# Parameter Estimation in Complex Nonlinear Dynamical Systems

Dissertation
Zur Erlangung des akademischen Grades
Doktoringenieur (Dr.-Ing.)

vorgelegt der Fakultät für Informatik und Automatisierung
der Technischen Universität Ilmenau

von M.Eng. Quoc Dong Vu
geboren am 27.12.1975 in Thaibinh

**Gutachter**

1. Prof. Dr.-Ing. habil. Pu Li
2. Prof. Dr.-Ing. habil. Christoph Ament
3. Prof. Dr. rer. nat. habil. Gerhard-Wilhelm Weber

Tag der Einreichung: 13.04.2015
Tag der wissenschaftlichen Aussprache: 02.10.2015

urn:nbn:de:gbv:ilm1-2015000394

# Parameter Estimation in Complex Nonlinear Dynamical Systems

A Dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Engineering (Dr.-Ing.)


Faculty of Computer Science and Automation

by M.Eng. Quoc Dong Vu
born on 27.12.1975 in Thaibinh

## Referees

1. Prof. Dr.-Ing. habil. Pu Li
2. Prof. Dr.-Ing. habil. Christoph Ament
3. Prof. Dr. rer. nat. habil. Gerhard-Wilhelm Weber

Date of submission: 13.04.2015
Date of scientific defense: 02.10.2015

# Acknowledgements

My dissertation would certainly never have been finished without the guidance of my advisor, help from friends, and support from my family during my study at the Group of Simulation and Optimal Processes (SOP), Ilmenau University of Technology.

Firstly, I would like to express my sincere gratitude to my advisor Professor Pu Li for the constant support of research, for his ideas, motivation, patience, and continuous encouragement. His effective guidance and firm requirement promoted me in all the long time of research and writing of this dissertation.

Besides my advisor, I truly thank the rest of my thesis committee: Professor Christoph Ament, Professor Gerhard-Wilhelm Weber, Professor Horst Puta, Professor Jens Haueisen, and Professor Daniel Baumgarten, for their time to review my thesis, their insightful comments and advices to improve it.

My completion of this thesis could not have been accomplished without the support of my current colleagues in our SOP Group, namely, Dr. Siegbert Hopfgarten, Dr. Abebe Geletu, Dr. Aouss Gabash, Mr. Evgeny Lazutkin, Mr. Xujiang Huang, Mr. Jens Hollandmoritz, Mr. Björn Töpper, Mr. Duc Dai Pham; as well as formers namely, Dr. Martin Bartl, Mr. Stefan Röll, Dr. Hui Zhang, Dr. Ines Mynttinen, Dr. Michael Klöppel, Mrs. Rim Abdul Jawad, Dr. Jasem Tamimi, Mr. Wolfgang Heß and Mrs. Rita Helm, with whom my stay at TU Ilmenau became a wonderful experience.

I would like to thank Professor Hongye Su, Professor Weirong Hong, and Dr Chao Zhao at Zhejiang University for their efficient cooperation in this research.

I greatly appreciate the financial support from Vietnamese Government (Project 322) and Thuringian Graduate Support Act (ThürGFVO) that funded parts of this research work. Additional support was provided by the German Academic Exchange Service (DAAD) for the short visits to Zhejiang University of China in 2008, 2009 and 2010.

I am very thankful to all of my loving Vietnamese friends with whom I shared so much brilliant times during my stay in Germany.

Last, but not the least, I would like to express my deepest gratitude to my family: to my beloved wife and son, to my parents and to my brother and sister for their great love and support during my study.

# Abstract

The aim of this dissertation is to develop mathematical/numerical approaches to parameter estimation in nonlinear dynamical systems that are modeled by ordinary differential equations or differential algebraic equations. Parameters in mathematical models often cannot be calculated by applying existing laws of nature or measured directly and therefore they need being obtained from experimental data through an estimation step. Numerical methods to parameter estimation are challenges due to undesirable characteristics, such as stiffness, ill-conditioning and correlations among parameters of model equations that cause computational intensiveness, convergence problems as well as non-uniqueness of the solution of the parameters. The goal of this dissertation is therefore two-fold: first to develop efficient estimation strategies and numerical algorithms which should be able to efficiently solve such challenging estimation problems, including multiple data profiles and large parameter sets, and second to develop a method for identifiability analysis to identify the correlations among parameters in complex model equations.

Direct strategies to solve parameter estimation problem, dynamic optimization problems, include direct sequential, direct simultaneous, direct multiple shooting, quasi-sequential, and combined multiple shooting and collocation strategy. This dissertation especially focuses on quasi-sequential strategy and combined multiple shooting and collocation strategy. This study couples the interior point method with the quasi-sequential strategy to solve dynamic optimization problems, particularly parameter estimation problems. Furthermore, an improvement of this method is developed to solve parameter estimation problems in that the reduced-space method of interior point strategy is used. In the previous work, combined multiple shooting and collocation strategy method was proved to be efficient to solve dynamic optimization problems with all constraints of states imposed only at the nodes of the discretization

grids. In this study, an improvement to combined multiple shooting and collocation strategy is made to impose all state values on constraints at all collocation points in order to improve the quality of the dynamic optimization problems.

To improve the quality of the parameter estimation solutions, multiple data-sets of measurement data usually are used. In this study, an extension to a dynamic three-stage estimation framework is made to the parameter estimation problem with a derivation to the quasi-sequential strategy algorithm. Due to the decomposition of the optimization variables, the proposed approach can efficiently solve time-dependent parameter estimation problems with multiple data profiles. A parallel computing strategy using the message passing interface (MPI) method is also applied successfully to boost computation efficiency.

The second challenging task in parameter estimation of nonlinear dynamic models is the identifiability of the parameters. The identifiability property of a model is used to answer the question whether the estimated parameters are unique. In this thesis, a systematic approach to identify both pairwise parameter correlations and higher order interrelationships among parameters in nonlinear dynamic models is developed. The correlation information obtained in this way clarifies both structural and practical non-identifiability. Moreover, this correlation analysis also shows that a minimum number of data sets, which corresponds to the maximum number of correlated parameters among the correlation groups, with different inputs for experimental design are needed to relieve the parameter correlations. The result of this correlation analysis provides a necessary condition for experimental design in order to collect suitable measurement data for unique parameter estimation.

