MODEL REDUCTION OF SYSTEMS EXHIBITING TWO-TIME SCALE BEHAVIOR OR PARAMETRIC UNCERTAINTY

A Dissertation

by

CHUILI SUN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

December 2006

Major Subject: Chemical Engineering

MODEL REDUCTION OF SYSTEMS EXHIBITING TWO-TIME SCALE BEHAVIOR OR PARAMETRIC UNCERTAINTY

A Dissertation

by

CHUILI SUN

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Juergen Hahn
Committee Members,	Mahmoud El-Halwagi
	Yu Ding
	Daniel F. Shantz
Head of Department,	N.K. Anand

December 2006

Major Subject: Chemical Engineering

ABSTRACT

Model Reduction of Systems Exhibiting Two-Time
Scale Behavior or Parametric Uncertainty. (December 2006)
Chuili Sun, B.S., Tsinghua University;
M.S., Tsinghua University
Chair of Advisory Committee: Dr. Juergen Hahn

Model reduction is motivated by the fact that complex process models may prevent the application of model-based process control. While extensive research on model reduction has been done in the past few decades, model reduction of systems exhibiting two-time scale behavior as well as parametric uncertainty has received little attention to date. This work addresses these types of problems in detail.

Systems with two-time scale behavior can be described by differential-algebraic equations (DAEs). A new technique based on projections and system identification is presented for reducing this type of system. This method reduces the order of the differential equations as well as the number and complexity of the algebraic equations. Additionally, the algebraic equations of the resulting system can be replaced by an explicit expression for the algebraic variables such as a feed-forward neural network or partial least squares. This last property is important insofar as the reduced model does not require a DAE solver for its solution, but system trajectories can instead be computed with regular ordinary differential equation (ODE) solvers.

For systems with uncertain parameters, two types of problems are investigated, including parameter reduction and parameter dependent model reduction. The parameter reduction problem is motivated by the fact that a large number of parameters exist in process models while some of them contribute little to a system's input-output

behavior. This portion of the work presents three novel methodologies which include (1) parameter reduction where the contribution is measured by Hankel singular values, (2) reduction of the parameter space via singular value decomposition, and (3) a combination of these two techniques.

Parameter dependent model reduction investigates how to incorporate the influence of parameters in the procedure of conventional model reductions. An approach augmenting the input vector to include the parameters are developed to solve this problem.

Finally, a nonlinear model predictive control scheme is developed in which the reduced models are used for the controller.

Examples are investigated to illustrate these techniques. The results show that excellent performance can be obtained for the reduced models.

To my parents, my husband and my daughter

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Dr. Juergen Hahn for his intelligent guidance and constant encouragement in my research program and the preparation of this dissertation. It is a great fortune for me to have the opportunity to work with Dr. Hahn.

I am deeply grateful to Dr. Mahmoud El-Halwagi. He has given me a lot of help as our graduate advisor; in addition, his suggestions on this work were so valuable. His kindness is most appreciated.

I would also like to thank the other members of my committee, Dr. Yu Ding and Dr. Daniel F. Shantz, for their time and assistance in helping me complete this work.

I very much appreciate all the help from my friends and fellow graduate students. I want to thank Mr. Abhay Singh, for his help and discussions on this work. I also want to thank all the members in our process system engineering group for their valuable questions during our group seminar. I wish all the best of luck in completing their graduate work and excelling in their professional careers.

NOMENCLATURE

Roman Letters

A, B, C, D	matrices for linear models in state space domain
$\bar{A}, \bar{B}, \bar{C}, \bar{D}$	matrices for linear models after a transformation
f, g, h	functions for nonlinear models in state space domain
$\bar{f}, \bar{g}, \bar{h}$	functions for nonlinear models after a transformation
G	matrix of transfer functions
n	number of states
P	projection matrix
T	transformation matrix
t	time
u	inputs
W_C	controllability gramian or covariance matrix
W_O	observability gramian or covariance matrix
$\bar{W}_C$	controllability gramian or covariance matrix after a transformation
$\bar{W}_O$	observability gramian or covariance matrix after a transformation
x	states
$\bar{x}$	states after a transformation
y	outputs
z	algebraic variables

Greek Letters and Symbols

δ	impulse function
λ	eigenvalue
$\Re$	set of real numbers
Σ	balanced gramian matrix
σ	singular value
θ	parameters
$\bar{\sigma}$	maximum singular value
$\|\| \ldots \|\|$	Euclidean norm
$\|\| \ldots \|\|_H$	Hankel norm
$\|\| \ldots \|\|_\infty$	infinity norm

Superscripts and Subscripts

0	initial state
i	general index
ss	steady state
T	transpose
-1	inverse
$-T$	tranpose of the inverse

Abreviations

ANN	artificial neural network
lim	limit
min	minimum
$CSTR$	continously stirred tank reactor
DAE	differential algebraic equation
$MIMO$	multi input, multi output
$MINLP$	mixed integer nonlinear programming
MPC	model predictive control
$NMPC$	nonlinear model predictive control
ODE	ordinary differential equation
PCA	principal component analysis
PDE	partial differential equation
POD	proper orthogonal decomposition
PLS	partial least squares
SCM	sensitivity covariance matrix
$SISO$	single input, single output
SVD	singular value decomposition

TABLE OF CONTENTS

CHAPTER Page

I INTRODUCTION 1

A. Motivation of Model Reduction 1
B. Current Status of Model Reduction 1
C. Objective of This Research 2
D. Dissertation Outline 3

II REVIEW OF MODEL REDUCTION AND RELATED ISSUES 6

A. Review of Model Reduction 6
1. Overview of Model Reduction 6
2. Overview of Balancing 8
a. Balancing for Linear Systems 8
b. Balancing for Nonlinear Systems 9
3. Singular Perturbation Method 12
4. Singular Value Decomposition 13
5. Proper Orthogonal Decomposition 14
B. System Identification 15
1. Artificial Neural Network 16
2. Partial Least Squares 16
a. Data Preprocessing 17
b. Calibrating a PLS Model 17
c. Data Prediction 18
C. Parameter Selection 18
1. Sensitivity Analysis 19

III DAE MODEL REDUCTION VIA PROJECTIONS AND NEURAL NETWORK 20

A. Overview 20
B. Model Reduction of DAE Systems 22
1. Computation of Transformations 23
a. Computation of Transformations via Balancing/SVD 23
b. Computation of Transformations via POD 25
c. Transformed System 26

CHAPTER Page

2. Order Reduction of Differential Equations 27
3. Reduction of Algebraic Equations 30
4. Further Reduction of Algebraic Equations via System Identification 31
5. Summary of the Model Reduction Procedure 33
C. Case Study: Distillation Column 34
1. Model Description 34
2. Data Sets Collection 34
3. Order Reduction of Both Differential and Algebraic Equations 37
4. Further Reduction of Algebraic Equations via Neural Network 40
D. Summary and Discussion 41

IV DAE MODEL REDUCTION VIA BALANCING AND PARTIAL LEAST SQUARES (PLS) 43

A. Model Reduction via Balancing/PLS for ODE Systems . . 43
1. Model Reduction Procedure 43
2. Implementation of PLS 46
B. Model Reduction via Balancing/PLS for DAE Systems . . 48
1. Model Reduction Procedure 48
2. Implementation of PLS 48
C. Case Studies 50
1. Example 1: Catalytic Fixed-Bed Reactor 50
a. Data Set Collection 50
b. PLS Implementation 51
c. Performance of Reduced System 52
2. Example 2: Distillation Column 53
a. Data Set Collection 54
b. PLS Implementation 54
c. Performance of Reduced System 55
D. Summary and Discussion 56

V PARAMETER REDUCTION FOR STABLE DYNAMICAL SYSTEMS BASED ON HANKEL SINGULAR VALUES AND SENSITIVITY ANALYSIS 57

A. Overview 57
B. Parameter Reduction for Linear Systems 59

CHAPTER Page

1. Method I: Parameter Reduction Based on Hankel Singular Values . . . 59
2. Method II: Parameter Reduction Based on Sensitivity Analysis . . . 63
a. Sensitivity Covariance Matrix . . . 63
b. Parameter Reduction Procedure . . . 65
c. Analysis of Error Bounds for Parameter Space Reduction . . . 67
3. Method III: Combination of Method I and II . . . 69
C. Extension to Nonlinear Systems . . . 69
1. Method I: Parameter Reduction Based on Hankel Singular Values . . . 70
2. Method II: Parameter Reduction Based on Sensitivity Analysis . . . 70
3. Method III: Combination of Method I and II . . . 72
D. Performance Evaluation . . . 72
1. Frequency Domain Analysis . . . 72
2. Time Domain Analysis Based on Monte Carlo Simulations . . . 73
E. Case Studies . . . 75
1. Example 1: A Randomly Generated Linear System . . . 76
2. Example 2: Two CSTRs in Series . . . 80
3. Example 3: Model of a Signal Transduction Pathway in Hepatocytes . . . 84
F. Summary and Discussion . . . 87

VI MODEL REDUCTION IN THE PRESENCE OF PARAMETRIC UNCERTAINTIES . . . 90

A. Overview . . . 90
B. Problem Statement . . . 91
C. Extension to Existing Approaches . . . 93
D. Discussion . . . 94
E. Case Studies . . . 96
1. Example 1: Two Nonlinear Reactors in Series . . . 96
2. Example 2: Catalytic Fixed-Bed Reactor . . . 99
F. Summary and Discussion . . . 100

VII APPLICATION OF NONLINEAR MODEL PREDICTIVE CONTROL BASED ON REDUCED ORDER MODELS . . . 103

CHAPTER Page

A. Overview of NMPC 104
1. The Optimization Problem 104
2. Process Constraints 105
3. Disturbance Estimation 105
4. The Algorithm Used in This Work 106
5. NMPC Scheme Based on Reduced Order Models . . . 107
B. Case Studies 107
1. Example 1: Catalytic Fixed-Bed Reactor 107
2. Example 2: Distillation Column System 110
C. Summary and Discussion 112

VIII CONCLUSIONS AND RECOMMENDATIONS 113

A. Contributions 115
B. Future Work 116

REFERENCES 117

APPENDIX A 129

APPENDIX B 133

APPENDIX C 134

VITA 152

LIST OF TABLES

TABLE		Page
I	Hankel singular values	37
II	Comparison of CPU times for reduced-order models	40
III	Comparison of CPU times for reduced-order models	55
IV	Comparison of CPU times for closed-loop control of fix-bed reactor model	109
V	Comparison of CPU times for closed-loop control of distillation column model	112
VI	Operating conditions and parameter values	132

LIST OF FIGURES

FIGURE		Page
1	A three layer feedforward neural network.	16
2	Relationship between remaining differential states and algebraic variables.	32
3	Flowchart of the model reduction procedure for DAE systems.	35
4	Concentration and temperature distributions in the distillation column.	36
5	Hankel singular values of the distillation column model.	38
6	Comparison between POD and balancing with different number of remaining differential equations.	39
7	Performance of presented method for DAE model reduction.	41
8	Steady state of reactor.	51
9	Prediction performance of PLS model for reactor system.	52
10	Comparison of reduced reactor systems' step responses.	53
11	Prediction performance of PLS model for distillation system.	54
12	Comparison of reduced distillation systems' step responses.	56
13	System decomposition diagram.	61
14	Monte Carlo simulation method	75
15	Hankel singular values for parameters (Method I), for example 1.	77
16	Relative errors between the original system and reduced systems for all three methods, for example 1. (a) Method I; (b) Method II; (c) Method III.	78

FIGURE Page

17 Eigenvalues of sensitivity covariance matrix (Method II), for example 1. 79

18 Hankel singular values for parameters (Method I), for example 2. 81

19 Eigenvalues of sensitivity covariance matrix (Method II), for example 2. 82

20 Comparison of statistics results for three methods, for example 2. (a) 4 parameters retained; (b) 5 parameters retained; (c) 6 parameters retained; (d) 7 parameters retained. Legend: solid line, Method I; dashed line, Method II; dotted line, Method III 83

21 The relative error set in one Monte Carlo simulation, for example 2. 84

22 Hankel singular values for parameters (Method I) and eigenvalues of sensitivity covariance matrix (Method II), for example 3. (a) Comparison of Hankel singular values in Method I; (b) Eigenvalues of sensitivity covariance matrix in Method II. 85

23 Comparison of statistics results for three methods, for example 3. 86

24 Comparison of output responses for original systems and reduced systems, for example 3. 87

25 Comparison of output responses for original systems and reduced systems, for example 3. 88

26 Input and parameter perturbations (scaled value). 98

27 Comparison of performance for conventional techniques and extended version. 99

28 Bound of the error for reducing the model from 120 states to 7 states for a range of frequencies. 101

FIGURE Page

29 Comparison of performance for conventional techniques and the extended version. 102

30 Block diagram of NMPC based on reduced order models 107

31 Temperature vs. time for set point change and disturbance rejection 108

32 Comparison of computed controller actions 109

33 Distillate concentration vs. time for set point change and disturbance rejection 110

34 Comparison of computed controller actions 111

35 Step response of example 1 142

36 Demo_1 result 148

37 Demo_2 result 149

38 Demo_3 result 151

CHAPTER I

INTRODUCTION

A. Motivation of Model Reduction

Model-based process control has become increasingly popular in the chemical process industries over the past few years [1]. Several factors have motivated this development, one of which is that highly accurate models can now be solved with modern dynamic simulators and powerful optimization algorithms. However, with the increasing accuracy of the models, the complexity of these models increases and computational requirements grow. Many dynamic models derived from first principles require extensive computation effort which makes their use questionable for real time model-based controllers. This presents a need for model reduction techniques. The objective of model reduction for the purpose of controller design is to reduce a high-order model to a lower order system which retains most of the input-output behavior.

B. Current Status of Model Reduction

Research on model reduction has been conducted for several decades. Initially, model reduction focused on linear models consisting of ordinary differential equations [2]–[7]. And the model reduction theory for linear systems is well-established. A survey of several different approaches for linear model reduction can be found in [8]–[10].

Besides the development of linear model reduction techniques, more recent work on model reduction has dealt with nonlinear models described by ordinary differential

---

The journal model is *IEEE Transactions on Automatic Control.*

equations [11]–[18] or partial differential equations [19]–[27]. Unlike for linear systems, no complete theory of model reduction for nonlinear systems exists and most of the steps for model reduction are still a subject of research for nonlinear models.

There exist many different model reduction techniques, for instance, balancing [7], [9], lumping [28]–[30] and singular perturbation [31], [32]. Some techniques, such as model reduction by truncation, singular perturbation, Hankel norm approximation, balancing, are reviewed in [33].

C. Objective of This Research

For systems with two-time scales, both differential equations and algebraic equations are necessary to describe the system behavior, resulting in differential-algebraic equation (DAE) systems. Usually, the number of algebraic equations in such models far exceeds the one for differential equations. This clearly presents a need for model reduction techniques that take the algebraic equations into account in addition to the differential equations.

For systems with parameter uncertainties, it is important to consider the parameter uncertainties during the model reduction procedures, otherwise, the uncertainties in the parameters can have a detrimental effect on the model. At the same time, most process systems include a large number of process parameters, however, some of them only provide a marginal contribution to the input-output behavior of the system. Therefore, reduction of these parameters will not affect the system much. It needs to be investigated how to select the relatively unimportant parameters and how to reduce them. This part of the work can be devided into two parts: the work for model reduction with parameter uncertainties and the work of reducing the parameter space.

Considering these objectives, model reduction techniques for DAE systems and systems with parameter uncertainties are investigated in this work. Creating a framework for nonlinear DAE model reduction is an important and challenging problem. The presented technique integrates the essential components for ODE model reduction techniques while it is extended to be applicable to DAE systems. This research lies at the interface of model reduction, differential-algebraic equation systems, and system identification. Several issues will be addressed in this part of research: (1) reducing the size of the model; (2) be applicable to linear as well as nonlinear systems; (3) retain important input-output information; (4) simplify to already existing methods under special circumstances, e.g. for linear systems consisting of only ordinary differential equations.

The work for parameter reduction follows similar concerns as the one for state reduction. Those parameters with more importance to the system will be retained while those with less importance can be neglected. The goal of the reduction procedure is to determine which parameters can be replaced by constant values, even if these are not accurately known, as compared to other parameters which have to be estimated from data. In order to determine the relative importance among all parameters, parameter sensitivity analysis is applied.

For the work of model reduction with parameter uncertainties, it is important to develop a way to incorporate the parameter uncertainties into the model reduction procedure.

D. Dissertation Outline

Chapter II will review model reduction techniques as well as some related issues, including system identification and parameter selection. The emphasis is on proce-

dures that use some type of balancing for the reduction, since this is the main focus of this dissertation.

Chapter III introduces a new technique for reducing differential-algebraic equation (DAE) systems. This method reduces the order of the differential equations as well as the number and complexity of the algebraic equations. Additionally, the algebraic equations of the resulting system can be replaced by an explicit expression for the algebraic variables such as a feedforward neural network. This technique is illustrated with a case study where responses of several different reduced-order models of a distillation column with 32 differential equations and 32 algebraic equations are compared.

The usage of Partial Least Squares (PLS) in model reduction is investigated in Chapter IV. PLS can be implemented on both DAE systems and ODE systems where a residualization method is used to derive a DAE system. Detail procedures for the PLS implementations are presented.

Chapter V introduces and evaluates three techniques to assess the importance of each parameter to the overall input/output description of the process under study, including a method based on Hankel singular values, a method via parameter sensitivity analysis and a method which combines the former two methods. Detail descriptions of these methods are presented for linear systems. And extensions to nonlinear systems are also addressed. Comparisons of the performance for all three methods are presented via three examples.

Chapter VI presents a technique to incorporate uncertain parameters during model reduction. This technique is capable of dealing with parametric uncertainty in the procedure of model reduction by augmenting the vector of inputs to the system with the uncertain parameters and by performing model reduction on the augmented system. A comparison between the presented technique and a conventional approach

is made via two examples.

Chapter VII evaluates the performance of reduced order models by implementing nonlinear model predictive control scheme in which the control move is computed based on the reduced order models.

Chapter VIII summarizes the presented results and identifies future extensions of this research.

CHAPTER II

REVIEW OF MODEL REDUCTION AND RELATED ISSUES

Because this research involves several issues related to model reduction, this chapter will give a comprehensive review for these topics, including a brief review of model reduction techniques, a more detailed review of one specific reduction technique, i.e., balancing, a review of some related issues for parameter estimation and selection, as well as a review on system identification.

A. Review of Model Reduction

1. Overview of Model Reduction

Model reduction is a useful tool to reduce complex systems for purposes such as controller design, online optimization as well as for getting a better understanding of a system's behavior [34]. Generally, the model reduction problem can be formulated as follows: given a full order model S, find a lower order model S_r, which is close to S in some sense [35]. For linear systems, some norms are usually applied to define this approximation, for instance, the H_∞ norm: $||S - S_r||_\infty$, or the hankel norm: $||S - S_r||_H$. For nonlinear systems, it is not easy to evaluate these norms. One way to evaluate the approximation is to compare the systems' responses by simulations, which is also applied in this research work.

For the past three decades, various model reduction techniques have been proposed. Some of them are briefly listed as follows:

- balancing [36], which exploits the balancing idea, to obtain a balanced realization. In this balanced realization, the controllability and observability gramians are identical and diagonal. Next a truncation or a residualization method is ap-

plied to reduce the system.

- singular perturbation approximation [37], which is applicable to systems which exhibit multi-time scale behavior. Singular perturbation approximates the model by setting the derivatives of very fast modes equal to zero.

- optimal Hankel norm approximation [38], which finds a reduced model to minimize the Hankel norm of the error between the original model and the reduced model.

- proper orthogonal decomposition [39]–[41], which applies singular value decomposition to obtain an orthogonal basis for snapshots of the system and retain the most important direction in state space.

- aggregation [42], [43], which selects a set of state variables from the original system using weighting factors and minimizes the selected index. This methods takes relatively low computational effort.

- Padé approximation [44], which is generally used for linear systems. The Padé approximation of the system's transfer function can be truncated to create a low-order transfer function of some different system which matches some number of derivatives of the original system's transfer function.

Most of these methods are for linear systems, and only some of them have been extended for nonlinear cases. While the theory for model reduction of linear systems has advanced significantly, reduction of nonlinear models is still an active area of research [45]–[47] with many unsolved problems [10]. A comprehensive literature review of model reduction was developed by Hahn [48]. The remainder of this section will give a brief review of several important model reduction techniques.

2. Overview of Balancing

For control-relevant reduction of linear systems, balanced model reduction has been a popular approach due to the fact that the modes that contribute the most to the input-output behavior of the system are identified and can then be retained in a subsequent model reduction step.

a. Balancing for Linear Systems

For a stable, linear, time-invariant system

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

it is possible to compute the linear controllability gramian, W_C, and the linear observability gramian, W_O, by the following formulae [49]

$$W_C = \int_0^\infty e^{At} B B^T e^{A^T t} dt, \quad W_O = \int_0^\infty e^{A^T t} C^T C e^{At} dt \tag{2.2}$$

The entries in the gramians provide a measure for the degree of controllability and observability of each state, i.e., the degree to which a state is influenced by the inputs and the effect that changes in this state have on the outputs. The eigenvalues of the quantity $W_C W_O$ can be computed and denoted as $\{\lambda_i \geq 0, i = 1, \cdots, n\}$. The square roots of these eigenvalues are called Hankel singular values [35], denoted as $\{\sigma_i \geq 0, i = 1, \cdots, n\}$.

The gramians can be used to compute a linear state transformation $\bar{x} = Tx$, such that the original system is transformed

$$\begin{aligned} \dot{\bar{x}} &= TAT^{-1}\bar{x} + TBu = \bar{A}\bar{x} + \bar{B}u \\ y &= CT^{-1}\bar{x} = \bar{C}\bar{x} \end{aligned} \tag{2.3}$$

into a balanced realization. One property of a balanced realization is that its controllability and observability gramians are equal and diagonal, where the entries along the diagonal are the Hankel singular values. These Hankel singular values provide a measure for the importance of the states to the input-output behavior of the system.

Balanced model reduction first transforms a system into a balanced form and then reduces the states corresponding to the smallest Hankel singular values, i.e., the states that contribute the least to the input-output behavior of the system. The reduction itself can be performed either by truncation where the reduced states are cut off [9], [35] or by residualization which corresponds to setting the time derivative of the less important states equal to zero [9]. For balanced truncation method, the error bound of balanced model reduction is given by [38]

$$\sigma_{m+1} \leq \|G(s) - G_r(s)\|_\infty \leq 2 \sum_{j=m+1}^{n} \sigma_j \tag{2.4}$$

where $G_r(s)$ refers to a system that has been balanced and only retains states 1 through j, and m is the number of retained states.

b. Balancing for Nonlinear Systems

The theory behind extending balancing from linear models to a certain class of nonlinear systems was first introduced by Scherpen [18]. However, the main drawback of her procedure is its extensive numerical requirement even for small systems [16], [17], [50]. An approximation to balancing of nonlinear systems has been introduced by Lall and coworkers via the concept of empirical gramians [13]. However, their procedure is restricted to nonlinear control affine systems and it also requires modifications for systems where an equilibrium point other than the origin is used [12]. In order to address these points, controllability and observability covariance matrices have been proposed by Hahn and Edgar as an extension of the empirical gramians [12]. These

covariance matrices can be used to compute a balancing-like transformation followed by a model reduction step via truncation or residualization.

Since it is not possible to compute nonlinear gramians for general systems described by

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= h(x) \end{aligned} \tag{2.5}$$

the concepts of controllability/observability covariance matrices have been introduced [12]

$$W_C = \sum_{i=1}^{p}\sum_{l=1}^{r}\sum_{m=1}^{s}\frac{1}{rsc_m^2}\int_0^\infty \Phi^{ilm}(t)dt \tag{2.6}$$

$$W_O = \sum_{l=1}^{r}\sum_{m=1}^{s}\frac{1}{rsc_m^2}\int_0^\infty T_l\Psi^{lm}(t)T_l^T dt \tag{2.7}$$

where $\Phi^{ilm}(t)$=$(x^{ilm}(t)-x_0^{ilm})(x^{ilm}(t)-x_0^{ilm})^T$, $\Psi_{ij}^{lm}(t)$= $(y^{ilm}(t)-y_0^{ilm})^T(y^{jlm}(t)-y_0^{jlm})$, T^n= $\{T_1, \ldots, T_r;\ T_i \in \Re^{n\times n},\ T_i^TT_i = I_n\ ,\ i = 1, .., r\}$, $M = \{c_1, \ldots, c_s; c_i \in \Re\ ,\ c_i > 0\ ,\ \mathrm{i} = 1, \ldots, s\}$, $E^n = \{e_1, \ldots, e_n$; standard unit vectors in $\Re^n\}$, r is the number of matrices for the excitation/perturbation directions, s is the number of different excitation/perturbation sizes for each direction, and p is the number of inputs/states of the system.

Both controllability and observability covariance matrices are symmetric and positive semi-definite, which guarantees the existence of a state transformation [35] $\bar{x} = Tx$ that makes them diagonal and equal in the directions in state space which

are contained in both covariance matrices corresponding to Σ_1:

$$TW_C T^T = \begin{bmatrix} \Sigma_1 & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.8}$$

$$(T^{-1})^T W_O T^{-1} = \begin{bmatrix} \Sigma_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \Sigma_3 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.9}$$

The transformed system is then given by

$$\begin{aligned} \dot{\bar{x}} &= Tf(T^{-1}\bar{x}, u) = \bar{f}(\bar{x}, u) \\ y &= h(T^{-1}\bar{x}) = \bar{h}(\bar{x}) \end{aligned} \tag{2.10}$$

A decision about the states to be retained in the reduced-order model can be made based upon the magnitude of the diagonal entries of the transformed covariance matrices. The reduction itself can then be performed by truncation

$$\begin{aligned} \dot{\bar{x}}_1 &= PTf(T^{-1}\bar{x}, u) \\ \bar{x}_2 &= \bar{x}_{2,ss} \\ y &= h(T^{-1}\bar{x}) \end{aligned} \quad , \quad \bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}, \quad P = \begin{bmatrix} I_{k\times k} & 0 \end{bmatrix} \tag{2.11}$$

where k refers to the number of retained states, P is the projection matrix, which has to be chosen to have appropriate dimensions for reducing the number of differential equations in the model.

One drawback of this technique is that the steady state behavior is not retained in the reduced system. To overcome this disadvantage, balanced residualization has

been introduced [48] where the derivatives of the less important states are set equal to zero

$$\begin{aligned} \dot{\bar{x}}_1 &= P_1 T f(T^{-1}\bar{x}, u) \\ 0 &= P_2 T f(T^{-1}\bar{x}, u) \\ y &= h(T^{-1}\bar{x}) \end{aligned} \tag{2.12}$$

where $\bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}$, $P_1 = [\ I_{k\times k} \quad 0\ ]$, $P_2 = [\ 0 \quad I_{(n-k)\times(n-k)}\ ]$, n: number of states, k: number of retained states

3. Singular Perturbation Method

Singular perturbation method has been used extensively over the last few decades. Since model reduction of DAE (differential and algebraic equation) systems is investigated in this work and one possible approach for obtaining a DAE system is from singular perturbation, some basic concepts will be reviewed here.

The singular perturbation technique was developed for systems which include both fast and slow modes. For these systems, the slow modes dominate while the fast modes can be neglected. This technique identifies those fast modes and eliminates them to produce a reduced-order system [51], [52].

The basic idea for singular perturbation is as follows:

Consider a linear, time-invariant system which described by

$$\begin{aligned} \begin{bmatrix} \dot{x} \\ \dot{z} \end{bmatrix} &= \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} u \\ y &= \begin{bmatrix} C_1 & C_2 \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} + Du \end{aligned} \tag{2.13}$$

where $x \in \Re^r$),$z \in \Re^{n-r}$.

In case of z describing fast and stable modes, this system can be approximated by setting $\dot{z}$ to zero. This yields the following system

$$\begin{aligned} \dot{x} &= A_{11}x + A_{12}z + B_1u \\ 0 &= A_{21}x + A_{22}z + B_2u \\ y &= C_1x + C_2z + Du \end{aligned} \tag{2.14}$$

And z can be represented as $z = -A_{22}^{-1}(A_{21}x + B_2u)$. Therefore, (2.14) can be rewritten as:

$$\begin{aligned} \dot{\bar{x}} &= \bar{A}\bar{x} + \bar{B}u \\ y &= \bar{C}\bar{x} + \bar{D}u \end{aligned} \tag{2.15}$$

where $\bar{A} = A_{11} - A_{12}A_{22}^{-1}A_{21}$, $\bar{B} = B_1 - A_{12}A_{22}^{-1}B_2$, $\bar{C} = C_1 - C_2A_{22}^{-1}A_{21}$, $\bar{D} = D - C_2A_{22}^{-1}B_2$

For this reduced system, one important property is that there is no steady state error compared with the full order system [33].

4. Singular Value Decomposition

Singular value decomposition (SVD) is an eigenvalue-like decomposition for rectangular matrices. Its mathematical formulation is as follows [53]: let A be a general real $m \times n$ matrix, the SVD of A is the factorization $A = U\Sigma V^T$, where $U \in \Re^{m\times m}$ and $V \in \Re^{n\times n}$ are orthogonal, and Σ is a unique diagonal matrix with real, non-negative elements (which are so called singular values) $\sigma_i, \quad i = 1, \cdots, \min(m, n)$ in descending order: $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_{\min(m,n)} \geq 0$. If A is complex, then its SVD is $A = U\Sigma V^H$, where U and V are unitary, and Σ is as before with real diagonal elements. The $\sigma_i's$ are called the singular values, the first [min(m, n)] columns of V are the right singular vectors and the first [min(m, n)] columns of U are the left singular vectors.

The direction corresponding to the largest singular values exhibits the largest variation. Therefore, SVD can identify dimensions based on their importance. And fewer most important dimensions can be used to approximate the original data.

5. Proper Orthogonal Decomposition

Proper orthogonal decomposition (POD) is a projection-based approach for nonlinear model reduction, also referred to as principal component analysis, or Karhunen-Loève expansion [24], which has also received extensive attention. POD can generate low order models for dynamical systems when combined with Galerkin projection procedure [39], [54]–[56]. POD is closely related to balancing and it is often used for order reduction of systems of differential equations, especially for distributed systems [24], [57]. While POD is easy to implement, it is not possible to make similar statements about control-relevance of the reduced model as can be done for balanced model reduction since POD does not take the state-to-output behavior into account.

POD makes use of simulation or experimental data of the states, so called "snapshots", measured at N discrete points in time. Each snapshot is represented by an $n\times 1$ vector $x(t_i)$ and the snapshot matrix is

$$X = [x(t_1), x(t_2), \cdots, x(t_N)] \in \Re^{n\times N} \tag{2.16}$$

which allows construction of the correlation matrix M:

$$M = XX^T \in \Re^{n\times n} \tag{2.17}$$

The principal directions of the data set can be extracted via singular value decomposition of M

$$M = U\Sigma V^*, \quad U = [u_1, u_2, \cdots, u_n] \in \Re^{n\times n}, \quad V = [v_1, v_2, \cdots, v_n] \in \Re^{n\times n} \tag{2.18}$$

where Σ is a diagonal matrix containing the singular values of M and U consists of the corresponding directions.

From the results returned by singular value decomposition it is possible to choose $T = U$ and perform truncation as shown in (2.11). The main difference between balancing and POD is that the transformation matrix T is determined from the input-to-state and state-to-output behavior for balanced reduction whereas POD computes T by extracting the main directions in which the system evolves in state space.

B. System Identification

This part gives a brief review on system identification [58]–[60] which stands for a class of black box modeling methods. System identification can be used to determine the system's input-output relationship by fitting experimental data, which does not look into the details of what is going on inside the system.

Three main components are necessary for system identification: input-output data, candidate model structures, and a performance criterion. Generally, the system identification process starts from planning an experiment to collect input-output data. The data then need to be preprocessed, e.g., removing outliers, filtering and scaling data. A candidate model structure is selected to obtain a best model in the sense of the criterion based on the input output data. The last step is model validation, which is to make the model useful in practice.

Two system identification techniques are briefly reviewed here: artificial neural networks and partial least squares.

1. Artificial Neural Network

An artificial neural network (ANN) [61]–[63] is a type of models mimicking the biological structures of the brain system. ANN is a powerful tool to map nonlinear relationships between process variables, which can be used for system identification. By use of process data, ANN can be trained to represent a nonlinear relationship and therefore can be used to predict process outputs via this relationship. This property is used in this research work. A simple feedforward neural network with one hidden layer shown as Fig. 1 [64] is a typical structure for this function.

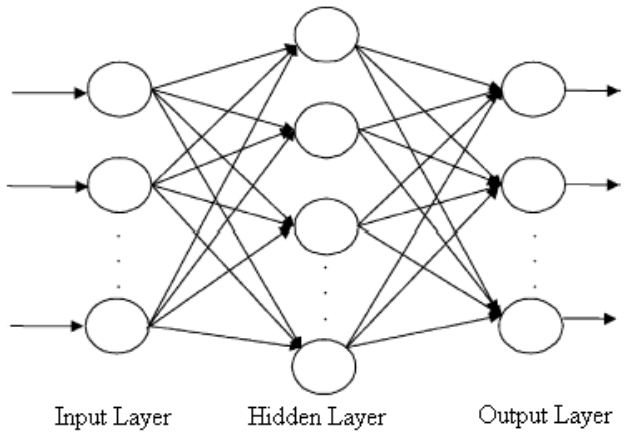

Fig. 1. A three layer feedforward neural network.

2. Partial Least Squares

PLS is a linear system identification method that determines a latent space with orthogonal principle factors to approximate the original input-output space. If an independent matrix X and a dependant matrix Y are given then a PLS model of the form $X = TP^T + E$, $Y = UQ^T + F$, can be constructed where T and U are the score

matrices, P and Q the loading matrices, and E and F are the residual matrices of X and Y, respectively. PLS extracts latent variables (also called principle factors) by analyzing the sample covariance matrix $(X^TY)(Y^TX)$. These latent variables can not only capture the variance of X, but also maximize the covariance between each X score and the corresponding Y score. The latent variables are orthogonal and therefore independent of one another. In most cases, the first few latent variables can capture the most useful information between X and Y. The NIPALS algorithm [65] is an efficient way to compute the PLS latent variables sequentially.

a. Data Preprocessing

To apply PLS, it is useful to scale the data sets to simplify the calculations. The scaled form of a specific variable x is given by

$$\hat{x} = \frac{x - \bar{x}}{\text{std}(x)} \tag{2.19}$$

where $\bar{x}$ is the mean value of the variable $x$and std (x) refers to the standard deviation.

All variables in the independent and dependent block are preprocessed in this way and all variables involved in the PLS procedure are scaled.

b. Calibrating a PLS Model

The main idea of PLS is to find an inner relationship between X and Y. This relationship can be approximated by their score matrices T and U with $U = B^*T$, where B is the regression matrix. The NIPALS algorithm is applied to construct a PLS model via an iterative procedure.

c. Data Prediction

Prediction of the dependent matrix from the independent matrix is the most important part of PLS. This is performed by multiplying the independent matrix by the regression coefficient matrix B which is obtained in the calibration step, i.e., $Y = B * X$.

