step 4.

The flowchart of using PSO in model order reduction shows in figure 3.2.

Example: Consider we have the linear system of order 6 as showing below:

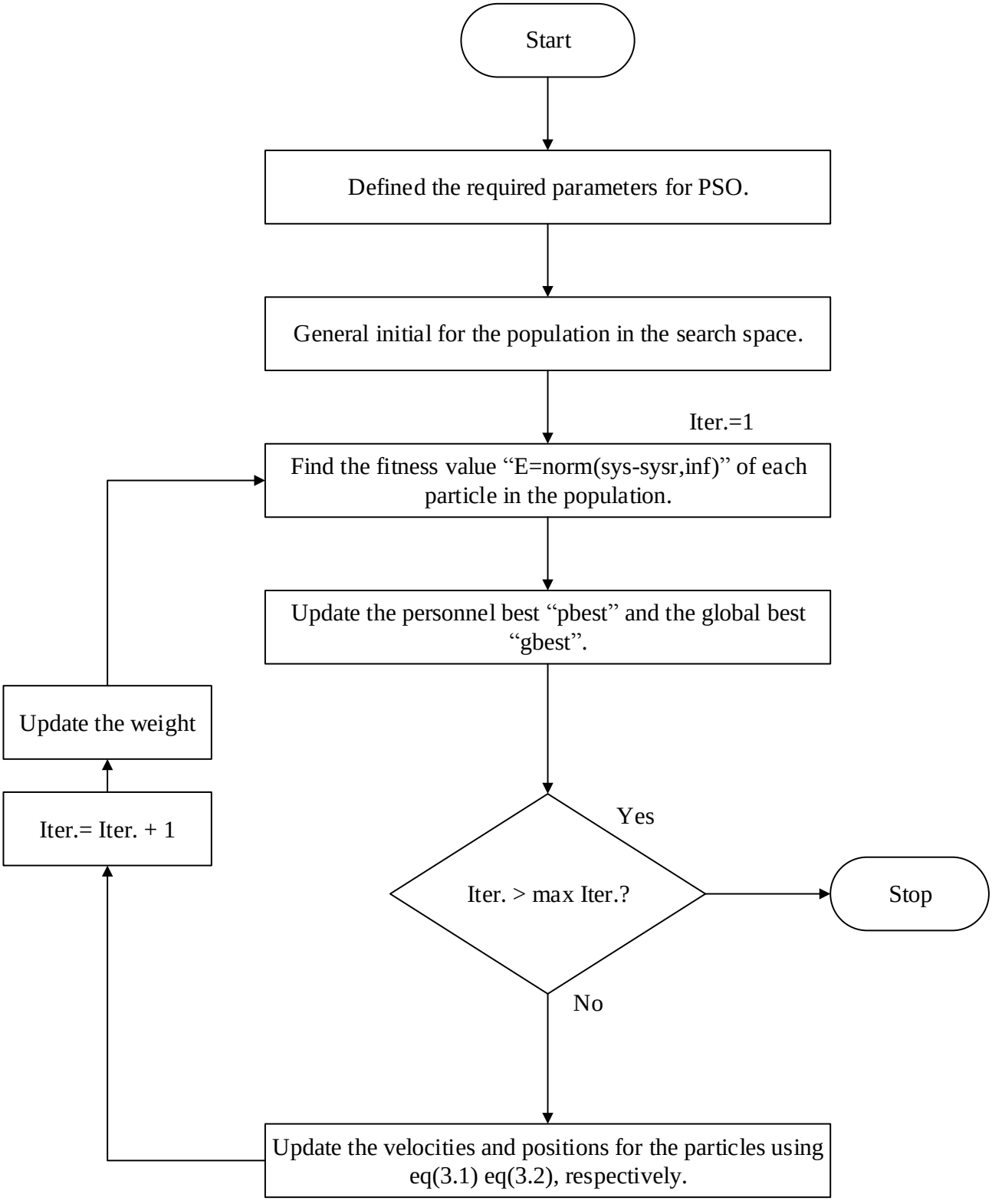

Figure 3.2: Flowchart of using the “PSO” in model-order reduction of linear systems

$$\dot{x} = \begin{bmatrix} -41 & -571 & -3491 & -10060 & -13100 & -6000 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} u \tag{3.6}$$

$$y = \begin{bmatrix} 2 & 70 & 762 & 3610 & 7700 & 6000 \end{bmatrix} x$$

The obtained system using "PSO" is of order 2, and it's parameters are as follows:

$$\dot{x}_r = \begin{bmatrix} -7.0291 & -3.7220 \\ -4.8124 & -3.9709 \end{bmatrix} x_r + \begin{bmatrix} 1.1779 \\ 0.4663 \end{bmatrix} u \tag{3.7}$$

$$y_o = \begin{bmatrix} 2.2553 & -1.4078 \end{bmatrix} x_r$$

We compare the output of both systems, the original and the obtained systems. Figure 3.3 shows that the behavior of the reduced system "with order 2" is close to the behavior of the original system "with order 6". The step signal is used as an input to both systems.

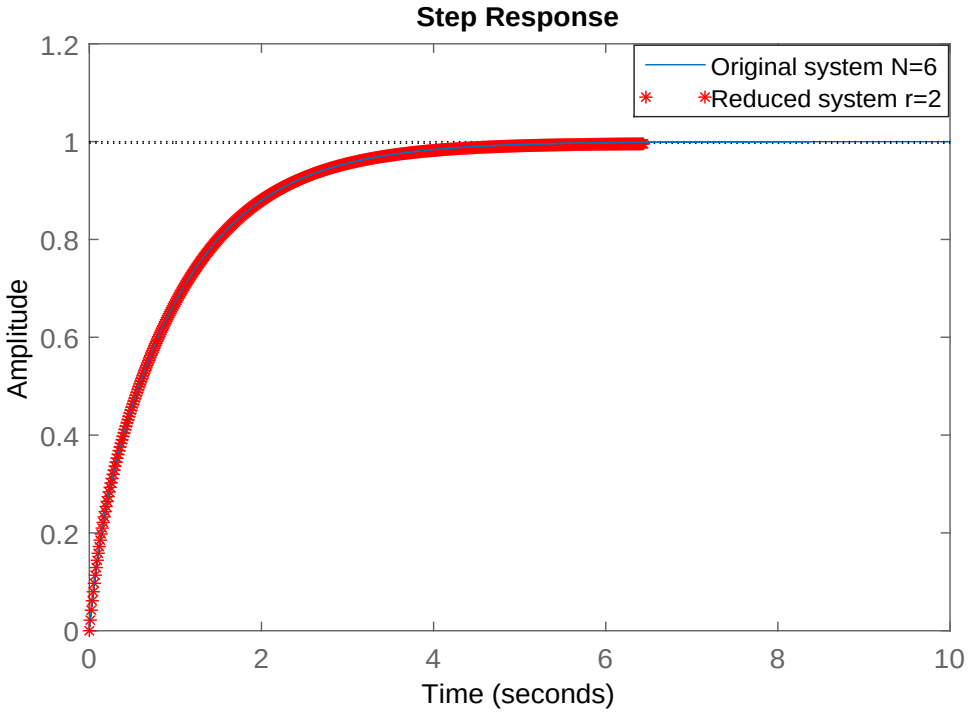

Figure 3.3: System outputs y and y_o

3.3 Model Reduction of Nonlinear Systems Using PSO-technique

In this section, the PSO is used to construct the projection matrix "V", which used to project the states of the original system which in turn produces a new system with less order. However, the behavior of the obtained system is closed to the behavior of the original system.

Consider the nonlinear system described by the following dynamics

$$\begin{aligned} \dot{x} &= f(x) + Bu \\ y &= Cx \end{aligned} \tag{3.8}$$

where $x \in \mathbb{R}^n$, f is a nonlinear function. Here, the first equation is called the "state equation", and the other equation is called "output equation".