# Zusammenfassung

Ziel der vorliegenden Dissertationsschrift ist es, mathematische bzw. numerische Verfahren zur Parameterschätzung für nichtlineare dynamische Systeme zu entwickeln, deren Modelle in Form von gewöhnlichen Differentialgleichungen oder differential-algebraischen Gleichungen vorliegen. Derartige Modelle zu validieren gelingt in der Regel nicht, indem Naturgesetze ausgenutzt werden können, vielmehr sind häufig aufwendige Messungen erforderlich, deren Datensätze dann auszuwerten sind. Numerische Verfahren zur Parameterschätzung unterliegen solchen Herausforderungen und unerwünschten Effekten wie Steifheit, schlechter Konditionierung oder Korrelationen zwischen zu schätzenden Parametern von Modellgleichungen, die rechenaufwendig sein, aber die auch schlechte Konvergenz bzw. keine Eindeutigkeit der Schätzung aufweisen können. Die Arbeit verfolgt daher zwei Ziele: erstens effektive Schätzstrategien und numerische Algorithmen zu entwickeln, die komplexe Parameter-Schätzprobleme lösen und dazu mit multiplen Datenprofilen bzw. mit großen Datensätzen umgehen können. Zweites Ziel ist es, eine Methode zur Identifizierbarkeit für korrelierte Parameter in komplexen Modellgleichungen zu entwickeln.

Eine leistungsfähige direkte Strategie zur Lösung von Parameter-Schätzaufgaben ist die Umwandlung in ein Problem der optimalen Steuerung. Dies schließt folgende Methoden ein: direkte sequentielle und quasi-sequentielle Verfahren, direkte simultane Strategien, direkte Mehrfach-Schießverfahren und kombinierte Mehrfach-Schießverfahren mit Kollokationsmethoden. Diese Arbeit orientiert besonders auf quasi-sequentielle Verfahren und kombinierte Mehrfach-Schießverfahren mit Kollokationsmethoden. Speziell zur Lösung von Parameterschätzproblemen wurde die Innere-Punkte-Verfahren mit dem quasi-sequentielle Verfahren gekoppelt. Eine weitere Verbesserung zur Lösung von Parameterschätzproblemen konnte erreicht werden, indem die „reduced-space“ Technik der Innere-Punkte-Verfahren benutzt wurde. Die Leistungsfähigkeit der kom-

binierte Mehrfach-Schießverfahren mit Kollokationsmethoden zur Lösung von Dynamischen Optimierungsproblemen war bisher damit verbunden, dass die Zustandsbeschränkungen nur in den Knoten des Diskretisierungsgitters eingehalten werden konnten. Mit dieser Arbeit konnte die kombinierte Mehrfach-Schießverfahren mit Kollokationsmethoden verbessert werden, so dass alle Zustandsgrößen die vorgegebenen Beschränkungen in allen Kollokationspunkten einhalten, was zu einer deutlichen Verbesserung des letztlich zu lösenden Optimalsteuerungsproblems zur Parameterschätzung führt.

Um die Qualität Parameterschätzung zu verbessern, werden üblicherweise mehrfache Messdatensätze benutzt. In der vorgelegten Dissertation wurde zur Parameterschätzung eine dynamische Drei-Stufen-Strategie mit einem eingebauten quasi-sequenziellen Verfahren entwickelt. Durch die Zerlegung der Optimierungsvariablen kann das vorgeschlagene Verfahren sehr effizient zeitabhängige Parameter–Schätzaufgaben mit mehrfachen Datenprofilen lösen. Zur Steigerung der Recheneffizienz wurde darüber hinaus erfolgreich eine Parallel-Rechner Strategie eingebaut, die das sog. „message passing interface“ (MPI) nutzt.

Eine zweite Herausforderung für die Parameterschätzung nichtlinearer dynamischer Modelle betrifft die Indentifizierbarkeit der Parameter. Damit verbunden ist die Frage nach der Eindeutigkeit der geschätzten Parameter. In dieser Arbeit wird auch ein systematisches Vorgehen zur Identifizierung paarweiser Korrelationen als auch zum Erkennen von Wechselwirkungen höherer Ordnung zwischen Parametern in nichtlinearen dynamischen Systemen vorgeschlagen. Damit lässt sich sowohl die strukturelle als auch eine praktische „Nichtidentifizierbarkeit“ klären. Darüber hinaus lässt sich durch eine Korrelationsanalyse darauf schließen, welche minimale Zahl von Datensätzen mit unterschiedlichen Eingängen zum Entwurf benötigt wird, um Parameterkorrelationen auszuschließen. Dies wiederum entspricht einer maximalen Zahl von korrelierten Parametern innerhalb der Korrelations–Gruppen. Im Ergebnis der Korrelationsanalyse erhält man eine notwendige Bedingung wie viele Messdaten für eine eindeutige Parameterschätzung benötigt werden.

# Contents

# List of Figures

# List of Tables

# Abbreviations

**AEs** algebraic equations. 16

**BDF** backward differentiation formula. 35–37, 47

**CMSC** combined multiple shooting and collocation strategy. xiv, xvii, 5, 6, 27, 71, 75, 76, 111, 112, 116, 117, 119, 123–127, 129, 130, 166, 167

**CSTR** continuous stirred tank reactor. 80, 89, 100

**DAE** differential algebraic equation. 4, 5, 25, 26, 33, 35, 36, 47, 64–66, 70, 87, 90, 92, 165

**DAEs** differential algebraic equations. xiii, 1, 2, 4, 5, 11, 13–17, 22, 23, 25, 27, 31, 35, 37, 45–47, 65, 71, 87, 88, 90

**DMS** direct multiple shooting strategy. 26, 27, 70, 71

**DOP** dynamic optimization problem. 1, 24, 64, 66, 71, 75, 81, 119, 123, 165

**DOPs** dynamic optimization problems. 2, 4, 5, 25, 26, 31, 65, 75, 83, 127, 131, 166, 167

**DSM** direct simultaneous strategy. 25–27, 65, 66, 70, 71

**DSQ** direct sequential strategy. 5, 24–27, 66, 68, 70

**EIV** error-in-variables. 76, 89, 90, 102, 105

**FIM** Fisher information matrix. 3

**GA** genetic algorithm. 90

**GLS** generalized least squares estimation. 20

**HPC** high performance computing. 99, 106

**IP** interior-point. 2, 4–6, 76, 79, 82, 83, 86, 89, 90, 92, 97, 98, 105, 106, 109, 110, 114, 116, 119, 123, 166

**IVP** initial value problem. 112

**IVPs** initial value problems. 31

**ML** maximum likelihood estimation. 21

**MPC** model predictive control. 88

**MPI** message passing interface. iv, 5, 99, 100, 105, 106, 112, 119, 166

**NLP** nonlinear programming. 2, 3, 5, 24–27, 31, 62–65, 67–70, 73, 75–79, 82, 87, 89–94, 98, 105, 106, 109–111, 113, 114, 116, 123, 167

**NLPs** Nonlinear Programming Problems. 4, 31, 75, 89, 93, 106, 167

**ODE** ordinary differential equation. 4, 26, 35, 47, 59, 70, 165

**ODEs** ordinary differential equations. 1, 4, 13, 14, 16, 35, 37, 47, 120, 147

**OpenMP** Open Multi-Processing. 99, 100

**PDEs** partial differential equations. 13

**PE** parameter estimation. xiv, 1–6, 16, 75, 76, 90, 100, 101, 108, 120, 122, 123, 165–167

**PSO** particle swarm optimization. 90

**QSQ** quasi-sequential strategy. 5, 6, 26, 27, 66, 71, 106, 116, 119, 166

**RS** reduced-space. 5, 6

**SQP** sequential quadratic programming. 4, 5, 27, 68, 69, 73, 75, 82, 83, 90, 113

# Chapter 1

# Introduction

## 1.1 Research Motivation

Mathematical models have been used to describe real world systems in a vast range of engineerings, such as chemical engineering, electrical engineering, mechanical engineering, and aerospace engineering, as well as in non-technical areas of natural sciences such as chemistry, physics, biology, medicine, and geo-sciences and economics, etc. Developing compact and accurate mathematical models for dynamic systems is essential in these fields for analyzing and simulating the system dynamics and implementation of optimization and control strategies [Gevers, 2006; Ljung, 2010; Nieman et al., 1971; Äström and Eykhoff, 1971].