C. Parameter Selection

Parameter reduction is a topic that is directly tied to first-principles modeling. The reason for this is that an existing model is used as the starting point and the number of parameters in this model is lowered during the process. The goal of the reduction procedure is to determine which parameters can be replaced by constant values, even if these are not accurately known, as compared to other parameters which have to be estimated from data. While techniques used for parameter reduction are often based on statistical techniques, i.e. singular value decomposition, the application of the methods is distinctly different from chemometric techniques as the goal is to keep the structure of the model intact.

While an extensive body of literature exists on model simplification and reduction (e.g. to name just a few selected papers: [4], [12], [13], [36]), the number of publications on reducing the number of parameters in a model is much more limited. One application of parameter reduction can be found in [66] where sensitivity analysis and principle component analysis are used to reduce the number of parameters in a model representing a complex metabolic network.

Parameter reduction is also closely related to parameter estimation [67]–[72]. For parameter estimation, only those parameters with greater estimability can be estimated from process data. Parameters with little contribution to the input-output

behavior can be neglected, as they can usually not be estimated in a realistic scenario. The difference between parameter estimation and parameter reduction lies in the fact that the former topic is to collect process information and develop algorithms to estimate parameters as accurate as possible, while the latter does not involve the estimation itself, but determines the relative importance of parameters and then reduces the less important ones. A common task for parameter estimation and parameter reduction is parameter selection. This topic has been extensively investigated in the literature using the concept of the Fisher information matrix [6], [73]–[77].

1. Sensitivity Analysis

Sensitivity analysis is a powerful tool for parameter selection, which investigates the effect that changes in the parameters have on the outputs of a system [78].

For a general ODE system given by

$$\begin{aligned} \dot{x} &= f(x, \theta, u) \\ y &= h(x, \theta, u) \end{aligned}, \quad \theta \in \Re^k \tag{2.20}$$

the sensitivity function (SF) of each parameter can be defined as

$$SF(\theta_j) = \frac{\partial y}{\partial \theta_j} \tag{2.21}$$

In order to obtain a general quantity, such that the values of sensitivity functions for each parameter can be compared, the dimensionless sensitivity function can be computed by

$$DSF(\theta_j) = \frac{\partial y}{\partial \theta_j} \cdot \frac{\theta_{j,0}}{y_0} = \frac{\partial \ln y}{\partial \ln \theta_j} \tag{2.22}$$

This quantity is equivalent to the sensitivity function in (2.21) when all variables are normalized.

CHAPTER III

DAE MODEL REDUCTION VIA PROJECTIONS AND NEURAL NETWORK

A. Overview

Differential-algebraic equation (DAE) systems are very common for describing process models, especially for models derived from first principles. They consist of differential as well as algebraic equations, resulting in differential-algebraic equation (DAE) systems [79], [80]. In these models, the differential equations, such as dynamic mass or energy balances, are used to describe the dynamic behavior while the algebraic equations represent phenomena at a faster time scale, such as thermodynamic equilibrium calculations. DAE can also result from singular perturbations of ordinary differential equation (ODE) systems [81], where the states are separated into slow and fast ones. Models described by DAE systems are often of high order resulting in difficulties for online control due to the extensive computational effort. Reducing the size of the model while retaining important system properties for controller design is the main goal of control-relevant model reduction.

While initial work on model reduction focused on linear systems, more recent research has dealt with nonlinear models described by sets of ordinary differential equations (ODE) [11], [13], [48] or partial differential equations (PDE) [19], [21]. At the same time, model reduction of DAE systems has received considerably less attention [10]. One may be inclined to argue that model reduction of DAE systems could consist of reduction of the differential equations by already existing techniques

---

Part of this chapter is reprinted from *Journal of Process Control*, Vol. 15, No. 6, C. Sun and J. Hahn, Reduction of stable differential algebraic equation systems via projections and system identification, pp. 639-650, Copyright (2005), with permission from Elsevier.

while the algebraic equations remain untouched. However, the number of algebraic equations in most models far exceeds the one for differential equations. This is one important aspect that must be taken into account when investigating methodologies for model reduction of DAE systems. The only approach for model reduction of nonlinear DAE systems for controller design before this research work was the technique presented by Löffler and Marquardt [82]. They generated state trajectories, computed a covariance matrix from data, extracted the principal components of the covariance matrix, and performed a projection on the original system. However, their method does not take the different character of the differential and the algebraic equations into account and different choices for generating the data set can lead to different principal directions.

A new technique is presented in this part of the work for reducing nonlinear DAE systems for controller design. The method reduces the order of the model by eliminating differential equations as well as the number and complexity of the algebraic equations via a procedure combining projections and neural network. This technique addresses both reduction of the algebraic and the differential equations and results in a system where the algebraic equations can be represented by an explicit expression, e.g., a feedforward neural network. This last property is important insofar as the reduced model does not require a DAE solver for its solution but can instead be computed by regular ODE solvers.

B. Model Reduction of DAE Systems

This research work focuses on DAE systems of the following form

$$\begin{aligned} \dot{x} &= f(x, z, u) \\ 0 &= g(x, z) \\ y &= h(x) \end{aligned} \tag{3.1}$$

where $x \in \Re^d$, represents a vector containing dynamic states of the system and $z \in \Re^{(n-d)}$, refers to the variables computed from the algebraic equations, where n is the total number of differential states and algebraic variables. It is assumed that $\partial g/\partial z \neq 0$, restricting this work to DAE systems with an index of at most 1 [79], [83]. However, this is not a serious drawback, since any process found in nature can be modeled by DAE systems of index 1 or lower. A second assumption for this work is that only one equilibrium point exists within an operating region for a constant input to ensure that the information contained in the covariance matrices represents the behavior of the system over the operating region.

The presented model reduction procedure combines elements from balanced model reduction with system identification techniques in order to reduce the differential as well as the algebraic equations while retaining control-relevant properties of the model. In a first step of the procedure, transformations are applied to both differential and algebraic equations. The transformation for the differential equations is computed via balancing of the covariance matrices or via POD, while the transformation matrix for the algebraic equations is obtained by singular value decomposition of the state covariance matrix computed for different excitations of the system. In a second step, both differential and algebraic equations of the transformed system can then be reduced via a truncation procedure. The model is further reduced by replac-

ing the algebraic equations with an identified model, i.e. a neural network is trained in order to represent the relationship between the remaining differential equations, i.e., the inputs to the neural network, and the remaining algebraic variables which are the outputs of the neural network.

The strong points of this procedure are that (1) the initial order reduction only retains states or variables with a significant contribution to the control-relevant input-output behavior of the system, (2) due to this initial step only states which contribute to the observed behavior are used for the system identification, (3) correlations among the states used for system identification have been eliminated in the order reduction step, (4) the implicit algebraic equations are replaced by an explicit expression simplifying the effort required for numerical solution, and (5) the method offers the potential to significantly reduce the number of differential as well as algebraic equations of a model.

A detailed description of the individual steps to be performed is presented in the following subsections.

1. Computation of Transformations

a. Computation of Transformations via Balancing/SVD

For ordinary differential equation (ODE) systems, the procedure of computing the transformation matrix via balancing of controllability and observability covariance matrices was presented in Hahn and Edgar [12]. However, since this procedure was derived for ODEs, it is not directly applicable to DAE systems due to the nature of the algebraic equations: a DAE system consisting of d differential equations and $(n - d)$ algebraic equations will only evolve in a subspace of dimension d, as the algebraic equations can not represent dynamic behavior. Therefore, it is not

meaningful to compute the controllability and observability covariance matrices for the entire systems (differential states and algebraic variables) due to the reason that both covariance matrices are defined for dynamic states only. Instead the following observations are made:

- The DAE system can be excited by changes in the inputs and a covariance matrix can be computed from the resulting trajectories. The controllability covariance matrix, W_C, for the states given by the differential equations is part of the computed covariance matrix.

- While a covariance matrix for the algebraic variables can also be computed from the state trajectories generated by excitations with the inputs, it is important to point out that this does not correspond to the controllability covariance matrix. The reason is that the algebraic variables do not exhibit dynamic behavior, e.g., they do not increase the rank of the covariance matrix for the differential and the algebraic equations if the algebraic equations are linear. Instead the covariance matrix for the algebraic variables serves as an indicator of the correlation among the algebraic variables and can thereby be used for reducing the number of algebraic variables.

- The observability covariance matrix, W_O, can also only be computed for states given by the differential equations. A perturbation of the algebraic variables would have the effect that the differential equations would also need to be perturbed in order to have consistent initial conditions. It can be shown that this will not increase the rank of the covariance matrix beyond the number of the states for a linear system.

Therefore, a modified version of the balancing procedure is presented for application to DAE systems: (1) the system is originally at steady state and is then

excited by changes in the inputs along the lines of the computation procedure for the controllability covariance matrix; data are collected along the trajectories generated by these excitations and a covariance matrix

$$W = \begin{bmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{bmatrix}, \qquad \begin{aligned} &W_{11} \in \Re^{d\times d}, \quad W_{12} \in \Re^{d\times(n-d)}, \\ &W_{21} \in \Re^{(n-d)\times d}, \quad W_{22} \in \Re^{(n-d)\times(n-d)}. \end{aligned} \tag{3.2}$$

is computed, where W_{11} is equal to W_C, the controllability covariance matrix of the differential states and W_{22} is the covariance matrix of the algebraic variables; (2) the observability covariance matrix, W_O, is computed for states described by the differential equations as shown in equation (2.2); (3) the transformation T_1 for the states, x, is computed from balancing W_C and W_O; (4) a singular value decomposition of W_{22}

$$W_{22} = U_2 \Sigma_2 V_2^* \tag{3.3}$$

is used to compute the transformation T_2 for the algebraic variables, z, where $T_2 = U_2$.

b. Computation of Transformations via POD

It is possible to use the same state trajectories for the correlation matrix as were used in the previous subsection, since the trajectories in both cases are generated by excitation of the system with the available inputs. The correlation matrix M is partitioned into four submatrices similar to what has been done for the matrix W in (3.2):

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}, \qquad \begin{aligned} &M_{11} \in \Re^{d\times d}, \quad M_{12} \in \Re^{d\times(n-d)}, \\ &M_{21} \in \Re^{(n-d)\times d}, \quad M_{22} \in \Re^{(n-d)\times(n-d)}. \end{aligned} \tag{3.4}$$

Transformations can be computed by applying singular value decomposition to

M_{11} and M_{22}, respectively, as shown in (3.5):

$$M_{11} = U_1 \Sigma_1 V_1^* \quad , \quad M_{22} = U_2 \Sigma_2 V_2^* \tag{3.5}$$

The unitary matrix $U_1$and U_2 serve as transformations T_1 and T_2 for the differential states and algebraic variables, respectively.

c. Transformed System

Regardless if the transformations have been computed from balancing, POD, or some other methods, the transformations T_1 and T_2

$$\begin{array}{ll} \bar{x} = T_1 x \Rightarrow x = T_1^{-1}\bar{x} & \bar{z} = T_2 z \Rightarrow z = T_2^{-1}\bar{z} \\ x, \quad \bar{x} \in \Re^d & z, \quad \bar{z} \in \Re^{n-d} \end{array} \quad , \tag{3.6}$$

can be applied to the original model ((3.1)), resulting in the transformed system

$$\begin{aligned} \dot{\bar{x}} &= T_1 f(T_1^{-1}\bar{x}, T_2^{-1}\bar{z}, u) = \bar{f}(\bar{x}, \bar{z}, u) \\ 0 &= g(T_1^{-1}\bar{x}, T_2^{-1}\bar{z}) = \bar{g}(\bar{x}, \bar{z}) \\ y &= h(T_1^{-1}\bar{x}) = \bar{h}(\bar{x}) \end{aligned} \tag{3.7}$$

where $\bar{f}, \quad \bar{g}, \quad \bar{h}$ represent nonlinear functions of the transformed system.

This transformed system has the same number of differential and algebraic equations as the original system and identical input-output behavior. However, the differential states as well as the algebraic variables are ordered in descending order with their importance to the control-relevant behavior of the model. Essentially, the system (3.7) is in a set of coordinates suitable for reducing the size of the model.

2. Order Reduction of Differential Equations

Once the system is transformed into a form suitable for model reduction, the number of differential equations is reduced by truncation while the algebraic variables remain unchanged, resulting in

$$\begin{aligned} \dot{\bar{x}}_1 &= P_1 \bar{f}(\bar{x}, \bar{z}, u) \\ \bar{x}_2 &= \bar{x}_{2,ss} \\ 0 &= \bar{g}(\bar{x}, \bar{z}) \\ y &= \bar{h}(\bar{x}) \end{aligned} \quad , \quad \bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}, \quad P_1 = [I_{k\times k} \quad 0] \tag{3.8}$$

where $\bar{x}_1$ contains the states of the reduced system, $\bar{x}_2$ represents the states that are reduced and k is the number of differential equations in the reduced-order model.

The error bounds resulting from the truncation of the differential equations for a system of linear DAEs are derived as follows. While a linear system can only approximate the nonlinear model locally, the following presentation serves as an illustration that the proposed model reduction takes the interaction of the differential and the algebraic equations into account. It can also be seen from that the transformation T_1, while being applied to the differential equations only, is dependent upon the differential equations, the algebraic equation as well as terms coupling the equations.

Consider a stable linear DAE system of the following form

$$\begin{aligned} \dot{x} &= Ax + A_z z + Bu \\ z &= Ex \\ y &= Cx \end{aligned} \tag{3.9}$$

where x is the vector of states and z represents a vector of algebraic variables.

Since the algebraic equations are given by an explicit expression for this system,

it is possible to substitute z with the expression $z = Ex$ to obtain an ODE system

$$\begin{aligned} \dot{x} &= (A + A_z E)x + Bu \\ y &= Cx \end{aligned} \tag{3.10}$$

As this system consists of stable linear ODEs, the controllability and observability covariance matrices reduce to the linear controllability and observability gramians, given by

$$W_C = \int_0^\infty e^{(A+AzE)t} B B^T e^{(A+AzE)^T t} dt, \quad W_O = \int_0^\infty e^{(A+AzE)^T t} C^T C e^{(A+AzE)t} dt \tag{3.11}$$

The balancing procedure determines a transformation T_1 which makes the controllability and observability gramians of the balanced system equal and diagonalizes them:

$$\bar{W}_C = T_1 W_C T_1^T = \bar{W}_O = (T_1^{-1})^T W_O T_1^{-1} = \Sigma = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n \end{bmatrix} \tag{3.12}$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0$ are the Hankel singular values of the system (3.10).

With the transformation $\bar{x} = T_1 x$, the balanced system is given by

$$\begin{aligned} \dot{\bar{x}} &= (\bar{A} + \overline{A_z E})\bar{x} + \bar{B}u \\ y &= \bar{C}\bar{x} \end{aligned} \tag{3.13}$$

where $\bar{A} = T_1 A T_1^{-1}$, $\overline{A_Z E} = T_1 A_Z E T_1^{-1}$, $\bar{B} = T_1 B$, $\bar{C} = C T_1^{-1}$

The state vector of the balanced system can be partitioned into two parts: states $\bar{x}_1$which contribute the most to the input-output behavior and less important states

$\bar{x}_2$. Accordingly, the system matrices can be partitioned:

$$\begin{aligned} \bar{A} &= \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix}, \quad \overline{A_Z E} = \begin{bmatrix} (\overline{A_Z E})_{11} & (\overline{A_Z E})_{12} \\ (\overline{A_Z E})_{21} & (\overline{A_Z E})_{22} \end{bmatrix} \\ \bar{B} &= \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \end{bmatrix}, \quad \bar{C} = \begin{bmatrix} \bar{C}_1 & \bar{C}_2 \end{bmatrix} \end{aligned} \tag{3.14}$$

Truncation of the system (3.13) results in

$$\begin{aligned} \dot{\bar{x}}_1 &= [\overline{A}_{11} + (\overline{A_Z E})_{11}]\bar{x}_1 + \overline{B}_1 u \\ y &= \overline{C_1}\bar{x}_1 \end{aligned} \tag{3.15}$$

where $\overline{A}_{11}$,$(\overline{A_Z E})_{11}$, $\overline{E}_{11}$, $\overline{B}_1$ and $\overline{C_1}$ have appropriate dimensions.

Because this procedure is identical to balanced truncation for the system (3.10), the error between the original system (3.9) and the reduced system (3.15) is bounded by [38]

$$\sigma_{k+1} \leq \|G(s) - G_r(s)\|_\infty \leq 2 \sum_{j=k+1}^{n} \sigma_j \tag{3.16}$$

where the σ_j represent Hankel singular values described in (3.12).

It can be concluded from the presented case that this reduction procedure as well as the error bounds depend not only on the matrices A, B, C, but also on A_Z and E. Therefore, the algebraic equations, i.e. given by E in this case, as well as part of the interaction between the differential and the algebraic equations, i.e. given by A_Z, have been taken into account for the reduction. Note that A_Z and E affect the reduction even if the only goal is to reduce the differential equations of a DAE system.

3. Reduction of Algebraic Equations

The reduction of the algebraic equations has to be specifically addressed for nonlinear DAE systems since the number of algebraic equations resulting from first-principles modeling usually far exceeds the number of differential equations. Similar to what can be observed for the differential states, there are some algebraic variables that dominate the observed behavior of the system, e.g., some variables will hardly move even for significant changes in the inputs or some variables are linearly dependent and do not need to be modeled separately. The described computation of the transformation T_2 for the algebraic variables ensures that the dominant components are contained in the first few algebraic variables of the system and that variables that are almost linearly dependent on these first few variables will be of lesser importance. It is then possible to reduce the number of algebraic equations via a truncation method with m algebraic variables retained in the model. The exact value of m has to be determined from the singular values contained in Σ_2 from (3.3) (if balancing is applied) or (3.5) (if POD is applied). Combined with truncation of the differential equations, the reduced system is given by

$$\begin{aligned} &\dot{\bar{x}}_1 = P_1 \bar{f}(\bar{x}, \bar{z}, u) \\ &\bar{x}_2 = \bar{x}_{2,ss} \\ &0 = P_2 \bar{g}(\bar{x}, \bar{z}) \\ &\bar{z}_2 = \bar{z}_{2,ss} \\ &y = \bar{h}(\bar{x}) \end{aligned} \quad , \quad \bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}, \quad P_1 = [\ I_{k\times k} \quad 0] \quad , \quad \bar{z} = \begin{bmatrix} \bar{z}_1 \\ \bar{z}_2 \end{bmatrix}, \quad P_2 = [\ I_{m\times m} \quad 0] \tag{3.17}$$

In this system, the differential states and algebraic variables of lesser importance have been truncated and replaced by constant values while the most important components of the model are retained. The resulting model (3.17) contains fewer differential and fewer algebraic equations than the original system (3.1).

4. Further Reduction of Algebraic Equations via System Identification

So far, the number of differential and algebraic equations has been reduced by the described technique. While the model reduction procedure could stop at this point, it is possible to obtain a more suitable system of even smaller size and lesser complexity. The reason for this is that the truncated differential states and algebraic variables can be removed from the algebraic equations and that the algebraic equations can be (locally) approximated by an explicit expression. Consider (3.18), which is part of the DAE system (3.17)

$$0 = P_2 \bar{g}(\bar{x}, \bar{z}) \tag{3.18}$$

where $\bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}$ and $\bar{z} = \begin{bmatrix} \bar{z}_1 \\ \bar{z}_2 \end{bmatrix}$, which allows to rewrite (3.18) as

$$0 = P_2 \bar{g}\left(\begin{bmatrix} \bar{x}_1 \\ \bar{x}_{2,ss} \end{bmatrix}, \begin{bmatrix} \bar{z}_1 \\ \bar{z}_{2,ss} \end{bmatrix} \right) \tag{3.19}$$

Since $\bar{x}_{2,ss}$ and $\bar{z}_{2,ss}$ are constant vectors and P_2 is a constant matrix, this can be rewritten as

$$0 = \hat{g}(\bar{x}_1, \bar{z}_1) \tag{3.20}$$

where $\hat{g}$ represents a new nonlinear function containing the variables $\bar{x}_1$ and $\bar{z}_1$ only.

Essentially, the set of states, $\bar{x}_1$, can be viewed as the inputs to the algebraic equations shown in (3.20), and the set of algebraic variables, $\bar{z}_1$, represent the outputs. Since the purpose of the algebraic equations is to represent this relationship between $\bar{x}_1$ and $\bar{z}_1$, it is possible to identify the effect of $\bar{x}_1$ on $\bar{z}_1$ and determine a suitable mathematical expression describing this behavior. The relationship between $\bar{x}_1$ and $\bar{z}_1$ needs to be retained in the reduced model to achieve a good approximation of the behavior of the original system. However, equation (3.20) is an implicit expression for $\bar{z}_1$ which may be non-trivial to solve. It would be desirable to replace this implicit

equation for the algebraic variables with an explicit expression. Since the inputs and the outputs of the algebraic equations are dependent in a static manner, it is sufficient to identify this static relationship between $\bar{x}_1$ and $\bar{z}_1$ as shown in (3.21)

$$\bar{z}_1 = \tilde{g}(\bar{x}_1) \tag{3.21}$$

resulting in the following system:

$$\begin{aligned} \dot{\bar{x}}_1 &= P_1 \bar{f}(\bar{x}, \bar{z}, u) = \hat{f}(\bar{x}_1, \bar{z}_1, u) \\ \bar{x}_2 &= \bar{x}_{2,ss} \\ \bar{z}_1 &= \tilde{g}(\bar{x}_1) \\ \bar{z}_2 &= \bar{z}_{2,ss} \\ y &= \bar{h}(\bar{x}) = \hat{h}(\bar{x}_1) \end{aligned} \tag{3.22}$$

where $\hat{f}, \quad \tilde{g}, \quad \hat{h}$ represent nonlinear functions of the reduced system.

Equation (3.22) represents an approximation to the system shown in (3.17) and the relationship between $\bar{x}_1$ and $\bar{z}_1$ is shown in Fig. 2, in which $\bar{z}_1$ is the output of an identified model which receives $\bar{x}_1$ as the input. $\bar{z}_1$ on the other hand serves as an input to the differential equations.

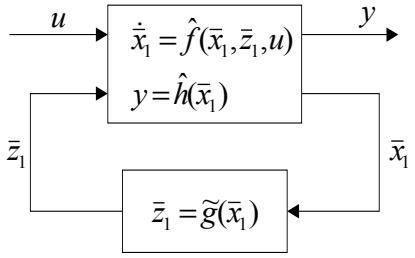

Fig. 2. Relationship between remaining differential states and algebraic variables.

It has to be noted that the relationship between $\bar{x}_1$ and $\bar{z}_1$ is usually nonlinear. One type of model that is able to take this property into account is a feedforward

neural network. The inputs of the neural network are given by $\bar{x}_1$ while the vector $\bar{z}_1$ serves as the outputs for approximating the set of equations $0 = \hat{g}(\bar{x}_1, \bar{z}_1)$ by $\bar{z}_1 = \tilde{g}(\bar{x}_1)$. The number of nodes in the hidden layer is a parameter that has to be determined for using this type of neural network for system identification. In the case where the dimension of $\bar{x}_1$ and $\bar{z}_1$ is significantly smaller than the number of states of the original system, the neural network will be small and can be easily trained. The data set used for training the neural network is a series of the state trajectories, but transformed into a new set of coordinates by the transformations T_1 and T_2. The first k differential states in the transformed system are extracted as the inputs of the ANN while the first m algebraic variables are the target outputs. The resulting reduced system is given by

$$\begin{aligned} \dot{\bar{x}}_1 &= \hat{f}(\bar{x}_1, \bar{z}_1, u) \\ \bar{z}_1 &= \tilde{g}(\bar{x}_1), \quad (\text{here : } \ \bar{z}_1 \ \text{ is expressed by use of } \bar{x}_1 \text{ via an ANN }) \\ y &= \hat{h}(\bar{x}_1) \end{aligned} \tag{3.23}$$

5. Summary of the Model Reduction Procedure

A summarizing description of the model reduction procedure is provided in this subsection.

In a first step, transformations are computed for both differential states and algebraic variables either via balancing or POD and these transformations are applied to the system. The resulting model has the same order and identical input-output behavior to the original system. However, it has the advantage that the important parts of the model can more easily be identified and extracted. Model reduction is performed in a subsequent step by truncation of the differential and algebraic equations. Further reduction of the complexity of the algebraic equations is obtained by system identification. In this work a feedforward neural network is used for system

identification because this type of model structure can accurately represent data from nonlinear systems and routines for its training, validation, and implementation are readily available. However, other models are equally valid and can be incorporated into the model reduction procedure without modifications. A flowchart summarizing the individual steps of the reduction procedure is presented in Fig. 3.

C. Case Study: Distillation Column

1. Model Description

Consider a distillation column with 30 trays for separation of a binary mixture of cyclohexane and heptane. The column has 32 differential states (concentrations of component A: cyclohexane) and symmetric product compositions. The Wilson equation is used for computation of the vapor-liquid equilibrium, resulting in a model with 32 differential equations and 32 algebraic equations. The feed stream is introduced in the middle of the column on stage 17 and has a composition of x_F= 0.5 and a temperature of 361.4 $^\circ$C. The distillate and bottom purities are x_D= 0.973, and x_B = 0.027, at temperatures of T_D = 354.2 $^\circ$C and T_B = 370.1 $^\circ$C, respectively. The concentration and temperature distributions along the height of the column are shown in Fig. 4, where the first elements represent the states corresponding to the condenser and the 32^{nd} elements refer to those of the reboiler. The reflux ratio is set to 3.0 and serves as the manipulated variable while the concentration of the distillate is the output of the system. The equations for this model are given in Appendix A.

2. Data Sets Collection

Four data sets were collected for comparing reduced-order models derived from balancing and POD, both of which include reduction of the algebraic equations via

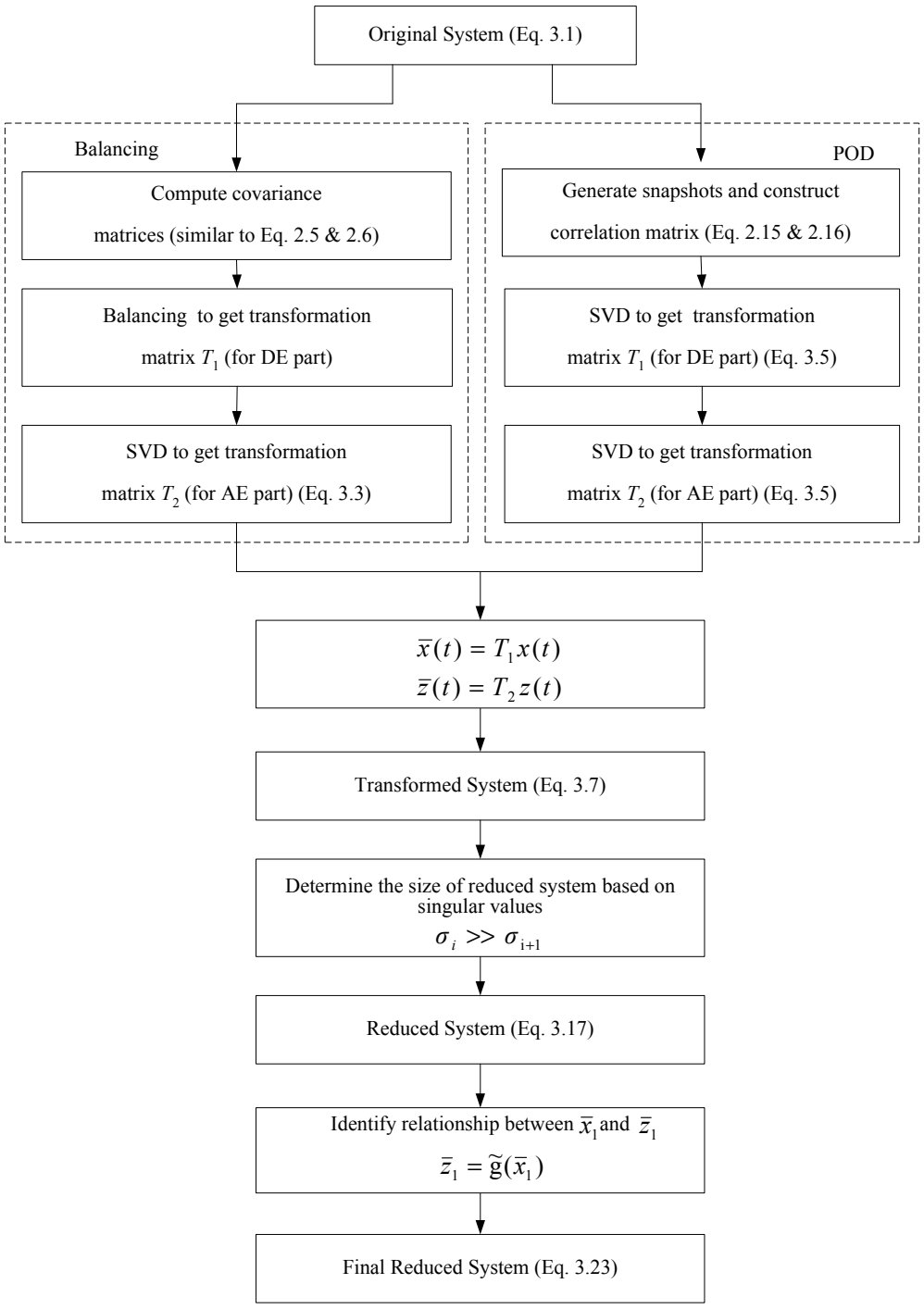

Fig. 3. Flowchart of the model reduction procedure for DAE systems.

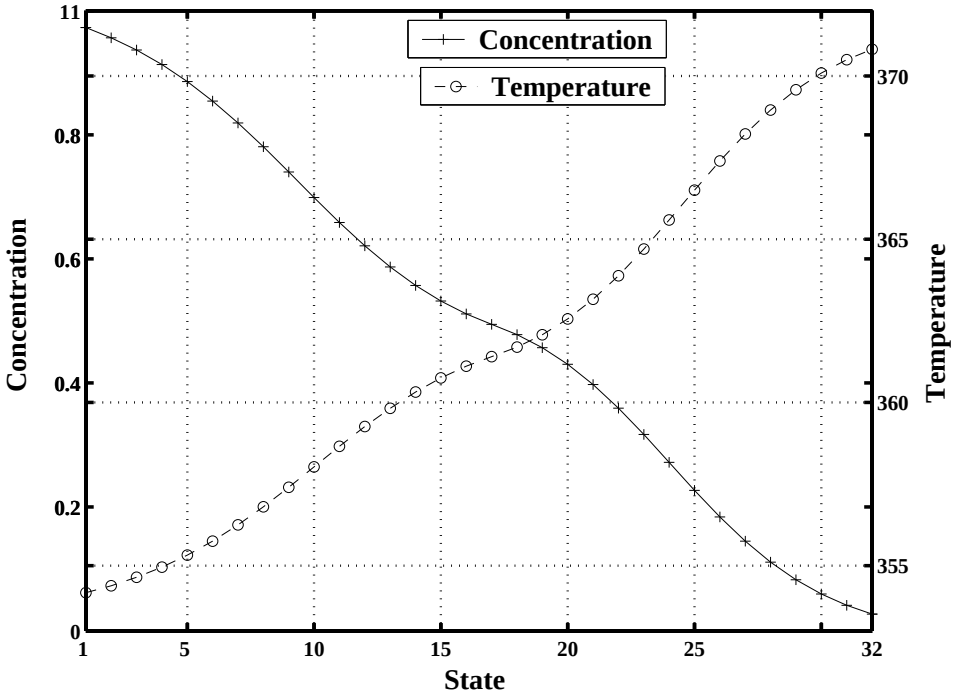

Fig. 4. Concentration and temperature distributions in the distillation column.

neural networks. One of the data sets consists of the snapshots for POD, two sets are required for computing covariance matrices for balancing, and the fourth is the training set for the neural network. Only state trajectories resulting from input excitations are required for the snapshots, which makes it a computationally efficient procedure. Balancing requires one data set describing the input-to-state behavior and one for the state-to-output behavior of the system. A covariance matrix for the data set generated by input excitations of $\pm 10\%$ around the nominal value is computed and the differential states were perturbed by $\pm 10\%$ to obtain the observability covariance matrix. A comprehensive data set is required for training the neural network since extrapolation does not hold for identified models. This is achieved by exciting the system with pulse inputs where the system can reach its new steady state before it is forced back to the original state. A combination of different excitation magni-

tudes, directions, and durations has to be used in order to ensure that all the main components of the system behavior are reflected in this data set.

3. Order Reduction of Both Differential and Algebraic Equations

The transformation matrices have been computed for the differential states and the algebraic variables via balancing or POD. When balancing is applied for the reduction of the differential equations, the main criterion to determine the number of states to be retained is based on the magnitude of the Hankel singular values of the balanced covariance matrices. Sorted by the magnitude from large to small, the first 10 Hankel singular values are shown in Fig. 5 and Table I lists the values of the first 6 as well as their corresponding percentages. For this example, truncated systems that contain 2, 3 and 5 states were investigated. For model reduction via POD, the number of the remaining differential states is determined based on the singular values contained in the matrix Σ_1 (see equation 3.3) and reduced systems with 2, 3 and 5 remaining differential equations were investigated. The singular values in the matrix Σ_2 (see equations 3.3 and 3.5) indicate that a system with 3 algebraic variables is sufficient for models reduced by either balancing or POD.