The objective of "MOR" is to obtain the reduced nonlinear system of the form

$$\begin{aligned} \dot{x}_r &= f_r(x_r) + B_r u \\ y_o &= C_r x_r \end{aligned} \tag{3.9}$$

where $x_r \in \mathbb{R}^r$, and f_r is a nonlinear function. The behavior "y_o" of the obtained nonlinear system is closer to the behavior of the original nonlinear system "y"

The main steps of using the Particle Swarm Optimization to obtain the reduced nonlinear system can be summarized as follows:

1: Select random values within the range specified for the positions $X(0)$ and velocities $V(0)$;

$$X(0) = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_i \\ \vdots \\ \vec{x}_N \end{bmatrix}$$

where $\vec{x}_i = \begin{bmatrix} x_{i1} & \dots & x_{ip} \end{bmatrix}$

V is the velocity matrix,

$$V(0) = \begin{bmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_i \\ \vdots \\ \vec{v}_N \end{bmatrix}$$

where $\vec{v}_i = \begin{bmatrix} v_{i1} & \dots & v_{ip} \end{bmatrix}$

2: Vectorize the rows of the matrix X to produce vectors $\vec{x}_1$, $\vec{x}_2$ ... $\vec{x}_N$. Choose the first vector "$\vec{x}_1$" and construct the projection matrix V, from this matrix we defined the second matrix U, where $U = (V^T P V)^{-1} V^T P$, such that $UV = I_r$.

3: Use the obtained matrices V and U, which we got from step 2, to project the original nonlinear system and get the reduced system

$$\begin{aligned} \dot{x}_r &= U f_r(Vx) + UBu \\ y_o &= CVx_r \end{aligned} \tag{3.10}$$

4: Compute the fitness value for this position, where the fitness value is the H_∞ of the error, i.e. $fitness\ value = \| y - y_o \|_\infty$.

5: Take the vectors "$\vec{x_2}$ ... $\vec{x_N}$, and do the same procedures as we did for $\vec{x_1}$.

Choose the lowest value of the fitness value to be the local and global best for initial values. P_{best} is the individual best,

$$P_{best} = \begin{bmatrix} \vec{p}_{best,1} \\ \vec{p}_{best,2} \\ \vdots \\ \vec{p}_{best,i} \\ \vdots \\ \vec{p}_{best,N} \end{bmatrix}$$

where $\vec{p}_{best,i} = \begin{bmatrix} p_{best,i1} & \dots & p_{best,ip} \end{bmatrix}$

The global best is defined as best of the personal best, i.e. $g_{best} = \begin{bmatrix} g_1 & g_2 & \dots & g_P \end{bmatrix}$

6: Set the iteration number as $t = 1$.

7: Update the velocities for all particles according to the equation (3.1).

8: Update the positions for all particles according to the equation (3.2).

9: Compute the fitness value of the new positions, and find the personnel and global best.

10: Stopping criterion:

Check if one of these criteria has achieved:

- The desired fitness value satisfied.

- Maximum number of iteration is met.

If one of them has been achieved, then stop and the solution is the global best. Otherwise, update the inertial weight, $t = t + 1$, and go to step 7.

To give a quick overview for the developed algorithm, the previous procedures of MOR for a class of nonlinear systems using the Particle Swarm Optimization can be summarized in the flowchart in figure 3.4.

3.4 Stability Analysis

The stability of the reduced system has been proved in Chapter 2, Section 2.4.

3.5 Simulation

To verify the developed algorithm, we reduced the nonlinear system that is used in [25] to the third order. The original nonlinear system is described below:

$$\begin{aligned} \dot{x_1} &= -2x_1 - x2 - 2u \\ \dot{x_2} &= -2x_2 - x3 + u \\ \dot{x_3} &= -2x_3 - x4 + sin(x_3) \\ \dot{x_4} &= -2x_4 + u \\ y &= 2x_1 + x_2 + x_3 + x_4 \end{aligned} \tag{3.11}$$

Our aim is to approximate this system using PSO to a reduced system of order three.