By applying *a priori* knowledge and existing laws of nature, mathematical models can be built in the form of ordinary differential equations or differential algebraic equations with many unknown coefficients or parameters that cannot be computed or measured directly. As a consequence, a complex DAEs or ordinary differential equations (ODEs) constrained optimization problem needs to be solved to estimate these parameters based on experimental data, leading to a parameter estimation problem.

With a popular objective function of least squares types, the PE of a DAEs system poses a dynamic optimization problem that can be solved by sophisticated numerical methods. Numerical methods to dynamic optimization problem (DOP) are challenges due to computational intensiveness and numerical difficulties that are caused by unde-

sirable characteristics of the models, such as stiffness, ill-conditioning and correlations among parameters of model equations (in the case of PE problem).

Direct numerical strategies to solve dynamic optimization problems (DOPs) use a discretization of the variables to transform the infinite-dimensional optimization problem to a finite dimensional nonlinear programming problem. In a control parameterization method, known as direct sequential strategy, only the control variables are discretized and the model equations are solved by using an appropriate integration method. In a complete discretization method, known as direct simultaneous strategy, both the state and the control variables are discretized. Some hybrid strategies utilize the advantages of both the direct sequential strategy and the direct simultaneous strategy. Modern direct numerical strategies include direct sequential strategy [Barton et al., 1998; Binder et al., 2001; Goh and Teo, 1988; Vassiliadis et al., 1994a,b], direct simultaneous strategy [Biegler, 2007; Biegler et al., 2002; Jockenhövel, 2004a], direct multiple shooting strategy [Bock and Plit, 1984; Plitt, 1981], direct quasi-sequential strategy [Hong et al., 2006], and combined multiple shooting with collocation strategy [Tamimi and Li, 2010].

In the study of Hong et al. [2006] the direct quasi-sequential strategy that coupled direct sequential with the collocation method was developed and applied successfully to a large-scale optimal control of a dynamic system. In that development, the method of sequential quadratic programming was applied to solve the resulting nonlinear programming (NLP) problems after the discretization. Recently the interior-point methods in both full-space and reduced-space modes have been well developed and widely used to solve NLP problems in mathematic as well as engineering areas due to its high efficiency [Bartlett et al., 2000; Byrd et al., 2000, 2006; Cervantes et al., 2000; Lau et al., 2009; Wächter and Biegler, 2005, 2006]. In those researches, the interior-point (IP) methods were applied to solve the DOPs in the direct simultaneous strategy. A further research on the application of the IP methods to other strategies to solve the PE problems, such as the direct quasi-sequential strategy, can be a promising task.

In [Tamimi and Li, 2010] a combined multiple shooting with collocation strategy was proved to be highly efficient to solve DOPs. In that approach, all constraints of state variables were imposed only at the nodes of the discretization grids while all their values between nodes were unconstrained. This fact can let the states violate the constraints. Therefore, a further study needs to be carried out in order to improve

the quality of the solution of the combined multiple shooting with collocation strategy strategy.

To achieve better results for parameter estimation, multiple data-sets of measurement data are usually used. In [Faber et al., 2003] a sequential approach was proposed to solve the large-scale parameter estimation problem of nonlinear steady-state models with multiple data-sets, where a nested three-stage computation was presented to decompose the problem. An extension of that approach to dynamic systems described by DAEs can be taken into account to utilize its advantages.

With an adequate discretization strategy and an appropriate NLP solver, the PE problem can be solved successfully. The next question that arises naturally is whether the estimated parameters are unique. A model with an infinite number of set of parameter solutions that give a good fit to the experimental data cannot be used. The uniqueness of the estimated parameters, which is termed as *identifiability*, depends on: (i) the characteristic of the model itself (*structural identifiability*), and (ii) the informativeness of the experimental data (*practical identifiability*).

Several approaches have been developed to assess the structural identifiability of nonlinear dynamic systems. Critical reviews of these approaches can be seen in [Chis, 2011; McLean and McAuley, 2012; Miao et al., 2011]. Structural identifiability problems may be due to insensitivities of the measured outputs to parameter changes, and/or in particular implicit functional relations between the parameters, which are termed as *parameter correlations*. Although several approaches have been developed to address structural identifiability problems, there is no approach applicable to every model [Chis, 2011]. By using these methods, the non-identifiable parameters can be shown but the cause and the type of the non-identifiability problem are still unknown. Moreover, these methods usually require strong mathematic background and expertize that can be difficult for modelers to handle. Due to their disadvantages, the application of existing approaches to high dimensional complex models can be limited [McLean and McAuley, 2012].

Practical identifiability properties can be found by results from fitting parameters to available data sets. In most previous studies, parameter correlations were detected by analyzing the sensitivity matrix and the Fischer information matrix (Fisher information matrix (FIM) ) [McLean and McAuley, 2012] in order to obtain local confidence regions of parameter solutions. Sensitivity analysis is well suitable to linear models but

it has limitations for highly nonlinear models [Dobre et al., 2012; Raue et al., 2011]. Yao et al. [2003] used the rank of the sensitivity matrix to determine the number of estimable parameters. However, the subsets of correlated parameters cannot be identified based on this result. Chu and Hahn [2007] proposed to check the parallel columns in the sensitivity matrix to determine parameter subsets in which the parameters are pairwise correlated. Quaiser and Mönnigmann [2009] proposed a method to rank the parameters from least estimable to most estimable. These methods, however, cannot identify parameter groups in which more than two parameters are correlated together, i.e., the corresponding columns in the sensitivity matrix are linearly dependent but not parallel. Such correlations present higher order interrelationships among parameters [McLean and McAuley, 2012].

Recently, Raue et al. [2009] used profile likelihood to detect non-identifiability for partially observable models. The parameter space is explored for each parameter by repeatedly fitting the model to a given data set, which then provides a likelihood-based confidence region for each parameter. The profile likelihood approach can also offer information on the correlated relations among the parameters [Bachmann et al., 2011; Hengl et al., 2007; Raue et al., 2009; Steiert et al., 2012] but it cannot show the exact types of these relations.