Table I. Hankel singular values

State	1	2	3	4	5	6
Singular Value	0.048	0.0023	0.0002	0.00006	0.000009	0.000006
% of sum	94.9%	4.5%	0.4%	0.12%	0.0178%	0.0119%

Since the only difference between balancing and POD is the procedure used for computing the state transformation for the differential equations, the performance of

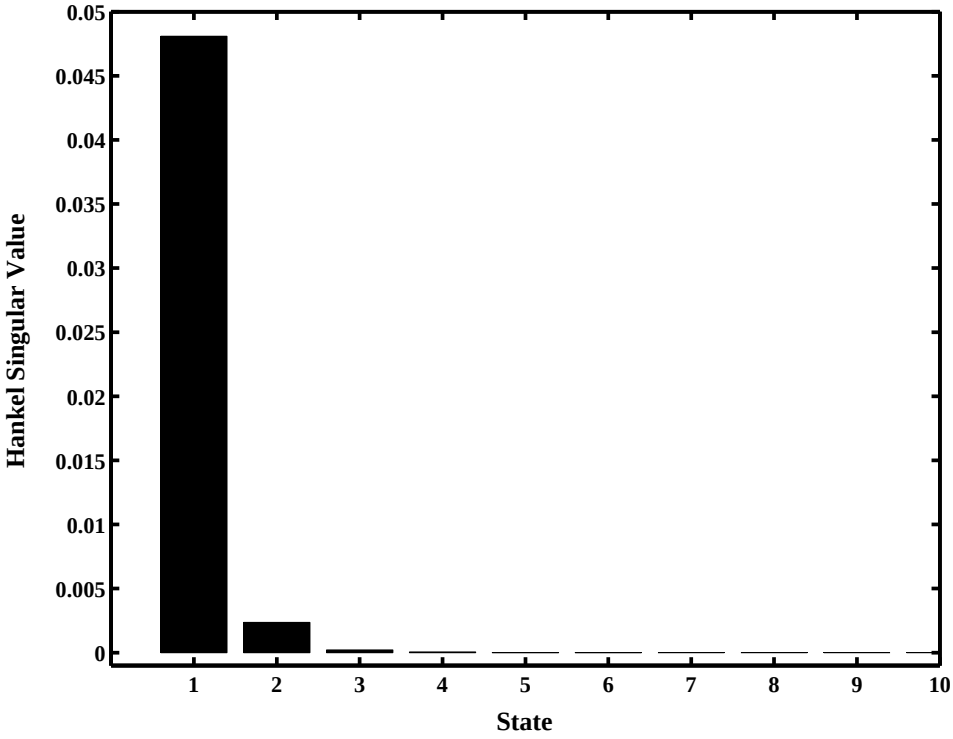

Fig. 5. Hankel singular values of the distillation column model.

these two methods is first compared where the algebraic equations remain untouched. The results are shown in Fig. 6, where the performance of the reduced systems using different reduction methods but also different number of states remaining in the model are compared. Since the simulation results of the reduced systems for the different cases were close to each other, the residuals between the results returned from the reduced order models and the full-order system are shown in Fig. 6 instead of the absolute values. Note that the scale of the y-axis is 10^{-4}, indicating excellent performance for all of the reduced-order models except one. Since there is a relatively large residual for the model with two states reduced by POD, the corresponding curve is only partially shown in Fig. 6. One observation from this figure is that balancing can result in a better approximation than POD if the same number of states is retained. This result is expected since balancing takes the input-to-state and

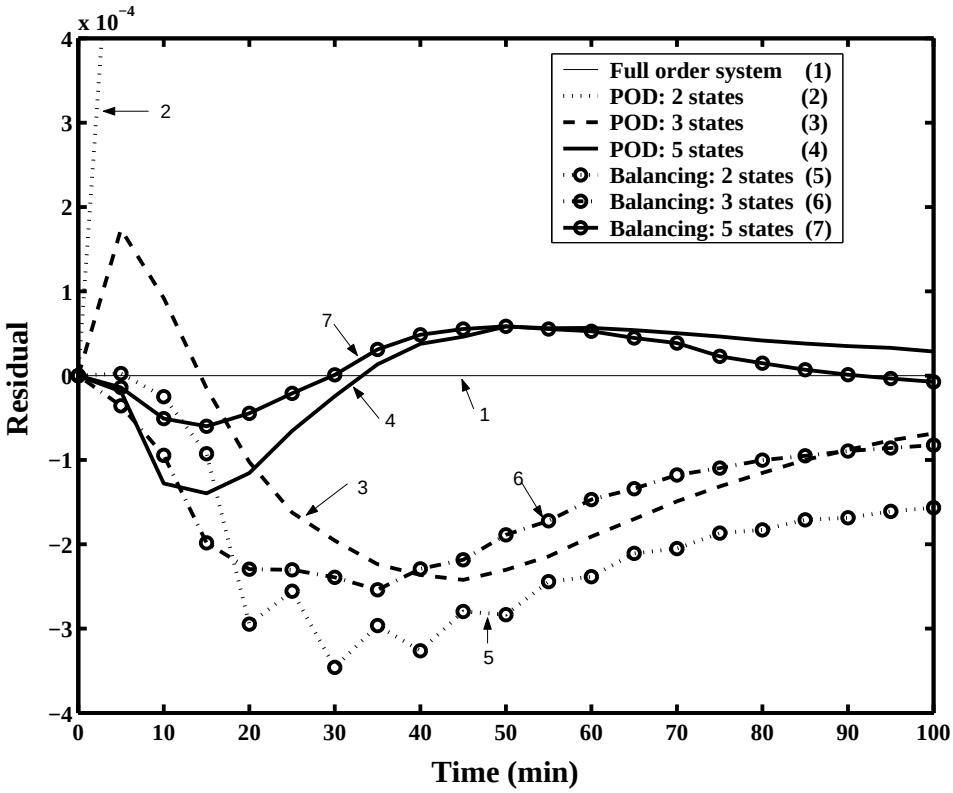

Fig. 6. Comparison between POD and balancing with different number of remaining differential equations.

the state-to-output behavior into account whereas POD neglects the state-to-output behavior. It has also been illustrated that systems with more states more closely approximate the original system. However, there is always a tradeoff between the quality of the approximation and the required computational effort, as is illustrated in Table II. Based on Fig. 6 and Table II, it can be concluded that the reduced system with 3 differential equations results in a very good approximation to the original system with a relatively small computational burden.

Table II. Comparison of CPU times for reduced-order models

States	2	3	5	32
Time (Seconds)	0.06	0.07	0.14	0.4

4. Further Reduction of Algebraic Equations via Neural Network

The neural network toolbox in MATLAB was used for this work and the chosen neural net contains 1 hidden layer and 1 output layer, with 5 nodes in the hidden layer and 3 nodes in the output layer. Hyperbolic tangent functions were used in the hidden layer and linear functions in the output layer. The network was trained using the Levenberg-Marquardt algorithm [84], [85]. After order reduction has been performed as described in subsection 4.3, the remaining differential states serve as the inputs to the neural network while the outputs are given by the remaining algebraic variables.

Three cases are compared to illustrate the performance of the presented method for model reduction of DAE system: (1) a linearized system with 32 differential equations and 32 algebraic variables; (2) only the differential equations are reduced by balanced truncation while the algebraic equations remain unchanged, resulting in a reduced system with 3 differential equations and 32 algebraic variables; (3) the system is reduced by the presented procedure, i.e., differential equations as well as algebraic equations are reduced by truncation and the effect that the states have on the remaining algebraic variables is identified by a neural network. This last reduced system contains 3 differential equations and a neural network with three inputs and three outputs. Fig. 7 shows a comparison of the performance of these three reduced-order systems for step changes in the input of -10% and +10%. Several

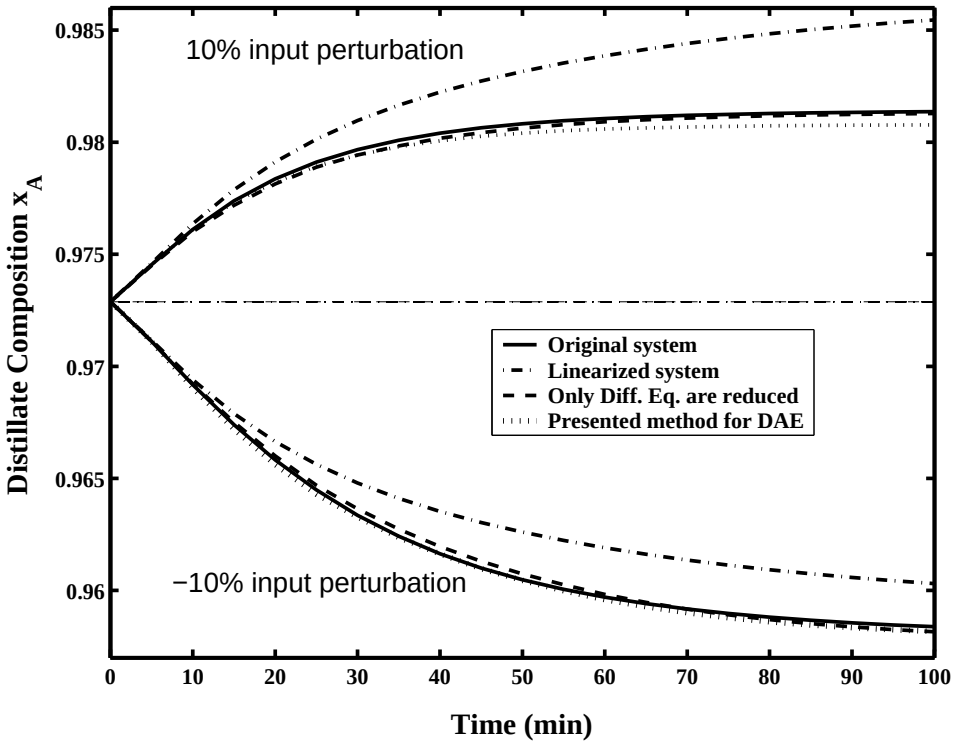

Fig. 7. Performance of presented method for DAE model reduction.

observations can be made based upon Fig. 7: (1) the upper and the lower trajectories are not symmetric, which illustrates the nonlinearity of the original system; (2) the performance of the linearized system is not as good as the ones using a nonlinear reduced model; (3) the reduced model including a neural network provides a good approximation to the full-order system. Although the reduced DAE system is a fairly small model, it exhibits better performance than case 1 and performance comparable to model 2 and to the original system.

D. Summary and Discussion

This chapter presents a new approach for the reduction of stable nonlinear DAE systems. The investigated technique performs order-reduction of the differential equa-

tions and reduces the size and complexity of the algebraic equations. The procedure for reducing the differential equations can be performed by balancing or POD, while a variety of identified model structures can be used for simplification of the algebraic equations. During the procedure, the interplay between the states and the algebraic variables has been taken into account as it is reflected in the computed projections.

The procedure has been illustrated by applying it to a model of a distillation column. A comparison of the results obtained from POD and balancing indicates that balancing performs better than POD since the input-to-state and the state-to-output behaviors are simultaneously taken into account while POD only uses the input-to-state behavior. The algebraic equations were further reduced by identifying a feedforward neural network resulting in a model of significantly smaller size that is also easier to simulate since the algebraic variables can be computed via an explicit expression.

CHAPTER IV

DAE MODEL REDUCTION VIA BALANCING AND PARTIAL LEAST SQUARES (PLS)

In the previous chapter, a technique combining projections and neural network was presented for DAE model reduction. The neural network was applied to approximate a nonlinear static relationship between differential states and algebraic variables. This operation can achieve a good approximation so that the reduced model has a satisfactory performance. Due to the nonlinearity nature, the neural network may be hard to train when the system is large scale. For those systems with a low to medium degree of nonlinearity and relatively larger size, some linear methods, e.g. partial least squares (PLS) as introduced by Wold [86], can also provide good performance for multivariate approximations [87].

In this chapter, a combination of balancing and PLS is investigated for nonlinear model reduction of DAE systems. Also, this method is studied on the residualization model derived from large scale ODE systems, which is actually a special DAE system. By combining a model-based method, i.e. nonlinear balancing, with a linear statistical approach, i.e. PLS, it is possible to reduce both size and complexity of a strongly nonlinear model while retaining the control-relevant input-output behavior.

A. Model Reduction via Balancing/PLS for ODE Systems

1. Model Reduction Procedure

As shown in Chapter II, reduced systems derived from balanced truncation cannot retain the steady state behavior, therefore, balanced residualization methods have been introduced. Balanced residualized models described by (2.12) can be rewritten

as

$$\begin{aligned} \dot{\bar{x}}_1 &= \bar{f}_1(\bar{x}_1, \bar{x}_2, u) \\ 0 &= \bar{f}_2(\bar{x}_1, \bar{x}_2, u) \\ y &= \bar{h}(\bar{x}_1, \bar{x}_2) \end{aligned} \tag{4.1}$$

While a residualized system will have better steady-state performance than a truncated system, it is only a reduced model in the sense that some differential equations have been replaced by algebraic ones, resulting a differential algebraic equation (DAE) system, where the retained states $\bar{x}_1$ are described by differential equations while the reduced states $\bar{x}_2$ can be represented as a vector of algebraic variables. A DAE solver is required to solve this type of model. However, solving a DAE system often requires a larger computational effort than solving an ODE system with the same number of states, which would defeat the purpose of nonlinear model reduction. Therefore, further reduction may be required for nonlinear balanced residualization.

The vector $\bar{x}_1$ from (4.1) contains states that are the most important ones for the input-output behavior of the system and $\bar{x}_2$ is only used as a correction term for the steady state behavior. As $\bar{x}_2$ only contributes to a lesser degree to the process' behavior it can be further reduced without significantly affecting the input-output behavior of the system.

Consider the implicit algebraic equation $0 = \bar{f}_2(\bar{x}_1, \bar{x}_2, u)$ in (4.1). $\bar{x}_2$ is dependent on $\bar{x}_1$ and on the input u in a static manner. Since $\bar{f}_2$ is nonlinear, it is hard to obtain an explicit expression for $\bar{x}_2$. However, a system identification technique can be applied to approximate $\bar{x}_2$ given $\bar{x}_1$ and u

$$\bar{x}_2 = \hat{g}(\bar{x}_1, u) \tag{4.2}$$

where $\hat{g}$ represents an identified expression, e.g. an artificial neural network (ANN) or a partial least squares (PLS) model. In this work, PLS is used to obtain this

approximation, as it is often sufficient to represent these states of lesser importance by a linear approximation. The reduced system is given by

$$\begin{aligned} &\dot{\bar{x}}_1 = \bar{f}_1(\bar{x}_1, \bar{x}_2, u) \\ &\bar{x}_2 \text{ is predicted by PLS using } (\bar{x}_1, u) \text{ as inputs} \\ &y = h(\bar{x}_1, \bar{x}_2) \end{aligned} \tag{4.3}$$

If the number of states to be reduced is large, then it can happen that the large number of resulting algebraic equations can cause numerical problems. However, as many of the algebraic variables do not contribute to the control-relevant behavior, it is possible to use a combination of residualization and truncation to address this situation and compute a reduced-order model of the following form:

$$\begin{aligned} \dot{\bar{x}}_1 &= P_1 T f(T^{-1}\bar{x}, u) \\ 0 &= P_2 T f(T^{-1}\bar{x}, u) \\ \bar{x}_3 &= \bar{x}_{3,ss} \\ y &= h(T^{-1}x) \end{aligned} \tag{4.4}$$

where $\bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \end{bmatrix}$, $P_1 = [\ I_{k\times k} \quad 0 \quad 0\ ]$, $P_2 = [\ 0 \quad I_{m\times m} \quad 0\ ]$, k: number of retained differential equations, m: number of retained algebraic variables

Following the same approximation procedure as in (4.2) and (4.3), the reduced system can be described by

$$\begin{aligned} &\dot{\bar{x}}_1 = \bar{f}_1(\bar{x}_1, \bar{x}_2, u) \\ &\bar{x}_2 \text{ is predicted by PLS using } (\bar{x}_1, u) \text{ as inputs} \\ &\bar{x}_3 = \bar{x}_{3,ss} \\ &y = h(\bar{x}_1, \bar{x}_2) \end{aligned} \tag{4.5}$$

Note that in the approximations (4.3) and (4.5), $\bar{x}_2$ also depends on the input u and, therefore, the independent matrix consists of $\bar{x}_1$ and u.

2. Implementation of PLS

To compute a PLS model which approximates $\bar{x}_2$ given values of $\bar{x}_1$ and u, the independent matrix is required to contain $\bar{x}_1$ and input u, and the dependent matrix consists of $\bar{x}_2$. The training data set can be collected by the following procedure:

Step 1: Simulate the original model

Uniformly distributed random perturbations of the inputs u are used to excite the original model starting from the steady state operating point. The values of the inputs and the states are recorded along each trajectory. For convenience of notation, the data matrix of states is represented by A and the input matrix is referred to as B in the following.

Step 2: Obtain the training data

Since $\bar{x}_1$ and $\bar{x}_2$ are states/variables of the transformed system, the data collected in step 1 also need to be transformed:

$$\bar{A} = TA \tag{4.6}$$

where T is the transformation matrix.

If the number of retained states is chosen to be equal to k, then the first k rows of $\bar{A}$ correspond to the data for $\bar{x}_1$, denoted as $\bar{X}_1$ and the following $(n-k)$ rows in (2.12) or m rows in (4.4) are data for $\bar{x}_2$, denoted by $\bar{X}_2$.

The independent matrix X and dependent matrix Y before scaling are con-

structed as in (4.7), respectively.

$$X = \begin{bmatrix} \bar{X}_1 \\ B \end{bmatrix} \quad Y = [\bar{X}_2] \tag{4.7}$$

Step 3: Scale the training data

X and Y are scaled to have zero mean and unit variance as shown in (2.19). For convenience of notation, the scaled matrices are still denoted as X and Y in the following.

After obtaining the training data set, the NIPALS algorithm is applied to construct the PLS model including the regression matrix, referred to as R. By use of the regression matrix, the reduced systems corresponding to (4.3) and (4.5) are given by (4.8) and (4.9), respectively.

$$\begin{aligned} \dot{\bar{x}}_1 &= \bar{f}_1(\bar{x}_1, \bar{x}_2, u) \\ \bar{x}_2 &= R * [\bar{x}_1, u] \\ y &= h(\bar{x}_1, \bar{x}_2) \end{aligned} \tag{4.8}$$

$$\begin{aligned} \dot{\bar{x}}_1 &= \bar{f}_1(\bar{x}_1, \bar{x}_2, u) \\ \bar{x}_2 &= R * [\bar{x}_1, u] \\ \bar{x}_3 &= \bar{x}_{3,ss} \\ y &= h(\bar{x}_1, \bar{x}_2) \end{aligned} \tag{4.9}$$

B. Model Reduction via Balancing/PLS for DAE Systems

1. Model Reduction Procedure

As in Chapter III, the DAE systems can be reduced as (3.22)

$$\begin{aligned}
\dot{\bar{x}}_1 &= P_1 \bar{f}(\bar{x}, \bar{z}, u) = \hat{f}(\bar{x}_1, \bar{z}_1, u) \\
\bar{x}_2 &= \bar{x}_{2,ss} \\
\bar{z}_1 &= \tilde{g}(\bar{x}_1) \\
\bar{z}_2 &= \bar{z}_{2,ss} \\
y &= \bar{h}(\bar{x}) = \hat{h}(\bar{x}_1)
\end{aligned} \tag{4.10}$$

where $\hat{f}, \quad \tilde{g}, \quad \hat{h}$ represent nonlinear functions of the reduced system.

Instead of using neural network to do the system identification, here, the relationship between $\bar{z}_1$ and $\bar{x}_1$ can be approximated by PLS

$$\begin{aligned}
&\dot{\bar{x}}_1 = \hat{f}(\bar{x}_1, \bar{z}_1, u) \\
&\bar{z}_1 \text{ is predicted by PLS using } \bar{x}_1 \text{ as input} \\
&y = \hat{h}(\bar{x}_1)
\end{aligned} \tag{4.11}$$

Note that $\bar{z}_1$ only depends on $\bar{x}_1$ and does not depend on the input u as the original system was a regular DAE system. Therefore, the input u does not need to be included in the independent matrix when computing the PLS model for DAE systems.

2. Implementation of PLS

The procedure to implement PLS for model reduction of the alebraic equations of a DAE system is similar to the one for ODE systems. Only minor modifications need to be performed for DAE systems for obtaining the training data set.

The state matrix A and input matrix B can be collected by simulating the original DAE system. Note that the matrix A not only consists of the matrix for the states but also includes the algebraic variables. The matrix A can be split into block matrices

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \tag{4.12}$$

where A_1: data for states, A_2: data for algebraic variables

The transformed data matrices can be computed by:

$$\bar{A}_1 = T_1 A_1, \quad \bar{A}_2 = T_2 A_2 \tag{4.13}$$

If the number of retained states is k, then the first k rows of $\bar{A}_1$ are the data matrix of $\bar{x}_1$, denoted as $\bar{X}_1$. Similarly, if the number of retained algebraic variables is m, then the first m rows of $\bar{A}_2$ correspond to the data matrix of $\bar{z}_1$, denoted as $\bar{Z}_1$. $\bar{X}_1$ and $\bar{Z}_1$ are in fact the independent matrix X and dependent matrix Y before scaling is applied. The final reduced system is given by (4.14) where the regression matrix R is computed via PLS.

$$\begin{aligned} \dot{\bar{x}}_1 &= \hat{f}(\bar{x}_1, \bar{z}_1, u) \\ \bar{x}_2 &= \bar{x}_{2,ss} \\ \bar{z}_1 &= R\bar{x}_1 \\ \bar{z}_2 &= \bar{z}_{2,ss} \\ y &= \hat{h}(\bar{x}_1) \end{aligned} \tag{4.14}$$

where $\bar{x} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix}$, $\bar{z} = \begin{bmatrix} \bar{z}_1 \\ \bar{z}_2 \end{bmatrix}$

C. Case Studies

Two examples are presented in this section to illustrate the use of PLS for nonlinear model reduction. One example is a catalytic fixed-bed reactor, described by 120 ODEs while the other example is a distillation column, consisting of 32 differential equations and 32 implicit algebraic equations.

1. Example 1: Catalytic Fixed-Bed Reactor

The reactor system is a multi-tubular reactor with a highly exothermic reaction that synthesizes phthalic anhydride from *o*-xylene [88]. Two partial differential equations are used to describe the mass and energy balances along the length of the reactor. A finite difference method is applied to discretize the partial differential equations into a set of ODEs consisting of 120 states, 60 of which describe the concentrations and 60 represent the temperatures at the discretization points. The steady state concentration and temperature distribution along the reactor length is shown in Fig. 8. It can be observed that there is a "hot spot" in the reactor, which is commonly found in exothermic processes. As the temperature at this hot spot is of primary interest, a measurement is located at the position where the hot spot occurs. The inlet temperature serves as the input variable.

a. Data Set Collection

The reduced reactor system is described by (4.4) and (4.5). It has been determined from the magnitude of the singular values that 25 states are sufficient for a good approximation of the original systems, i.e., $k = 25$. Therefore, the reduced system consists of 25 differential equations. Ten additional states are approximated via PLS, i.e., $m = 10$, to obtain a good approximation at steady state. The other 85

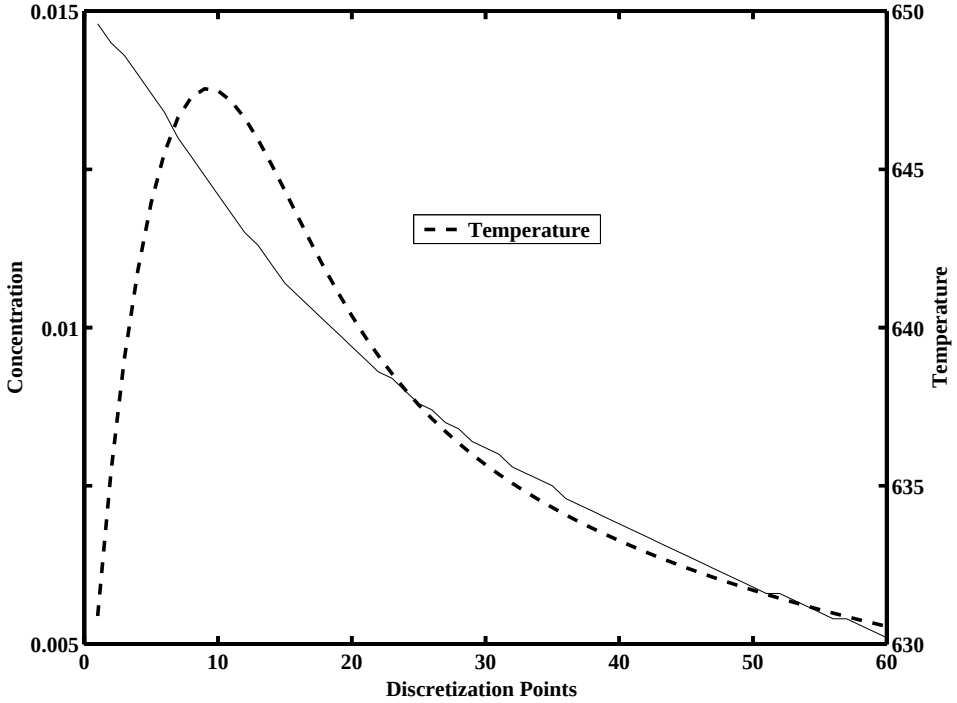

Fig. 8. Steady state of reactor.

states are truncated, i.e., kept as constants. Due to the high degree of nonlinearity for the reactor model, only perturbations of the input up to ±5% are acceptable for the stability of the operating point. The training data set is collected and preprocessed as described in subsection A-2.

b. PLS Implementation

After the PLS model is obtained by use of the training data set, this model is evaluated on test data which is generated by exciting the system with a series of random input perturbations. The prediction error of the PLS model, i.e., the error between the original value of one dependent variable and the corresponding predicted value, is shown in Fig. 9. Note that this dependent variable corresponds to a variable of the transformed system. The upper graph in Fig. 9 is the absolute value and

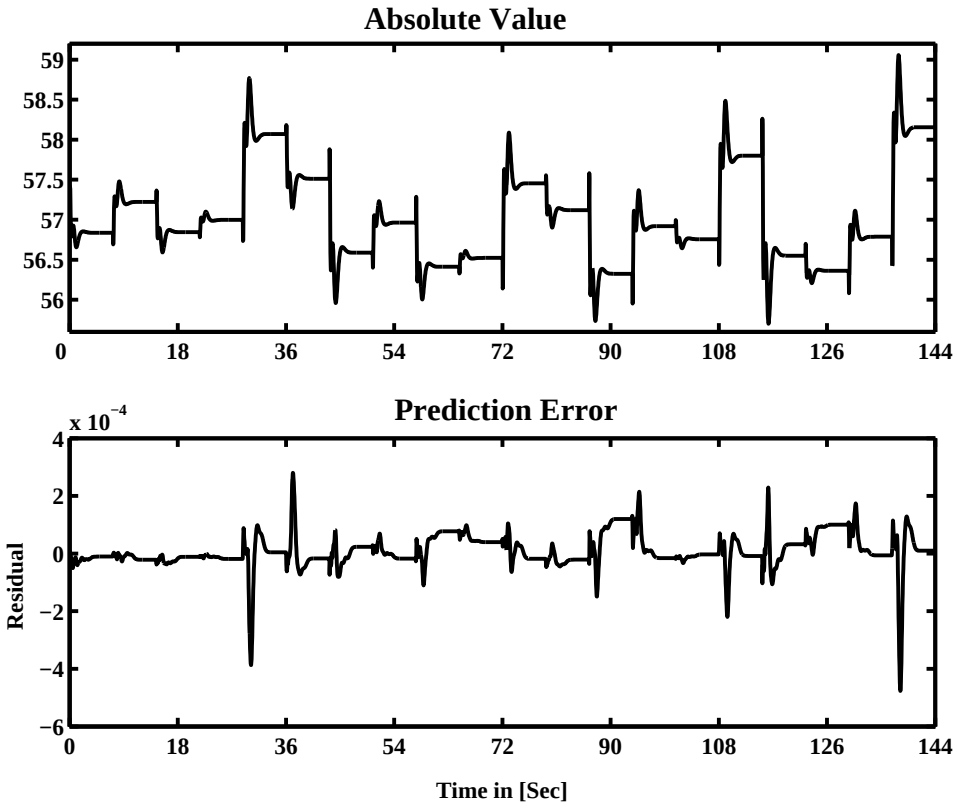

Fig. 9. Prediction performance of PLS model for reactor system.

the lower graph shows the prediction error. Based on the small prediction error, it can be concluded that the PLS model provides a good fit and can be used for model reduction of the reactor system.

c. Performance of Reduced System

To show the performance of the reduced system via the presented technique, a comparison between the reduced system derived from truncation (model 1) and a reduced system resulting from residualization via PLS (model 2) is given in Fig. 10. In this figure, the step responses of these reduced systems and the original system with an input perturbation of 3% are compared. The graph in the upper subplot depicts the outputs of the original system and the lower graphs represent the residuals

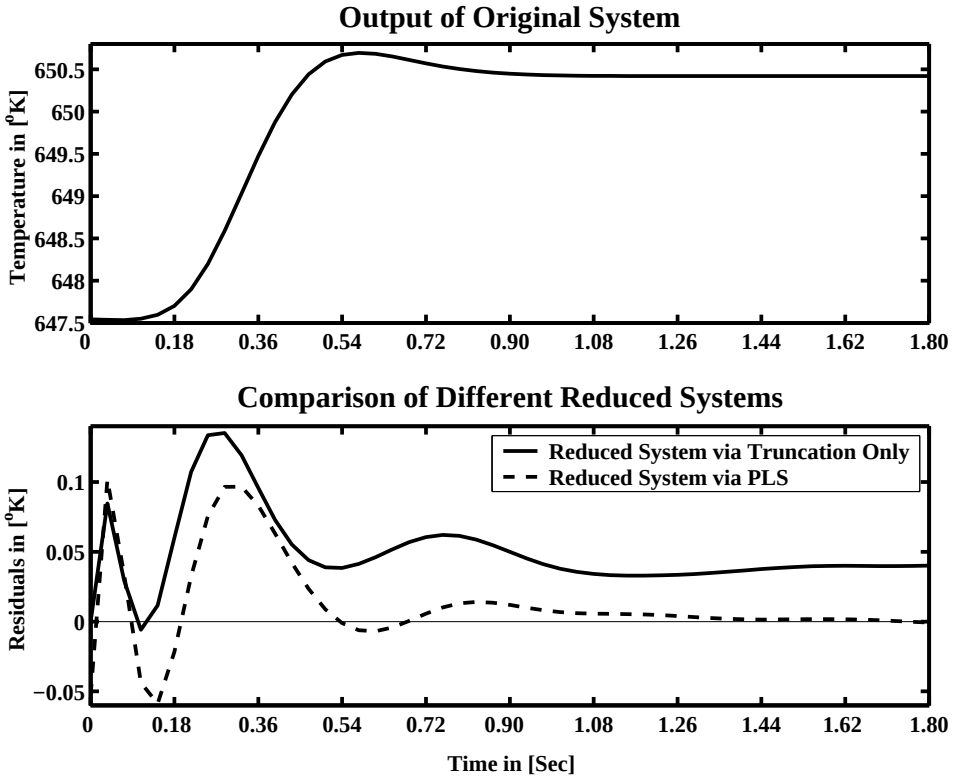

Fig. 10. Comparison of reduced reactor systems' step responses.

between this output and the outputs of the reduced systems. It can be seen that the performance of model 2 is much better than that of model 1, especially with regard to the steady state behavior. These results illustrate that while PLS is a linear system identification technique, it can return excellent results if it is included in a model reduction procedure that combines PLS and nonlinear balancing.

2. Example 2: Distillation Column

The distillation column model investigated in Chapter III is revisited here, which is a DAE system consisting of 32 differential equations and 32 algebraic equations.

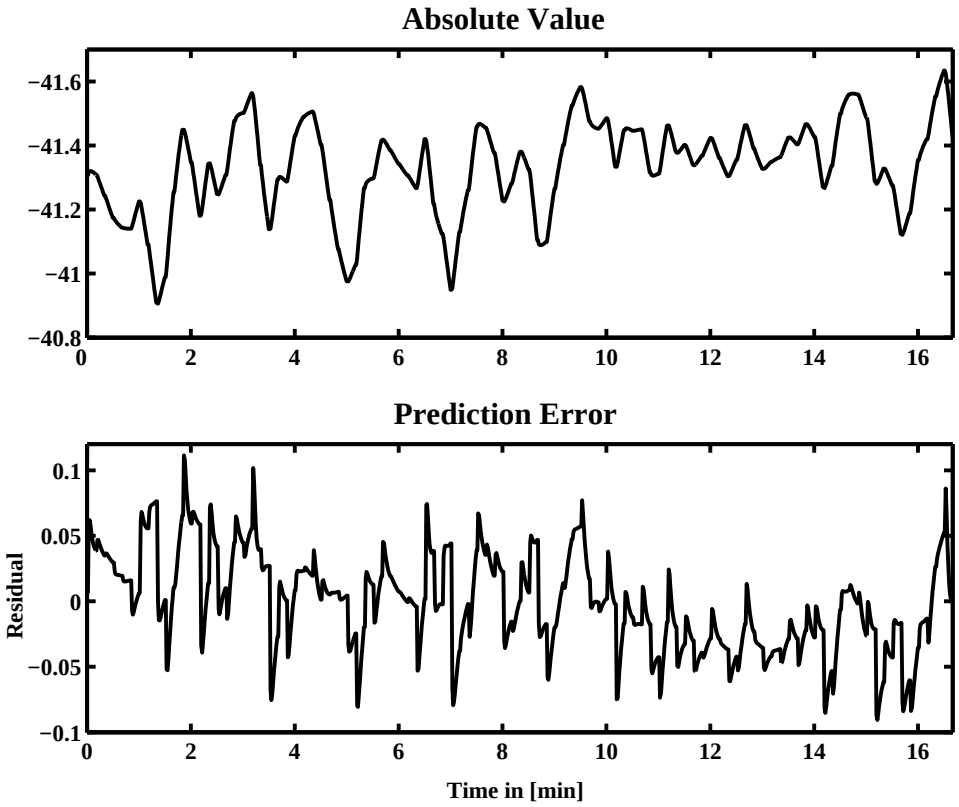

Fig. 11. Prediction performance of PLS model for distillation system.

a. Data Set Collection

Based on the singular values it is determined that 3 states are sufficient for approximating the 32 differential equations and 3 algebraic variables are retained, i.e., $k = m = 3$. The data set is collected by exciting the distillation column with a series of input perturbations of up to $\pm 15\%$.

b. PLS Implementation

Similar to the reactor model, the prediction error is shown in Fig. 11. This error is sufficiently small so that the PLS model can be used to predict the retained algebraic variables from the values of the states of the reduced-order model.

c. Performance of Reduced System

Given a -10% input perturbation, the step responses of three systems are compared: 1) a reduced system where only the differential equations are reduced via balanced truncation (model 1); this method is computationally efficient but will have steady state offset; 2) a reduced system where only the differential equations are reduced via balanced residualization (model 2); the procedure will result in no steady state offset but is computationally not the most efficient; 3) a reduced system where both differential and algebraic equations are reduced, and the retained algebraic variables are approximated by PLS (model 3). A comparison is shown in Fig. 12. The performance for model 3 is significantly better than that of model 1 and comparable with model 2. Moreover, a comparison of the computation times is provided in Table III, where the computational effort of all of the reduced systems is significantly smaller than for the original system (model 4). Additionally, model 3 is easier to solve than model 2 while it has a comparable degree of accuracy. It can be concluded that a combination of PLS and balancing performs very well for model reduction of DAE systems and represents a good trade-off between accuracy and the computational effort required for its solution.

Table III. Comparison of CPU times for reduced-order models

Model	1	2	3	4
Time (Seconds)	0.13	0.23	0.15	0.43

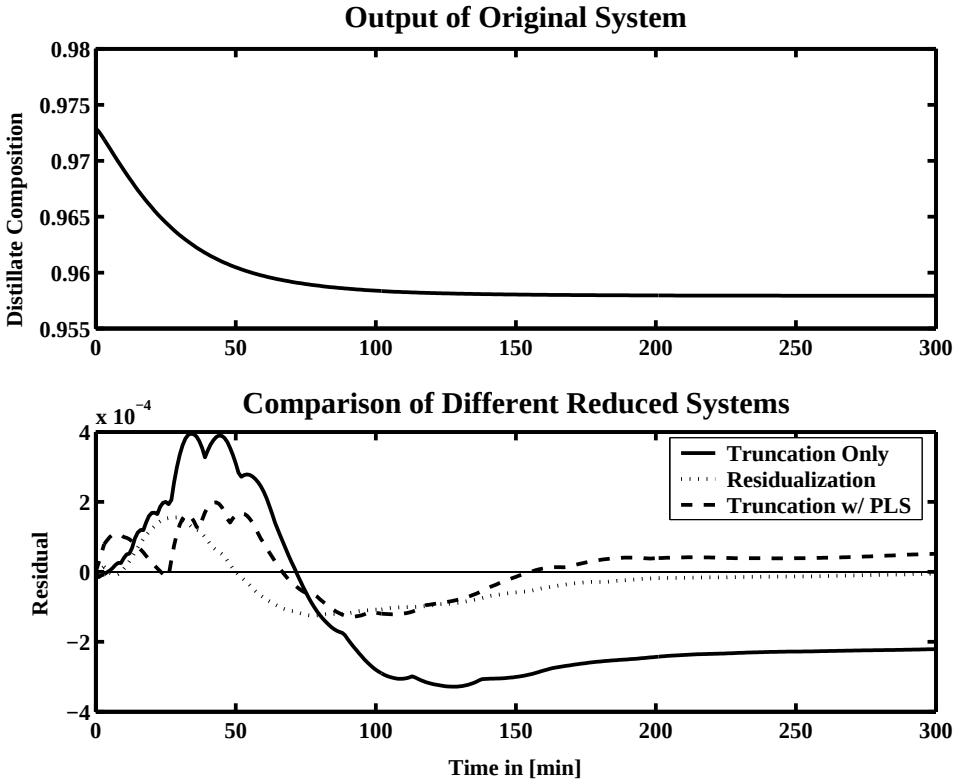

Fig. 12. Comparison of reduced distillation systems' step responses.