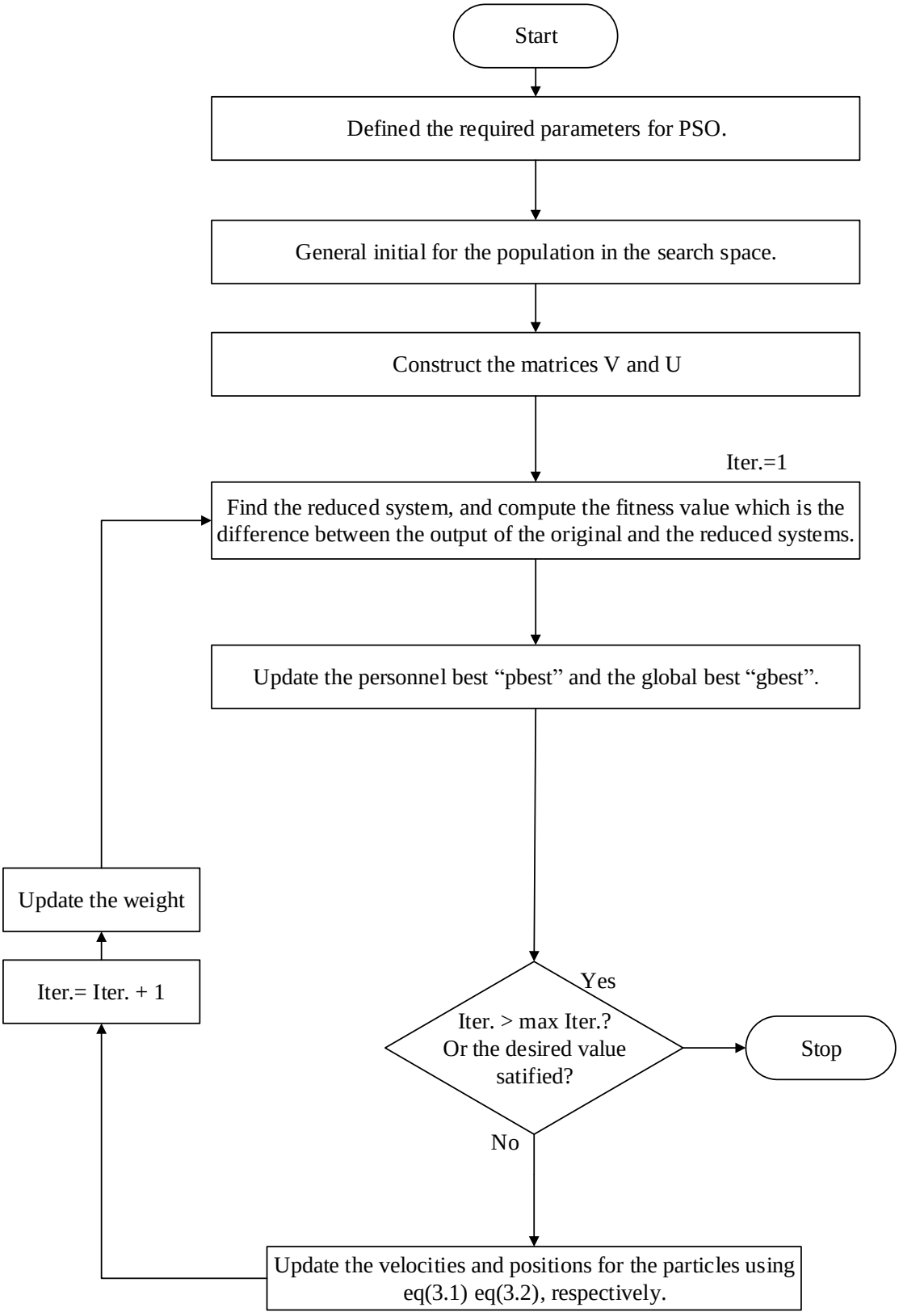

Figure 3.4: Flowchart of Model Order Reduction of Nonlinear Systems

Figure 3.5 shows the response of both systems, the original and the reduced

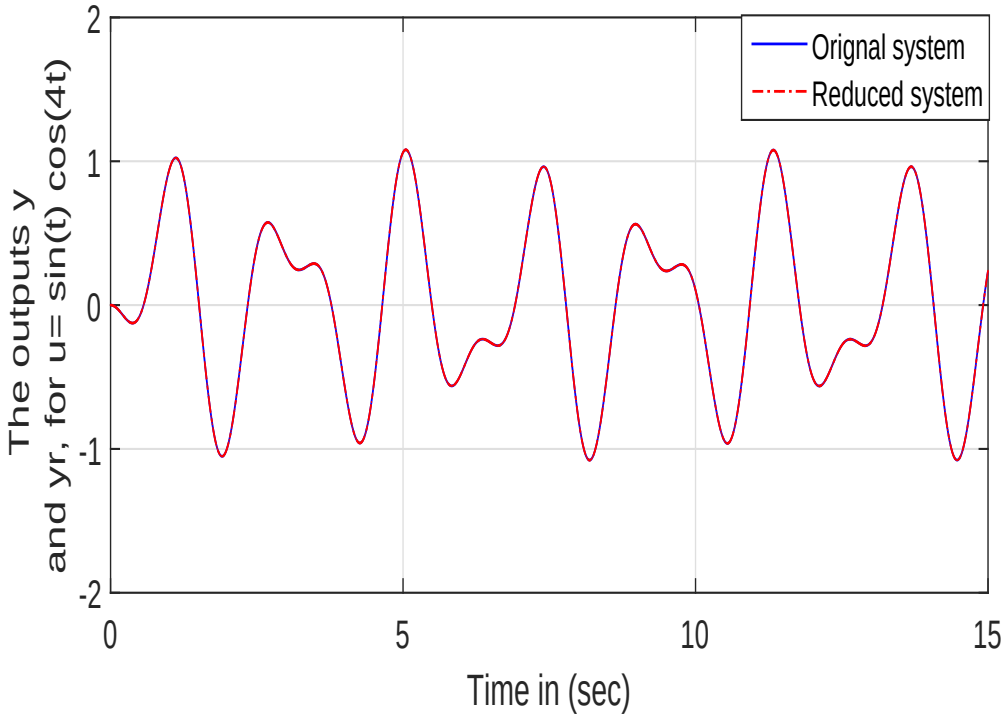

Figure 3.5: Outputs for both systems, the original and the obtained systems, using the developed approach.

systems, using a bounded input, where $u = 2sin(t)cos(4t)$. From Figure 3.5, it is apparent that the reduced nonlinear system using the "PSO" gives a good approximation to the original nonlinear system described in equation (3.11).

3.6 Conclusion

To conclude, a new efficient and practical technique is developed for doing MOR for a class of nonlinear systems. The developed approach uses the projection technique directly to the states of the original system. The Particle Swarm Optimization "PSO" is used to construct the projection matrix "V", which in turn

project the states of the original system. The fitness value of the PSO is the H_∞ of the error "E". Moreover, the reduced system using this approach is proved mathematically to be a stable system. Finally, to verify the efficiency of this approach, we used it to gain a reduced system, and its output was compared to the output of the original system.

CHAPTER 4

ASSESSMENT OF THE DEVELOPED PROCEDURES

4.1 Introduction

Chapter 2 and Chapter 3 presented a theoretical analysis for the developed model-reduction algorithms and the stability analysis of these algorithms was studied. Since the PSO-based algorithm gives better result than the projection-based method, then the PSO-based approach is applied to two electrical circuits. The PSO-based method has been validated by comparing the simulation of the output for the reduced system with the original one. In addition, this method is compared to three existing methods, quadratic reduction method [11], direct nonlinear reduction with variational analysis [18], and quadratic bilinear systems [76], to investigate the efficiency and the accuracy of the developed approach.

4.2 Nonlinear Circuit Example 2

In this example, we use the same circuit as in Section (2.5.1), but the dimension of this circuit is increased to 20. The equation of the circuit can be written as follows:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_6 \end{pmatrix} = \begin{pmatrix} -g(x_1) - g(x_1 - x_2) \\ g(x_1 - x_2) - g(x_2 - x_3) \\ \vdots \\ g(x_{18} - x_{19}) - g(x_{19} - x_{20}) \\ g(x_{19} - x_{20}) \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} u \tag{4.1}$$

where $g(x) = e^{40x} + x - 1$

4.3 Nonlinear Circuit Example 3

Figure (4.1) shows the structure of another nonlinear electrical circuit. The I-V characteristic of the nonlinearity is given by $i = \psi(x)$. The differential equations of this circuit is given as:

$$\begin{aligned} \dot{x}_1 &= \frac{1}{R1}(E - x_1) - \frac{1}{Rc}(x_1 - x_2) - \psi(x_1) \\ \dot{x}_2 &= \frac{1}{Rc}(x_1 - x_2) - \frac{1}{Rc}(x_2 - x_3) - \psi(x_2) \\ &\vdots \\ \dot{x}_{19} &= \frac{1}{Rc}(x_{18} - x_{19}) - \frac{1}{Rc}(x_{19} - x_{20}) - \psi(x_{19}) \\ \dot{x}_{20} &= \frac{1}{Rc}(x_{19} - x_{20}) - \frac{1}{R2}(x_{20} - E) - \psi(x_{20}) \end{aligned} \tag{4.2}$$

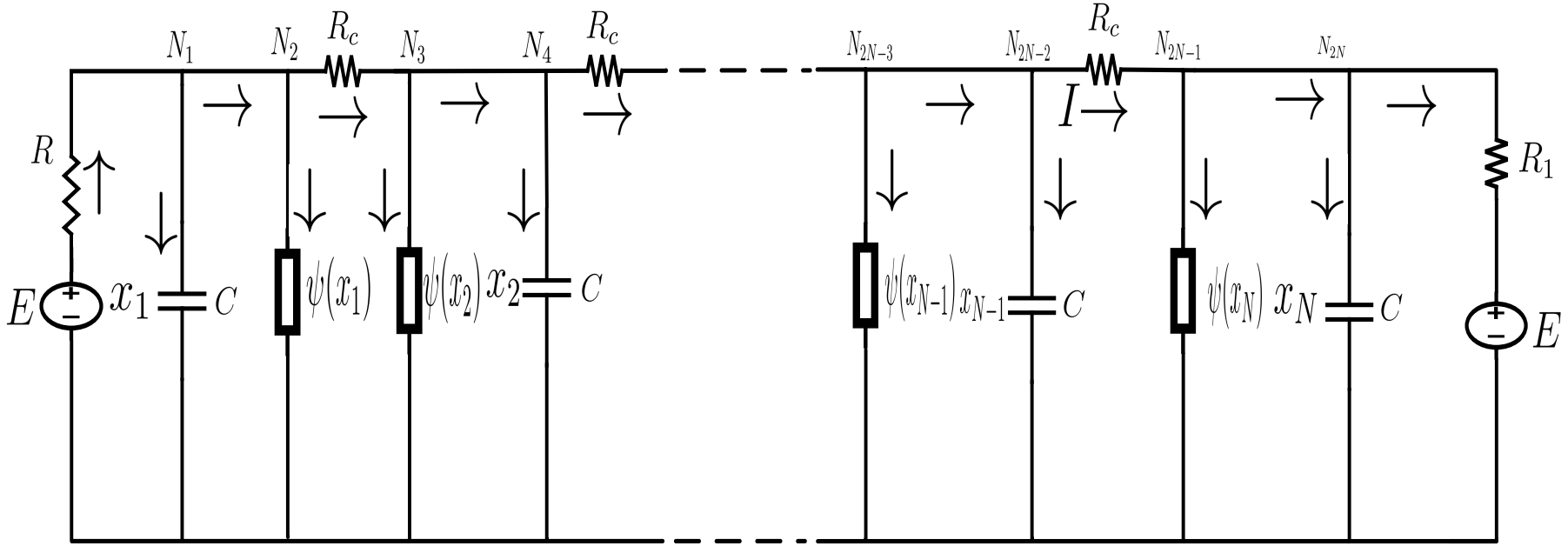

Figure 4.1: Nonlinear circuit example 2

where E is the input, $\psi(x)$ is the nonlinear function, and $\psi(x) = 10^{-3}(17.76x - 103.79x^2 + 299.62x^3)$. We can define the output of this circuit (I) as follows:

$$y = I = \begin{pmatrix} 0 & 0 & 0 & 0 & \frac{1}{Rc} & \frac{-1}{Rc} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{19} \\ x_{20} \end{pmatrix} \tag{4.3}$$

In this chapter, the PSO-based method has been applied to these two electrical circuits in comparison with three existing studies. The original systems are of order 20, and the reduced systems are of order 4. The following sections contain a brief summary of these studies with comparison to the new approach.