In summary, through the above analysis it can be seen that further studies still need to be done in order to find new methods that can easily address the identifiability problem of complex dynamic models, especially mathematically to figure out the type of the correlation among parameters in a non-identifiability model.

## 1.2 Structure and Contribution of the Thesis

**Chapter 2: Parameter Estimation Theory: A review**

Chapter 2 gives a review over system identification problems and its sub-area PE problem of ordinary differential equation (ODE) or, for more complicated, differential algebraic equation (DAE) systems. The problem formulation for parameter estimation is presented, i.e., explains how to form PE problems with equation systems described by ODEs or DAEs, objective function and constraints.

**Chapter 3: Direct methods to Dynamic Optimization Problems**

In this Chapter, direct methods to solve DOPs will be introduced. A brief review on numerical direct methods that have been developed to solve DOPs are introduced. Each method was briefly explained with its strength and weakness. Methods to solve the ODE and DAE systems, especially the orthogonal collocation method on finite elements, are presented. Two common used numerical methods, sequential quadratic programming (SQP) and interior-point (IP), to solve the resulting Nonlinear Programming Problems (NLPs) are also described.

**Chapter 4: Improved methods to Dynamic Optimization Problems**

This dissertation focuses on two methods used to solve DOPs for parameter estimation: direct sequential strategy (DSQ) coupled with the collocation method that developed by Hong et al. [2006] and CMSC that developed by Tamimi and Li [2010]. In [Hong et al., 2006], the DSQ with the collocation method was developed and applied successfully to optimal control of a large-scale dynamic system. In that development, numerical method SQP was applied to solve the resulted NLP problems after discretization task. Thanks to the development of the numerical methods to NLP problems, the IP methods are more and more widely used in mathematic as well as engineering areas due to its high efficiency to solve appropriate problems. In this dissertation, the IP method will be coupled with the quasi-sequential strategy (QSQ) to solve DOPs, particularly PE problems. Mathematical derivations are made and the strengths of the method are explained in [Hong et al., 2009]. Furthermore, an improvement of this method is developed to solve PE problems in [Vu and Li, 2010] in which the reduced-space (RS) method of IP strategy is used.

The CMSC method [Tamimi and Li, 2010] was proved to be efficient to solve DOPs. In [Tamimi and Li, 2010], all constraints of states of the DAE model were imposed only at the nodes of the discretization grids while all their values between nodes were let free. This can lead to state constraints violations inside the time intervals. In this thesis, an improvement is made to impose constraints for all state values at all collocation points in order to improve the quality of the DOPs. A parallel computation strategy that utilizes MPI programming is also applied to decrease the computation time.

To improve the quality of the PE problems, multiple data-sets of measurement data

usually need to be used. Faber et al. [2003] studied a three-stage framework for the estimation of nonlinear steady-state systems with multiple data-sets by making use of the optimality condition of the sub-NLP problems. In this study, an extension of the work in [Faber et al., 2003] is made to the PE problem of dynamic systems described by DAEs with a derivation to the DSQ algorithm for dynamic PE problems. As a result, a dynamic three-stage estimation framework is developed. Due to the decomposition of the optimization variables, the proposed approach can solve time-dependent PE problems with multiple data profiles by IP solvers and parallel computation strategy.

**Contributions:**

The main contributions in Chapter 4 of this thesis can be summarized as follows:

1. interior-point (IP) coupled with quasi-sequential strategy (QSQ) method
   (a) IP coupled with QSQ method in full space mode in Section 4.1, which was published in [Hong et al., 2009].
   (b) IP coupled with QSQ method in reduced-space (RS) mode in Section 4.2, which was published in [Vu and Li, 2010].
2. Parameter Estimation Problems framework with Multiple Datasets
   (a) A dynamic three-stage decomposition of the optimization variables framework is developed in Section 4.3 which was partially published in [Vu et al., 2010; Zhao et al., 2013].
3. Improvement of combined multiple shooting and collocation strategy (CMSC)
   (a) Improvement of CMSC in that all collocation points are imposed by constraints with parallel computation strategy in Section 4.4. [1]

**Chapter 5: Identification of parameter correlations**

One of the challenging tasks in PE of nonlinear dynamic models is the identifiability of the parameters in the relevant problem. The identifiability property of a model is

[1] Vu, Q. D. and Li, P. (2012), An improved direct multiple shooting approach combined with collocation and parallel computing to handle path constraints in dynamic nonlinear optimization, 5th International Conference on High Performance Scientific Computing, March 5-9, 2012, Hanoi, Vietnam.

used to answer the question whether or not the estimated parameters are unique. For example, a biological model usually contains a large number of correlated parameters leading to non-identifiability problems. Although many approaches have been developed to address both structural and practical non-identifiability problems, very few studies have been made to systematically investigate parameter correlations.

In this thesis, an approach to identify both pairwise parameter correlations and higher order interrelationships among parameters in nonlinear dynamic models is developed [Li and Vu, 2013]. The correlation information obtained in this way clarifies both structural and practical non-identifiability. Moreover, this correlation analysis also shows that a minimum number of data sets, which corresponds to the maximum number of correlated parameters among the correlation groups, with different inputs for experimental design are needed to relieve the parameter correlations.

The information of this identifiability analysis in biological models gives a deeper insight into the cause of non-identifiability problems. The result of this correlation analysis provides a necessary condition for optimal experimental design in order to collect suitable measurement data for unique parameter estimation.

**Contributions:**

The main contributions of Chapter 5 of this thesis can be summarized as follows:

1. An approach to identify both pairwise parameter correlations and higher order interrelationships among parameters in nonlinear dynamic models in Chapter 5

   (a) The information of pairwise and higher order interrelationships among parameters in biological models gives a deeper insight into the cause of non-identifiability problems, which was published in [Li and Vu, 2013].

   (b) The result of correlation analysis provides a necessary condition for experimental design in order to acquire suitable measurement data for unique parameter estimation, which was published in [Li and Vu, 2013].

**Chapter 6: Conclusions and future work**

Finally, Chapter 6 draws the conclusions of this dissertation and highlights some recommendations for future developments.