D. Summary and Discussion

This chapter presents a new technique for nonlinear model reduction by using a combination of nonlinear balancing and partial least squares (PLS). The reduction procedure can be applied for model reduction of nonlinear ODE and DAE systems. The technique has been illustrated in two case studies, where the reduced-order models were significantly smaller and faster to solve than the original model, while they provide an excellent approximation to the input-output behavior of the original system.

CHAPTER V

PARAMETER REDUCTION FOR STABLE DYNAMICAL SYSTEMS BASED ON HANKEL SINGULAR VALUES AND SENSITIVITY ANALYSIS

Chemical process models consist of not only process variables but also process parameters. The previous work on model reduction is mainly focusing on reducing the number of process variables and little attention has been given to reducing the number of parameters. This chapter addresses the problem of reducing insignificant or redundant information in parameter sets associated with fundamental models of dynamical systems. Whilst these parameters are important for describing physical or chemical relationships between the process variables, some of them only provide a marginal contribution to the input-output behavior of the system. Consequently, the identification of the latter parameters is difficult on the basis of recorded process data. Since this important issue has not attracted considerable attention over the past decade, this part of work introduces and evaluates three techniques to assess the importance of each parameter to the overall input/output description of the process under study.

A. Overview

Chemical processes usually contain a large number of parameters, all of which are only known to a certain degree. However, it is not always necessary to identify the values of all of the parameters from data for building a model to be used for monitoring and control. Instead, a reduction of the parameter set can be performed such that

---

Part of this chapter is reprinted from *Chemical Engineering Science*, Vol. 61, No. 16, C. Sun and J. Hahn, Parameter reduction for stable dynamical systems based on Hankel singular values and sensitivity analysis, pp. 5393-5403, Copyright (2006), with permission from Elsevier.

only a limited number of them need to be estimated. This paper addresses this point and presents three new techniques for reducing the parameter set of fundamental models.

The first methodology is based upon identifying the contribution that a parameter has to the input-output behavior of a system. The importance of each parameter is measured by the Hankel singular values of the balanced system if the parameter is used as input. Explicit bounds on the strongest contribution that a parameter has, even if it is time-varying, can be computed for the linear case and used as a measure for reducing/retaining a specific parameter. The advantages that this technique has are that (a) the physical meaning of the parameters is retained in the reduction and (b) explicit bounds for the contribution of a parameter to the system behavior can be computed for linear systems. The disadvantage is that this method can be very conservative.

The second technique is based upon determining "interactions" between parameters via singular value decomposition of a sensitivity covariance matrix. Unlike the first technique, the emphasis on the reduction for this method is on reducing the dimension of the parameter space of the system. The result can be a significantly smaller parameter set to be retained in the model. The disadvantage is that the physical interpretation of the parameters is lost during the procedure.

As the first methodology can be very conservative, while the second technique looses the physical interpretation of the parameters, a combination of the two techniques is presented. In a first step the contribution of individual parameters to the system behavior is evaluated. Only the parameters that have significant contributions will then be used for reduction of the parameter space. This combination allows that many parameters can still retain their physical interpretation, while at the same time a reduction of the parameters comparable to that of the second method can be

achieved.

B. Parameter Reduction for Linear Systems

This section presents three new approaches for reducing the parameter set of models. The advantages and drawbacks of each technique are discussed in detail.

The state space realization of a stable linear system with parameters can be expressed as

$$\begin{aligned} \dot{x} &= Ax + A_\theta \theta + Bu \\ y &= Cx + C_\theta \theta + Du \end{aligned} \tag{5.1}$$

where $x \in \Re^n$ refers to a vector of the states of the system, $u \in \Re^l$ is a vector of inputs, $y \in \Re^m$ represents a vector containing the measured variables, and $\theta \in \Re^s$ refers to the parameters of the system. Without loss of generality, suppose that all variables have been normalized. Also, assume that the eigenvalues of the matrix A have negative real parts resulting in a stable system.

1. Method I: Parameter Reduction Based on Hankel Singular Values

As the value of parameters can lie anywhere within their uncertainty intervals, or may in some cases even change with time within this interval, it is possible to treat the parameters as additional inputs to the system. As such they influence the input-output behavior of a system.

In (5.1), there are s parameters θ_j, $j = 1, \ldots, s$, which allows to rewrite (5.1) as

$$\begin{aligned} \dot{x} &= Ax + \sum_{j=1}^{s} A_{\theta,j}\theta_j + Bu \\ y &= Cx + \sum_{j=1}^{s} C_{\theta,j}\theta_j + Du \end{aligned}, \qquad \begin{aligned} A_\theta &= \begin{bmatrix} A_{\theta,1} & A_{\theta,2} & \cdots & A_{\theta,s} \end{bmatrix} \\ C_\theta &= \begin{bmatrix} C_{\theta,1} & C_{\theta,2} & \cdots & C_{\theta,s} \end{bmatrix} \end{aligned}, \tag{5.2}$$

For linear systems, the contributions of inputs and each parameter for input

output behavior can be superposed. Therefore, (5.2) is equivalent to a series of state space representations

$$\begin{aligned} \dot{x}_1 &= Ax_1 + Bu \\ y_1 &= Cx_1 + Du \end{aligned} \tag{5.3}$$

$$\begin{aligned} \dot{x}_2 &= Ax_2 + A_{\theta,1}\theta_1 \ , \\ y_2 &= Cx_2 + C_{\theta,1}\theta_1 \end{aligned} \tag{5.4}$$

$$\vdots$$

$$\begin{aligned} \dot{x}_{s+1} &= Ax_{s+1} + A_{\theta,s}\theta_s \\ y_{s+1} &= Cx_{s+1} + C_{\theta,s}\theta_s \end{aligned} \tag{5.5}$$

where $x_i, i = 1, \cdots, s+1$ represents the state vector resulting from the inputs or one parameter at a time, and $y_i, i = 1, \cdots, s+1$ refers to the corresponding output vector.

This decomposition is shown in Fig. 13. It follows that $x = \sum_{i=1}^{s+1} x_i$ and $y = \sum_{i=1}^{s+1} y_i$. Therefore, the system (5.1) can be regarded as a sum of these s+1 subsystems. For each subsystem, the controllability gramian $W_{C,i}$, $i = 1, \ldots, s+1$, can be computed. $W_{C,1}$ corresponds to the controllability gramian representing the input-to-state behavior for excitation with the inputs u (subsystem (5.3)). $W_{C,2}$ to $W_{C,s+1}$ on the other hand contain information about the input-to-state behavior if the system is only excited by variations in one of the parameters at a time (subsystems (5.4)-(5.5)). Based on (2.2), the controllability gramian for the original system, W_C, is a sum of the controllability gramian for each subsystem, $W_{C,i}$,$i = 1, \ldots, s+1$, if the parameters are also viewed as inputs to the system

$$W_C = \sum_{i=1}^{s+1} W_{C,i} \tag{5.6}$$

Since the observability gramian W_O, does not depend upon the input or the

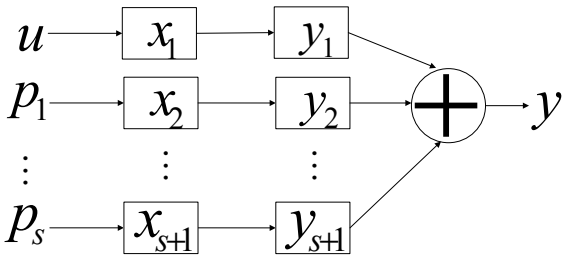

Fig. 13. System decomposition diagram.

parameters, the observability gramian of the original system is the same as that of each subsystem. Therefore, the observability gramian only needs to be computed once for this investigation.

The Hankel singular values for each subsystem can be computed based on $W_{C,i}$,$i = 1, \ldots, s{+}1$, and W_O. These Hankel singular values can be used to determine the contribution of individual states to the input-output behavior of each subsystem. Based upon the error bound for model reduction of a balanced system shown in (2.4), it can be concluded that the contributions of inputs to the input-output behavior are bounded by

$$\sigma_{1,u} \leq \|G_u(s)\|_\infty \leq 2\sum_{i=1}^{n} \sigma_{i,u} \tag{5.7}$$

where $G_u(s)$ denotes the difference between the original system and the system with the input removed from the original system, $\sigma_{i,u}$refers to the i-th Hankel singular value of the subsystem (5.3).

Similar bounds can be established for the contribution of each parameter to the input-output behavior of the system

$$\sigma_{1,\theta_j} \leq \left\|G_{\theta_j}(s)\right\|_\infty \leq 2\sum_{i=1}^{n} \sigma_{i,\theta_j} \tag{5.8}$$

where $G_{\theta_j}(s)$ denotes the difference between the original system and the system with the parameter θ_j removed from the original system, σ_{i,θ_j}refers to the i-th Hankel singular value of each subsystem in (5.4)-(5.5).

It should be noted that the contribution that individual states make to the input-output behavior are not as important for this work as the overall contribution of inputs to the system. Accordingly, the error bounds presented in (5.7) and (5.8) refer to the error that will incur if the inputs or individual parameters are eliminated from the system. Following this argument, a parameter θ_j can be safely assumed to be constant if

$$\sigma_{1,u} >> 2\sum_{i=1}^{n}\sigma_{i,\theta_j} \tag{5.9}$$

The reason behind this argument is that the maximal error that is incurred by setting a parameter to its constant value is significantly smaller than the smallest error that can result from neglecting the effect that changes in the inputs have on the system. It should be noted that this is a very conservative condition. In most cases, the parameters that satisfy $\sigma_{1,u} \approx 2\sum_{i=1}^{n}\sigma_{i,\theta_j}$ can also be reduced. This point can be verified by checking the performance of the reduced system *a posteriori*.

Based upon the presented argument, it is possible to compute explicit upper and lower error bounds for reducing the parameter set. The overall error bound is not given by the sum of the individual bounds as parameter may interact with one another, resulting in an error bound which is likely smaller than the sum of the error bounds. Instead the upper and lower bounds can be computed as follows: Suppose there are k parameters reduced in a model. Without loss of generality, these parameters are referred to as $\{\theta_r,\, r = 1,\, 2,\, \ldots,\, k\}$. The error bound for neglecting

these parameters for the input-output behavior of the system can be given by

$$\sigma_{1,\,\theta_r} \leq \left\| G_{\theta_r}(s) \right\|_\infty \leq 2 \sum_{i=1}^{n} \sigma_{i,\,\theta_r} \tag{5.10}$$

where σ_{i,θ_r} represents the i-th Hankel singular values corresponding to a system excited by changes in these k parameters.

The strong points of this type of procedure for reducing the set of parameters are that

- The physical meaning of the parameters is retained during the procedure as it is apparent which parameters have been reduced based on the corresponding Hankel singular values.
- Explicit error bounds can be given for the effect that neglecting changes in the parameters have on the input-output behavior of the system.
- The computed error bounds are not just valid for the steady-state behavior, but will also hold if the parameters vary with time within their uncertainty range.

A drawback of this method is that it can be very conservative, especially if the condition given by (5.9) is used for determining which parameters to neglect.

2. Method II: Parameter Reduction Based on Sensitivity Analysis

a. Sensitivity Covariance Matrix

Define the sensitivity functions for the parameters: $x_\theta = \frac{\partial x}{\partial \theta} \in \Re^{n \times s}$, $u_\theta = \frac{\partial u}{\partial \theta} \in \Re^{l \times s}$, $y_\theta = \frac{\partial y}{\partial \theta} \in \Re^{m \times s}$. By differentiating (5.1), the parameter variation system can be obtained (Tomovic and Vukobratovic, 1972)

$$\begin{aligned} \dot{x}_\theta &= \tfrac{d}{dt}\left(\tfrac{\partial x}{\partial \theta}\right) = \tfrac{\partial}{\partial \theta}\left(\tfrac{dx}{dt}\right) = \tfrac{\partial}{\partial \theta}\left(\dot{x}\right) \\ &= \tfrac{\partial}{\partial \theta}\left(Ax + A_\theta \theta + Bu\right) = Ax_\theta + A_\theta + Bu_\theta \end{aligned} \tag{5.11}$$

$$y_\theta = \frac{\partial y}{\partial \theta} = \frac{\partial}{\partial \theta}(Cx + C_\theta \theta + Du) = Cx_\theta + C_\theta + Du_\theta \tag{5.12}$$

It is apparent that if the original system (5.1) is stable, then the system given by the parameter variation equation is also stable, since the location of the eigenvalues of the matrix A also determines stability of the system given by (5.11) and (5.12).

It is assumed that inputs do not depend on parameters, thus $u_\theta = 0$. Solving these differential equations with initial conditions of zero, the solution is given by [89]

$$x_\theta = -A^{-1}A_\theta + e^{At}A^{-1}A_\theta \tag{5.13}$$

$$y_\theta = -CA^{-1}A_\theta + Ce^{At}A^{-1}A_\theta + C_\theta \tag{5.14}$$

With a change of notation, (5.14) results in

$$y_\theta = y_{\theta,ss} + Ce^{At}M \tag{5.15}$$

where $M = A^{-1}A_\theta$ and$y_{\theta,ss} = -CA^{-1}A_\theta + C_\theta$ is the final steady state value of y_θ.

And the following quantity is defined for ease of notation

$$\bar{y}_\theta = y_\theta - y_{\theta,ss} = Ce^{At}M \tag{5.16}$$

The Sensitivity Covariance Matrix (SCM) can then be computed from the following expression

$$\begin{aligned} \int_0^\infty \bar{y}_\theta^T \bar{y}_\theta dt &= \int_0^\infty \left(Ce^{At}M\right)^T \left(Ce^{At}M\right) dt \\ &= M^T \left(\int_0^\infty e^{A^T t} C^T C e^{At} dt\right) M = M^T W_O M \in \Re^{s\times s} \end{aligned} \tag{5.17}$$

where W_o is the observability gramian.

This covariance matrix contains information about the influence the parameters have on the outputs of the system. It should be noted that it not only contains information about the effect of individual parameters on the system but also contains

information about "interactions" of parameters. Also, this quantity is related to the observability gramian of the linear system, W_o.

b. Parameter Reduction Procedure

The information contained in the SCM allows to extract the principle directions in the parameter space that will give the largest contribution to the parameter-output behavior. Extracting the most important directions in the parameter space, unlike the approach presented in Section 3.1, allows to take the "interactions" among the parameters into account, i.e. if the effect of changes in two parameters cannot be uniquely attributed to one of the two, and can result in a significantly reduced parameter set. This section describes how the information contained in the SCM can be extracted and interpreted.

Since the sensitivity covariance matrix defined in (5.17) is a positive semi-definite matrix, all eigenvalues are non-negative real numbers. Furthermore, due to its symmetry, this matrix can be diagonalized by an orthogonal matrix even if multiple eigenvalues exist [49]. Therefore, singular value decomposition can be applied to this sensitivity covariance matrix

$$\text{SCM} = T^T \Lambda T \tag{5.18}$$

where $T \in \Re^{s \times s}$, an orthogonal matrix; superscript ^{T} indicates the transpose; and $\Lambda = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq 0$ are eigenvalues of SCM.

Starting from (5.18), it can be shown that

$$\Lambda = (T^{-1})^T \text{SCM} (T^{-1}) \tag{5.19}$$

A new parameter space can be obtained by a coordinate transformation $\bar{\theta} = T\theta$, where $\bar{\theta}$ is a parameter vector in the new parameter space. Since T is invertible, $\theta = T^{-1}\bar{\theta}$. Substituting into (5.1), the system in the new parameter space is given by

$$\begin{aligned} \dot{x} &= Ax + \overline{A_\theta}\bar{\theta} + Bu \\ y &= Cx + \bar{C}_\theta\bar{\theta} + Du \end{aligned} \tag{5.20}$$

where $\overline{A_\theta} = A_\theta T^{-1}$, $\overline{C_\theta} = C_\theta T^{-1}$. In this system, $\bar{M} = A^{-1}\overline{A_\theta} = A^{-1}A_\theta T^{-1} = MT^{-1}$, where $\bar{M}$ is the quantity in the new system corresponding to M in (5.14).

The sensitivity covariance matrix for this system where the parameter space has been transformed is given by

$$\overline{\mathrm{SCM}} = \bar{M}^T W_O \bar{M} = (T^{-1})^T M^T W_O M T^{-1} = (T^{-1})^T \mathrm{SCM}(T^{-1}) \tag{5.21}$$

Compared to (5.19), it is obvious that $\overline{\mathrm{SCM}} = \Lambda$. Therefore, $\overline{\mathrm{SCM}}$ is a diagonal matrix where the diagonal entries are the eigenvalues of the SCM in a descending order. Since the diagonal entries of $\overline{\mathrm{SCM}}$ provide a measure of the importance of the corresponding parameters in the new parameter space, the parameters can be classified as belonging to an important and a less important category. The transformed parameter vector $\bar{\theta}$ can be partitioned into two parts, $\bar{\theta}_1$ and $\bar{\theta}_2$, as shown in (5.22). $\bar{\theta}_1$ represents the more important parameters, which should be retained during parameter reduction and $\bar{\theta}_2$are of lesser importance and will be reduced.

$$\bar{\theta} = \begin{bmatrix} \bar{\theta}_1 \\ \\ \bar{\theta}_2 \end{bmatrix} \tag{5.22}$$

One important task is the determination of the number of parameters in $\bar{\theta}_1$ and $\bar{\theta}_2$. It is possible to use a trial and error procedure where a trade-off between the number of parameters to be retained and the quality of the approximation is found.

A good rule of thumb is to determine if there exist gaps in the magnitude of the eigenvalues of the SCM and then test several responses where the cut-off between retained/reduced parameters is placed at these gaps. However, this can only be used as an *a priori* rule and in practice it is always necessary to evaluate the performance of the system of reduced parameters.

System (5.20) can be rewritten as

$$\begin{aligned} \dot{x} &= Ax + \begin{bmatrix} \overline{A_{\theta_1}} & \overline{A_{\theta_2}} \end{bmatrix} \begin{bmatrix} \bar{\theta}_1 \\ \bar{\theta}_2 \end{bmatrix} + Bu \\ y &= Cx + \begin{bmatrix} \bar{C}_{\theta_1} & \bar{C}_{\theta_2} \end{bmatrix} \begin{bmatrix} \bar{\theta}_1 \\ \bar{\theta}_2 \end{bmatrix} + Du \end{aligned} \tag{5.23}$$

where the parameters can be reduced by truncation of the parameter vector $\bar{\theta}$.

$$\begin{aligned} \dot{x} &= Ax + \bar{A}_{\theta_1}\bar{\theta}_1 + Bu \\ y &= Cx + \bar{C}_{\theta_1}\bar{\theta}_1 + Du \end{aligned} \tag{5.24}$$

c. Analysis of Error Bounds for Parameter Space Reduction

While the coordinate transformation of the parameters $\bar{\theta} = T\theta$ does not result in an error, the truncation (going from (5.23) to (5.24)) will result in the new system approximating the original one. The error resulting from parameter reduction can be quantified by the H^∞ norm as follows:

$$\left\|\bar{G}(s) - \bar{G}_r(s)\right\|_\infty = \lambda_{\max}^{\frac{1}{2}}\{[\bar{G}(j\omega) - \bar{G}_r(j\omega)]^H[\bar{G}(j\omega) - \bar{G}_r(j\omega)]\} \tag{5.25}$$

where

$$\begin{aligned}
&\bar{G}(j\omega) - \bar{G}_r(j\omega) = C(j\omega I - A)^{-1}[\ B \quad \overline{A_{\theta 1}} \quad \overline{A_{\theta 2}}\ ] + [\ D \quad \overline{C_{\theta 1}} \quad \overline{C_{\theta 2}}\ ] \\
&-\{C(j\omega I - A)^{-1}[\ B \quad \overline{A_{\theta 1}} \quad 0\ ] + [\ D \quad \overline{C_{\theta 1}} \quad 0\ ]\} \\
&= C(j\omega I - A)^{-1}[\ 0 \quad 0 \quad \overline{A_{\theta 2}}\ ] + [\ 0 \quad 0 \quad \overline{C_{\theta 2}}\ ]
\end{aligned} \tag{5.26}$$

and $\lambda_{\max}^{\frac{1}{2}}$ represents the square root of the largest eigenvalue.

Therefore, the truncation error results in

$$\begin{aligned}
&\left\|\bar{G}(s) - \bar{G}_r(s)\right\|_\infty = \lambda_{\max}^{\frac{1}{2}}\{[\bar{G}(j\omega) - \bar{G}_r(j\omega)]^H[\bar{G}(j\omega) - \bar{G}_r(j\omega)]\} \\
&= \lambda_{\max}^{\frac{1}{2}}\{[C(j\omega I - A)^{-1}[\ 0 \quad 0 \quad \overline{A_{\theta 2}}\ ] + [\ 0 \quad 0 \quad \overline{C_{\theta 2}}\ ]]^H \\
&\qquad [C(j\omega I - A)^{-1}[\ 0 \quad 0 \quad \overline{A_{\theta 2}}\ ] + [\ 0 \quad 0 \quad \overline{C_{\theta 2}}\ ]]\} \\
&= \lambda_{\max}^{\frac{1}{2}} \left\{ \begin{array}{lll} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & [C(j\omega I - A)^{-1}\overline{A_{\theta 2}} + \overline{C_{\theta 2}}]^H[C(j\omega I - A)^{-1}\overline{A_{\theta 2}} + \overline{C_{\theta 2}}] \end{array} \right\} \\
&= \lambda_{\max}^{\frac{1}{2}}\{[C(j\omega I - A)^{-1}\overline{A_{\theta 2}} + \overline{C_{\theta 2}}]^H[C(j\omega I - A)^{-1}\overline{A_{\theta 2}} + \overline{C_{\theta 2}}]\}
\end{aligned} \tag{5.27}$$

It should be noted that expression (5.27) has to be evaluated over the entire frequency range to determine the error bound, unlike the error bound for the first technique which was given by (5.10).

The strong points of this second technique are that

- it is possible to determine not just if a parameter has an important effect on the system response, but also if the effects of variations in several parameters can be uniquely attributed to the individual parameters.

- the reduced set of parameter can be significantly smaller than the original set.

The main drawback is that the physical meaning of the parameters is lost during this procedure.

3. Method III: Combination of Method I and II

As Methods I and II both have their advantages and drawbacks, it is possible to combine them to avoid some of the disadvantages. A third technique is proposed here which performs the following steps:

(1) The sensitivity of the outputs with respect to changes in the parameters is investigated via the Hankel singular value-based technique. This investigation will keep all the parameters at their nominal values if changes in them will not result in significantly different behavior of the outputs.

(2) Compute the SCM for the reduced set of parameters from (1) and perform parameter space reduction.

The advantages that this combination of the techniques offers are that, the physical meaning of all reduced parameters in step (1) are retained (as stated in the discussions for Method I), while it is possible to substantially reduce the number of parameters due to the reduction from step (2).

C. Extension to Nonlinear Systems

As any system is nonlinear to some degree, it is important to extend the parameter reduction techniques to nonlinear systems. While the extension itself is made in this work, no proofs are provided for the nonlinear case, as several of the arguments made in Section 3 are based upon existing Results from linear systems theory, e.g. the theory and application of nonlinear balancing is not at a state where a meaningful proof of the extension of the first method can be provided. Due to this the extension of the methods to nonlinear system can only be guaranteed to work locally and that the presented error bounds are not guaranteed to be valid for the nonlinear case over an operating region.

1. Method I: Parameter Reduction Based on Hankel Singular Values

Linear gramians are used to compute Hankel singular values for linear systems. In the case of nonlinear systems, covariance matrices [12] can be computed numerically based on the system trajectories. The controllability covariance matrices for inputs and each parameter are obtained by exciting the system with various input signals. To compute the observability covariance matrix, all inputs and parameters are set to their nominal values while each state is given an initial condition perturbation. The controllability and observability covariance matrix can be balanced [11] and the Hankel singular values corresponding to the system with inputs and excited by each parameter can be computed. The less important parameters can be reduced based on the same criterion described for linear systems.

2. Method II: Parameter Reduction Based on Sensitivity Analysis

In this subsection, the sensitivity analysis-based method for nonlinear systems is presented. The nonlinear systems under investigation are given by

$$\dot{x} = f(x, \theta, u) y = g(x, \theta, u) \tag{5.28}$$

with $x(0) = x_0$.

Similar to the linear systems, the parameter variation system is given by

$$\dot{x}_\theta = \frac{d}{dt}\left(\frac{\partial x}{\partial \theta}\right) = \frac{\partial}{\partial \theta}\left(\frac{dx}{dt}\right) = \frac{\partial}{\partial \theta}\left(\dot{x}\right) = \frac{\partial}{\partial \theta}\left(f(x, \theta, u)\right) = J_{f,x} x_\theta + J_{f,u} u_\theta + J_{f,\theta} \tag{5.29}$$

$$y_\theta = \frac{\partial y}{\partial \theta} = \frac{\partial}{\partial \theta}(g(x, \theta, u)) = J_{g,x} x_\theta + J_{g,u} u_\theta + J_{g,\theta} \tag{5.30}$$

where $J_{f,x}$, $J_{f,u}$, $J_{g,x}$, $J_{g,u}$ are Jacobian matrices of the vector functions f, g and with respect to the state and input vectors, $J_{f,p} J_{g,p}$ are Jacobian matrices of the vector functions f, g with respect to the parameter vector θ.

Since all Jacobian matrices are functions of x, u and θ, it is necessary for (5.28) and (5.29) to be solved simultaneously. The set of equations need to be integrated over the time interval [0 T], with initial conditions$x(0) = x_0$ and $x_\theta(0) = 0$ and u_θ held at zero.

The sensitivity covariance matrix can then be computed from

$$\text{SCM} = \int_0^\infty \bar{y}_\theta^T \bar{y}_\theta dt = \int_0^\infty (y_\theta - y_{\theta,ss})^T (y_\theta - y_{\theta,ss}) dt \tag{5.31}$$

After the sensitivity covariance matrix has been obtained, the parameter reduction procedure can be implemented similar to the linear system case: (1) singular value decomposition is applied to compute an invertible transformation matrix T; (2) the linear transformation relationship $\bar{\theta} = T\theta$ is introduced, such that the system in the new parameter space is given by

$$\begin{aligned} \dot{x} &= f(x, T^{-1}\bar{\theta}, u) \\ y &= g(x, T^{-1}\bar{\theta}, u) \end{aligned} \tag{5.32}$$

In this new parameter space, the parameters are sorted based on their relative importance, from large to small. Therefore, $\bar{\theta}$ can also be partitioned into two parts, similar to what was done in (5.22). The last step is to truncate the less important states $\bar{\theta}_2$ via a projection matrix P leading to the reduced system

$$\begin{aligned} \dot{x} &= f(x, PT^{-1}\bar{\theta}, u) \\ y &= g(x, PT^{-1}\bar{\theta}, u) \\ \bar{\theta}_2 &= \bar{\theta}_{2,\,ss} \end{aligned} \tag{5.33}$$

where $P = [\ I_{k\times k} \quad 0_{k\times(s-k)}\ ]$, k is the number of parameters retained.

3. Method III: Combination of Method I and II

The combination method is also applicable to nonlinear systems. The elements introduced in Section 4.1 and 4.2 can be combined in the same manner as described for linear systems.

D. Performance Evaluation

Evaluating the performance of systems where the parameter set has been reduced is an important step of the parameter reduction procedure. While the error bounds presented in Section 3 allow an *a priori* estimate of the error resulting from the procedure, it should be noted that these can be very conservative bounds. Additionally, these error bounds are not guaranteed to hold for nonlinear systems. To address these points, any parameter reduction procedure needs to include performance evaluation of the reduced system.

Two approaches are presented in this work for performance evaluation. One approach is based upon frequency response analysis, which is only applicable to linear systems. The other one analyzes responses in the time domain based on Monte Carlo simulations. This second method can be used for linear as well as for nonlinear systems.

1. Frequency Domain Analysis

Transfer functions can be used to describe the effect that changes in the inputs/parameters of the system have on the outputs. These transfer functions can be bundled in a matrix of transfer functions, where the original inputs and the parameters serve as the inputs of the transfer functions. The relative error due to parameter

reduction is then given by

$$\Delta = \frac{\|G(s) - G_r(s)\|_\infty}{\|G(s)\|_\infty} \tag{5.34}$$

The individual components of Eq. (5.34) can be evaluated by (Skogestad and Postlethwaite, 1996)

$$\|G(s)\|_\infty = \bar{\sigma}[G(j\omega)] \tag{5.35}$$

where $\bar{\sigma}[G(j\omega)]$ represents the largest singular value of $G(j\omega)$ over the entire spectrum of frequencies.

Therefore, once the reduced system is obtained, the singular values for both $G(j\omega)$ and $[G(j\omega) - G_r(j\omega)]$ can be computed. The relative error Δ can then be computed by evaluating the largest singular value of the numerator and the denominator in Eq. (5.34).

2. Time Domain Analysis Based on Monte Carlo Simulations

As frequency response analysis cannot be applied to nonlinear systems, a Monte Carlo-based technique [90] is presented here. This method is based upon varying the inputs and the parameters of the system, computing the trajectories of the outputs in the time domain, and comparing the results of the systems with reduced parameter sets to the behavior of the original process. A variety of different inputs and changes in the parameters has to be considered for this technique. This is achieved by randomly varying the inputs and the parameters within their bounds and computing a large number of trajectories (usually greater than 1000). The average error introduced by reduction of the parameter set can then be computed from this run.

Since Monte-Carlo-based techniques rely on stochastic variations of some of the variables/inputs, it is important to also determine confidence intervals for the average error introduced by the reduction procedure. This is done by repeating the Monte

Carlo simulations several times and determining a mean of the average error as well as a standard deviation of the average error

$$Err = \frac{\sigma}{Mean(|y|)} \tag{5.36}$$

where $\sigma = \sqrt{\frac{1}{N-1}\sum_{k=1}^{N}[(y(k) - y_r(k))]^2}$ is the standard deviation. $Mean(|y|) = \frac{1}{N}\sum_{k=1}^{N}|y(k)|$ is the mean of absolute value of the responses of the original system, y represents the output of the original system and y_r refers to that of the reduced system, and N is the sample length for one simulation.

Choosing the number of simulations used for each Monte Carlo simulation as well as the number of repeated Monte Carlo simulations is a key component influencing this evaluation method. In general, it can be said that if both numbers are sufficiently large then the standard deviation will be small.

A detailed procedure is as follows:

(1) Generate random values of inputs and the parameters;

(2) Excite the system and collect the system's trajectories;

(3) Compute the error as in (5.36);

(4) Repeat the above procedure some times, for instance, 5000 times. Then there are 5000 errors recorded.

(5) Choose some threshold values, for instance, 0.0005, 0.001, 0.005, 0.01, 0.05 (these values are used in this work).

(6) Compute the percentages that the errors are less than the threshold values, for example, if there are 3000 errors are less than 0.0005, then the percentage corresponding to the threshold value 0.0005 is 60%. For each threshold value, there is one percentage.

The above procedures are regarded as one monte carlo experiment.

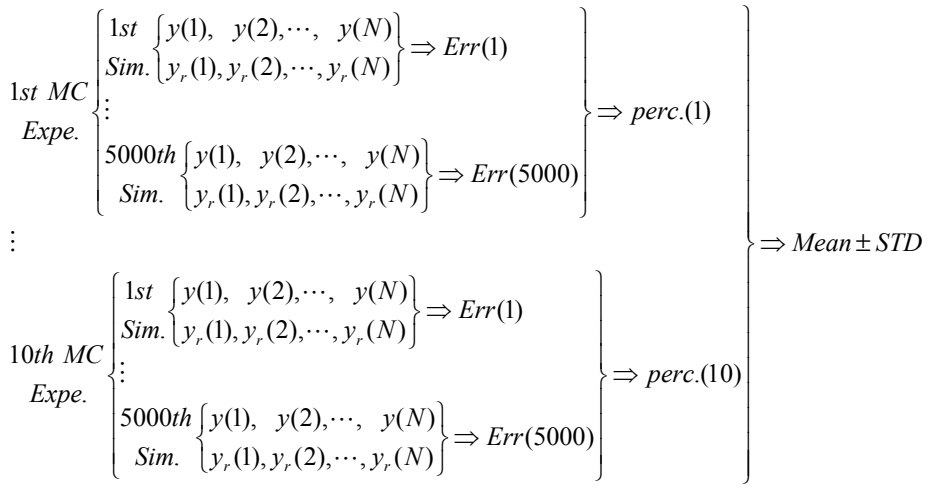

Fig. 14. Monte Carlo simulation method

(7) Do several such monte carlo experiment, for instance, 10, and recording the percentages for each threshold value in each experiment.

(8) For each threshold value, collect all the percentages in each experiment, compute the mean and standard deviation for these percentages, then the confidence interval can be expressed in the form of Mean ± Standard deviation.

This procedure is shown in Fig. 14.

E. Case Studies

Three examples are studied to illustrate the presented techniques. The first example is a randomly generated linear system where the performance is evaluated based upon frequency domain analysis. The second example is a system of two continuously stirred tank reactors in series, which is a system consisting of four nonlinear differential equations. A third model of a signal transduction model is also investigated in order to show that the presented methods are applicable to larger size systems. Monte Carlo simulations are used for performance evaluation for the latter two systems.

1. Example 1: A Randomly Generated Linear System

This example is used to illustrate the parameter reduction techniques for linear systems. This system is randomly generated with 1 input, 1 output, 40 states and 100 parameters. Results of parameter reduction are presented for all three methods.