4.4 Comparison with Quadratic Reduction Method

This method uses Taylor expansion to expand the terms of nonlinear systems. The idea of this method is based on the quadratic approximation of the nonlinear systems by deleting the parts of Taylor expansion which are greater than two degree. In other words, this method obtains a quadratic reduced system for the

original nonlinear system [11]. The nonlinear function can be written as follows:

$$f(x(t)) = f(0) + A_1 x(t) + A_2(x(t) \otimes x(t)) + A_3(x((t) \otimes x(t) \otimes x(t)) + ... \quad (4.4)$$

If the nonlinear function "$f(x)$" was approximated by the first two terms of the Taylor expansion (4.4), then the following quadratic nonlinear system

$$\begin{aligned} \dot{x} &= Ax + x^T W x + Bu(t) \\ y &= Cx(t) \end{aligned} \quad (4.5)$$

is the approximated system to the original nonlinear system (4.5). Where $A \in \mathbb{R}^{n \times n}$, and W is $3 - D$ array or an $n \times n \times n$ tensor. If we approximate $x \approx V x_r$, then the reduced will be as follows:

$$\begin{aligned} \dot{x}_r &= A_r x_r + A_r V T A^{-1} x_r^T V^T W V x_r + A_r V^T B u(t) \\ y &= CV x_r \end{aligned} \quad (4.6)$$

where $A_r = (V^T A^{-1} V)^{-1}$, the V matrix is obtained using the following:

$$spancolumn\{V\} = span\{A^{-1}B, ..., A^{-q}B\}$$

Since the quadratic system is a more precise approximation of the original nonlinear system than the linearized system, then this method is more precise than the linearization method [11].

For the nonlinear electrical circuit in example 2, a specific current source is used

to be a step input ($u = 0$, when $t \leq 3$, else $u = 1$, and $t \in [0, 10]$). For a clear comparison between the reduced methods, the output of the original nonlinear systems is plotted for the nonlinear circuit in example 2 in Section (4.2) with size 20 using the PSO-base method and the reduced quadratic approximation method, and the dimension of the reduced order is 4. From this example, it is clear that the reduced system using the developed approach is more accurate than the reduced system that obtained using the quadratic method (Figure (4.2)). The response of example 3 mentioned is Section (4.3) is in Figure (4.3). In this

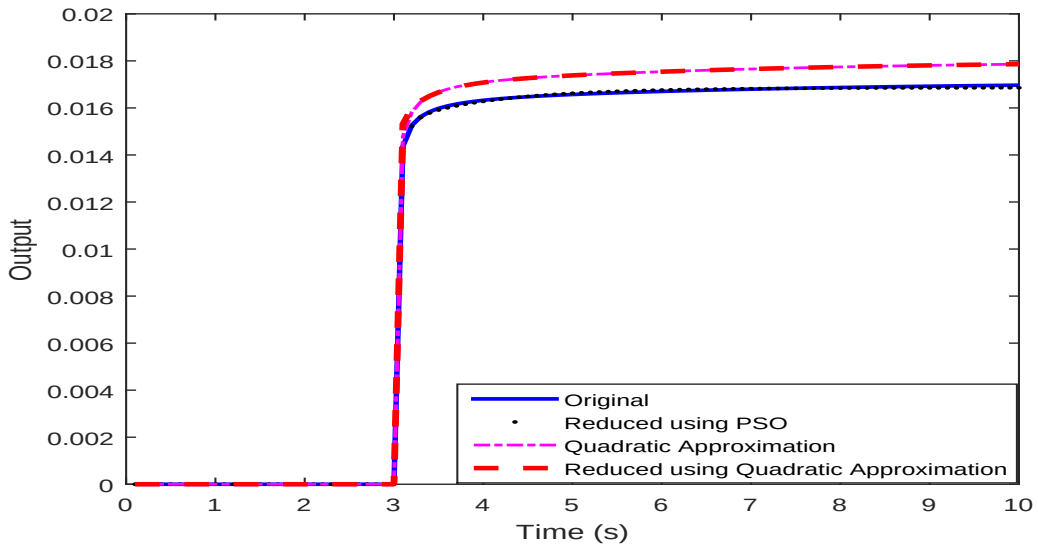

Figure 4.2: Comparison of the system response in example 2 using PSO-based method vs. Quadratic Reduction Method

figure, the dimension of the reduced order is 4, and both, the PSO-base method and the quadratic nonlinear system [11], are used to obtain the reduced system. In this example, a current source is used to be a step input ($u = 0$, when $t \leq 0$, else $u = 1$, and $t \in [0, 10]$). Figure (4.3) shows that the reduced system obtained using the PSO-base method is more accurate than the quadratic system. In other

words, the output of the reduced system obtained using the developed approach is much closer to the output of the original system.

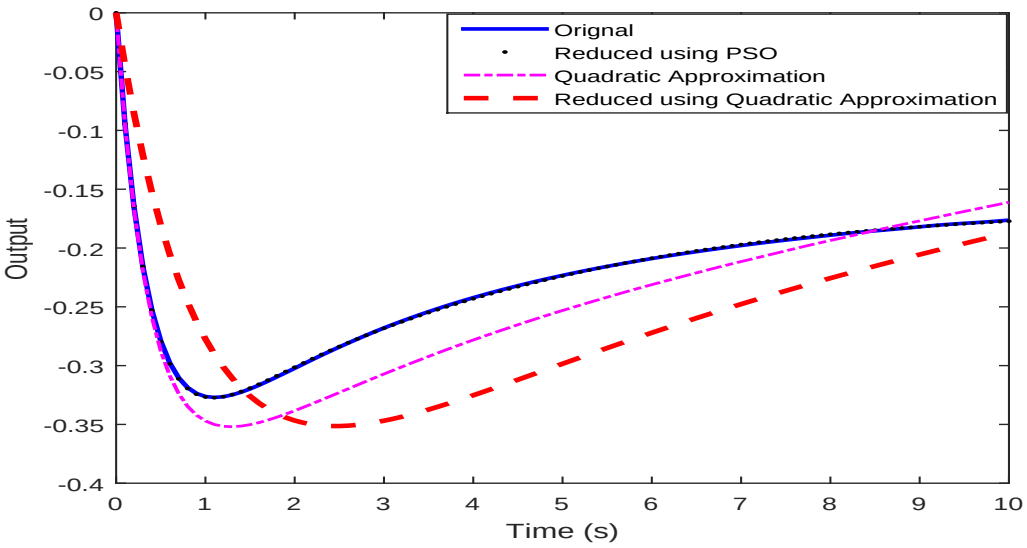

Figure 4.3: Comparison of the system response in example 3 using PSO-based method vs. Quadratic Reduction Method