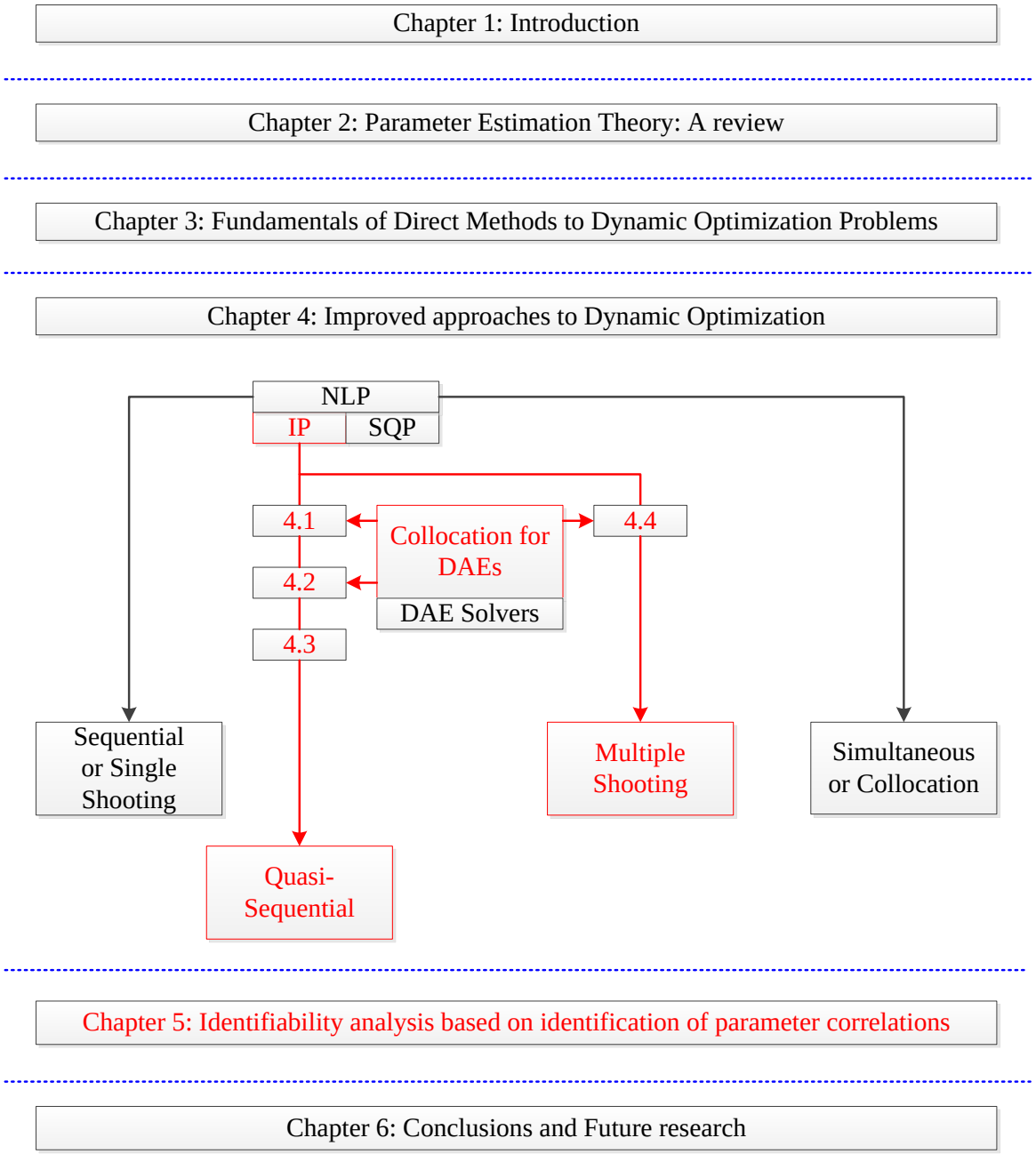



Fig. 1.1 Structure and Contribution of the Thesis[1].

[1]The parts with contribution are highlighted in red color.

## 1.3 Publications

### 1.3.1 Journal Papers

I. Zhao, C., Vu, Q., and Li, P. (2013). A quasi-sequential parameter estimation for nonlinear dynamic systems based on multiple data profiles. *Korean Journal of Chemical Engineering*, 30(2):269–277.

II. Li, P. and Vu, Q. (2013). Identification of parameter correlations for parameter estimation in dynamic biological models. *BMC Systems Biology*, 7(1).

### 1.3.2 Proceedings

I. Hong, W., Tan, P., Vu, Q. D., and Li, P. (2009). An improved quasi-sequential approach to large-scale dynamic process optimization. In $10^{th}$ *International Symposium on Process Systems Engineering: Part A* (C. A. O. d. N. Rita Maria de Brito Alves and E. C. Biscaia, eds.), vol. 27 of *Computer Aided Chemical Engineering*, pages 249 – 254. Elsevier.

II. Vu, Q. D., Zhao, C., Li, P., and Su, H. (2010). An efficient parameter identification approach of large-scale dynamic systems by a quasi-sequential interior-point method based on multiple data-sets. In $2^{nd}$ *International Conference on Engineering Optimization EngOpt2010*, published on CD, 8 pages.

III. Vu, Q. D. and Li, P. (2010). A reduced-space interior-point quasi-sequential approach to nonlinear optimization of large-scale dynamic systems. In *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2010 IEEE RIVF International Conference on Computing and Communication Technologies*, pages 1–6.

# Chapter 2

# Parameter Estimation Theory: A review

The use of mathematical models for process analysis, optimal process design, monitoring and control has become increasingly attractive in recent years. Mathematical models can be used to describe diverse systems ranging from technical areas of process engineering, electrical engineering, mechanical engineering, aerospace engineering and chemical engineering, to non-technical areas of natural sciences as chemistry, physics, biology, medicine, as well as geo-sciences and economics, etc. Developing rigorous dynamic process models with a highly predictive quality is essential for successful implementation of optimization and advanced process control techniques since these applications depend heavily on model parameter values estimated based on experimental data [Abdellatif et al., 2013; Gauss, 1809; Gauss and Stewart, 1995; Isermann and Münchhof, 2010; Walter and Pronzato, 1997]. Therefore, parameter estimation is a critical step in the development and update of a process model. Rigorous parametric modeling of a dynamic process usually leads to a nonlinear DAEs system with up to thousands of variables and unknown parameters. As a consequence, a complex DAEs constrained optimization problem needs to be solved for carrying out a parameter estimation task. Correlations among unknown parameters usually result in ill-posed optimization problems. Therefore, it is desirable to develop efficient estimation strategies and numerical algorithms which should be able to solve such challenging estimation problems, including multiple data profiles and large parameter sets.

## 2.1 System Identification Problems

System identification, i.e., obtaining a satisfactory mathematical model within a suitable model structure from measured time series data of the inputs and outputs of a dynamical system, plays an important role in many branches of engineering and science. In a mathematical model the relationships between quantities, such as concentrations of a chemical reaction, distances of a motion, currents in electrical systems, flows in a dynamic fluid, and so forth, that can be observed in the physical (chemical) system are described as mathematical relations. Significant developments have been made in the past decades by researchers and engineers from such diverse fields based on system and control theory [Isermann and Münchhof, 2010; Ljung, 1999, 2010; Ljung and Glad, 1994a; Maine et al., 1985; Åström and Eykhoff, 1971; Walter and Pronzato, 1997; Wiener, 1965], system biology [Villaverde and Banga, 2013], signal processing, communications and information theory [Giannakis and Serpedin, 2001]. Depend on the *a priori* knowledge and physical insight about the system, the models can be distinguished between three color-coded levels: *White-box, Grey-box* and *Black-box* as depicted in Fig. 2.1.