For Method I, the Hankel singular values corresponding to each parameter and input can be computed. A comparison between the error bounds computed by twice the sum of total Hankel singular values for each parameter and the largest Hankel singular value of the input is shown in Fig. 15. In this figure, each stem represents the value of $2\sum_{i=1}^{n}\sigma_{i,p_j}$ for each parameter. The dashed line indicates a threshold value, $\sigma_{1,u}$. The parameters in the areas above the dashed line are important and should be retained. Those in the areas deep below the dashed line only have a very small contribution to the system behavior and are considered relatively unimportant. Those located in the middle area are to be further investigated. A trial and error procedure is necessary to determine how many parameters should be retained considering both performance and size of the parameter set. Based on the relative error Δ defined in Eq. (5.34), a preliminary criterion can be applied to determine the number of parameters to be retained. In order to illustrate this concept, the graph of the relative errors Δ for several reduced systems is shown in Fig. 16-a. This figure contains four plots, each of which represents one case: 10, 12, 15 or 20 parameters which are reduced. It can be observed that the more parameters are reduced, the larger the relative error becomes. If the relative error is assumed to be smaller than 0.15, then 20 parameters can be reduced, however, if more stringent condition needs to be satisfied, then fewer parameters should be reduced to guarantee the performance.

For Method II, the logarithm of the eigenvalues of the sensitivity covariance matrix is shown in Fig. 17. Based on the distributions of eigenvalues, the number

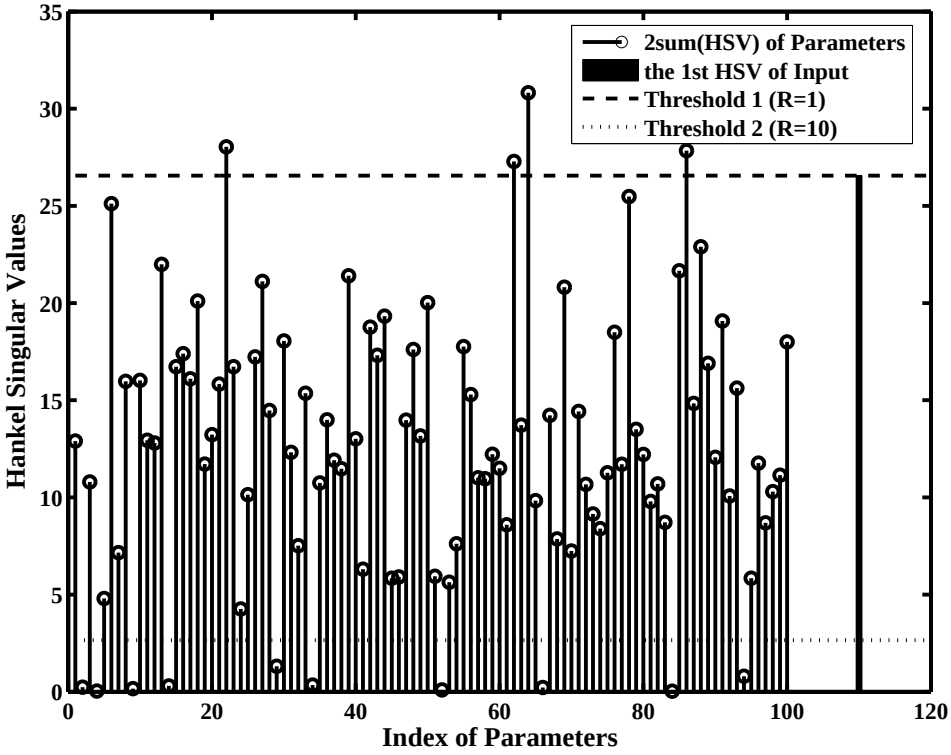

Fig. 15. Hankel singular values for parameters (Method I), for example 1.

of retained parameters can be roughly determined, to be in the range of 15 to 25. Similar to Method I, the relative errors for several reduced systems are shown in Fig. 16-b, with 85, 84, 82 and 78 retained parameters. Compared to Fig. 16-a, it can be concluded that the reduced system with 85 reduced parameters by use of the sensitivity analysis-based method can have a comparable performance to that of the reduced system by using Hankel singular values with only 12 reduced parameters. It should also be noted that the relative errors are smaller at low frequencies while much larger in high frequencies. This is important insofar as parameters usually only change slowly with time and retaining the low frequency behavior is more important than achieving a good approximation in the high frequency range. The overall performance of this method is very good for this example.

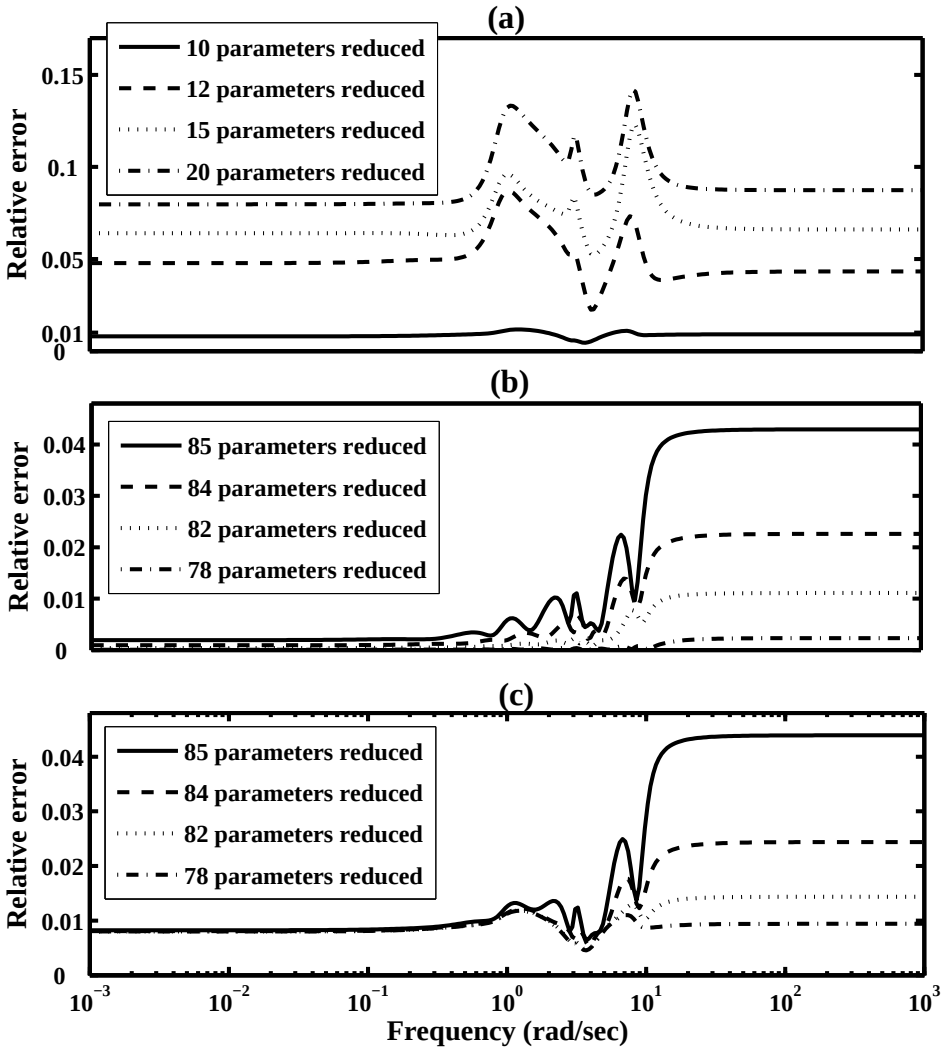

Fig. 16. Relative errors between the original system and reduced systems for all three methods, for example 1. (a) Method I; (b) Method II; (c) Method III.

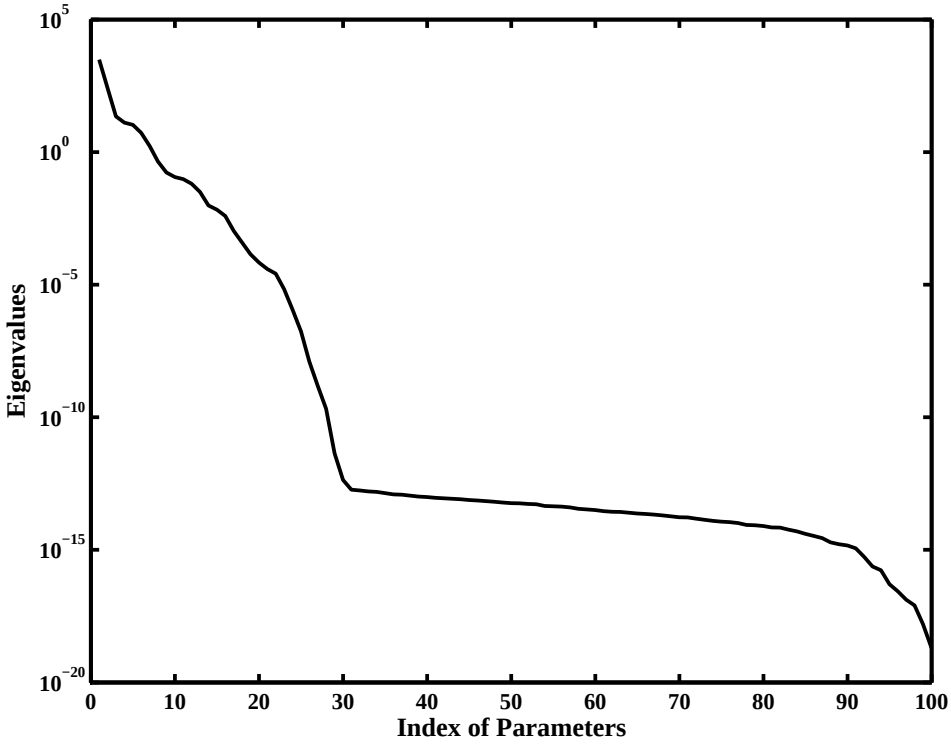

Fig. 17. Eigenvalues of sensitivity covariance matrix (Method II), for example 1.

The same guidelines are followed for Method III. In the initial screening step, only those parameters that satisfied the condition of $\sigma_{1,u} \Big/ 2\sum_{i=1}^{n} \sigma_{i,p_j} \geq 10$ are to be reduced. Thus an initially reduced system can be obtained with 10 parameters reduced (refer to Fig. 2-a). Parameter reduction via Method II is then performed on this reduced set of parameters. To give a comparison with Method II, the relative errors of reduced system are presented in Fig. 16-c. From Fig. 16-b and Fig. 16-c, it can be seen that the error resulting from the initial screening step is the main error resource, while in the high frequencies, the error resulting from the sensitivity analysis step is more important. Overally, if the same number of parameters reduced, the performance of Method III is a slightly worse than Method II for this example. However, by combining the two methods it is possible to keep the physical meaning

of at least some of the parameters intact.

2. Example 2: Two CSTRs in Series

This example deals with a system of two continuously stirred tank reactors in series which was studied in [91]. The model equations are shown in Appendix B. This system has one input, the coolant flow rate and one output, the effluent concentration from the second tank. This model contains four states, which are the temperatures and concentrations of the products in each tank. The system also contains 14 parameters,

$$\theta = [q, C_{Af}, T_f, T_{cf}, V_1, V_2, \rho, \rho_c, C_p, C_{pc}, -\Delta H, \frac{E}{R}, hA_1, hA_2]^T \tag{5.37}$$

some of which are physical constants, which depend upon the properties of the streams while some others are equipment related or even related to the operating conditions. The operation conditions and nominal values of parameters are given in (Henson and Seborg, 1990).

The three presented methods are implemented on this example problem. A comparison of Hankel singular values based upon covariance matrices computed for the system excited by the input and the parameters is shown in Fig. 18. In this figure, the sequence of parameters is consistent with that of the vector θ. Based on this figure, it can be concluded that the relative importance of the parameters is

$$\frac{E}{R} > T_f > T_{cf} > C_{Af} > -\Delta H > q > \rho = C_p > \rho_c = C_{pc} > V_1 > hA_1 > V_2 > hA_2 \tag{5.38}$$

It also can be seen that there are three parameters, hA_1, V_2 and hA_2, whose Hankel singular values are very small and can be neglected.

The sensitivity covariance matrix is obtained for method II based on the system trajectories. The eigenvalues of the sensitivity covariance matrix in descending order

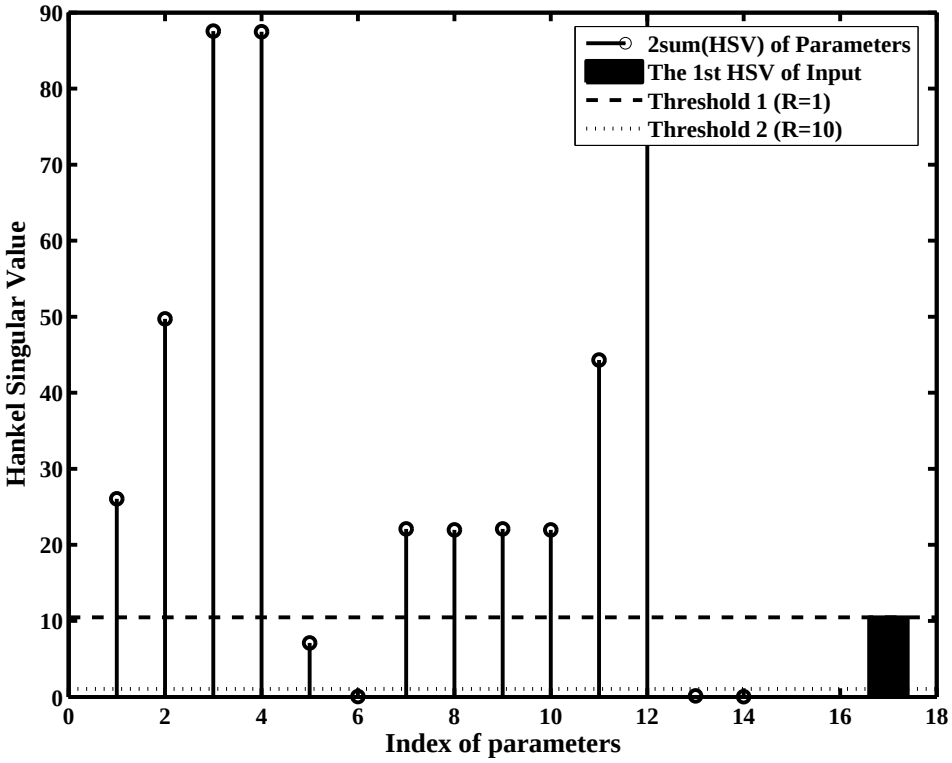

Fig. 18. Hankel singular values for parameters (Method I), for example 2.

are shown in Fig. 19.

For method III, the initial screening step, based on the Hankel singular values, determined that only hA_1, V_2 and hA_2 are to be reduced. They keep their nominal values in the later procedures. Parameter space reduction is then performed on the 11 remaining parameters.

Model performance evaluation for this nonlinear example is conducted in the time domain by use of Monte Carlo simulations. Each Monte Carlo experiment included simulating the system with variations in the inputs and the parameters 5,000 times. The variations of the inputs and parameters are generated randomly in their ranges based on a uniform distribution. The errors between the original system and reduced systems can be computed as defined in (5.36). Based upon the results for 5000

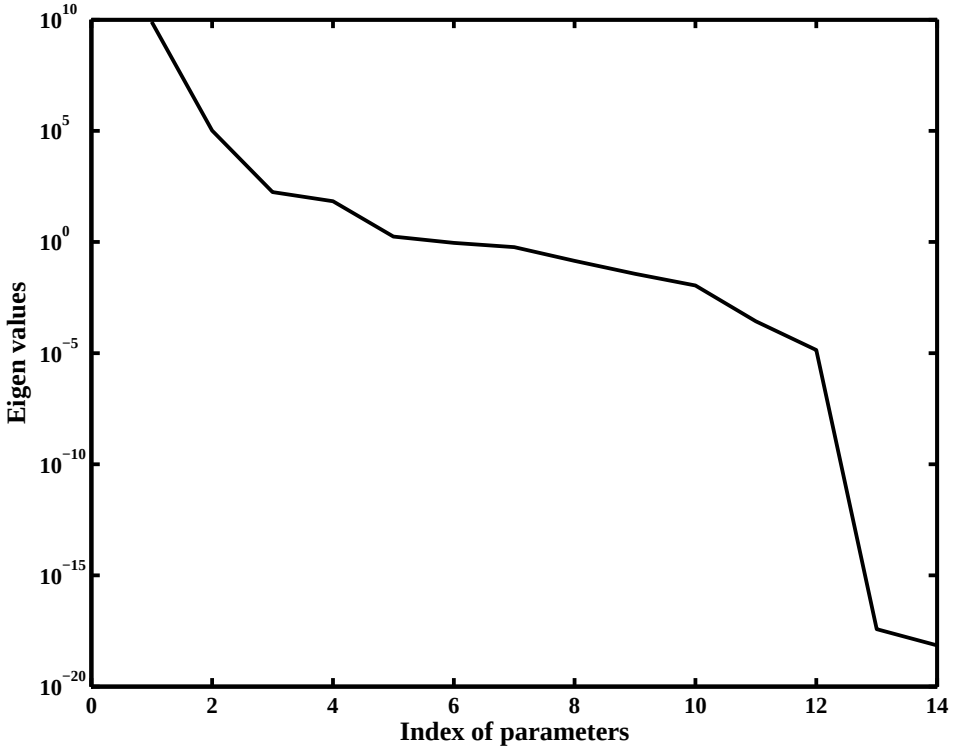

Fig. 19. Eigenvalues of sensitivity covariance matrix (Method II), for example 2.

simulations, the percentage of the errors below threshold values of 0.0005, 0.001, 0.005, 0.01 and 0.05 is recorded. While this Monte Carlo run determined value, it is important to repeat this experiment several times to statistically evaluate the results. For this work, the experiment has been repeated 10 times and standard deviations of the mean value of the error bound have been obtained. For example, for each experiment, the percentages of the error below 0.01 for the system reduced by the combination method were determined to be 0.9936, 0.9938, 0.9936, 0.9928, 0.9936, 0.9926, 0.9932, 0.9950, 0.9938 and 0.9932, respectively resulting in a percentage of 0.9935±0.00066. This procedure was repeated for each threshold value and each reduction technique. Fig. 20 shows a comparison of the statistical results for the three methods for the four cases where 4, 5, 6 and 7 parameters are to be retained.

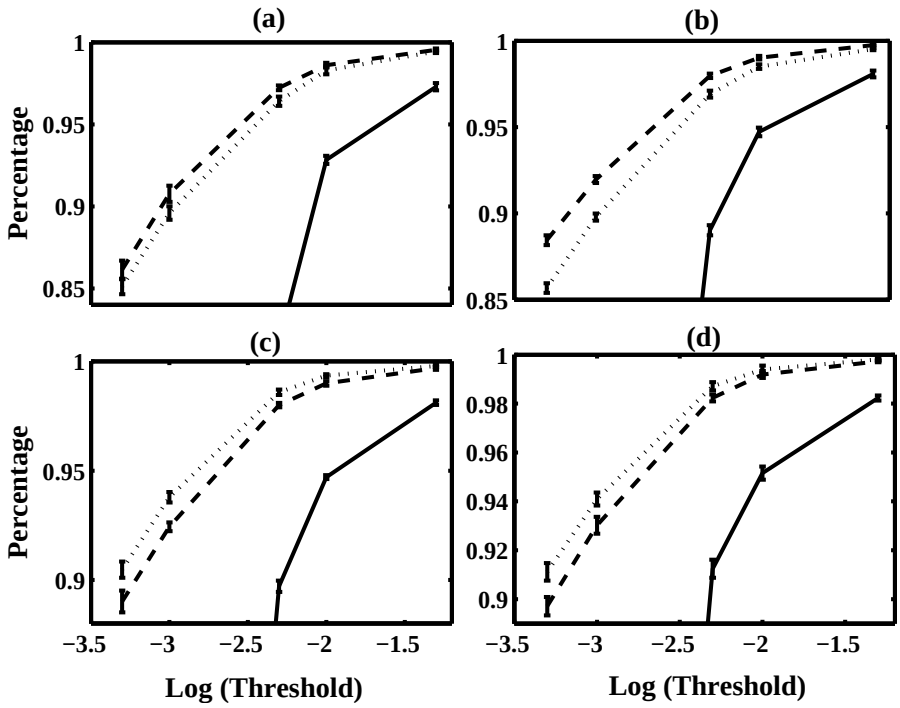

Fig. 20. Comparison of statistics results for three methods, for example 2. (a) 4 parameters retained; (b) 5 parameters retained; (c) 6 parameters retained; (d) 7 parameters retained. Legend: solid line, Method I; dashed line, Method II; dotted line, Method III

From these figures, it can be concluded that all three methods achieve a satisfactory performance for the reduced system. Additionally, it can be seen that if the same number of parameters needs to be retained, that the sensitivity analysis method and the combination method have a better performance than Method I (Hankel singular value-based technique). This point is also illustrated by Fig. 21, in which the relative errors for three methods are plotted for one Monte Carlo experiment. It can also be concluded for this example that Method II and Method III have comparable performance.

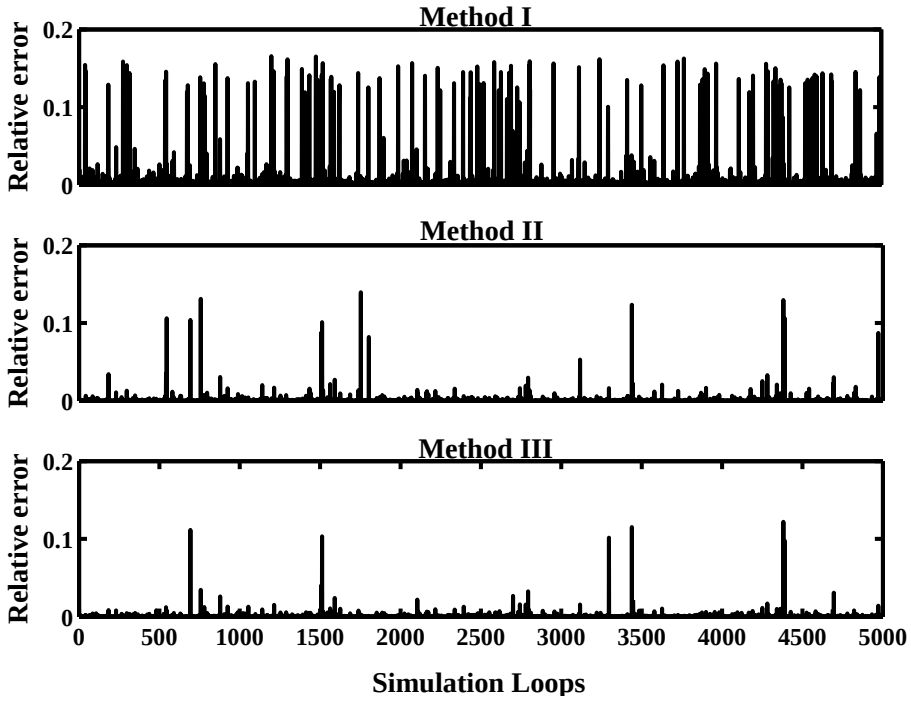

Fig. 21. The relative error set in one Monte Carlo simulation, for example 2.

3. Example 3: Model of a Signal Transduction Pathway in Hepatocytes

This example describes the JAK/STAT pathway used for signal transduction in hepatocytes in response to injury or inflammation [92]. This model consists of 32 states, 39 parameters, 1 input (the concentration of interleukin 6) and 1 output (the concentration of SOCS). While the values of all 39 parameters are taken from the literature [93], these value will have to be confirmed in experiments when different types of cells and experiments are used. Since it is highly unlikely that all 39 parameters can be estimated from available experimental data, it is important to reduce the number of parameters, such that only the more significant ones are retained in the model.

In a first step, Method I was applied to this system. A comparison of the Hankel singular values corresponding to each parameter is shown in Fig. 22-a. Based on

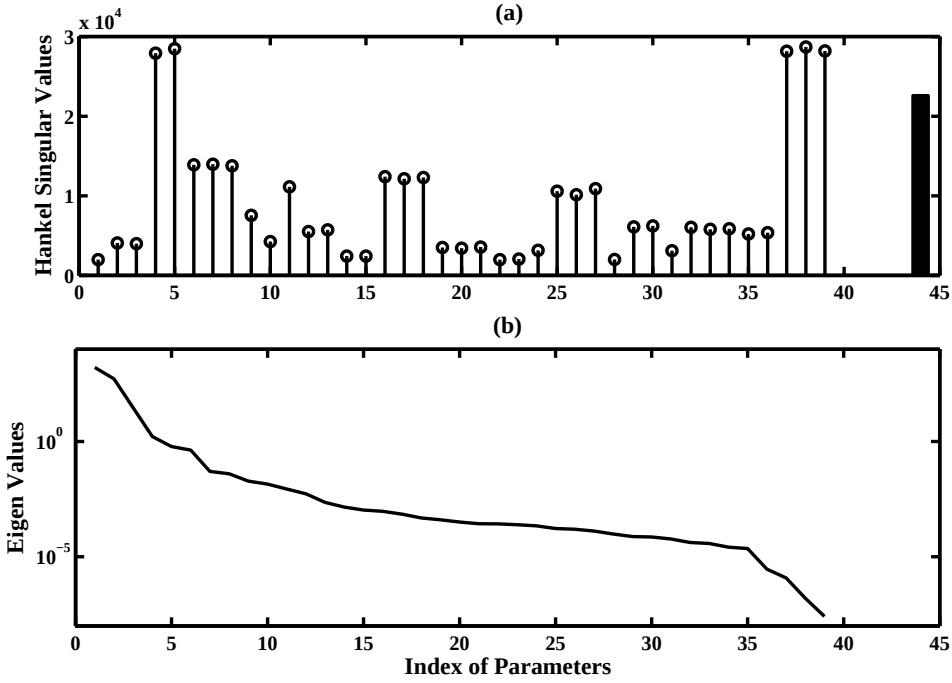

Fig. 22. Hankel singular values for parameters (Method I) and eigenvalues of sensitivity covariance matrix (Method II), for example 3. (a) Comparison of Hankel singular values in Method I; (b) Eigenvalues of sensitivity covariance matrix in Method II.

this comparison, the parameters can be sorted by their relative importance to the experimental measurements:

$$\begin{aligned} &kr0 > kr2 > gp80 > kf0 > kf2 > kr3 > kf3 > k4 > kf9 > k10 > kr9 > \\ &k6 > k16 > kf15 > kr15 > kf5 > k18b > k18a > k22 > kf21 > k20 > \\ &kr7 > kf7 > k23 > kr21 > kr5 > kf1 > kr1 > k12 > kf11 > kr11 > \\ &k14 > k19 > kr8 > kf8 > kr13 > k17 > kf13 > k0 \end{aligned} \tag{5.39}$$

The set of parameters to be reduced can be taken from this list. For example, if 5 parameters need to be reduced, then the 5 least important parameters $kf8$, $kr13$, $k17$, $kf13$, $k0$ should be removed from the model.

The second technique was also applied to this model, where the sensitivity co-

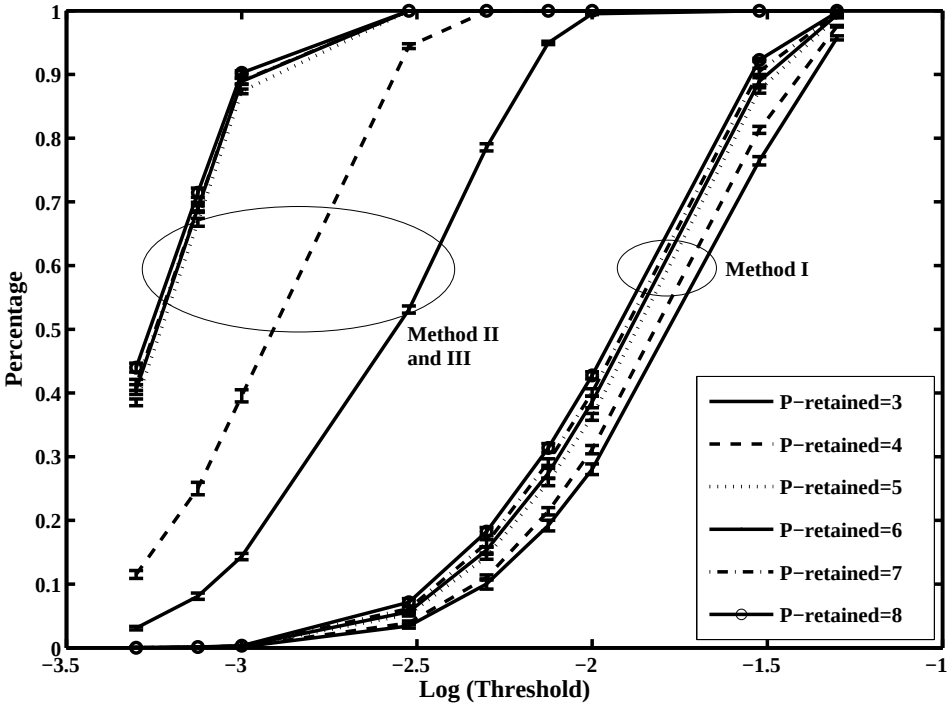

Fig. 23. Comparison of statistics results for three methods, for example 3.

variance matrix is obtained from simulations of system trajectories. The eigenvalues of the sensitivity covariance matrix are shown in descending order in Fig. 22-b.

The initial screening step of the third technique determined that only $kr13$, $k17$, $kf13$, $k0$ are to be reduced based upon the Hankel singular values. These parameters are set constant to their nominal values and parameter space reduction is then performed on the 35 remaining parameters.

To show the performance of these methods, the same evaluation procedure as in the example 2 was conducted. Fig. 23 shows the comparison of the statistics results for the three methods for the six cases where 3 to 8 parameters are retained.

Several observations can be made from this figure: (1) it can be concluded that all three methods achieve a satisfactory performance for the reduced system at a larger threshold value; (2) not surprisingly, the performance is getting better with

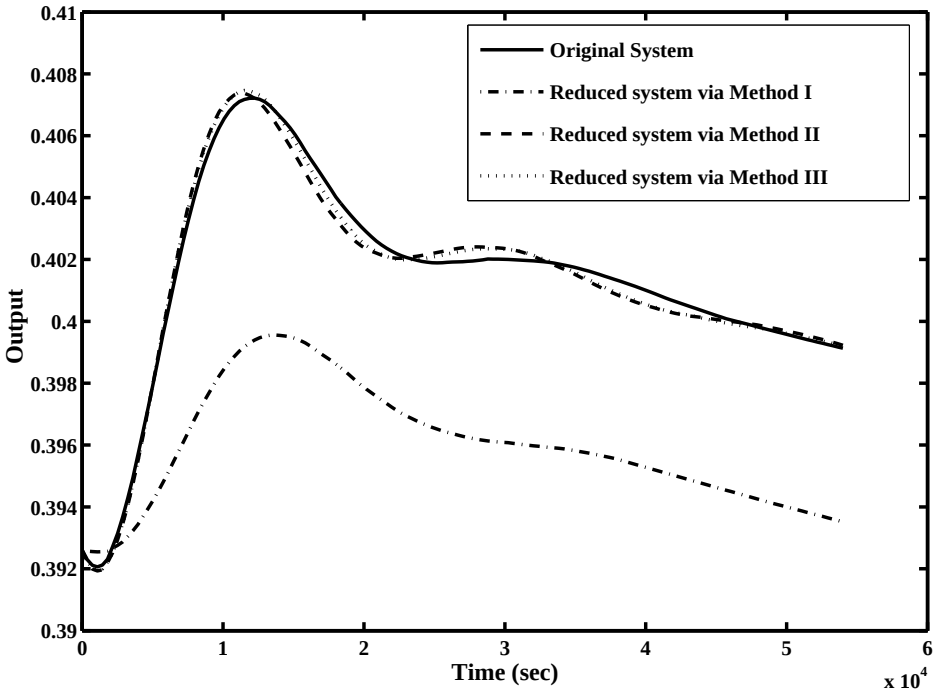

Fig. 24. Comparison of output responses for original systems and reduced systems, for example 3.

an increasing number of retained parameters; (3) the performance of Method II and III is quite similar and both result in better approximations for the same number of retained parameters. This point is also illustrated in Fig. 24, where the output responses are compared for the original system and systems reduced by these three methods (6 parameters are retained). Also Fig. 25 shows the relative errors for three methods for one Monte Carlo experiment.

F. Summary and Discussion

Three techniques for reducing the parameter set of fundamental models are presented in this chapter. The methods are originally developed for linear systems and their extension to nonlinear systems is presented.

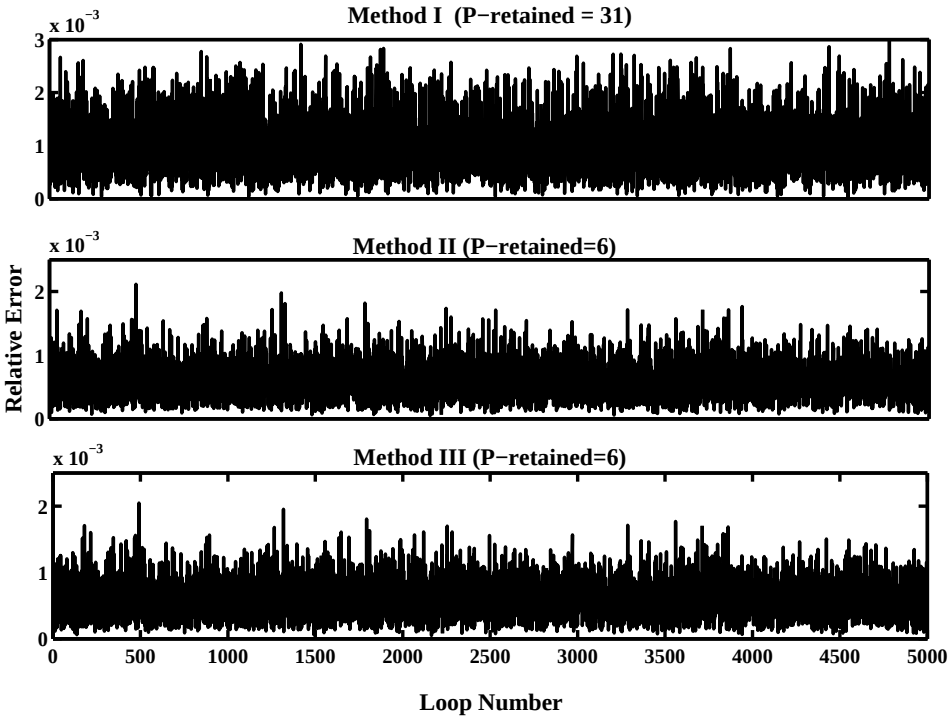

Fig. 25. Comparison of output responses for original systems and reduced systems, for example 3.

One technique focuses on determining the relative importance of parameters for the system behavior based on Hankel singular values computed for each parameter. The parameters corresponding to large values for the sum of the Hankel singular values are classified as important parameters. The less important parameters can then be reduced directly from the original system. The second technique is based on parameter space reduction, which allows reduction of a significant number of parameters while retaining most of the system behavior. This method is based on analysis of the sensitivity covariance matrix. The relative importance of parameters are investigated via the eigenvalues of the sensitivity covariance matrix. A linear transformation of the parameter space is performed and the less important directions in parameter space are reduced. Considering the advantages and drawbacks of these two methods, a

combination method of these two is also developed in this work. An initial screening step similar to analysis via Hankel singular values is applied followed by a reduction of the parameter space spanned by the remaining parameters.