4.5 Comparison with Direct Nonlinear Reduction with Variational Analysis

The model order reduction that based on the variational analysis theory started by transformed the original nonlinear system into several linear systems, then model reduction is applied to each of these linear systems. Consider there exists the following nonlinear system:

$$\begin{aligned} \dot{x} &= f(x) + Bu \\ y &= Cx \end{aligned} \tag{4.7}$$

If we change the input of the system (4.7) to be an input of the form $\alpha u(t)$, then the system will be:

$$\begin{aligned} \dot{x} &= f(x) + B(\alpha u) \\ y &= Cx \end{aligned} \tag{4.8}$$

where α is an arbitrary scalar. We can write the expansion of the state $x(t)$ in the parameter α is the following form:

$$x(t) = \alpha x_1(t) + \alpha^2 x_2(t) + \alpha^3 x_3(t) + \dots \tag{4.9}$$

The Taylor series of the nonlinear function "$f(x)$" can be written in the Kronecker form as follows:

$$f(x(t)) = f(0) + A_1 x(t) + A_2(x(t) \otimes x(t)) + A_3(x((t) \otimes x(t) \otimes x(t)) + ... \tag{4.10}$$

By substituting equation (4.10) and equation (4.9) in equation (4.8), we get the following:

$$\begin{aligned} \alpha x_1(t) + \alpha^2 x_2(t) + \alpha^3 x_3(t) + \dots = & \alpha A_1 x_1(t) + \alpha^2 [A_1 x_2 + A_2(x_1 \otimes x_1)] \\ & + \dots + B(\alpha u) \end{aligned} \tag{4.11}$$

Since the coefficient of like powers of α are equal, then equation (4.11) is equivalent to the following:

$$\dot{x}_1(t) = A_1 x_1 + Bu(t) \tag{4.12}$$

$$\dot{x}_2(t) = A_1 x_2 + A_2(x_1 \otimes x_1) \tag{4.13}$$

$$\dot{x}_3(t) = A_1 x_3 + A_2(x_1 \otimes x_2 + x_2 \otimes x_1) + A_3(x_1 \otimes x_1 \otimes x_1) \tag{4.14}$$

In this technique, the MOR technique is applied to the linear systems (4.12) (4.13) and (4.14), instead of applying it to the original system. This method started by using the Taylor expansion of the nonlinear function "$f(x)$" to approximate the original nonlinear system (4.7) into the second order

$$\begin{aligned} \dot{x}(t) &= A_1 x + A_2(x \otimes x) + Bu \\ y(t) &= Cx \end{aligned} \tag{4.15}$$

or to the third order system

$$\begin{aligned} \dot{x}(t) &= A_1 x + A_2(x \otimes x) + A_3(x \otimes x \otimes x) + Bu \\ y(t) &= Cx \end{aligned} \tag{4.16}$$

In equation (4.9), if we put $\alpha = 1$, then the solution of equation (4.15) is equivalent to the following:

$$x(t) = x_1(t) + x_2(t) \tag{4.17}$$

also, the solution of equation (4.16) is equivalent to the following:

$$x(t) = x_1(t) + x_2(t) + x_3(t) \tag{4.18}$$

The main advantage of the direct order reduction method is the development of a single projection matrix "V" to reduce the whole system, instead of reducing the individual linear systems (4.12), (4.13), and (4.14). This single matrix will be used to reduce the system (4.15) (if we approximate the nonlinear to the second order) or to reduce the system (4.16) (if we approximate the nonlinear system to the third order) [18].

The construction of the projection matrix V can be summarized as follows [18]:

- Firstly, construct the projection matrix V_1 based on the first linear system (4.12), i.e. $spancolum\{V_1\} = span\{A_1^{-1}b, A_1^{-2}b, \ldots, A_1^{-q_1}b\}$. Then, approximate x_1 as $x_1 \approx V_1 x_{r1}$.

- Secondly, construct the projection matrix V_2 based on the second linear matrix (4.13), i.e. $spancolum\{V_2\} = span\{A_1^{-1}A_2, A_1^{-2}A_2, \ldots, A_1^{-q_1}A_2\}$. Then, we make an approximation of x_2 as $x_2 \approx V_2 x_{r2}$. The same procedures can be followed to approximate x_3. Then we have $x(t) = V_1x_1 + V_2x_2 + V_3x_3$.

- Thirdly, the projection matrix "V" can be written as: $spancolum\{V\} = spancolum\{V_1, V_2, V_3\}$

- Finally, the reduced system using this approach can be written as follows:

$$\dot{x}_r = V^T A_1 V x_r + V^T A_2 (V x_r \otimes V x_r) + V^T A_3 (V x_r \otimes V x_r \otimes V x_r) + V^T b u$$

$$y_o(t) = CV x_r \tag{4.19}$$

Figure (4.4) shows the response of the nonlinear circuit in example 2. In this example, a specific current source is used as input,($u = 0$, when $t \leq 3$, else $u = 1$, and $t \in [0, 10]$). For a clear comparison between the reduced methods, the output of the nonlinear circuit in example 2 [in Section (4.2)] is plotted with size 20, and the dimension of the reduced-order system is 4. The PSO-based method and the direct nonlinear reduction with variational analysis [18] are compared to each other. The reduced system using the direct nonlinear reduction with variational analysis gives a good result. However, the developed approach is found to give more accurate result. The simulation of example 3, that was mentioned in Section (4.3), is shown in Figure (4.5). The input of this circuit is u, where $u = 0$, when $t \leq 0$, else $u = 1$, and $t \in [0, 10]$. It is clear from the figure that the reduced system, of order 4, obtained using the PSO-based method gives a good approximation to the original nonlinear system than the direct nonlinear reduction with variational analysis [18].

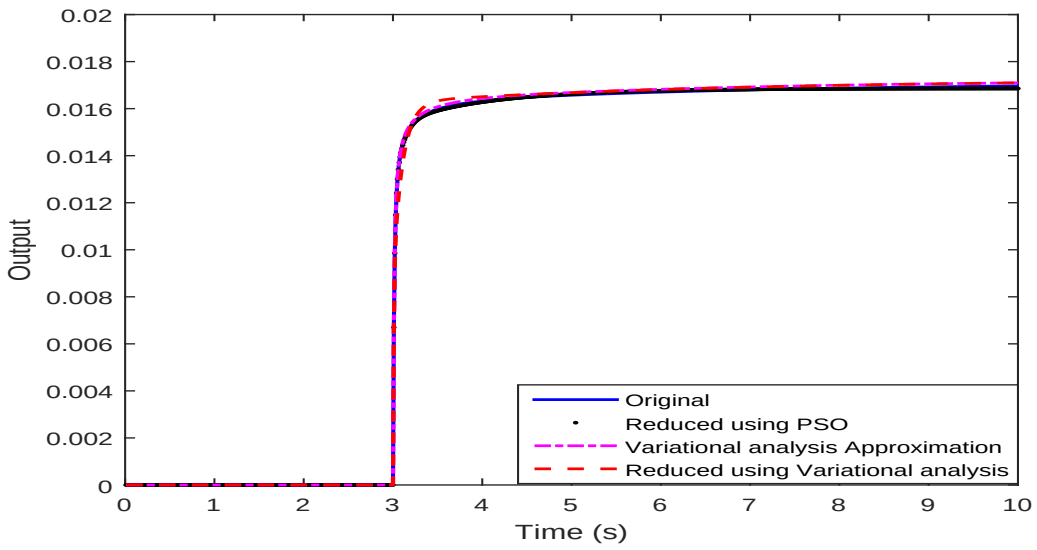

Figure 4.4: Comparison of the system response in example 2 using PSO-based method vs. Direct Nonlinear Reduction with Variational Analysis