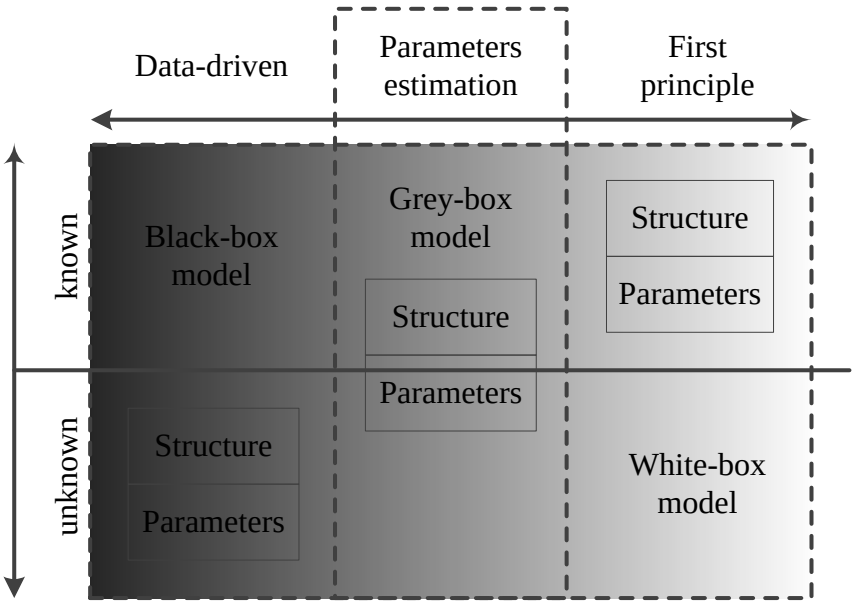



Fig. 2.1 Block diagram of transition of system identification levels

### White-box model

Mathematical models describing the dynamical system of interest can be built using the first principle laws of physics, chemistry, biology, etc. In this case, the model is perfectly known, i.e., the mathematical equations that describe relations between the states of physical systems are explicit and their parameters are totally known. This type of modeling requires specialist *a priori* knowledge and physical insight which might be lacking. Therefore, developing such models can be very difficult, time-consuming and, in the case of large-scale systems, impossible. This means that a pure white-box model does not exist in reality.

### Black-box model

In contrast to the first type of models, in Black-box models there is no *a priori* physical insight available. The model is only a transportation mean to transfer information from inputs to outputs of the considered system. The model structure can be chosen among well-known types depending on the purpose of the identification task. One system can be described in several structures which do not reflect any internal physical relations that happen inside the model. In general, black-box models have disadvantages in the extrapolation due to its lack of flexibility. Black-box models are suitable for specific purpose of identification of systems of interest. More information is available in [Juditsky et al., 1995; Sjöberg et al., 1995] and references therein.

### Grey-box model

Lying between the two above extreme cases, grey-box models provide (internal) physical representation of system but several parameters are missing and need to be determined from measured data. Grey-box (in some literatures, gray-box is used instead) models therefore inherit both advantages and disadvantages of the two extreme cases. In grey-box modeling, *a priori* knowledge concerning the system is used to set up a structure of the model and then the physical insight of the system can be manifested by a system of differential and/or algebraic equations, e.g., ODEs, partial differential equations (PDEs) and DAEs. Grey-box provides a higher flexibility, which means grey-box is a more generic model that can be used to do simulation and extract rules that describe the behavior of the system. Even grey-box can be used to form a unique black-box presentation of the system of interest but the vice versa is not necessarily true. A grey-box model therefore is widely used, e.g., model-based control and simu-

lation, and thus grey-box modeling becomes natural framework for modeling dynamic systems. Reviews of system identification are given in [Gevers, 2006; Ljung, 2010]. Typical representatives for grey-box models are continuous-time nonlinear state-space models which are main objects of this thesis.