All three techniques are illustrated via three examples. While each method can result in a good approximation, the number of parameters reduced via Method II and III is significantly larger than for the first technique if comparable performance is to be achieved. Both Method II and the combination method exhibited comparable performance in the shown examples, where the combination technique has the advantage that the physical interpretation of some parameters is retained in the model.

CHAPTER VI

MODEL REDUCTION IN THE PRESENCE OF PARAMETRIC UNCERTAINTIES

This chapter describes how to extend a certain class of existing model reduction techniques to take into account uncertainty in model parameters. The key idea of this extension is that the reduced-order model should not only contain the model parameters, but that the reduction procedure itself has to be geared for dealing with parametric uncertainty. This goal is achieved by augmenting the vector of inputs to the system with the uncertain parameters and by performing model reduction on the augmented system. It is shown that error bounds for the reduced order model can be computed if the underlying system is linear with respect to the states, parameters, and inputs. A comparison between the presented technique and a conventional approach is made via two examples.

A. Overview

For systems with parameter uncertainties, the most important parameters can be determined and retained in the systems after the parameter reduction step. When this parameter-reduced system needs to be reduced due to large number of states, those retained parameters of the original model can usually still be found in the reduced system since the structure of the model is retained in the reduced-order model. If the values of parameters change then this change can be directly incorporated in the reduced-order model. However, while the parameters itself are still contained in the

---

Part of this chapter is reprinted from *Journal of Process Control*, Vol. 16, No. 6, C. Sun and J. Hahn, Model reduction in the presence of uncertainty in model parameters, pp. 645-649, Copyright (2006), with permission from Elsevier.

model, it is usually not taken into account if the effects that changes in the parameters have on the model behavior are retained. This chapter will address this deficiency and show how a certain class of model reduction procedures can be extended so that the effects that changes in the parameters have on the model behavior are retained. The result is a model reduction procedure that will be especially useful for systems that contain uncertainties in the model parameters.

B. Problem Statement

This presented approach deals with stable systems described by explicit nonlinear ordinary differential equations of the following form

$$\begin{aligned} \dot{x} &= f(x, \theta, u) \\ y &= g(x, \theta, u) \end{aligned} \tag{6.1}$$

where x represents a vector of the states, u is an input vector, and θ represents a vector containing the model parameters. The focus of this work will be on model reduction techniques based upon a coordinate transformation

$$\bar{x} = Tx \tag{6.2}$$

followed by truncation of less important states:

$$\begin{aligned} \dot{\bar{x}}_1 &= PTf(T^{-1}\bar{x}, \theta, u) \\ \bar{x}_2 &= \bar{x}_{2,ss} \\ y &= g(T^{-1}\bar{x}, \theta, u) \end{aligned} \tag{6.3}$$

where $\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}$, $P = \begin{bmatrix} I & 0 \end{bmatrix}$.

This class of nonlinear model reduction techniques includes proper orthogonal

decomposition (POD) [24], [40] and nonlinear balanced truncation [11], [14]. The linear counterparts [36] form a special case of this problem formulation.

It should be noted that the reduced-order model shown in equation (6.3) still contains the terms $f(\dots)$, $g(\dots)$, and that the parameters, θ, are also retained in the truncated system. One may be inclined to assume that if the values of any of the parameters change, it may be sufficient to include a change in these parameters in equation (6.3). However, this reasoning would disregard that the model reduction procedure may have dramatically changed the effect that changes in the parameters have on the system behavior. The explanation for this can be found in equation (6.2) where the transformation matrix T is computed such that the states are arranged with regard to their contribution to a certain behavior that one wants to retain. For POD, T is computed such that the states are arranged with regard to their contribution to the effect that the inputs u have on the states ($u \rightarrow x$). For balancing, T is chosen such that each state contributes more to the input-output behavior ($u \rightarrow y$) than any subsequent ones. However, neither of these commonly used procedures makes use of the parameters for computing T. In fact, if a reduced model contains uncertainty in the parameters, then the reduced model is only appropriate for its use if

- the uncertainty is very small, or if
- the uncertainty has little effect on the behavior of the process, or if
- it happens that the effect that changes in the parameters have involved a "similar" subspace as the one retained by the reduction procedure.

It should be noted that the last case can happen by pure chance, but neither balancing or POD (as they are commonly applied) directly address this point.

The purpose of this work is to show how balancing and/or POD can be extended such that the effect that uncertainty in the parameters have on the model will be

retained. This is achieved by incorporating parametric uncertainty into the procedure used for computing the state transformation T.

C. Extension to Existing Approaches

The effect that changes in the parameters have on the transformation matrix T can be addressed by lumping the inputs and the parameters into a new vector

$$\tilde{u} = \begin{pmatrix} u \\ \theta \end{pmatrix} \tag{6.4}$$

where the system has to be formulated such that the bounds for the inputs and the parameters

$$u_{\min} \leq u \leq u_{\max}, \quad \theta_{\min} \leq \theta \leq \theta_{\max} \tag{6.5}$$

have the same magnitude after reformulation.

The original system can then be rewritten as

$$\begin{aligned} \dot{x} &= f(x, \tilde{u}) \\ y &= g(x, \tilde{u}) \end{aligned} \tag{6.6}$$

where $\tilde{u}$ represents a vector of inputs to the system. The correlation matrix for POD or the empirical gramians for balancing can then be computed for the system given by equation (6.6) instead of the original system (6.1). The state transformation T is then computed from the correlation matrix or by balancing the empirical gramians. Lumping the parameters and the inputs together in one vector will ensure that the state transformation T will depend upon the effect that changes in the parameters θ have on the system in addition to capturing the effect that excitations with the inputs u have.

While this extension is straightforward, it can nevertheless have a significant

effect on the quality of the reduced model if it contains uncertainty in the model parameters. This is achieved by retaining the directions in state space that are important for representing the effect that changes in the parameters have on the response of the system in addition to retaining directions in state space important for representing the effect of changes in the inputs.

D. Discussion

It is possible to derive error bounds for balanced truncation using the presented procedure for linear systems under parametric uncertainty. It should be noted that the error bounds that are computed for systems where parametric uncertainty is neglected during the procedure are not valid if the parameters are not known exactly.

A linearized version of system (6.1) is given by

$$\begin{aligned} \dot{x} &= Ax + Bu + E\theta \\ y &= Cx + Du + F\theta \end{aligned} \tag{6.7}$$

Lumping the inputs and the parameters together results in the system

$$\begin{aligned} \dot{x} &= Ax + \tilde{B}\tilde{u} \\ y &= Cx + \tilde{D}\tilde{u} \end{aligned} \tag{6.8}$$

with

$$\begin{aligned} \tilde{B} &= \begin{bmatrix} B & E \end{bmatrix} \\ \tilde{D} &= \begin{bmatrix} D & F \end{bmatrix} \end{aligned} \tag{6.9}$$

The empirical controllability gramian for a nonlinear system reduces to the linear controllability gramian, $\tilde{W}_C$, for this type of system which is computed by solving the

following Lyapunov equation [35]

$$A\tilde{W}_C + \tilde{W}_C A^T = -\tilde{B}\tilde{B}^T = \quad - \quad \begin{bmatrix} B & E \end{bmatrix} \quad \begin{bmatrix} B^T \\ E^T \end{bmatrix} = -BB^T - EE^T \tag{6.10}$$

Note that if the uncertainty in the parameter had been neglected for the reduction procedure then the controllability gramian would have been computed from (6.11):

$$AW_C + W_C A^T = -BB^T \tag{6.11}$$

The observability gramian is unaffected by the presented procedure and can be computed by solving the following Lyapunov equation:

$$A^T W_O + W_O A = -C^T C \tag{6.12}$$

The coordinate transformation T which takes uncertainty in the model parameter into account is then computed from balancing $\tilde{W}_C$ and W_O, whereas the conventional approach balances W_C and W_O and neglects the effect of model uncertainty. Accordingly, the Hankel singular values will also be different for both reduced systems.

Applying the linear transformation $\bar{x} = Tx$, the transformed system is given by

$$\begin{aligned} \dot{\bar{x}} &= TAT^{-1}\bar{x} + TBu + TE\theta = \bar{A}\bar{x} + \bar{B}u + \bar{E}\theta \\ y &= CT^{-1}\bar{x} + Du + F\theta = \bar{C}\bar{x} + Du + F\theta \end{aligned} \tag{6.13}$$

and model truncation results in

$$\begin{aligned} \dot{\bar{x}}_1 &= \overline{A}_{11}\bar{x}_1 + \overline{B}_1 u + \bar{E}_1\theta \\ y &= \overline{C_1}\bar{x}_1 + Du + F\theta \end{aligned} \tag{6.14}$$

where

$$\bar{A} = TAT^{-1} = \begin{bmatrix} \bar{A}_{11} & \bar{A}_{12} \\ \bar{A}_{21} & \bar{A}_{22} \end{bmatrix}, \quad \bar{B} = TB = \begin{bmatrix} \bar{B}_1 \\ \bar{B}_2 \end{bmatrix}$$
$$\bar{C} = CT^{-1} = \begin{bmatrix} \bar{C}_1 & \bar{C}_2 \end{bmatrix}, \quad \bar{E} = TE = \begin{bmatrix} \bar{E}_1 \\ \bar{E}_2 \end{bmatrix} \tag{6.15}$$

The error for the reduced order system that includes uncertainty in the model parameters is given by

$$\|G(s) - G_r(s)\|_\infty = \left\| \begin{array}{cc} \bar{C}(sI - \bar{A})^{-1}[\bar{B} & \bar{E}] - \bar{C}_1(sI - \bar{A}_{11})^{-1}[\bar{B}_1 \quad \bar{E}_1] \end{array} \right\|_\infty \tag{6.16}$$

and whose bound can be computed to be [38]

$$\|G(s) - G_r(s)\|_\infty \leq 2 \sum_{j=k+1}^{n} \tilde{\sigma}_j \tag{6.17}$$

where the $\tilde{\sigma}_j$s represent the Hankel singular values corresponding to the system given by equation (6.8).

It should be noted that if a conventional balancing procedure, rather than the extension presented here, is used that it is not possible to derive a closed form solution for the upper bound if uncertainty is present in the model parameters.

E. Case Studies

1. Example 1: Two Nonlinear Reactors in Series

This section presents a comparison between the extended approach and the regular balancing procedure for a system of two CSTRs with uncertainty in a model parameter. The process model was the same example as in Chapter V. This model has four states, one input which is the coolant flow rate q_c and one output, the effluent concentration from the second tank. The value of the parameter E/R is assumed to

be not precisely known. The input has a nominal value of $q_{c0} = 100\text{L}/\min$ and can be changed by ± 10 L/min and the parameter's value is $(E/R)_0 = 10^4 K \pm 10K$.

The reason for choosing a relatively small system for this first example is that it is possible to present the state transformation matrices, and that significant changes can already be seen for the two techniques even when only a small number of states is reduced.

The percentages of the changes of the inputs and parameters are $\pm 1\%$. Covariance matrices are computed for this operating region, first using the conventional approach and then by making use of the presented extension. The transformation matrix computed from the conventional approach results in

$$T_1 = \begin{bmatrix} 0.2492 & -1.8592 & 0.3931 & -1.0448 \\ -0.1318 & -26.4654 & -0.6511 & -7.9172 \\ -0.0265 & -5.8592 & -0.5379 & -23.6948 \\ -2.7405 & -44.6690 & 0.3459 & -10.4925 \end{bmatrix} \tag{6.18}$$

whereas the one computed by the presented method is

$$T_2 = \begin{bmatrix} -0.1866 & -7.7041 & 0.0069 & -1.7991 \\ 0.2088 & 3.4629 & 0.0403 & -0.1640 \\ -0.0197 & 2.3825 & 0.0497 & 0.3556 \\ 0.0645 & -0.4507 & 0.0151 & 5.7952 \end{bmatrix} \tag{6.19}$$

It can be seen that the transformation matrices shown in equations (6.18) and (6.19) are significantly different as they retain different directions in state space. Using either transformation, the system was reduced to 3 states, resulting in the reduced system (I) using conventional balancing and the reduced system (II) for the extended version presented in this paper.

A comparison of the performance in the time domain has been made. A series

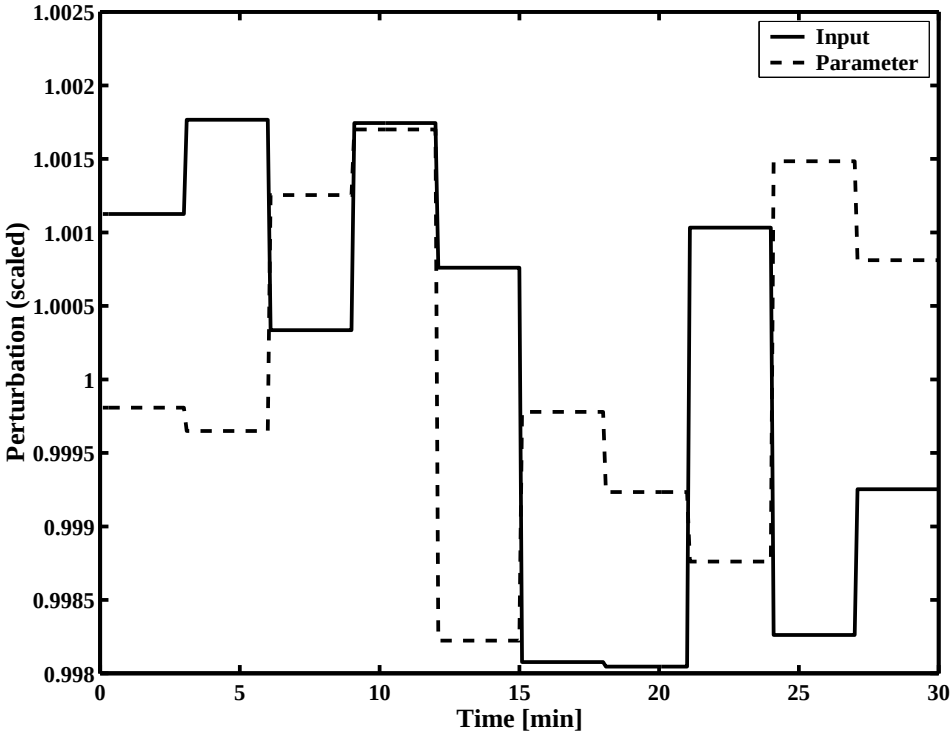

Fig. 26. Input and parameter perturbations (scaled value).

of combinations of random input and parameter perturbations are used to excite the systems within a portion of the operating region. The reason for varying the input and the parameter over time is to show a range of different conditions rather than one scenario. These perturbations are shown in Fig. 26. The output responses for the original system and two reduced systems are shown in Fig. 27-a. Fig. 27-b presents the residuals computed from the response of the original system and both reduced systems in order to show the comparison more clearly. It can be concluded from the example that model reduction techniques that take uncertainty in the model parameters into account can result in significantly better performance if the values of parameters of the system are not precisely known.

Additional simulations where the operating region encompasses the entire $\pm 1\%$ change of the inputs and parameters have also been performed. The performance for

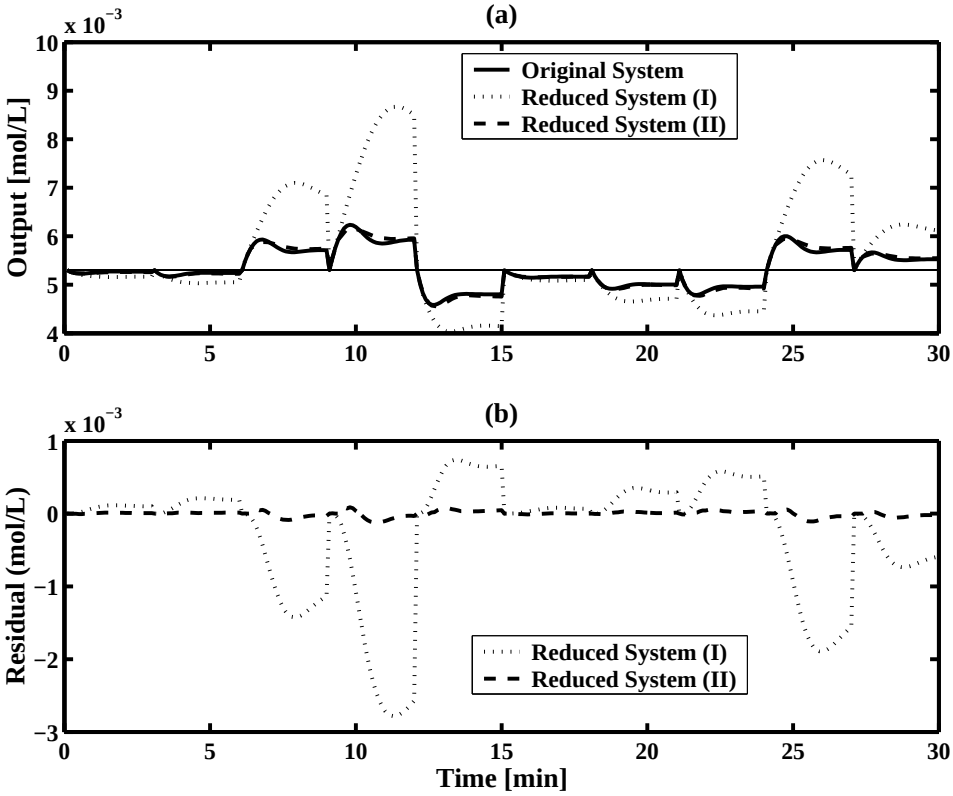

Fig. 27. Comparison of performance for conventional techniques and extended version.

the extended method was virtually identical to the one presented here, however, the response of the system reduced by the conventional technique degraded so strongly, that the figure has not been included.

2. Example 2: Catalytic Fixed-Bed Reactor

The reactor example used in Chapter IV is revisited. The difference is that the linearized version of this model is investigated here in order to compute the error bounds shown in equation (6.16).

The value of the heat of reaction is assumed to be not precisely known for this case study. The input has a nominal value of $u_0 = 625K$ and the parameter's nominal value is $H_0 = 1285409\ kJ/kmol$.

The percentages of the changes of the inputs and parameters are taken to be within $\pm 10\%$ around the nominal value. As the system is linearized, it is possible to compute linear controllability and observability gramians for this process. This is done once for the system with just one input and additional for the system where both the input and the parameter are viewed as inputs to the system. It is possible in both cases to reduce the system from originally 120 states to 7 states while retaining an excellent approximation of the original system. The two reduced models are named Reduced System (I) and (II), respectively.

Fig. 28 illustrates the approximation error given by expression inside of the H_∞ norm in (6.16) for both a conventional balancing technique and for the extended methods introduced in this work over a range of frequencies. It can be seen that the presented approach results in a significantly smaller error at any frequency. Fig. 29 shows a comparison of simulation results for the original system, the system reduced by the conventional approach (I), and the model reduced by the technique presented in this work (II), for varying inputs and changes in the value of the parameter. It can be concluded that the results obtained by the presented technique can result in a significantly more accurate reduced model if uncertainty is present in the model parameters.

F. Summary and Discussion

This chapter presented an extension to a certain class of model reduction procedures, i.e. POD and balancing, for cases where the model contains uncertainty in the model parameters. It was shown that simply retaining the variables representing model parameters may not necessarily be sufficient for addressing parametric uncertainty. Rather, the uncertainty has to be directly incorporated into the model

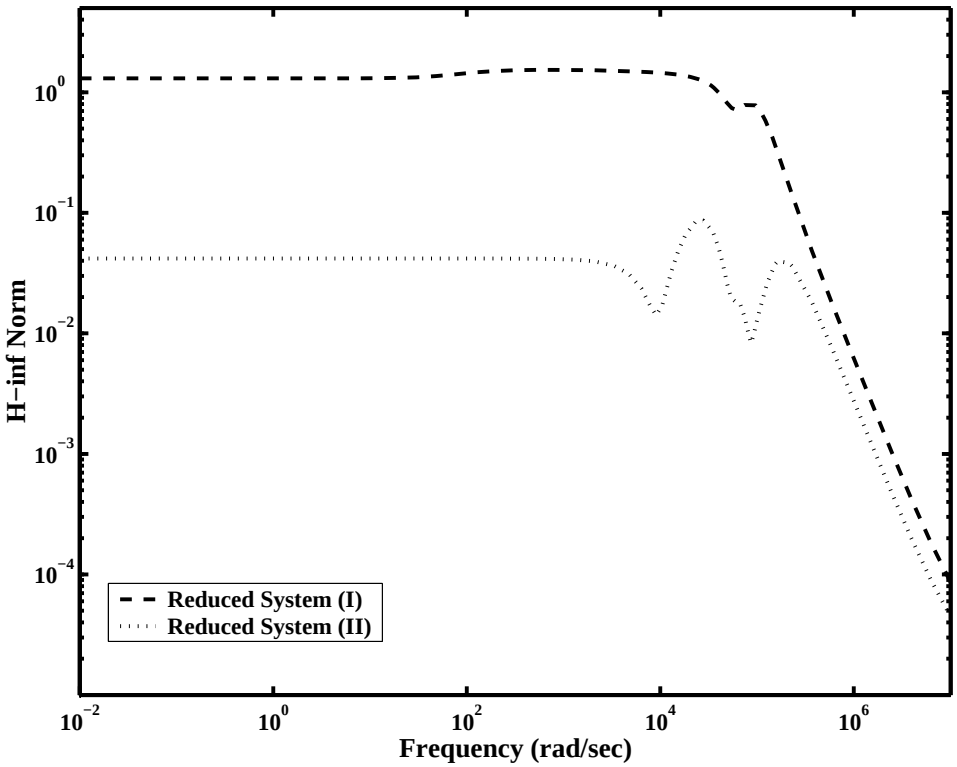

Fig. 28. Bound of the error for reducing the model from 120 states to 7 states for a range of frequencies.

reduction procedure. The presented extension addresses this issue by augmenting the input vector with the parameters. The resulting augmented vector can then be used for exciting the system and computing empirical gramians (balancing) or the correlation matrix (POD).

An error bound for the performance of the reduction procedure, when an extension to balanced truncation is used, has been derived for linearized systems where uncertainty in the model parameters is present. Additionally, the performance of incorporating parametric uncertainty into the model reduction procedure has been illustrated in two case studies for varying inputs to the system and changes in the parameter.

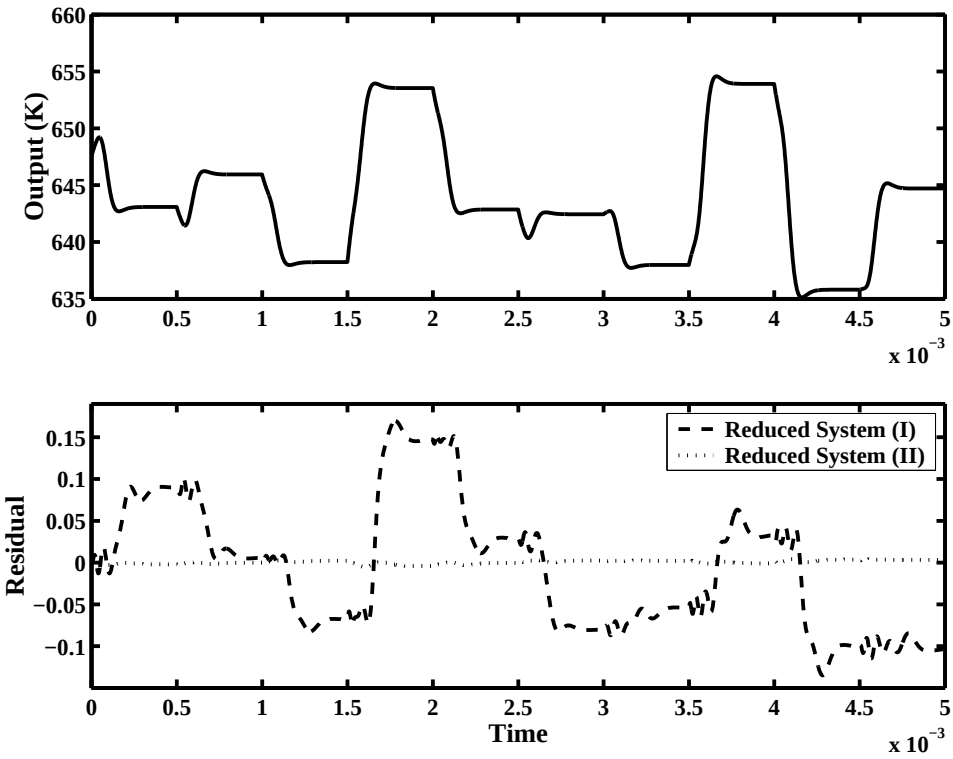

Fig. 29. Comparison of performance for conventional techniques and the extended version.

CHAPTER VII

APPLICATION OF NONLINEAR MODEL PREDICTIVE CONTROL BASED ON REDUCED ORDER MODELS

The most important property for a reduced model is to represent the input-ouput behavior of a model correctly, when it is applied within a controller. In order to test this a model predictive control (MPC) algorithm was used to compute control moves. MPC is an important control strategy in industry, in which a process model is used to predict process behavior [94]. Through an online optimization procedure, a sequence of control moves is selected to minimize a cost function based on the reference trajectory over a prediction horizon, possibly subject to constraints on the manipulated inputs and outputs. Generally, MPC consists of three components: process models, reference trajectories and recessive online optimization. Based on the type of the process models, MPC can be classified into linear model predictive control (LMPC) and nonlinear model predictive control (NMPC). LMPC has been exclusively developed in the past three decades [95], however, LMPC is precluded in processes sufficiently nonlinear. With the development of more accurate nonlinear process models, NMPC is gaining wider acceptance. On the other hand, with the complexity of the nonlinear process models, the difficulty to implementing NMPC increases due to the computational effort. That's why model reduction is motivated. Therefore, this chapter gives some applications of NMPC based on the reduced order models to show that model reduction is an efficient strategy to overcome this problem of NMPC.

A. Overview of NMPC

Since the main focus of this work is to show the performance of NMPC based on the reduced order models instead of the theoretical development of NMPC, the principles of NMPC are only briefly reviewed.

1. The Optimization Problem

The optimization problem for NMPC is [94], [96]:

$$\min_{u(k|k),u(k+1|k),\cdots,u(k+M-1|k)} J = \phi[y(k+P|k)]+\sum_{j=0}^{P-1} L[y(k+j|k),u(k+j|k),\Delta u(k+j|k)] \tag{7.1}$$

where M is the control horizon, P is the prediction horizon, $u(k+j|k)$ is the input values which is calculated by use of information available at time k, $y(k+j|k)$ is the output values which is calculated by use of information available at time k, $\Delta u(k+j|k) = u(k+j|k)-u(k+j-1|k)$, ϕ and L are possibly nonlinear functions. Quadratic functions of the following form is are commonly used for ϕ and L:

$$\phi = [y(k+P|k) - y_s(k)]^T Q[y(k+P|k) - y_s(k)] \tag{7.2}$$

$$\begin{aligned} L = {} & [y(k+j|k) - y_s(k)]^T Q[y(k+j|k) - y_s(k)]+ \\ & [u(k+j|k) - u_s(k)]^T S[u(k+j|k) - u_s(k)]+ \\ & [\Delta u(k+j|k)]^T R[\Delta u(k+j|k)] \end{aligned} \tag{7.3}$$

where $y_s(k)$, $u_s(k)$ are steady state targets for y and u, respectively. Q, R and S are positive definite weighting matrices.

The sequence of control moves are obtained by solving this optimization problem: $u(k|k)$, $u(k+1|k)$, $\cdots$, $u(k+M-1|k)$. Among this sequence of control moves, only the first one is implemented $u(k) = u(k|k)$. Then the next set of pro-

cess measurements are obtained and the process parameters are updated. Another optimization problem will then be solved to compute the next control move.

2. Process Constraints

Because there exists actuator limitation as well as operational limitations, input and output constraints are often present in MPC algorithms [94]:

$$u_{\min} \leq u \leq u_{\max} \tag{7.4}$$

$$\Delta u_{\min} \leq u \leq \Delta u_{\max} \tag{7.5}$$

$$y_{\min} \leq y \leq y_{\max} \tag{7.6}$$

where $(\cdot)_{\min}$, $(\cdot)_{\max}$ refer to the minimum and maximum values.

3. Disturbance Estimation

Due to the presence of unmeasured disturbance and modeling errors, the controller can exhibit steady state offset. To overcome this problem, integral action is incorporated to generate the output targets $y_s(k)$ with disturbance estimation. One way to estimate the disturbance is as follows, which is also used in this work.

$$\hat{d} = \frac{1}{N_d} \sum_{k=1}^{N_d} (y_{k,m} - \hat{y}_{k,p}) \tag{7.7}$$

where N_d represents a design horizon for the purpose of smoothing out the influence of measurement noise. y_k is the measurement value and $\hat{y}_k$ is the estimated output. If there is only one measurement, then $\hat{d}$ is a scalar at one time point, otherwise, $\hat{d}$ is a vector.

4. The Algorithm Used in This Work

Based on the disturbance estimation, the optimization problem is modified as follows.

$$y_s(k) = y_{sp}(k) - \hat{d}(k) \tag{7.8}$$

where $y_{sp}(k)$ are set points for the output variables.

The penalty on the inputs is eliminated (S=0), resulting in the quadratic function L:

$$\begin{aligned} L = &[y(k+j|k) - y_{sp}(k) + \hat{d}(k)]^T Q[y(k+j|k) - y_{sp}(k) + \hat{d}(k)] + \\ &[\Delta u(k+j|k)]^T R[\Delta u(k+j|k)] \end{aligned} \tag{7.9}$$

The quadratic function ϕ becomes:

$$\phi = [y(k+P|k) - y_{sp}(k) + \hat{d}(k)]^T Q[y(k+P|k) - y_{sp}(k) + \hat{d}(k)] \tag{7.10}$$

The cost function J can then be rewritten as:

$$\begin{aligned} J = &\sum_{j=0}^{P-1} [y(k+j|k) - y_{sp}(k) + \hat{d}(k)]^T Q[y(k+j|k) - y_{sp}(k) + \hat{d}(k)] \\ &+ \sum_{j=0}^{P_c-1} \Delta u(k+j|k)^T R \Delta u(k+j|k) \end{aligned} \tag{7.11}$$

Generally, the last item uses the control horizon instead of the prediction horizon, that is:

$$\begin{aligned} J = &\sum_{j=0}^{P-1} [y(k+j|k) - y_{sp}(k) + \hat{d}(k)]^T Q[y(k+j|k) - y_{sp}(k) + \hat{d}(k)] \\ &+ \sum_{j=0}^{M_c-1} \Delta u(k+j|k)^T R \Delta u(k+j|k) \end{aligned} \tag{7.12}$$

This is the cost function used in this work.

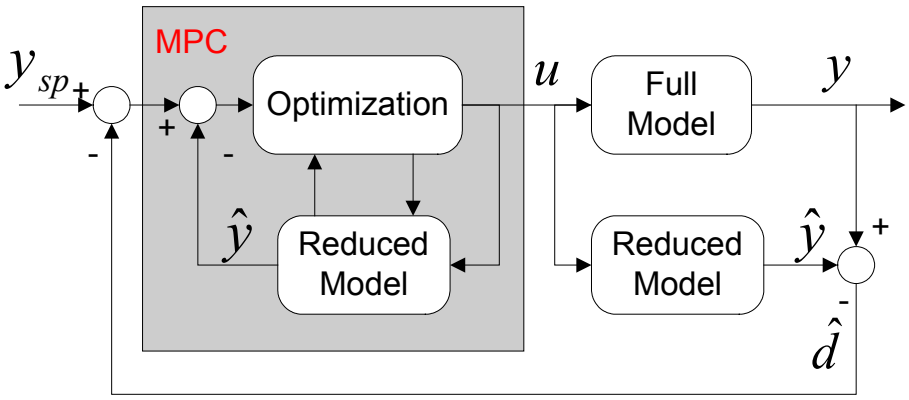

Fig. 30. Block diagram of NMPC based on reduced order models

5. NMPC Scheme Based on Reduced Order Models

Since the purpose of implementing NMPC here is to test the reduced order models, a NMPC scheme based on reduced order models is developed. The main difference between the general NMPC and the NMPC based on the reduced order models is that in the optimization procedure, the output of the reduced order models will be used instead of the output of the full order model. The block diagram of this NMPC scheme is shown in Fig. 30.

B. Case Studies

Two examples are used to illustrate the implementation of NMPC, one is for an ODE system and the other is for a DAE system. The close loop performances are compared.

1. Example 1: Catalytic Fixed-Bed Reactor

This example is used to investigate a NMPC controller implemented on an ODE system. This reactor system is the same as the one used in Chapter IV. With the

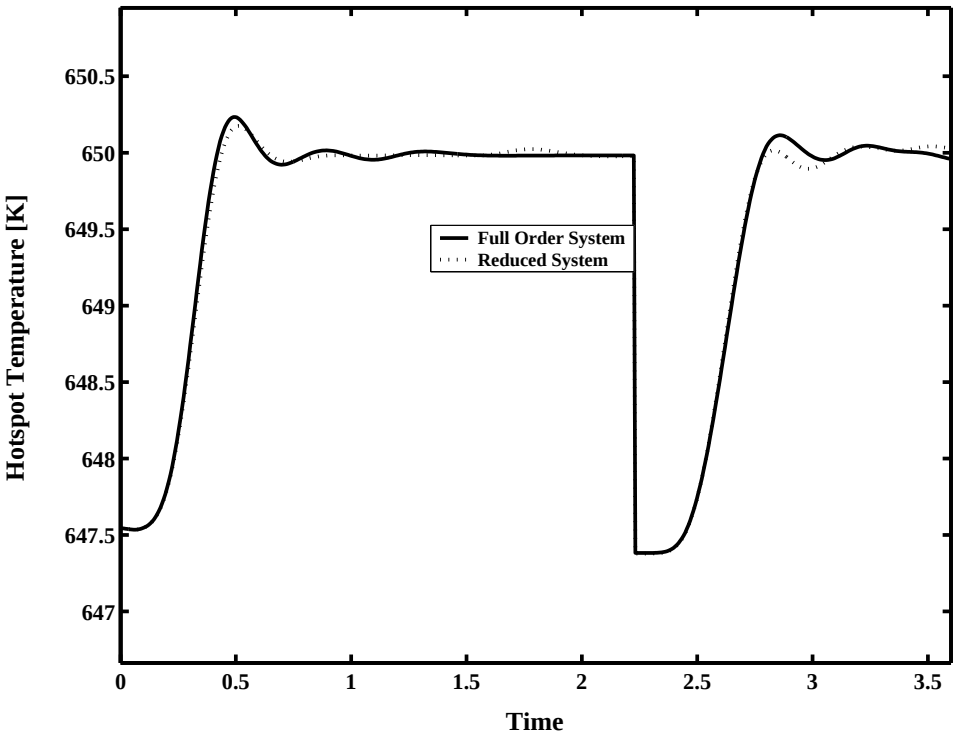

Fig. 31. Temperature vs. time for set point change and disturbance rejection

NMPC controller included, the system forms a close-loop system. The close-loop responses are computed and a comparison of the output performance is conducted.