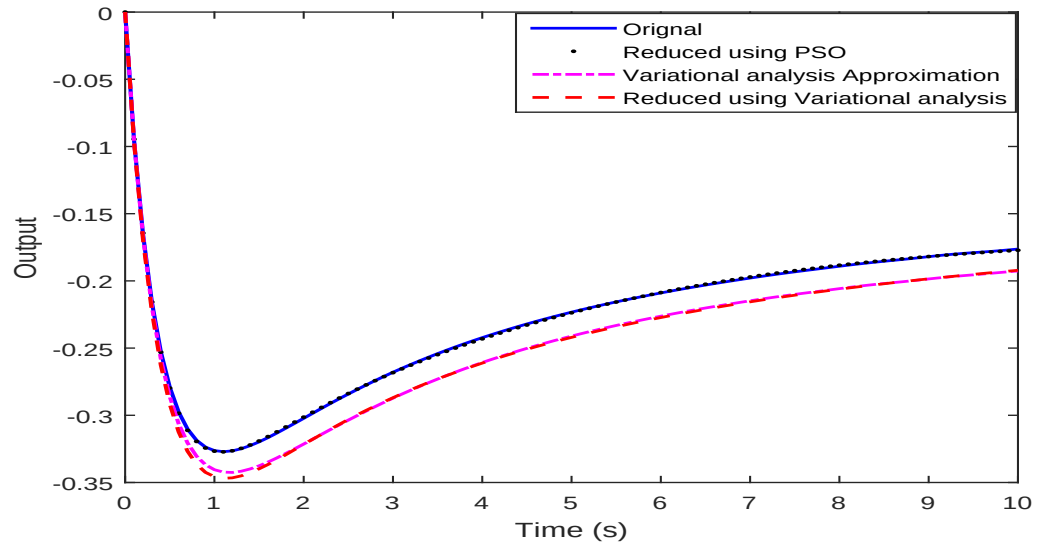

Figure 4.5: Comparison of the system response in example 3 using PSO-based method vs. Direct Nonlinear Reduction with Variational Analysis

4.6 Comparison with Model Reduction for Quadratic Bilinear Systems

As can be illustrated from the previous section, the Taylor expansion of the nonlinear system (4.7) can be written in a Kronecker product of the state x, i.e.

$$f(x) = A_1 x + A_2(x \otimes x) + A_3(x \otimes x \otimes x) + \dots \tag{4.20}$$

Since the bilinearization system is derived by approximating the nonlinear system using the two terms of equation (4.20), the following approximation of the function $f(x)$ is obtained as follows:

$$f(x) = A_1 x + A_2(x \otimes x) \tag{4.21}$$

Then, the bilinearization system can be represented as follows:

$$\begin{aligned} \dot{x}(t) &= A_{\otimes} x_{\otimes} + N_{\otimes} x_{\otimes} u(t) + B_{\otimes} u(t) \\ y(t) &= C_{\otimes} x_{\otimes} \end{aligned} \tag{4.22}$$

Where

$$x_{\otimes} = \begin{bmatrix} x(t) \\ x(t) \otimes x(t) \end{bmatrix}, B_{\otimes} = \begin{bmatrix} B \\ 0 \end{bmatrix}, C_{\otimes} = \begin{bmatrix} C \\ 0 \end{bmatrix}$$

$$A_{\otimes} = \begin{bmatrix} A_1 & A_2 \\ 0 & A_1 \otimes I + I \otimes A_1 \end{bmatrix}, N_{\otimes} = \begin{bmatrix} 0 & 0 \\ B \otimes I + I \otimes B & 0 \end{bmatrix}$$

Mian IIyas Ahmed et al. [76] proposed a new technique for model reduction of quadratic bilinear systems. In this technique, first the nonlinear system (4.7) is approximated using the quadratic bilinear system as follows:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Nx(t)u(t) + Hx(t) \otimes x(t) + Bu(t) \\ y(t) &= Cx(t) \end{aligned} \tag{4.23}$$

where $A, N \in \mathbb{R}^{n \times n}$, $H \in \mathbb{R}^{n \times n^2}$, $B, C \in \mathbb{R}^n$. The next step is to project the approximated system using two matrices $V, W^T \in \mathbb{R}^{n \times r}$. The reduced matrices of this technique are showing below:

$$A_r = W^T AV \ , \ H_r = W^T H(V \otimes V), \ , \ N_r = W^T NV \ , \ B_r = W^T \ , \ C_r = CV \tag{4.24}$$

The matrices V and W^T are constructed as follows [76]:

$$range(V) = span\{(\sigma_1 I - A)^{-1} B, (\sigma_2 I - A)^{-1} NV^1, \ldots, (\sigma_r I - A)^{-1} NV^{r-1}\}.$$

$$range(W) = span\{(\sigma_1 I - A)^{-T} C^T, (\sigma_2 I - A)^{-T} N^T W^1, \ldots, (\sigma_r I - A)^{-T} N^T W^{r-1}\}.$$

In example 2, the PSO-based method is used to reduce the order of the nonlinear transmission line circuit with 20 nodes, see Figure (2.2). In this example, a specific current source is used as input ($u = 0,$ when $t \leq 3$, else $u = 1$, and $t \in [0, 10]$). Figure (4.6) shows a comparison of the reduced systems of order 4 that obtained using the PSO-based method and the quadratic bilinear system method [76], in order to examine the accuracy and the efficiency of the developed approach. It is apparent from the this figure that the reduced system using the PSO-based method provides more accurate results than the quadratic bilinear system method [76]. The circuit in example 3, that was mentioned in Section (4.3), is used to

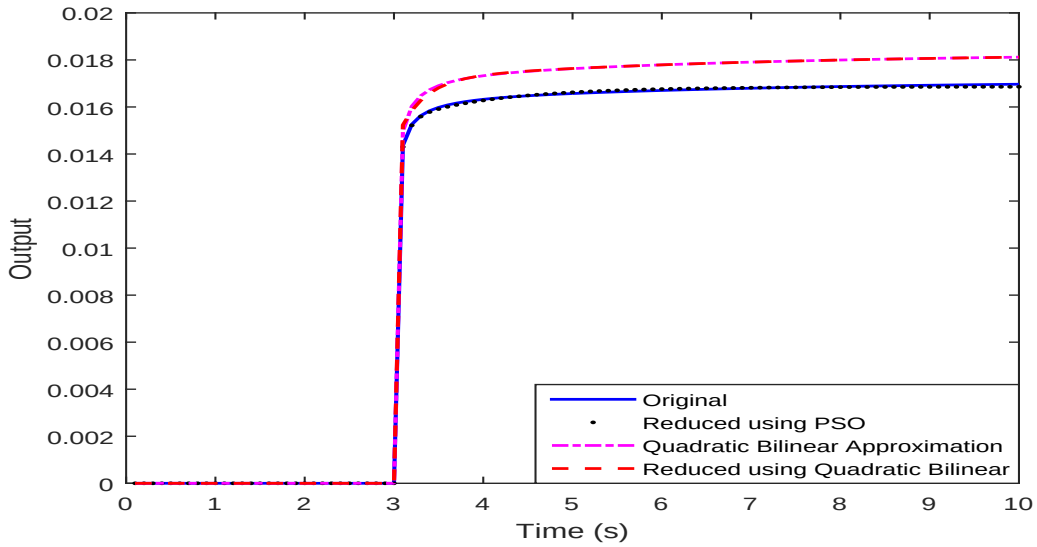

Figure 4.6: Comparison of the system response in example 2 using PSO-based method vs. Model Reduction for Quadratic Bilinear Systems

test the accuracy of the developed approach. Here, a specific current source is used as input ($u = 0,$ when $t < 0$, else $u = 1$, and $t \in [0, 10]$). Figure (4.7) shows that the quadratic bilinear systems method [76] failed to save the stability of the reduced system. However, the reduced system of order 4 that obtained using the

PSO-based approach gives a good approximation to the original nonlinear system.