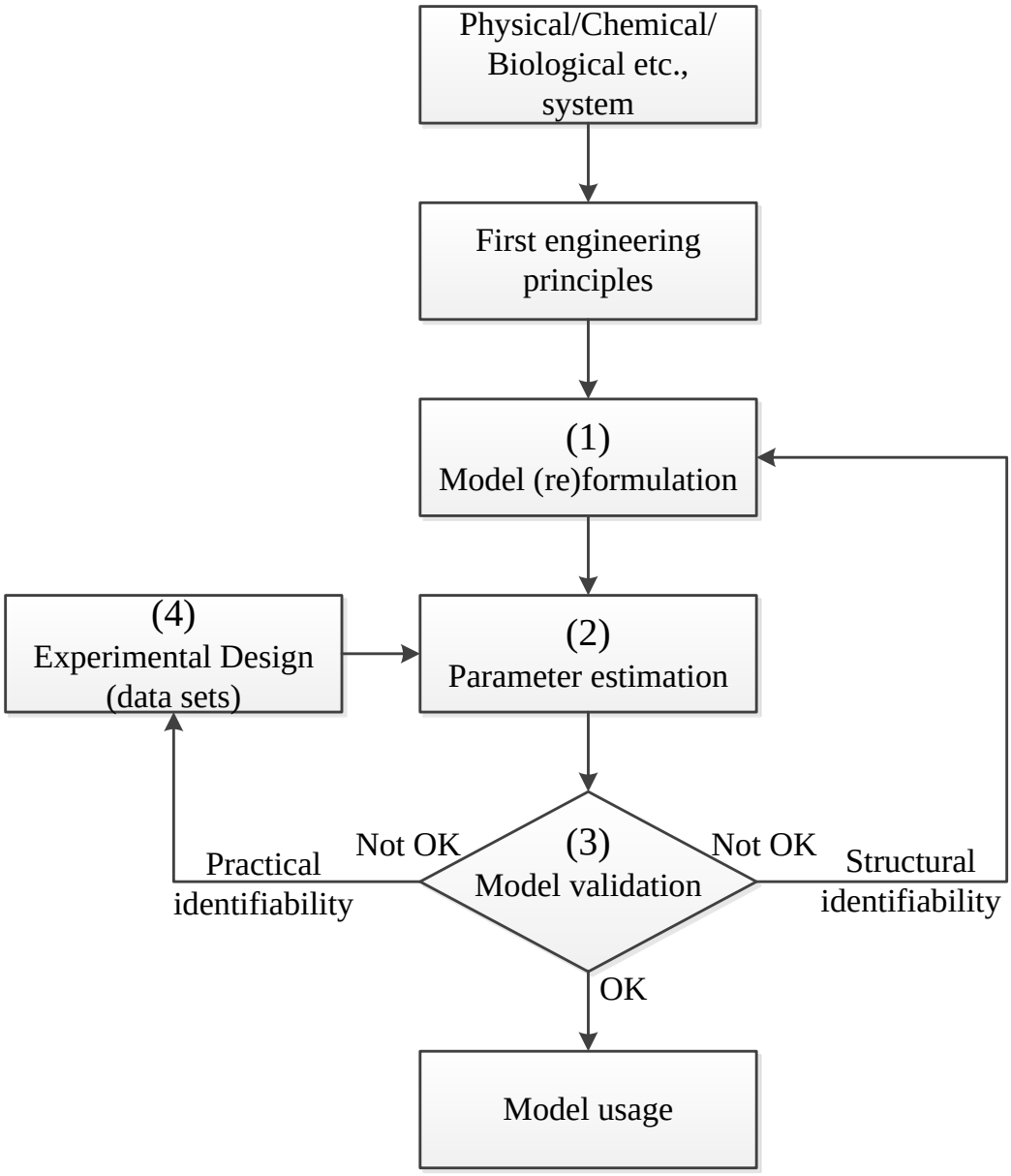



Fig. 2.2 A simple diagram of grey-box system modeling

## 2.2 Parameter Estimation of DAEs systems

Parameter Estimation of DAEs systems is an interesting and most widely used sub-area of *system identification* field. In the context of physical (and chemical) modeling and simulation, the DAEs (or ODEs) that express the process dynamic phenomena usually contain unknown coefficients that need to be estimated by using numerical optimization to fit the mathematic model to observed data sets [Schittkowski, 2013]. Figure 2.3 shows the block diagram of the parameter estimation of a DAEs system.

This fitting can be done by the Bayesian statistical framework [Efron, 2013; Girolami, 2008; Klein and Morelli, 2006] and the Frequentist approach. Until now Frequentist is still the approach most used and thus from now on we only focus on this method. For further reading, comparisons of the Frequentist and Bayesian approaches to estimation can be seen in [Aguilar, 2013; Efron, 2012; Raue et al., 2012; Samaniego, 2010]. Fig. 2.2 show the main four steps in the parameter estimation task. Those are:

1. The **Model (re)formulation** step in that a mathematical model is built from *a priori* physical knowledge with unknown parameters.

2. The **Parameter estimation** step in that the unknown parameters are estimated by utilizing optimization methods that can minimize the residuals between the measurement data (obtained from real process) and the corresponding outputs of the mathematic model. This step can be done *in silico* before the real experimental process are conducted. Suppose that the optimization solver converges at the end of this step, one or several sets of parameters are obtained.

3. The **Model validation** step is then used to check the quality of the estimation step. The validation data must be independently measured from the data that are used in the estimation step. The most important question raised here is that whether the estimated parameter set is unique. This question forms the *identifiability* problem. A model is said to be *identifiable* if there exists an unique solution of the parameter estimation problem, otherwise it is *non-identifiable*. Certainly a non-identifiable model is not reliable and useless. The identifiability of a model can be classified into two levels: *structural identifiability* and *practical identifiability*. Structural identifiability means that the model is identifiable with *ideal continuous noise free* observations that can be *in silico* done by producing

simulated data. Structural identifiability expresses that it is a property of the model itself and this property of course depends on how the model is constructed. Structural identifiability should be checked before conducting real experiments. Obviously, structural identifiability is necessary but not sufficient to affirm an accurate estimation of the model parameters from experimental data. In contrast, practical identifiability associates with *real sparse noisy* measurements and it certainly depends on how the practical experiments are conducted. Practical identifiability in principle can be solved by means of suitable experimental design.

4. The **Experimental design** is an important step in which real measured data sets are produced. The quality of the measured data is usually affected by many factors, such as the type of input signals, experiment conditions, quality and quantity of sensors, etc. The data sets should be pretreated before being passed to the estimation step. Due to the practical identifiability problem, this step may be repeated until a suitable parameter set of the model is obtained.

Until now these main four steps in PE problem still challenge researchers in many aspects. The first and last steps require the modelers to have deep knowledge in the field of modeling, measurement techniques, etc., which are out of the scope of this dissertation. The second and third steps are conducted in this thesis and new contributions are presented in Chapter 4 and Chapter 5, respectively.

**The model equation**

DAEs represent a powerful way of modeling dynamical systems [Biegler et al., 2012]. DAEs model can be used to describe a first principle lumped parameter system, whose state variables are described by ODEs together with supplementary algebraic equations (AEs) to express dynamic phenomena such as thermodynamic equilibriums, mass and energy transfers and reaction kinetics. A general explicit index-1 DAEs model can be described as follows:

$$\dot{\mathbf{z}}(t) = f(\mathbf{z}(t), \tilde{\mathbf{y}}(t), \mathbf{u}(t), \boldsymbol{\theta}(t), \boldsymbol{\Pi}) \tag{2.1a}$$

$$0 = g(\mathbf{z}(t), \tilde{\mathbf{y}}(t), \mathbf{u}(t), \boldsymbol{\theta}(t), \boldsymbol{\Pi}) \tag{2.1b}$$

$$\bar{\mathbf{y}}(t) = h(\mathbf{z}(t), \tilde{\mathbf{y}}(t), \mathbf{u}(t), \boldsymbol{\beta}) \tag{2.1c}$$