Two models are compared in the following. One is a nonlinear model of the process itself, with 120 states, which will be the reference trajectory, since it contains no model mismatch and no other model can result in better controller performance. The other is a nonlinear reduced model with 20 states. Fig. 31 represents the trajectories generated by a set point change to $650K$ for the temperature at the hot spot. In addition to this, the reactor temperature is affected by an output disturbance of $2.6K$. Fig. 32 represents the optimal control moves computed for this case study for each of these two models. This example tests both set point tracking and disturbance rejection. From Fig. 31, it can be seen that the reduced order model can achieve a very comparable performance to the full-order system.

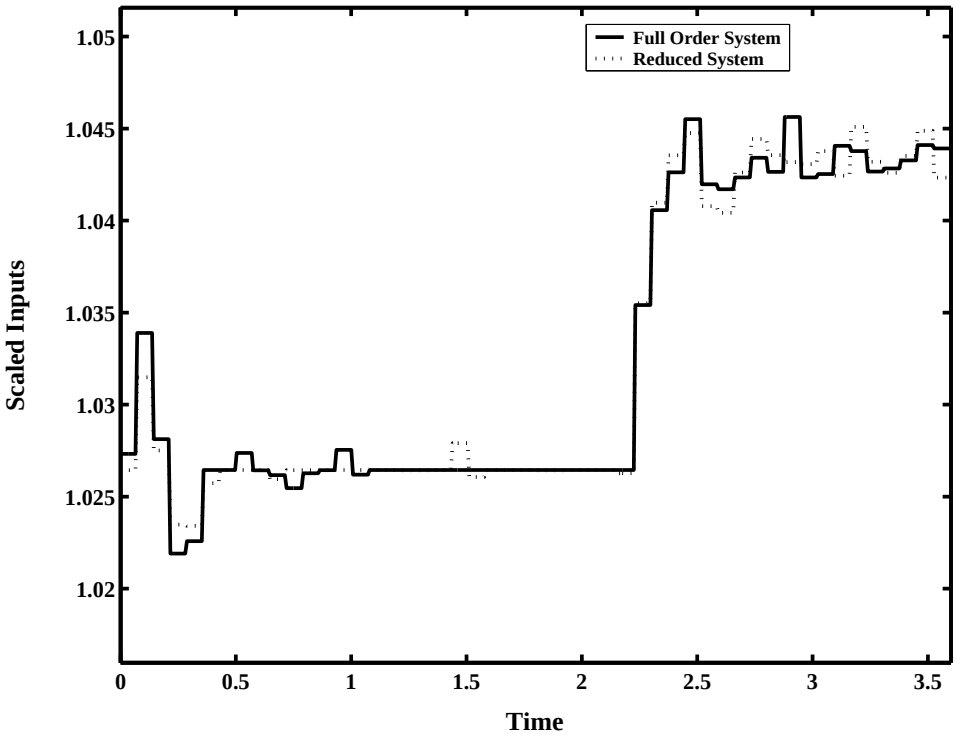

Fig. 32. Comparison of computed controller actions

Table IV. Comparison of CPU times for closed-loop control of fix-bed reactor model

Model	Full-order system	Reduced system
Time (Seconds)	221.8	127.6

The computation times for these two controllers are given in Table IV. It can be concluded that the controller based upon the reduced-order model results in similar performance as the one based upon the full-order model. However, it is apparent that the reduced-order controller requires much less computation time in order to achieve this performance, which shows the advantage of the reduced order models in process control.

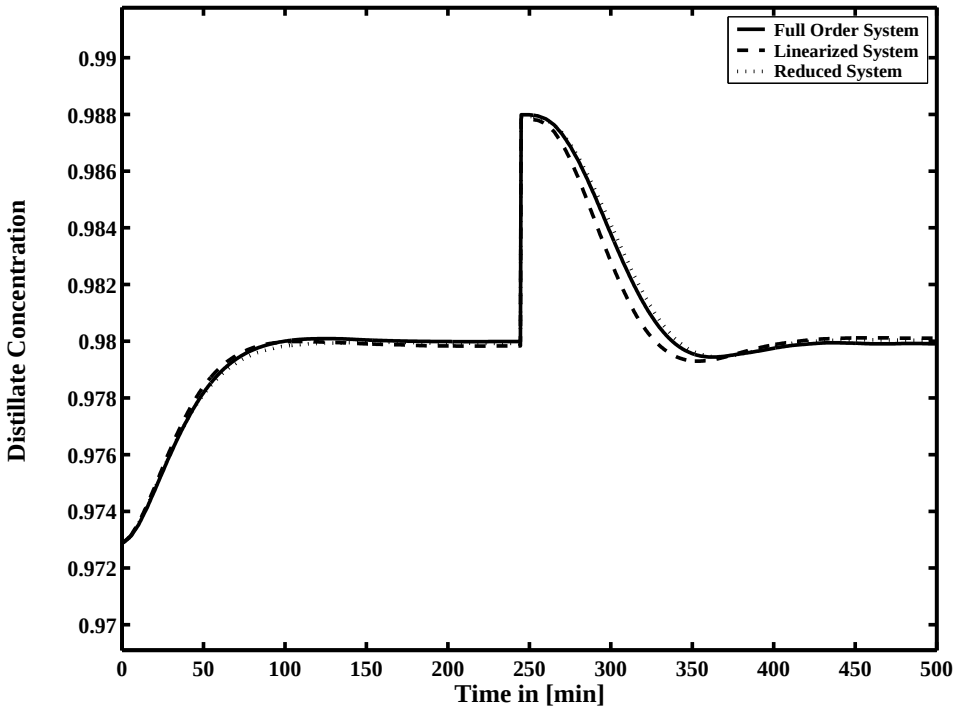

Fig. 33. Distillate concentration vs. time for set point change and disturbance rejection

2. Example 2: Distillation Column System

A NMPC controller implemented on a DAE system is investigated here. The distillation column model with 32 differential equations and 32 algebraic equations which is studied in Chapter III is revisited.

The open loop response has been studied in Chapter III, shown in Fig. 7. From this figure, it can be concluded that the reduced order model has a similar performance to the original model and much better than the linearized model.

The achievable closed-loop response of this model is studied here, in a similar fashion as was done for the reactor example.

Three different models are compared in the following. The first one is a nonlinear model of the process itself, which will be the reference trajectory, since it contains no model mismatch and no other model can result in better controller performance.

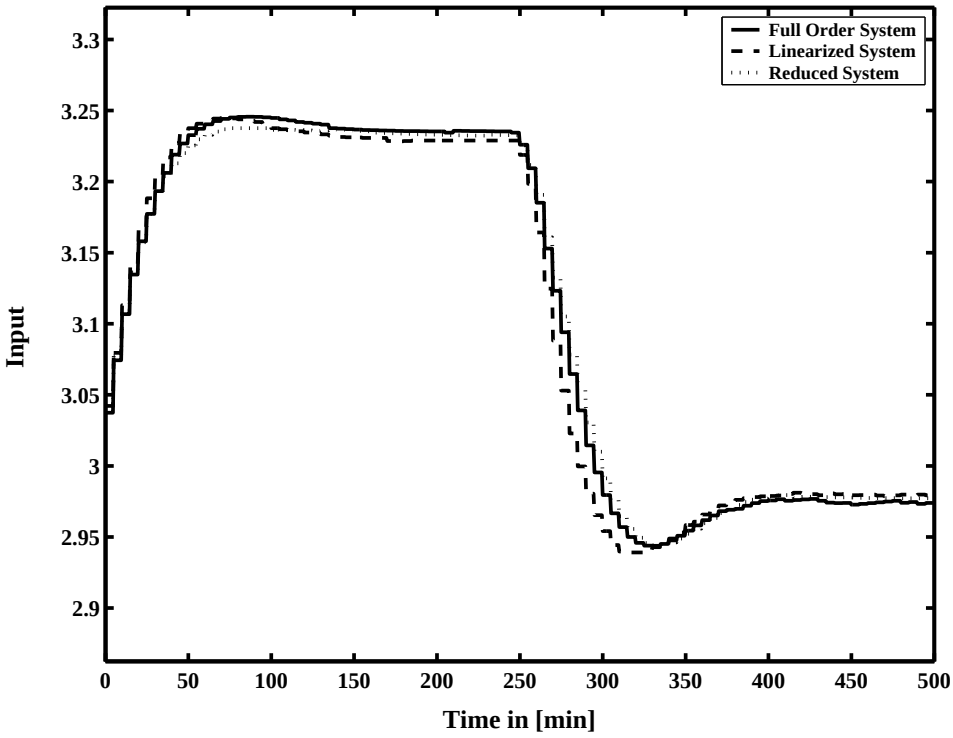

Fig. 34. Comparison of computed controller actions

The second model is a linearized version of the original nonlinear model. The third model is a nonlinear reduced model via the technique presented in Chapter III. Fig. 33 represents the trajectories generated by a set point change to 0.98 for the distillate concentration. In addition to this, after 250 minutes of operation, the distillate concentration gets hit by an output disturbance of 0.08. Fig. 34 represents the optimal control moves computed for this case study for these three models. This example also tests set point tracking and disturbance rejection. It can be seen that both the linearized model and the reduced model result in a controller that closely matches the performance achieved by a controller based upon the full-order nonlinear model. And it is also obvious that the nonlinear reduced order model performs better than the linearized model, which illustrates that the NMPC based upon the reduced order models is outperforms the LMPC based upon the linearized model.

Table V. Comparison of CPU times for closed-loop control of distillation column model

Model	Full-order system	Linearized system	Reduced system
Time (Seconds)	264.1	66.2	81.8

The computation times for the three controllers are given in Table V. Similar to the reactor system, the controller based upon the reduced-order model results in similar performance as the one based upon the full-order model. Also the controller based on the reduced-order model requires much less computation time in order to achieve this performance.

C. Summary and Discussion

This chapter tested the performance of reduced order models in a NMPC frame work. The principles of NMPC were briefly reviewed in this chapter and two examples are investigated to illustrate the implementation of NMPC. One example is a catalytic fix-bed reactor system which is an ODE system. The other is a distillation column model which is a DAE system. Both examples show that NMPC based on the reduced order models can achieve a very close performance with the one based on the nonlinear full-order system. The computational effort decreases significantly when the reduced order system is used. Additionally, the distillation column example also shows the reduced order system performs better than the linearized system.

CHAPTER VIII

CONCLUSIONS AND RECOMMENDATIONS

This research presents new techniques for the reduction of nonlinear systems exhibiting two-time scale behavior and systems with parameter uncertainties. Systems with two-time scale behavior are described by differential algebraic equations (DAE). The proposed reduction method for this type of system is based on a combination of projection methods and system identification techniques. This approach performs order-reduction of the differential equations and reduces the size and complexity of the algebraic equations. The procedure for reducing the differential equations can be performed by balancing, while a variety of identified model structures can be used for simplification of the algebraic equations. During the procedure, the interplay between the states and the algebraic variables has been taken into account as it is reflected in the computed projections. The case studies shows that the reduced system via this presented technique can achieve a satisfactory performance while it has a much smaller size.

For systems with parameter uncertainties, two main problems are studied in this work. The first one is parameter reduction. Three techniques for reducing the parameter set of fundamental models are presented in this work, one of which focuses on determining the relative importance of parameters for the system behavior based on Hankel singular values computed for each parameter. The parameters corresponding to large values for the sum of the Hankel singular values are classified as important parameters. The less important parameters can then be reduced directly from the original system. The second technique is based on parameter space reduction, which allows reduction of a significant number of parameters while retaining most of the system behavior. This method is based on analysis of the sensitivity covariance ma-

trix. The relative importance of parameters are investigated via the eigenvalues of the sensitivity covariance matrix. A linear transformation of the parameter space is performed and the less important directions in parameter space are reduced. Considering the advantages and drawbacks of these two methods, a combination method of these two is also developed in this work. An initial screening step similar to analysis via Hankel singular values is applied followed by a reduction of the parameter space spanned by the remaining parameters. It shows that all three methods can result in good approximations. On the other hand, both the second and the third method perform better than the first method whose advantage is that the parameters retain their physical meanings throughout the reduction procedure. The latter two methods exhibit comparable performance, however, the last method has a potential advantage over the second method since the physical interpretation of some parameters is retained in the model.

A second problem for systems with parameter uncertainties is to derive a procedure for state reduction where parameter uncertainties are taken into account. A technique involving lumping of the system inputs and the parameters into an augmented input vector is presented to address this problem. This approach takes into account uncertainties in regarding parameters as one kind of process inputs. The resulting of the reduced-order models perform much better than if the parameter uncertainties were ignored.

As a last contribution, the closed-loop behavior of systems after reduction was also investigated in this work. It has been illustrated that the controllers based on the reduced models can achieve performance comparable to those based on the full order models and at the same time require significantly less computation time.

A. Contributions

Specifically, the contributions of this dissertation are listed in chronological order:

1. Revised the $MATLAB^{TM}$ code for computing empirical gramians or covariance matrices, which increases the computation speed by about a factor of 50-60. For instance, originally a computation of observability gramian requires 315.3 seconds, now it only takes 5.4 seconds.

2. This research proposed and implemented a new approach based on a combination of projection methods and system identification techniques for DAE model reduction. Artificial neural network or partial least squares were applied to do the system identification.

3. A partitioning of covariance matrices for differential states and algebraic variables was proposed.

4. An error bound for the model reduction of linear DAE system was derived.

5. The application of Partial least squares was investigated in both ODE and DAE model reduction.

6. A new approach based on Hankel singular values was proposed and implemented for parameter reduction and the error bound for reducing parameters was presented.

7. A new technique based on the combination of Hankel singular values and parameter sensitivity analysis was investigated.

8. This research investigated the problem of model reduction with parameter uncertainties. A technique lumping the inputs and parameters as an augmented input vector was presented.

9. A nonlinear model predictive control strategy was applied to test the performance of the reduced order models.

10. Robust nonlinear model reduction routine programs were developed in the $MATLAB^{TM}$ environment. A documentation is included in Appendix C.

B. Future Work

Several extensions of the presented work are possible.

1. For DAE model reduction, further investigation should focus on required modifications for fully implicit DAEs or higher index DAEs.

2. For DAE model reduction, an optimization algorithm should be developed to find optimal transformation matrices T_1, T_2 to minimize $\|S_1 - S_2\|_\infty$, where S_1 represents a quantity for the original system and S_2 for the reduced system.

3. Nonlinear transformations should be investigated. That is, the linear transformation $\bar{x} = Tx$ used in this work should be further extended to $\bar{x} = h(x)$.

4. Model reduction techniques for DAE or PDE systems with uncertain parameters should be investigated.

5. Applications to controller and observer design should be investigated for systems with reduced parameter sets.

REFERENCES

[1] S.J. Qin and T.A. Badgwell, "A survey of industrial model predictive control technology," *Control Eng. Practice*, vol. 11, no. 7, pp. 733–764, July 2003.

[2] B.D.O. Anderson and Y. Liu, "Controller reduction: Concepts and approaches," *IEEE Transactions on Automatic Control*, vol. 34, no. 8, pp. 802–812, August 1989.

[3] A.C. Antoulas, D.C. Sorensen, and S. Gugercin, "A survey of model reduction methods for large-scale systems," in *Structured Matrices in Operator Theory, Numerical Analysis, Control, Signal and Image Processing*. Providence, RI: AMS publications, 2001.

[4] L. Fortuna, G. Nunnari, and A. Gallo, *Model Order Reduction Techniques with Application in Electrical Engineering*, New York: Springer-Verlag, 1992.

[5] A.J. Laub, M.T. Heath, C.C. Paige, and R.C. Ward, "Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms," *IEEE Trans. Automatic Control*, vol. 32, no. 2, pp. 115–121, February 1987.

[6] R.N. Mehra, "Optimal inputs for linear system identification," *IEEE Transactions on Automatic Control*, vol. AC-19, no. 3, pp. 192–200, June 1974.

[7] R. Samar, I. Postlethwaite, and D.W. Gu, "Model reduction with balanced realization," *Int. J. of Control*, vol. 62, no. 1, pp. 33–64, July 1995.

[8] P. Th. L. M. van Woerkom, "Mathematical model of flexible spacecraft dynamics: A survey of order reduction approaches," in *Automatic Control in Space*, pp. 127–135. Oxford and New York: Pergamon Press, 1985.

[9] S. Skogestad and I. Postlethwaite, *Multivariable feedback control*, New York: John Wiley & Sons, 1997.

[10] W. Marquardt, "Nonlinear model reduction for optimization based control of transient chemical processes," in *Proc. CPC VI*, 2001, pp. 30–60.

[11] J. Hahn and T.F. Edgar, "Balanced approach to minimal realization and model reduction of stable nonlinear systems," *I&EC Research*, vol. 41, no. 9, pp. 2204–2212, May 2002.

[12] J. Hahn, T.F. Edgar, and W. Marquardt, "Controllability and observability covariance matrices for the analysis and order reduction of stable nonlinear systems," *Journal of Process Control*, vol. 13, no. 2, March 2003.

[13] S. Lall, J.E. Marsden, and S. Glavaski, "Empirical model reduction of controlled nonlinear systems," in *IFAC World Congress*, 1999.

[14] S. Lall, J.E. Marsden, and S. Glavaski, "A subspace approach to balanced truncation for model reduction of nonlinear control systems," *Intern. J. on Robust and Nonlinear Control*, vol. 12, no. 6, pp. 519–535, May 2002.

[15] K.S. Lee, Y. Eom, J.W. Chung, J. Choi, and D. Yang, "A control-relevant model reduction technique for nonlinear systems," *Computers Chem. Engng.*, vol. 24, no. 2-7, pp. 309–315, July 2000.

[16] A. Newman and P.S. Krishnaprasad, "Computation for nonlinear balancing," in *IEEE Proc. 37th Conference on Decision and Control*, 1998.

[17] A. Newman and P.S. Krishnaprasad, "Computing balanced realizations for nonlinear systems," in *Proc. 14th Int. Symp. Math. Theory Networks and Systems*, 2000.

[18] J.M.A. Scherpen, "Balancing of nonlinear systems," in *Proc. Eur. Contr. Conf. in Groningen*, 1993.

[19] J. Baker and P.D. Christofides, "Finite-dimensional approximation and control of non-linear parabolic pde systems," *International Journal of Control*, vol. 73, no. 5, pp. 439–456, March 2000.

[20] G. Berkooz, P. Holmes, and J.L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Ann. Rev. Fluid Mechanics*, vol. 25, pp. 539–575, January 1993.

[21] P.D. Christofides and P. Daoutidis, "Finite-dimensional control of parabolic pde systems using approximate inertial manifolds," *J. Math. Anal. Appl.*, vol. 216, pp. 398–420, December 1997.

[22] A. Kienle, "Low-order dynamic model for ideal multicomponent distillation processes using nonlinear wave propagation theory," *Chem. Eng. Sci.*, vol. 55, no. 10, pp. 1817–1828, May 2000.

[23] W. Marquardt, "Traveling waves in chemical processes," *Int. Chem. Eng.*, vol. 30, no. 4, pp. 585–606, October 1990.

[24] H.M. Park and D.H. Cho, "The use of karhunen-loève decomposition for the modeling of distributed parameter systems," *Chemical Engineering Science*, vol. 51, no. 1, pp. 81–98, January 1996.

[25] S.Y. Shvartsman and I.G. Kevrekidis, "Nonlinear model reduction for control of distributed systems: a computer-assisted study," *AIChE Journal*, vol. 44, no. 7, pp. 1579–1595, July 1998.

[26] S.Y. Shvartsman, C. Theodoropoulos, R. Rico-Martinez, I.G. Kevrekidis, E.S. Titi, and T.J. Mountziaris, "Order reduction for nonlinear dynamic models of distributed reacting systems," *Journal of Process Control*, vol. 10, no. 2-3, pp. 177–184, April 2000.

[27] D. Zheng, A. Hoo, and M.J. Piovoso, "Low-order model identification of distributed parameter system by a combination of singular value decomposition and the karhunen-loève expansion," *I&EC Research*, vol. 41, no. 6, pp. 1545–1556, March 2002.

[28] S.M. Jacob, B. Gross, S.E. Voltz, and V.W. Weekman, "A lumping and reaction scheme for catalytic cracking," *AIChE Journal*, vol. 22, no. 4, pp. 701–713, July 1976.

[29] J. Wei and J.C.W. Kuo, "Lumping analysis in monomolecular reaction systems I: Analysis of the exactly lumpable system," *I&EC fundamentals*, vol. 8, no. 1, pp. 114–123, February 1969.

[30] J.C.W. Kuo and J. Wei, "Lumping analysis in monomolecular reaction systems II: Analysis of approximately lumpable system," *I&EC fundamentals*, vol. 8, no. 1, pp. 124–133, February 1969.

[31] G.A. Robertson and I.T. Cameron, "Analysis of dynamic process models for structural insight and model reduction - part 1," *Computers Chem. Engng.*, vol. 21, no. 5, pp. 455–473, December 1996.

[32] G.A. Robertson and I.T. Cameron, "Analysis of dynamic process models for structural insight and model reduction - part 2," *Computers Chem. Engng.*, vol. 21, no. 5, pp. 475–488, December 1997.

[33] G. Obinata and B. D.O. Anderson, *Model reduction for control system design*, London: Springer-Verlag, 2001.

[34] A.Y. Orbak, O.S. Turkay, E. Eskinat, and K. Youcef-Toumi, "Model reduction in the physical domain," in *Proc. IMechE 2003*, 2003, pp. 481–496.

[35] K. Zhou, J.C. Doyle, and K. Glover, *Robust and Optimal Control*, Upper Saddle River, NJ: Prentice-Hall, 1996.

[36] B.C. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Transactions on Automatic Control*, vol. 26, no. 1, pp. 17–32, February 1981.

[37] Y. Liu and B.D.O. Anderson, "Singular perturbation approximation of balanced systems," in *IEEE Proc. 28th Conference on Decision and Control*, 1989, pp. 1355–1360.

[38] K. Glover, "All optimal Hankel-norm approximations of linear multivariable systems and their L^{∞} error bounds," *Int. J. of Control*, vol. 39, no. 6, pp. 1115–1193, June 1984.

[39] J.L. Lumley, "The structure of inhomogeneous turbulence," in *Atmospheric Turbulence and Wave Propagation*, pp. 166–178. Moscow: Nauka, 1967.

[40] M. Rathinam and L.R. Petzold, "A new look at proper orthogonal decomposition," *SIAM J. Numer. Anal.*, vol. 41, no. 5, pp. 1893–1925, October 2003.

[41] S.S. Ravindran, "A reduced-order approach for optimal control of fluids using proper orthogonal decomposition," *Internat. J. Numer. Methods Fluids*, vol. 34, no. 5, pp. 425–448, November 2000.

[42] M. Aoki, "Some approximation methods for estimation and control of large scale systems," *IEEE Transactions on Automatic Control*, vol. 23, no. 2, pp. 173–182, April 1978.

[43] J.M. Siret, G. Michailesco, and P. Bertrand, "Representation of linear dynamical system by aggregated models," *International Journal of Control*, vol. 26, no. 1, pp. 121–128, July 1977.

[44] Y. Shamash, "Stable reduced order model using padé-type approximation," *IEEE Transactions on Automatic Control*, vol. 19, no. 5, pp. 615–616, October 1974.

[45] K. Gallivan, A. Vandendorpe, and P. Van Dooren, "Sylvester equations and projection-based model reduction," *Journal of Computational and Applied Mathematics*, vol. 162, no. 1, pp. 213–229, January 2004.

[46] A. Kumar, P.D. Christofides, and P. Daoutidis, "Singular perturbation modeling of nonlinear processes with nonexplicit timescale multiplicity," *Chemical Engineering Science*, vol. 53, no. 8, pp. 1491–1504, April 1998.

[47] B. Lohmann, "Application of model order reduction to a hydropneumatic vehicle suspension," *IEEE Transactions on Control Systems Technology*, vol. 3, no. 1, pp. 102–109, March 1995.

[48] J. Hahn and T.F. Edgar, "An improved approach to the reduction of nonlinear models using balancing of empirical gramians," *Computers Chem. Engng.*, vol. 26, no. 10, pp. 1379–1397, October 2002.

[49] F.W. Fairman, *Linear Control Theory: The State Space Approach*, New York: John Wiley & Sons, 1998.

[50] A. Newman and P.S. Krishnaprasad, "Nonlinear model reduction for RTCVD," in *IEEE Proc. 32nd Conference on Information Sciences and Systems*, 1998.

[51] P.V. Kokotovic and R.A. Yackel, "Singular perturbation of linear regulators: Basic theorems," *IEEE Transactions on Automatic Control*, vol. 17, no. 1, pp. 29–37, February 1972.

[52] R.D. Milne, "The analysis of weakly coupled dynamical systems," *International Journal of Control*, vol. 2, no. 2, pp. 171–199, August 1965.

[53] M.J. Greenacre, *Theory and application of correspondence analysis*, London: Academic Press, 1984.

[54] N. Aubry, P. Holmes, J.L. Lumley, and E. Stone, "The dynamics of coherent structures in the wall region of a turbulent boundary layer," *J. Fluid Mech.*, vol. 192, pp. 115–173, July 1988.

[55] N. Aubry, R. Guyonnet, and R. Lima, "Spatio-temperal analysis of complex signals: theory and applications," *J. Stat. Phys.*, vol. 64, no. 3/4, pp. 683–739, August 1991.

[56] N. Aubry, W. Lian, and E.S. Titi, "Preserving symmetries in the proper orthogonal decomposition," *SIAM J. Sci. Compu.*, vol. 14, no. 2, pp. 483–505, March 1993.

[57] M.D. Graham and I.G. Kevrekidis, "Alternative approaches to the karhunen-loève decomposition for model reduction and data analysis," *Computers Chem. Engng.*, vol. 20, no. 5, pp. 495–506, May 1996.

[58] L. Ljung and T. Glad, *Modeling of Dynamic Systems*, Englewood Cliffs, NJ: Prentice Hall, 1994.

[59] L. Ljung, *System Identification - Theory for the User*, Upper Saddle River, NJ: Prentice Hall, 1999.

[60] T. Söderström and P. Stoica, *System Identification*, London: Prentice Hall International, 1989.

[61] I. Aleksander and H. Morton, *An introduction to neural computing*, Boston, MA: International Thomson Computer Press, 1990.

[62] L.V. Fausett, *Fundamentals of neural networks*, Upper Saddle River, NJ: Prentice Hall, 1994.

[63] S.S. Haykin, *Neural networks: a comprehensive foundation*, Upper Saddle River, NJ: Prentice Hall, 1998.

[64] D.E. Rumelhart, D.E. Hinton, and R.J. Williams, *Learning internal representations by error propagation in parallel distributed processing*, pp. 318–362, Cambridge, MA: MIT Press, 1986.

[65] P. Geladi and B.R. Kowalski, "Partial least squares regression: A tutorial," *Analytica Chimica Acta.*, vol. 185, pp. 1–17, July 1986.

[66] D. Degenring, C. Froemel, G. Dikta, and R. Takors, "Sensitivity analysis for the reduction of complex metabolism models," *Journal of Process Control*, vol. 14, no. 7, pp. 729–745, October 2004.

[67] Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and Vision Computing*, vol. 15, no. 1, pp. 59–76, January 1997.

[68] P.D. Roberts and T.W.C. Williams, "On an algorithm for combined system optimisation and parameter estimation," *Automatica*, vol. 17, no. 1, pp. 199–209, January 1981.

[69] S. Chen, M. Brdy, and P.D. Roberts, "An integrated system optimisation and parameter estimation technique for hierarchical control of steady-state systems," *Int. J. Syst. Sci.*, vol. 17, no. 8, pp. 1209–1228, August 1986.

[70] S. Krishnan, G. W. Barton, and J. D. Perkin, "Robust parameter estimation in online optimization - part i: Methodology and simulated case study," *Computers Chem. Engng.*, vol. 16, no. 6, pp. 545–562, June 1992.

[71] K.Z. Yao, B.M. Shaw, B. Kou, K.B. Mcauley, and D.W. Bacon, "Modeling ethylene/butene copolymerization with multi-site catalysts: parameter estimability and experimental design," *Polymer Reaction Engineering*, vol. 11, no. 3, pp. 563–588, 2003.

[72] K.G. Gadkar, J. Varner, and F.J. Doyle III, "Model identification of signal transduction networks from data using a state regulator problem," *System Biology*, vol. 2, no. 1, pp. 17–30, March 2005.

[73] G.C. Goodwin and R.L. Payne, *Dynamic system identification: Experimental design and data analysis*, San Diego, CA: Academic Press, 1977.

[74] J.A. Jacquez and P. Greif, "Numerical parameter identifiability and estimability: Integrating identifiability, estimability, and optimal sampling design," *Math. Biosci.*, vol. 77, no. 1-2, pp. 201–227, December 1985.

[75] E. Walter and L. Pronzato, "Qualitative and quantitative experimental design for phenomenological models - a survey," *Automatica*, vol. 26, no. 2, pp. 195–213,

March 1990.

[76] S. R. Weijers and P.A. Vanrolleghem, "A procedure for selecting best identifiable parameters in calibrating activated sludge model no. 1 to full-scale plant data," *Water Science and Technology*, vol. 36, no. 5, pp. 69–79, 1997.

[77] R. Li, M.A. Henson, and M.J. Kurtz, "Selection of model parameters for off-line parameter estimation," *IEEE Transactions Control Systems Technology*, vol. 12, no. 3, pp. 402–412, May 2004.

[78] A. Saltelli, K. Chan, and E. M. Scott, *Sensitivity Analysis*, New York: John Wiley & Sons, 2000.

[79] K.E. Brenan, S.L. Campbell, and L.R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, New York: North-Holland, 1989.

[80] G.D. Byrne and P.R. Ponzi, "Differential-algebraic systems, their application and solutions," *Computers and Chemical Engineering*, vol. 12, no. 5, pp. 377–382, May 1988.

[81] P.D. Christofides and P. Daoutidis, "Feedback control of two-time-scale nonlinear systems," *International Journal of Control*, vol. 63, no. 5, pp. 965–994, March 1996.

[82] H.P. Löffler and W. Marquardt, "Order reduction of nonlinear differential-algebraic process models," *Journal of Process Control*, vol. 1, no. 1, pp. 32–40, January 1991.

[83] P. Daoutidis and A. Kumar, *Control of nonlinear differential algebraic equation systems*, New York: Chapman & Hall/CRC, 1999.

[84] K. Levenberg, "A method for the solution of certain problems in least squares," *Quarterly of Applied Mathematics*, vol. 2, no. 2, pp. 164–168, July 1944.

[85] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J. Appl. Math.*, vol. 11, no. 2, pp. 431–441, June 1963.

[86] H. Wold, "Partial least squares," in *Encyclopedia of Statistical Sciences*, pp. 581–591. New York: Wiley, 1985.

[87] G.López-Cueto, M. Ostra, and C. Ubide, "Linear and non-linear multivariate calibration methods for multicomponent kinetic determinations in cases of severe non-linearity," *Analytica Chimica Acta*, vol. 405, no. 1-2, pp. 285–295, January 2000.

[88] P. J. Van Welsenaere and G. F. Froment, "Parametric sensitivity and runaway in fixed bed catalytic reactors," *Chemical Engineering Science*, vol. 25, pp. 1503–1516, October 1970.

[89] M.E. Sperline, *Ordinary differential equations: solutions and applications*, Washington, D.C.: University Press of America, 1981.

[90] G.S. Fishman, *Monte Carlo: concepts, algorithms, and applications*, New York: Springer-Verlag, 1996.

[91] M.A. Henson and D.E. Seborg, "Input-output linearization of general nonlinear processes," *AIChE Journal*, vol. 36, no. 11, pp. 1753–1757, November 1990.

[92] A.K. Singh, A. Jayaraman, and J. Hahn, "Modeling regulatory mechanisms in il-6 signal transduction in hepatocytes," *Biotechnology & Bioengineering*, In Press, 2006.

[93] S. Yamada, S. Shiono, A. Joo, and A. Yoshimura, "Control mechanism of jak/stat signal transduction pathway," *FEBS Letters*, vol. 534, no. 1-3, pp. 190–196, January 2003.

[94] M.A. Henson, "Nonlinear model predictive control: Current status and future directions," *Computers Chem. Engng.*, vol. 23, no. 2, pp. 187–202, December 1998.

[95] S.J. Qin and T.A. Badgwell, "An overview of nonlinear model predictive control applications," in *Nonlinear model predictive control*, pp. 369–392. Boston, MA: Birkhäuser Verlag, 2000.

[96] E.S. Meadows and J.B. Rawlings, "Model predictive control," in *Nonlinear process control*, pp. 233–310. Englewood Cliffs, NJ: Prentice-Hall, 1997.

[97] M.A. Henson and D.E. Seborg, *Nonlinear Process Control*, Upper Saddle River, NJ: Prentice-Hall, 1997.

APPENDIX A

DISTILLATION COLUMN MODEL

The following equations describe the distillation column used for separation of a binary mixture.