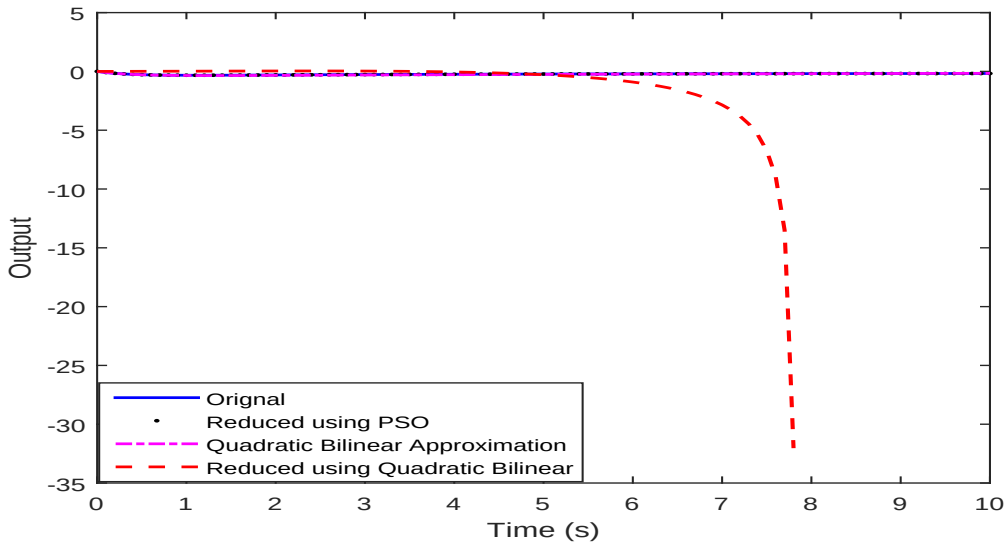

Figure 4.7: Comparison of the system response in example 3 using PSO-based method vs. Model Reduction for Quadratic Bilinear Systems

4.7 Conclusion

In this chapter, two nonlinear electrical circuits have been used to test the PSO-based method. This approach has been validated by comparing the simulation of the output for these examples using its output with the original nonlinear model. In addition, a brief summary was given regarding three existing approaches and the PSO-based method was compared to these methods in order to investigate the accuracy and the efficiency of the developed approach. It could be observed that the PSO-based method gave more accurate results than these existing approaches.

REFERENCES

[1] A. C. Antoulas. *Advances in design and control: Approximation of largescale dynamical systems*, volume 6. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2005.

[2] A. C. Antoulas, D. C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary Mathematics*, 280:193–219, 2001.

[3] B. Besselink, N. van. de. Wouw, and H. Nijmeijer. Model reduction of nonlinear systems with bounded incremental $\mathcal{L}_2$ gain. In *50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, pages 7170–7175. IEEE, 2011.

[4] S. Ibrir and M. Bettayeb. Model reduction of a class of discrete-time nonlinear systems. *Applied Mathematics and Computation*, 250:78–93, 2015.

[5] S. Ibrir and M. Bettayeb. Model reduction procedure for stable nonlinear systems. In *GCC Conference and Exhibition (GCCCE), 2015 IEEE 8th*, pages 1–6, 2015.

[6] S. Al-Baiyat, M. Bettayeb, and U. Al-Saggaf. New model reduction scheme for bilinear systems. *International Journal of Systems Science*, 25(10):1631–1642, 1994.

[7] K. Fujimoto and D. Tsubakino. Computation of nonlinear balanced realization and model reduction based on Taylor series expansion. *Systems and Control Letters*, 57(4):283–289, 2008.

[8] J. M. A. Scherpen. Balancing for nonlinear systems. *Systems and Control Letters*, 21(2):143 – 153, 1993.

[9] J. Scherpen. H_∞ balancing for nonlinear systems. *International Journal of Robust and Nonlinear Control*, 6(2):645–668, 1996.

[10] Q. Wang. Macro-modeling of micro-electrical-mechanical system devices. Master's thesis, MIT , 1998.

[11] Y. Chen. Model order reduction for nonlinear systems. Master's thesis, MIT, 1999.

[12] J. R. Phillips. Projection frameworks for model reduction of weakly nonlinear systems. In *Design Automation Conference*, pages 184–189, 2000.

[13] Z. Bai and D. Skoogh. A projection method for model reduction of bilinear dynamical systems. *Linear Algebra and its Applications*, 415(2-3):406 – 425, 2006.

[14] L. Feng and P. Benner. A note on projection techniques for model order reduction of bilinear systems. In *Numerical Analysis and Applied Mathematics (AIP Conference Proceedings)*, volume 936, pages 208–211, 2007.

[15] U. Baur, P. Benner, and L. Feng. Model order reduction for linear and nonlinear systems: a system-theoretic perspective. *Archives of Computational Methods in Engineering*, 21(4):331–358, 2014.

[16] J. R. Phillips. Automated extraction of fionlinear circuit macromodels. *IEEE Custom Integrated Circuits Conference*, pages 451–454, 2000.

[17] W. J. Rugh. *Nonlinear System Theory.* Johns Hopkins University Press, 1981.

[18] L. Feng, X. Zeng, Ch. Chiang, D. Zhou, and Q. Fang. Direct nonlinear order reduction with variational analysis. In *Proceedings of the Conference on Design, Automation and Test in Europe*, 2004.

[19] M. Rewienski and J. White. A trajectory piecewise-linear approach to model order reduction and fast simulation of nonlinear circuits and micromachined devices. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(2):155–170, Feb 2003.

[20] M. Rewienski and J. White. Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations. *Linear Algebra and its Applications*, 415(2-3):426 – 454, 2006.

[21] P. K. Gunupudi and M. S. Nakhla. Nonlinear circuit-reduction of high-speed interconnect networks using congruent transformation techniques. *IEEE Transactions on Advanced Packaging*, 24(3):317–325, Aug 2001.

[22] L. Sirovich. Turbulence and the dynamics of coherent structures. part i: Coherent structures. *Quarterly of Applied Mathematics*, 45(3):561–571, Aug 1987.

[23] S. Glavaski, J. E. Marsden, and R. M. Murray. Model reduction, centering, and the karhunen-lo'eve expansion. In *the IEEE Conference on Decision and Control*, pages 2071–2076, 1998.

[24] M. Meyer and H. G. Matthies. Efficient model reduction in non-linear dynamics using the karhunen-loeve expansion and dual-weighted-residual methods. *Computational Mechanics*, 31(1-2):179–191, Aug 2003.

[25] S. Ibrir and M. Bettayeb. Model reduction of a class of nonlinear systems: a convex-optimization approach. *IMA Journal of Mathematical Control and Information*, 2013.

[26] P. Benner, M. Hinze, and E. J. W. ter. Maten, editors. *Model reduction for circuit simulation.* Lecture Notes in Electrical Engineering 74. Springer, 2011.

[27] S. X.-D. Tan and L. He. *Advanced model order reduction techniques in VLSI design.* Cambridge University Press, 2007.

[28] A. Varga. Balancing free square-root algorithm for computing singular perturbation approximations. In *Proceedings of the 30th IEEE Conference on Decision and Control*, volume 2, pages 1062–1065, 1991.

[29] S. Gugercin and A. C. Antoulas. A survey of model reduction by balanced truncation and some new results. *International Journal of Control*, 77(8):748–766, 2004.

[30] M. Heinkenschloss, T. Reis, and A. C. Antoulas. Balanced truncation model reduction for systems with inhomogeneous initial conditions. *Automatica*, 47(3):559–564, 2011.

[31] P. Benner and J. Saak. Efficient balancing-based MOR for large-scale second-order systems. *Mathematical and Computer Modelling of Dynamical Systems*, 17(2):123–143, 2011.

[32] K. Gallivan, E. Grimme, and P. V. Dooren. Asymptotic waveform evaluation via a lanczos method. *Applied Mathematics Letters*, 7(5):75–80, 1994.

[33] P. Feldmannand and R. W. Freund. Efficient linear circuit analysis by pade approximation via the lanczos process. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(5):639–649, May 1995.

[34] I. M. Jaimoukha and E. M. Kasenally. Implicitly restarted krylov subspace methods for stable partial realizations. *SIAM Journal on Matrix Analysis and Applications*, 18(3):633–652, 1997.

[35] E. J. Grimme. *Krylov Projection Methods for Model Reduction.* PhD thesis, ECE Dept., U. of Illinois, Urbana-Champaign, 1997.