Component balances (differential equations):

Condenser:

$$\frac{dx_{A,1}}{dt} = \frac{1}{A_{Cond}} V(y_{A,2} - x_{A,1})$$

Trays in the rectification section (i= 2, ..., 16):

$$\frac{dx_{A,i}}{dt} = \frac{1}{A_{Tray}} [L_1(x_{A,i-1} - x_{A,i}) - V(y_{A,i} - y_{A,i+1})]$$

Feed tray:

$$\frac{dx_{A,17}}{dt} = \frac{1}{A_{Tray}} [F x_{A,Feed} + L_1 x_{A,16} - L_2 x_{A,17} - V(y_{A,17} - y_{A,18})]$$

Trays in the stripping section (i= 18, ..., 31):

$$\frac{dx_{A,i}}{dt} = \frac{1}{A_{Tray}} [L_2(x_{A,i-1} - x_{A,i}) - V(y_{A,i} - y_{A,i+1})]$$

Reboiler:

$$\frac{dx_{A,32}}{dt} = \frac{1}{A_{Reboiler}} [L_2 x_{A,31} - (F - D) x_{A,32} - V y_{A,32}]$$

Implicit algebraic equations:

For condenser, all trays, and reboiler:

$$0 = [x_{A,i}\gamma_{A,i} P_{SatA,i} + (1 - x_{A,i})\gamma_{B,i} P_{SatB,i} - P]/P, \qquad i = 1, \cdots, 32$$

Further equations:

$$V = L_1 + D$$

$$L_2 = F + L_1$$

$$L_1 = RR \times D$$

$$P_{SatA,i} = \exp[a_1 + a_2/T_i + a_3 \log(T_i) + a_4 T_i^{a_5}], \quad i = 1, \cdots, 32$$

$$P_{SatB,i} = \exp[b_1 + b_2/T_i + b_3 \log(T_i) + b_4 T_i^{b_5}], \quad i = 1, \cdots, 32$$

$$\gamma_{A,i} = \exp\left\{-\log\left[x_{A,i} + L_{12}(1 - x_{A,i})\right] + (1 - x_{A,i})\left[\frac{L_{12}}{x_{A,i} + L_{12}(1 - x_{A,i})} - \frac{L_{21}}{L_{21}x_{A,i} + (1 - x_{A,i})}\right]\right\}$$
$$i = 1, \cdots, 32$$

$$\gamma_{B,i} = \exp\left\{-\log\left[(1 - x_{A,i}) + L_{21}x_{A,i}\right] + x_{A,i}\left[\frac{L_{21}}{L_{21}x_{A,i} + (1 - x_{A,i})} - \frac{L_{12}}{x_{A,i} + L_{12}(1 - x_{A,i})}\right]\right\}$$
$$i = 1, \cdots, 32$$

$$y_{A,i} = x_{A,i}\gamma_{A,i}P_{SatA,i}/P, \quad i = 1, \cdots, 32$$

Variable description:

A_{Cond}	total molar holdup in the condenser
A_{Tray}	total molar holdup on each tray
$A_{Reboiler}$	total molar holdup in the reboiler
F	feed flowrate
D	distillate flowrate
RR	reflux ratio
L_1	flowrate of the liquid in the rectification section
L_2	flowrate of the liquid in the stripping section
V	vapor flowrate in the column
$x_{A,i}$	liquid composition of component A on the $i-$th stage
$x_{A,Feed}$	feed composition of component A
$y_{A,i}$	vapor composition of component A on the $i-$th stage
T_i	temperature of $i-$th tray
$\alpha_{A,B}$	relative volatility
$\gamma_{A,i}$	activity coefficient of component A
$\gamma_{B,i}$	activity coefficient of component B
$P_{SatA,i}$	saturated vapor pressures of component A
$P_{SatB,i}$	saturated vapor pressure of component B
P	pressure
L_{12}, L_{21}	Wilson activity coefficient model parameters
$a_1, a_2, \cdots, a_5$	parameters from DIPPR Database (empirical fit) for component A
$b_1, b_2, \cdots, b_5$	parameters from DIPPR Database (empirical fit) for component B

Table VI. Operating conditions and parameter values

F	0.4 mol/L	$x_{A,Feed}$	0.5	D	0.2 mol/L	RR	3.0
A_{Cond}	0.5	A_{Tray}	0.25	$A_{Reboiler}$	1.0	P	101000 Pa
a_1	51.087	a_2	-5226.4	a_3	-4.2278	a_4	9.7554e(-18)
a_5	6.0	b_1	87.829	b_2	-6996.4	b_3	-9.8802
b_4	7.2099e(-6)	b_5	2.0	L_{12}	1.618147	L_{21}	0.502535

APPENDIX B

CONTINUOUS STIRRED TANK REACTOR IN SERIES

This model consists of four nonlinear ordinary differential equations [91]:

$$\frac{dC_{A1}}{dt} = \frac{q}{V_1}(C_{Af} - C_{A1}) - k_0 C_{A1} \exp(-\frac{E}{RT_1})$$

$$\frac{dT_1}{dt} = \frac{q}{V_1}(T_f - T_1) + \frac{(-\triangle H)k_0 C_{A1}}{\rho C_p} \exp(-\frac{E}{RT_1}) +$$

$$\frac{\rho_c C_{pc}}{\rho C_p V_1} q_c \left[1 - \exp(-\frac{UA_1}{q_c \rho_c C_{pc}})\right] (T_{cf} - T_1)$$

$$\frac{dC_{A2}}{dt} = \frac{q}{V_2}(C_{A1} - C_{A2}) - k_0 C_{A2} \exp(-\frac{E}{RT_2})$$

$$\frac{dT_2}{dt} = \frac{q}{V_2}(T_1 - T_2) + \frac{(-\triangle H)k_0 C_{A2}}{\rho C_p} \exp(-\frac{E}{RT_2}) +$$

$$\frac{\rho_c C_{pc}}{\rho C_p V_2} q_c \left[1 - \exp(-\frac{UA_2}{q_c \rho_c C_{pc}})\right] \left[(T_1 - T_2 + \exp(-\frac{UA_1}{q_c \rho_c C_{pc}})(T_c f - T_1)\right]$$

APPENDIX C

NONLINEAR MODEL REDUCTION ROUTINES IN MATLAB

Balancing is an important approach for model reduction of controlled systems which consists of two steps: the first step is to find a transformation that balances the controllability and observability gramians [9] in order to determine which states have the greatest contribution to the input-output behavior. The next step is to perform a Galerkin projection onto the states corresponding to the largest singular values of the balanced gramians for the region of interest in state-space [13]. In order to perform model reduction via balancing, three components are required: a controllability gramian, an observability gramian, and a transformation matrix which balances the system. For control-affine nonlinear systems, it is possible to compute empirical gramians instead of linear gramians. For general nonlinear systems, covariance matrices should be used. The MATLAB code presented on this web page includes routines for computing empirical gramians or covariance matrices and a routine for computing the transformation matrix that balances the system. The algorithms used to compute these quantities were presented in [12], [48]. Detailed descriptions of these routines are presented. Additionally, several examples are given as a demonstration on how to use these routines.

A. Descriptions of Routines

1. Routine for Computing Controllability Gramian or Covariance Matrix

function yhat = ctrl_gram_cov (OdeFcn, Tspan, ParaVector, Cm, flag)

This function computes the controllability gramian or the controllability covariance matrix for stable dynamical systems by making use of data collected along system

trajectories. To use this function, the following input arguments are required:

- OdeFcn: the ode function of the system written by MATLAB. The form can be: Function xdot = OdeFcn(t, x)
- Tspan: [start time, end time, sampleLength], which is used for integration of the OdeFcn. The end time should be long enough for the system to get close to a steady state. This value can be estimated by simulating the system and checking how long it takes to reach the steady state.
- ParaVector: system parameters, including InputNumber p, StateNumber n, OutputNumber k, Orientation Number r (generally r=2) and length q
- Cm: CmValue, and s = size(Cm), the perturbation size of inputs.
- flag == 0, gramian; flag ∼= 0, covariance matrix
- The output argument yhat represents the controllability gramian (flag = 0) or the controllability covariance matrix (flag ∼= 0)

This routine applies impulse input perturbations for computing empirical controllability gramians or applies step input perturbations for computing controllability covariance matrices.

2. Routine for Computing Observability Gramian or Covariance Matrix

function xhat = obsv_gram_cov (OdeFcn, Tspan, ParaVector, Cm, OutputIndex, xss, flag)

This function computes the observability gramian or observability covariance matrix for stable dynamical systems by making use of data collected along system trajectories. To use this function, the following input arguments are required:

- OdeFcn, Tspan, ParaVector, Cm, flag: these arguments are the same as those in the function ctrl_gram_cov.

- OutputIndex: the indices of output corresponding to states. Here, assume the outputs are one or more of the state exactly, for instance, if there is one output which is the first state, then the OutputIndex is 1. And if there are two outputs, one output is the first state and the other is the fifth state, then the OutputIndex term should be [1 5].

- xss: the initial condition of the states.

- The output argument xhat represents the observability gramian (flag = 0) or covariance matrix (flag ∼= 0)

This routine applies initial condition perturbations in each state to compute the empirical observability gramian or the observability covariance matrix.

3. Routines for Unscaled Systems

Routine 1 and 2 are designed to compute gramians or covariance matrices for scaled systems. For unscaled systems, the following two routines can be used:

function yhat = ctrl_gram_cov_unscaled (OdeFcn, Tspan, ParaVector, Cm, uss, xss, flag)

function xhat = obsv_gram_cov_unscaled (OdeFcn, Tspan, ParaVector, Cm, OutputIndex, uss, xss, flag)

The only difference between the routines for scaled systems and unscaled systems lies in the input arguments. For ctrl_gram_cov_unscaled, uss and xss are needed which represent the input values at steady state and the initial conditions. For obsv_gram_cov_unscaled, uss and xss are also needed.

4. Routine for Computing the Transformation Matrix for Balancing

function [Trans, invTrans, Wc, Wo, svd_Wc, svd_Wo] = bal_realization(yhat, xhat, n)

This function computes the transformation matrix that balances the system and also computes the balanced gramians (or the balanced covariance matrices) and the Hankel singular values, when the controllability and observability gramian or covariance matrix are known. The input argument yhat represents the (empirical) controllability gramian or covariance matrix, xhat refers to the (empirical) observablility gramian or covariance matrix, and n is the number of states. The output arguments are as follows:

- Trans: the transformation matrix
- invTrans: the inverse of the transformation matrix
- Wc: balanced controllability gramian/covariance matrix
- Wo: balanced observability gramian/covariance matrix
- svd_Wc: singular values
- svd_Wo: singular values

B. Model Reduction Procedure

1. Compute Necessary Quantities for Model Reduction

Gramians (or covariance matrices) and the transformation transformation are required for balanced model reduction. These quantities can be computed by use of the above mentioned routines. The routines for unscaled systems are mainly for

verifying these routines by comparison against the MATLAB commands for linear systems. In practice, the routines for scaled systems are applied as it needs to be taken into account that a state changing by orders of magnitude can be more important than a state which hardly changes, even though its steady state may have a smaller absolute value. One important problem to discuss here is how a scaled system is obtained.

For a general nonlinear system

$$\begin{aligned} \dot{x} &= f(x,u) \\ y &= g(x,u) \end{aligned} \tag{C.1}$$

Let x_{ss}, u_{ss}represent the steady state values of x and u. Introduce two quantities: $T_x = diag(x_{ss})$and $T_u = diag(u_{ss})$, then the scaled system is:

$$\left.\begin{aligned} \dot{x} &= f(x,u) \\ y &= g(x,u) \end{aligned}\right\} \xrightarrow{\tilde{x}=T_x^{-1}x, \tilde{u}=T_u^{-1}} \left\{\begin{aligned} \dot{\tilde{x}} &= T_x^{-1} f(T_x\tilde{x}, T_u\tilde{u}) \\ y &= g(T_x\tilde{x}, T_u\tilde{u}) \end{aligned}\right. \tag{C.2}$$

2. Balanced System

With the balancing transformation matrix T (= "Trans"), the balanced system is given by:

$$\left.\begin{aligned} \dot{x} &= f(x,u) \\ y &= g(x,u) \end{aligned}\right\} \xrightarrow{\bar{x}=Tx} \left\{\begin{aligned} \dot{\bar{x}} &= T f(T^{-1}\bar{x}, u) \\ y &= g(T^{-1}\bar{x}, u) \end{aligned}\right. \tag{C.3}$$

Generally the scaled system is used for the computation of T resulting in:

$$\left.\begin{aligned} \dot{\tilde{x}} &= T_x^{-1} f(T_x\tilde{x}, T_u\tilde{u}) \\ y &= g(T_x\tilde{x}, T_u\tilde{u}) \end{aligned}\right\} \xrightarrow{\bar{x}=T\tilde{x}} \left\{\begin{aligned} \dot{\bar{x}} &= T T_x^{-1} f(T_x T^{-1}\bar{x}, T_u\tilde{u}) \\ y &= g(T_x T^{-1}\bar{x}, T_u\tilde{u}) \end{aligned}\right. \tag{C.4}$$

3. Model Reduction Implementation

After obtaining a balanced system, it needs to be determined how many states can be reduced and which reduction method to use. The former problem can be solved by a trial and error procedure while taking into account the magnitude of the Hankel singular values of the states to be reduced. The answer to the latter question is that balanced truncation is the method of choice for nonlinear systems as other techniques, e.g. balanced residualization, can lead to systems which may be smaller but significantly harder to solve.

The state vector of a balanced system can be divided into two parts: relatively important states $\bar{x}_1$ and relatively unimportant states $\bar{x}_2$. The reduced model is:

$$\begin{aligned} \dot{\bar{x}}_1 &= PTf(T^{-1}\bar{x}, u) \\ \dot{\bar{x}}_2 &= \bar{x}_{2ss} \\ y &= g(T^{-1}\bar{x}, u) \end{aligned} \tag{C.5}$$

where P =[I_k 0], k is the number of retained states.

Or for the scaled system, the reduced model is:

$$\begin{aligned} \dot{\tilde{x}}_1 &= PTT_x^{-1}f(T_xT^{-1}\bar{x}, T_u\tilde{u}) \\ \dot{\tilde{x}}_2 &= \dot{\tilde{x}}_{2ss} \\ y &= g(T_xT^{-1}\bar{x}, T_u\tilde{u}) \end{aligned} \tag{C.6}$$

C. Demonstration Examples

Several case studies are presented in this documentation. The procedure is mainly demonstrated on the first example and lesser detail is provided for the remaining two examples.

1. Example 1: a Linear System

This example is included to show that the presented method will simplify to linear balanced truncation if the system under study is linear. Also, this example can be used to illustrate the procedure.

a. Model Description and Input Arguments Determining

$$\dot{x} = Ax + Bu \ , \qquad x \in \Re^n$$
$$y = Cx + Du$$

where, $A = \begin{bmatrix} \text{-2.0000 0 0} \\ \text{1.0000 -1.1000 0} \\ \text{0 0.1000 -1.0000} \end{bmatrix}$, $B = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$, $C = [0\ 0\ 1]$, $D = [0]$.

If the initial value for input $u_{ss} = 2$, then $x_{ss} = -A^{-1}Bu_{ss} = \begin{bmatrix} 2.0000 \\ 1.8182 \\ 0.1818 \end{bmatrix}$.

Therefore, the quantities for scaling the system are:

$T_x = diag(x_{ss}) = \begin{bmatrix} \text{2.0000 0 0} \\ \text{0 1.8182 0} \\ \text{0 0 0.1818} \end{bmatrix}$ and $T_u = diag(u_{ss}) = [2]$.

The scaled system is then given by $\tilde{A} = T_x^{-1}AT_x = \begin{bmatrix} \text{-2.0000 0 0} \\ \text{1.1000 -1.1000 0} \\ \text{0 1.0000 -1.0000} \end{bmatrix}$,

$\tilde{B} = T_x^{-1}BT_u = \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix}$, $\tilde{C} = CT_x = [\ 0\ 0\ 0.1818]$, $\tilde{D} = [0]$.

The code to obtain the scaled system from the unscaled system is:

---

tx = diag(xss);

tu = diag(uss);

Abar = inv(tx)*A*tx;

Bbar = inv(tx)*B*tu;

Cbar = C*tx;

Dbar = [0];

sysbar = ss(Abar,Bbar,Cbar,Dbar);

This system has 1 input, 1 output, and 3 states. Therefore, p = 1, n = 3, k = 1. Select r = 2, q = 1000. Then "ParaVector" = [1 3 1 2 1000]. uss = 2, xss = [2.0 1.8182 0.1818]^T, and "OutputIndex" = 3.

The step response of this system is shown as Fig. 35.

From Fig. 35, it can be seen that it requires about 6 seconds for the system to get close to steady state, therefore, the variable of "end time" can be 6 if the "start time" is 0. "SampleLength" can be 0.1. Therefore, "Tspan" = [0 6 0.1].

b. Model Reduction Demonstration

(i) Compute empirical controllability gramian

The input arguments are:

"ParaVector" = [1 3 1 2 1000], "Tspan" = [0 6 0.1], "Cm" = 0.1, s = size(Cm) = 1, flag = 0.

Then the controllability gramian can be computed as:

yhat = ctrl_gram_cov (OdeFcn, Tspan, ParaVector, Cm, flag) for the scaled system.

yhat = ctrl_gram_cov_unscaled (OdeFcn, Tspan, ParaVector, Cm, flag) for the

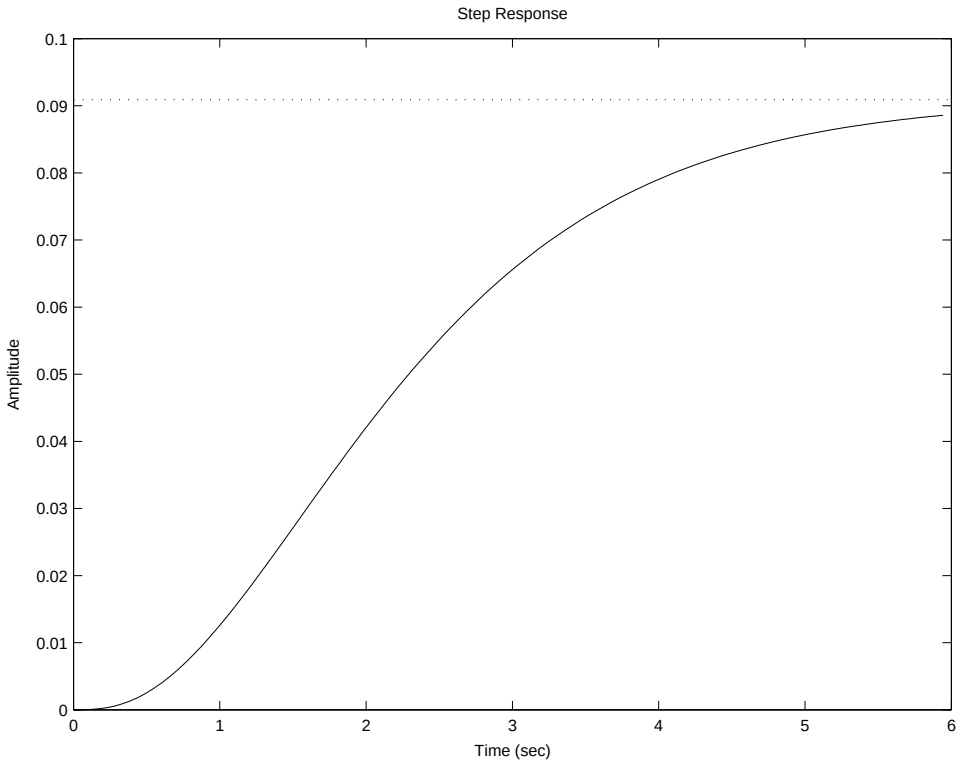

Fig. 35. Step response of example 1

unscaled system.

If the controllability covariance matrix is desired, then the "flag" needs to be set to any integer other than 0, for instance, "flag" = 1.

The result of the (empirical) controllability gramian for the unscaled system is

$$Ctrl\_gram\_unscaled = \begin{bmatrix} 1.0053 & 0.3232 & 0.0109 \\ 0.3232 & 0.2911 & 0.0190 \\ 0.0109 & 0.0190 & 0.0019 \end{bmatrix}$$

By use of the MATLAB command "gram", the controllability gramian for the

unscaled system is [command: gram(sys(A,B,C,D), 'c')]

$$Ctrl\_gram\_unscaled\_matlab = \begin{bmatrix} 1.0000 & 0.3226 & 0.0108 \\ 0.3226 & 0.2933 & 0.0191 \\ 0.0108 & 0.0191 & 0.0019 \end{bmatrix}$$

The result of the (empirical) controllability gramian for the scaled system is

$$Ctrl\_gram\_scaled = \begin{bmatrix} 0.9995 & 0.3561 & 0.1199 \\ 0.3561 & 0.3525 & 0.2299 \\ 0.1199 & 0.2299 & 0.2295 \end{bmatrix}$$

And by use of the MATLAB command "gram", the controllability gramian for the scaled system is [command: gram(sys($\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$), 'c')]

$$Ctrl\_gram\_scaled\_matlab = \begin{bmatrix} 1.0000 & 0.3548 & 0.1183 \\ 0.3548 & 0.3548 & 0.2309 \\ 0.1183 & 0.2309 & 0.2309 \end{bmatrix}$$

It can be seen that *Ctrl_gram_unscaled* and *Ctrl_gram_unscaled_matlab* return results that are very close to one another. There only exist a very small difference between them. Similarly, the results returned by *Ctrl_gram_scaled* and *Ctrl_gram_scaled_matlab* are also very close.

(ii) Compute observability gramian

With all input arguments already known, the observability gramian is:

xhat = obsv_gram_cov (OdeFcn, Tspan, ParaVector, Cm, OutputIndex, xss, flag) for scaled system, or

xhat = obsv_gram_cov_unscaled (OdeFcn, Tspan, ParaVector, Cm, OutputIndex, uss, xss, flag) for unscaled systems.

The result returned for the observability gramian for the unscaled system is

$$Obsv\_gram\_unscaled = \begin{bmatrix} 0.0005 & 0.0009 & 0.0079 \\ 0.0009 & 0.0021 & 0.0235 \\ 0.0079 & 0.0235 & 0.5528 \end{bmatrix}$$

By use of the MATLAB command "gram", the observability gramian for unscaled system is [command: gram(sys(A,B,C,D), 'o')]

$$Obsv\_gram\_unscaled\_matlab = \begin{bmatrix} 0.0005 & 0.0010 & 0.0079 \\ 0.0010 & 0.0022 & 0.0238 \\ 0.0079 & 0.0238 & 0.5000 \end{bmatrix}$$

The result of observability gramian for the scaled system is

$$Obsv\_gram\_scaled = \begin{bmatrix} 0.0019 & 0.0034 & 0.0029 \\ 0.0034 & 0.0071 & 0.0078 \\ 0.0029 & 0.0078 & 0.0184 \end{bmatrix}$$

And by use of the MATLAB command "gram", the observability gramian for the scaled system is [command: gram(sys($\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$), 'o')]

$$Obsv\_gram\_scaled\_matlab = \begin{bmatrix} 0.0019 & 0.0035 & 0.0029 \\ 0.0035 & 0.0072 & 0.0079 \\ 0.0029 & 0.0079 & 0.0165 \end{bmatrix}$$

It can be seen that *Obsv_gram_unscaled* is quite close to *Obsv_gram_matlab* considering the assumptions made above. The same is true for the commands *Obsvl_gram_scaled* and *Obsv_gram_scaled_matlab*.

(iii) Compute balanced transformation matrix

After the controllability gramian yhat and observability gramian xhat are ob-

tained, the transformation matrix can be computed by the following command:

[Trans, invTrans, Wc, Wo, svd_Wc, svd_Wo] = bal_realization(yhat, xhat, n)

The result returned for the transformation matrix for the unscaled system is:

$$Trans\_unscaled = \begin{bmatrix} \text{-0.0760 -0.1808 -2.6316} \\ \text{0.0886 0.0983 -2.9249} \\ \text{0.0471 -0.1658 1.4489} \end{bmatrix}$$

By use of the MATLAB command "balreal", the transformation matrix for the unscaled system is [command: balreal(sys(A,B,C,D))]

$$Trans\_unscaled\_matlab = \begin{bmatrix} \text{-0.0773 -0.1845 -2.5300} \\ \text{0.0887 0.0859 -2.7758} \\ \text{0.0434 -0.1528 1.3434} \end{bmatrix}$$

These quantities are very close.

Also, for the scaled system, the result is:

$$Trans\_scaled = \begin{bmatrix} \text{-0.1071 -0.2322 -0.3396} \\ \text{-0.1256 -0.1270 0.3753} \\ \text{-0.0666 0.2114 -0.1839} \end{bmatrix}$$

By use of the MATLAB command "balreal", the transformation matrix for the scaled system is [command: balreal(sys($\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D}$))]

$$Trans\_scaled\_matlab = \begin{bmatrix} \text{-0.1093 -0.2372 -0.3253} \\ \text{0.1254 0.1105 -0.3569} \\ \text{0.0614 -0.1965 0.1727} \end{bmatrix}$$

These quantities are also very close despite the signs. The signs are not a problem as long as they are consistent in the whole procedure.

By comparing the results computed by the presented code with those computed directly by MATLAB, it can be seen that our code is applicable to linear system and the nonlinear gramians have reduced to linear gramians in as the system under study is linear. One point which should be mentioned is that the code for unscaled systems is used to compute these quantities and compared them with the results returned by MATLAB. However, it is important to scale the system so that the importance of each state can be compared based on a uniform standard. Therefore, the code for a scaled system should generally be used to compute gramians.

(iv) Generate the reduced order system

If the odefile of the scaled system is as follows,

```
function xdot = lin_example_scaled(t,x)
global ud uss xss a b c d;
x = diag(xss)*x;
u = diag(uss)*ud;
xdot = a*x + b*u;
xdot = diag(1./xss)*xdot;
```

---

then the reduced order system can be obtained by modifying the above ode function:

```
function xdot = lin_example_scaled_red(t,x)
global ud uss xss a b c d red_n Trans invTrans n;
x=diag(xss)*invTrans*x;
u = diag(uss)*ud;
xdot = a*x + b*u;
```

```
xdot = diag(1./xss)*xdot;
xdot(1:red_n,1) = Trans(1:red_n,:)*xdot;
xdot(red_n+1:n,1) = zeros(n - red_n,1);
```

In this ode function, Trans refers to the balanced transformation matrix, invTrans is its inverse, red_n is the number of retained states.

(v) Test the reduced order systems by comparing step response with the full-order system

It should be noted that when the reduced-order system is integrated, the initial value should be transformed by use of the balanced transformation matrix (initvalue = Trans*ones(n,1);). And the output should be transformed back to the original coordinate (y = (invTrans*y')').

For instance,

```
% retaining 2 states
red_n = 2;
% initial states
initvalue = Trans*ones(n,1);
% integrate the reduced system
[t, y] = ode15s('lin_example_scaled_red',[0 10], initvalue);
y = (invTrans*y')';
```

A comparison between the reduced system and the full order system is shown in Fig. 36.

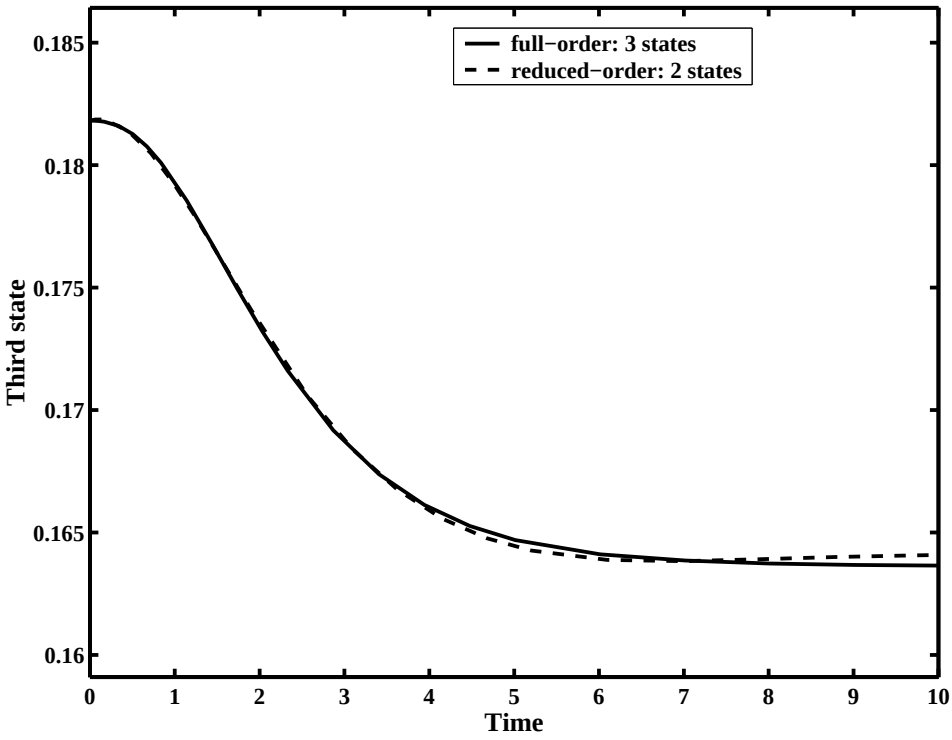

Fig. 36. Demo_1 result

2. Example 2: Distillation Column Model

Consider a distillation column with 30 trays for the separation of a binary mixture [48]. The column has 32 states and is assumed to have a constant relative volatility of 1.6, and symmetric product compositions. The feed stream is introduced at the middle of the column on stage 17 and has a composition of $x\mathrm{F} = 0.5$. Distillate and bottoms purities are $x\mathrm{D} = 0.935$ and $x\mathrm{B} = 0.065$, respectively. The reflux ratio is set to 3.0 and can be controlled and the purity of the distillate is measured.

The input arguments are:

"Tspan" = [0 125 1], "ParaVector" = [1 32 1 2 1000], cm = 0.1, uss = 3,

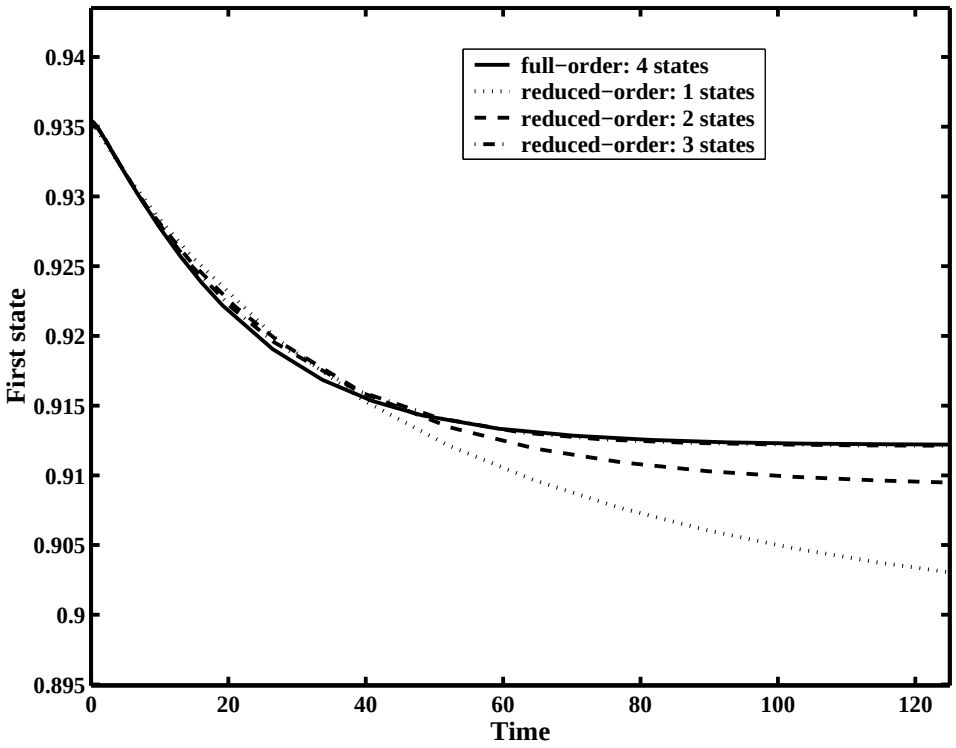

Fig. 37. Demo_2 result

OutputIndex = 1, flag = 0.

$$Svd\_scaled = \begin{bmatrix} 0.11599 \\ 0.0048545 \\ 0.0007854 \\ 0.00012426 \\ 2.1448e\text{-}005 \\ \text{others} \end{bmatrix}$$

The result is shown in Fig. 37.

3. Example 3: a CSTR System

This example is a system of two CSTRs in series, which was presented in the nonlinear process control book edited by Henson and Seborg [97]. The model has four states, one input which is the coolant flow rate q_c and one output, the effluent concentration from the second tank. The input has a nominal value of $q_{c0} = 100\text{L}/\text{ min}$.

Since this system is not control-affine, the empirical gramian cannot be obtained. In this case, the covariance matrices are computed.

The input arguments are:

"Tspan" = [0 10 0.1], "ParaVector" = [1 4 1 2 1000], cm = 0.01, uss = 100, xss = [0.0882 441.2193 0.0053 449.4746]^T, OutputIndex = 3, flag = 1.

The results for the scaled system are:

$$Ctrl\_covariance\_scaled = \begin{bmatrix} 4.3026 & -0.1809 & 7.8843 & -0.1515 \\ -0.1809 & 0.0078 & -0.3192 & 0.0060 \\ 7.8843 & -0.3192 & 15.3951 & -0.3073 \\ -0.1515 & 0.0060 & -0.3073 & 0.0063 \end{bmatrix}$$

$$Obsv\_covariance\_scaled = \begin{bmatrix} 0.0000 & 0.0011 & 0.0000 & 0.0002 \\ 0.0011 & 0.0425 & 0.0000 & 0.0087 \\ 0.0000 & 0.0000 & 0.0000 & -0.0000 \\ 0.0002 & 0.0087 & -0.0000 & 0.0029 \end{bmatrix}$$

$$Trans\_scaled = \begin{bmatrix} 0.0241 & 1.2795 & -0.0073 & 0.3472 \\ 0.0732 & 2.5367 & 0.0236 & 0.2518 \\ 0.1291 & 1.9599 & -0.0054 & 1.0097 \\ 0.1708 & 1.7444 & -0.1017 & -2.5225 \end{bmatrix}$$

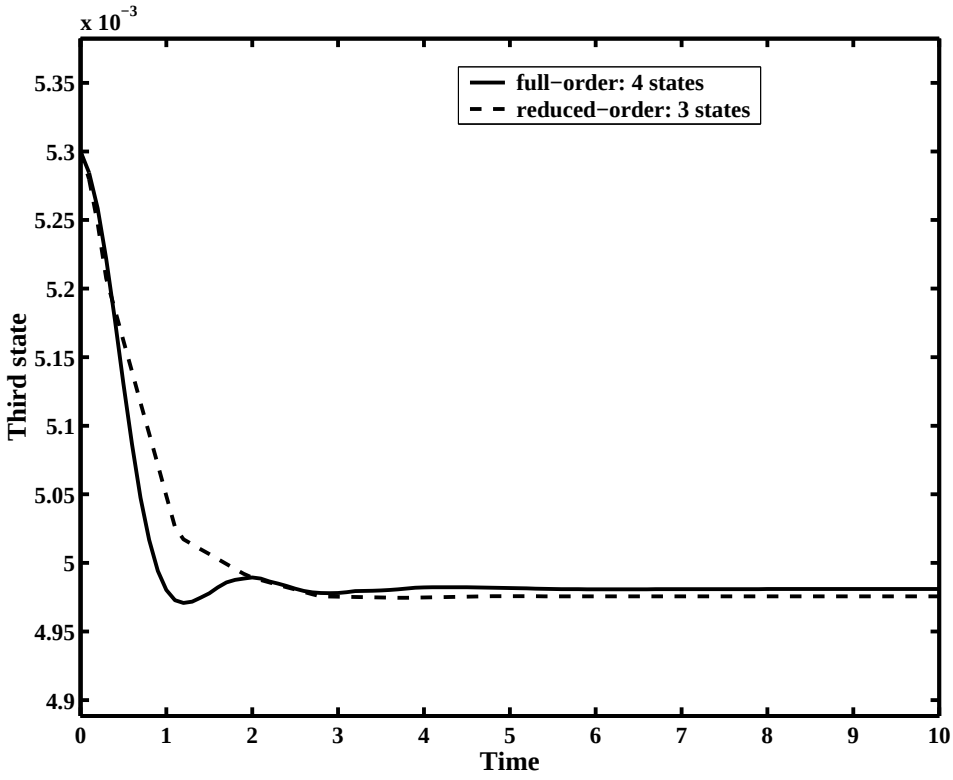

Fig. 38. Demo_3 result

$$Svd\_scaled = \begin{bmatrix} 0.0134 \\ 0.0028 \\ 0.0007 \\ 0.0001 \end{bmatrix}$$

The result is shown in Fig. 38.

D. Summary

This Appendix presents a tutorial on how to perform nonlinear model reduction via balanced truncation using the Matlab routines. Basic principles and descriptions of routine functions are presented. Several case studies are used to demonstrate these routine functions, both for linear and nonlinear systems.

VITA

Chuili Sun received her Bachelor of Science degree in Automation from Tsinghua University in June 1999. She received her Master of Science degree in control theory and control engineering from the same university in June 2002. She entered the graduate program in chemical engineering at Texas A&M University in September 2002. She received her Ph.D. degree from Texas A&M University in December 2006. Her research interests include process modeling, nonlinear model reduction and model-based process control.

Ms. Sun can be reached c/o Dr. Juergen Hahn at Artie McFerrin Department of Chemical Engineering, Texas A&M University, 3122 TAMU, College Station, TX 77843. Her email address is treacysun@gmail.com.