[36] B. Lohmann and B. Salimbahrami. Introduction to krylov subspace methods in model order reduction. *Methods and Applications in Automation*, pages 1–13, 2000.

[37] P. Heres and W. Schilders. Practical issues of model order reduction with krylov-subspace methods. In *Progress in Electromagnetic Research Symposium: PIERS 2004*, 2005.

[38] S. Gugercin. An iterative SVD-Krylov based method for model reduction of large-scale dynamical systems. In *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, pages 5905–5910, Seville, Spain, December 2005.

[39] R. Freund. Recent advances in structure-preserving model order reduction. In *Simulation and Verification of Electronic and Biological Systems*, pages 43–70. Springer Netherlands, 2011.

[40] M. Bettayeb, L. M. Silverman, and M. G. Safonov. Optimal approximation of continuous time systems. In *Proceedings of the 19th IEEE Conference on Decision and Control*, pages 195–198, Albuquerque, NM, December 1980.

[41] L. M. Silverman and M. Bettayeb. Optimal approximation of linear systems. In *Proceedings of the Joint Automatic Control Conference*, volume 2, pages FA8–A, San Francisco, California, August 1980.

[42] D. Kavranoglu and M. Bettayeb. Characterization of the solution to the optimal H_∞ model reduction problem. *Systems and Control Letters*, 20(2):99–107, 1993.

[43] K. M. Grigoriadis. Optimal H_∞ model reduction via linear matrix inequalities : Continuous and discrete-time cases. *Systems and Control Letters*, 26(5):321 – 333, 1995.

[44] J. C. Geromel, F. R. Kawaoka, and R. G. Egas. Model reduction of discrete time systems through linear matrix inequalities. *International Journal of Control*, 77(10):978–984, 2004.

[45] K. Glover. All optimal hankel-norm approximations of linear multivariable systems and their L^∞ error bounds. *International Journal of Control*, 39(6):1115–1193, 1984.

[46] M. G. Safonov, R. Y. Chiang, and D. N. Limebeer. Optimal Hankel model reduction for nonminimal systems. *IEEE Transactions on Automatic Control*, 35(4):496–502, 1990.

[47] C. B. Vishwakarma and R. Prasad. Time domain model order reduction using Hankel matrix approach. *Journal of the Franklin Institute*, 351(6):3445 – 3456, 2014.

[48] W-Y. Yan and J. Lam. An approximate approach to H_2 optimal model reduction. *IEEE Transactions on Automatic Control*, 47(7):1341–1358, 1999.

[49] L. Zhang, B. Huang, and J. Lam. H_∞ model reduction of Markovian jump linear systems. *Systems and Control Letters*, 50(2):103–118, 2003.

[50] Y. Ebihara and T. Hagiwara. On model reduction using LMI's. In *Proceedings of the American Control Conference*, volume 1, pages 774–779, June 2004.

[51] M. Kanno. H_2 model reduction using lmis. In *44th IEEE Conference on Decision and Control and 2005 European Control Conference. CDC-ECC '05.*, pages 5887–5892, Dec 2005.

[52] L. Zhang, P. Shi, E. Boukas, and C. Wang. H_∞model reduction for uncertain switched linear discrete-time systems. *Automatica*, 44(11):2944 – 2949, 2008.

[53] B. C. Moore. Principal component analysis in linear systems: Controllability, observability and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981.

[54] U. M. Al-Saggaf and G. F. Franklin. An error bound for a discrete reduced order model of a linear multivariable system. *IEEE Transactions on Automatic Control*, 32(9):815–819, 1987.

[55] K. C. Sou and A. Rantzer. Controller reduction via minimum rank matrix approximation. *Automatica*, 48(6):1069–1076, 2012.

[56] B. Haasdonk and M. Ohlberger. Efficient reduced models and a posteriori error estimation for parametrized dynamical systems by offline/online decomposition. *Mathematical and Computer Modelling of Dynamical Systems*, 17(2):145–161, 2011.

[57] B. Luitel and G. K. Venayagamoorthy. Particle swarm optimization with quantum infusion for system identification. *Engineering Applications of Artificial Intelligence*, 23(5):635–649, 2010.

[58] M. Clerc. *Particle swarm optimization*. ISTE Ltd, 2006.

[59] R. Salim and M. Bettayeb. H_2 optimal model reduction using genetic algorithms and particle swarm optimization. In *6th International Symposium on Mechatronics and its Applications ISMA'09.*, pages 1–6. IEEE, 2009.

[60] R. Salim and M. Bettayeb. H_2 and H_∞ optimal model reduction using genetic algorithms. *Journal of the Franklin Institute*, 348(7):1177–1191, 2011.

[61] E. J. Davison. A method for simplifying linear dynamic systems. *IEEE Transactions on Automatic Control*, 11(1):93–101, Jan 1966.

[62] A. J. Laub, M. T. Heath, C. Paige, and R. C. Ward. Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms. *IEEE Transactions on Automatic Control*, 32(2):115–122, 1987.

[63] J. Hahn and T. F. Edgar. An improved method for nonlinear model reduction using balancing of empirical gramians. *Computers and Chemical Engineering*, 26(10):1379–1397, 2002.

[64] G. A. Murad, I. Postlethwaite, and D. W. Gu. Model reduction for non-minimal state-space systems. In *Proceedings of the 36th IEEE Conference on Decision and Control*, volume 4, pages 3304–3310, 1997.

[65] M. Abu. Hatab. Model order reduction using LMI. Master's thesis, The Islamic University of Gaza , 2009.

[66] W. H.A. Schilders, H. A. Vorst, and J. Rommes. *Model order reduction: theory, research aspects and applications*, volume 13. Springer, 2008.

[67] T. Penzl. Algorithms for model reduction of large dynamical systems. *Linear Algebra and its Applications*, 415(2-3):322–343, 2006.

[68] Biswa Nath Datta. Chapter 14 - internal balancing and model reduction. In Biswa Nath Datta, editor, *Numerical Methods for Linear Control Systems*, pages 601 – 646. Academic Press, San Diego, 2004.

[69] W. H.A. Schilders, H. A. Vorst, and J. Rommes. *Model order reduction: theory, research aspects and applications*. Springer.

[70] P. Benner, E. S. Quintana-Orti, and G. Quintana-Orti. Computing optimal hankel norm approximations of large-scale systems. In *43rd IEEE Conference on Decision and Control. CDC*, volume 3, 2004.

[71] P. Van. Dooren, K. A. Gallivan, and P. A. Absil. H_2-optimal model reduction of {MIMO} systems. *Applied Mathematics Letters*, 21(12):1267 – 1273, 2008.

[72] D. A. Wilson. Optimum solution of model-reduction problem. *Proceedings of the Institution of Electrical Engineers*, 117(6):1161–1165, June 1970.

[73] H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, Englewood Cliffs, NJ, 2nd edition, 1996.

[74] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings IEEE International Conference on Neural Networks*, volume 4, 1995.

[75] E. Ozcan and C. K. Mohan. Analysis of a simple particle swarm optimization system. *Intelligent engineering systems through artificial neural networks*, 8(4):253 – 258, 1998.

[76] M. I. Ahmed, P. Benner, and L. Fenf. A new interpolatory model reduction approach for quadratic bilinear descriptor systems. *PAMM*.

Vitae

Name: Saeed Salim Ahmed Ba Rayyan

Nationality: Yemeni

Date of Birth: 18/06/1987

Email: *ssabdoan@hotmail.com*

Permenant Address: Yemen-Hadhramout-Mukalla

Academic Background:

M.S in Electrical Engineering, December 2016

King Fahd University of Petroleum and Minerals

Dhahran, Saudi Arabia.

B.S in PElectronic and Communication Engineering, January 2012

Hadhramout University for Science and Technology,

Hadhramout, Republic of Yemen.

Publications: S. Ibrir and Saeed Ba Rayyan, Model-Order Reduction of a Class of Nonlinear Systems Based on Particle Swarm Optimization, in progress.