$$\mathbf{z}(t_0) = \mathbf{z}_0 \tag{2.1d}$$

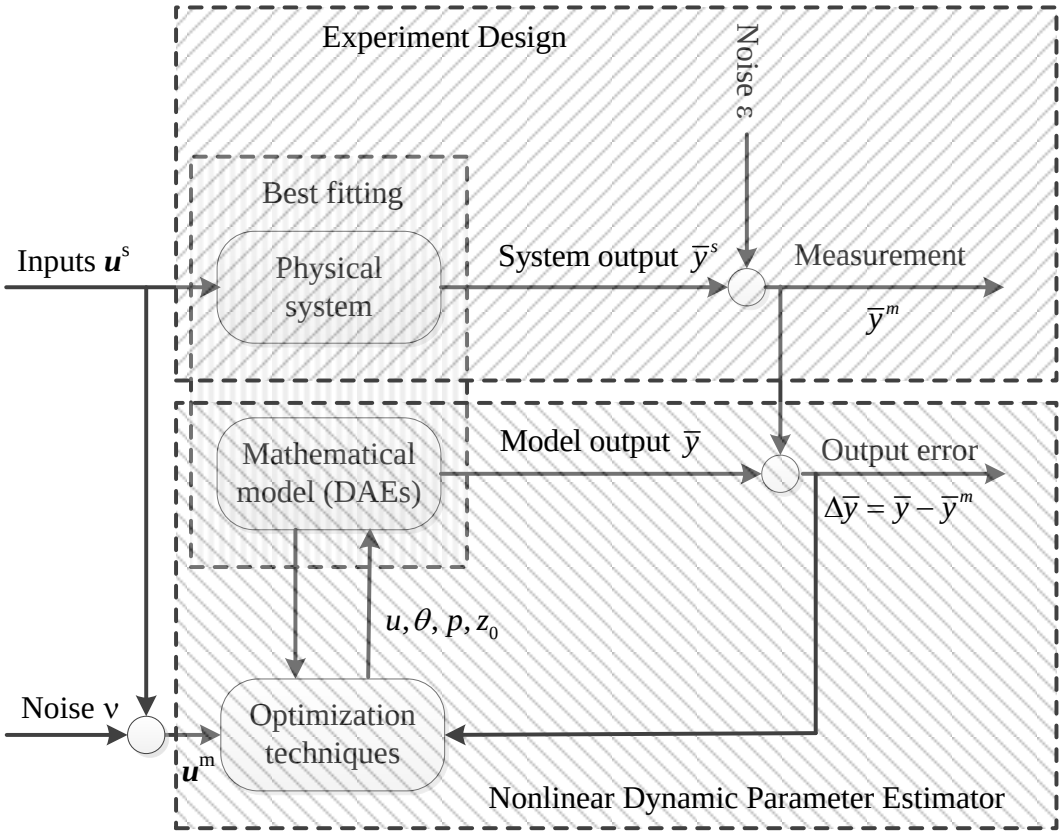



Fig. 2.3 Block diagram of the parameter estimation of DAEs system procedure

where

| | |
|---|---|
| (2.1a) | are differential equations |
| (2.1b) | are algebraic equations |
| (2.1c) | are output equations |
| $\mathbf{z}(t) \in R^{n_z}$ | is a vector of differential state variables |
| $\tilde{\mathbf{y}}(t) \in R^{n_{\tilde{y}}}$ | is a vector of algebraic state variables |
| $\mathbf{u}(t) \in R^{n_u}$ | is a vector of system input or control variables |
| $\boldsymbol{\theta}(t) \in R^{n_\theta}$ | is a vector of unknown time-dependent parameters |
| $\boldsymbol{\Pi} \in R^{n_\Pi}$ | is a vector of unknown time-independent parameters |
| $\boldsymbol{\beta} \in R^{n_\beta}$ | is a vector of additional unknown time-independent parameters |
| $\bar{\mathbf{y}}(t) \in R^{n_{\bar{y}}}$ | is a vector of output (measured) variables |
| $\mathbf{z}_0$ | is a vector of initial values of differential states. These values can be known *a priori*; otherwise they need to be estimated together with parameters |

The model equations (2.1) are assumed to accurately describe input-output relations and the interaction between internal states of the system of interest. Equations (2.1) describe a system in continuous time *State-space* form as a standard model for dynamic systems. In the observation equations (2.1c), the output vector $\bar{\mathbf{y}}(t)$ can be some internal states if so desired or even be pools of several states with additional parameters such as scaling or offset parameters $\mathbf{q}$. During an experiment within a time horizon $[t_0, t_f]$ , the output vector is measured at discrete time point $t_i$ such that $t_0 \leq t_i \leq t_f$,

$$\bar{\mathbf{y}}_i^m = \bar{\mathbf{y}}(t_i) + \boldsymbol{\varepsilon}_i$$

where

$\bar{\mathbf{y}}_i^m$ denotes the measured output vector of the $i^{th}$ measurement points,

$i = 1, \cdots, NL; NL$ is the number of measurement points,

$\bar{\mathbf{y}}(t_i)$ is a vector of system outputs,

$\boldsymbol{\varepsilon}_i$ is a vector of measurement noises associated with the $i^{th}$ measurement value.

In addition, the input vector $\mathbf{u}(t)$ in each experiment can be measured at time point $t_i$,

$$\mathbf{u}_i^m = \mathbf{u}(t_i) + \boldsymbol{\upsilon}_i$$

where

$\mathbf{u}_i^m$ denotes the measured input vector of the $i^{th}$ measurement points,

$\boldsymbol{\upsilon}_i$ is a vector of the random measurement noises.

The input vector $\mathbf{u}(t)$ can be set at *a priori* known values, or fixed to the measurements $\mathbf{u}_i^m(t)$ or, in the case that it is not precisely known, should be estimated together with unknown parameters. In order to make all the input and output measurements to be unique we define:

$$\boldsymbol{\emptyset} = \begin{bmatrix} \bar{\mathbf{y}} \\ \mathbf{u} \end{bmatrix} \; ; \quad \boldsymbol{\chi}^m = \begin{bmatrix} \bar{\mathbf{y}}^m \\ \mathbf{u}^m \end{bmatrix} \quad \text{and} \quad \boldsymbol{\varsigma} = \begin{bmatrix} \boldsymbol{\varepsilon} \\ \boldsymbol{\upsilon} \end{bmatrix}$$

so that we have:

$$\boldsymbol{\chi}^m(t_i) = \boldsymbol{\chi}(t_i) + \boldsymbol{\varsigma}_i, \quad \text{or for shot} \quad \boldsymbol{\chi}_{\mathrm{i}}^{\mathrm{m}} = \boldsymbol{\chi}_{\mathrm{i}} + \boldsymbol{\varsigma}_{\mathrm{i}}$$

In order to compact Eqs. (2.1a), (2.1b) and (2.1c) to be in more general form, we can define $\mathbf{y} = [\tilde{\mathbf{y}} \;\; \bar{\mathbf{y}}]^T$ as a vector of algebraic variables, where $y \in R^{n_y}$ and $n_y = n_{\tilde{y}} + n_{\bar{y}}$; $\mathbf{p} = [\boldsymbol{\Pi} \;\; \boldsymbol{\beta}]^T$ as a vector of parameters, where $p \in R^{n_p}$ and $n_p = n_\Pi + n_\beta$.
Then Eq. (2.1) can be rewritten as follow

$$\dot{\mathbf{z}}(t) = F(\mathbf{z}(t), \mathbf{y}(t), \mathbf{u}(t), \boldsymbol{\theta}(t), \mathbf{p}) \tag{2.2a}$$

$$0 = G(\mathbf{z}(t), \mathbf{y}(t), \mathbf{u}(t), \boldsymbol{\theta}(t), \mathbf{p}) \tag{2.2b}$$

$$\mathbf{z}(t_0) = \mathbf{z}_0 \tag{2.2c}$$

### The objective function

Parameter estimation tackles the task of estimating the parameter vector that makes the mathematical model best fit to the measurement vector. This task is usually done by minimizing a scalar function called objective function that includes the differences or residuals $\mathbf{e}_i$ between input-output of the model calculated and the measured values:

$$\mathbf{e}_i = [\boldsymbol{\chi}_i^m - \boldsymbol{\chi}_i]$$

where $\boldsymbol{\chi} = [\bar{\mathbf{y}}\ \ \mathbf{u}]^T$ in that $\bar{\mathbf{y}}$ is calculated using the estimated parameters, $\mathbf{u}$ denotes the vector of estimated inputs (in the case that it is unknown).

The type of the objective function plays an important role in parameter estimation due to the fact that it relates not only to the parameter values but also to the statistical properties of the parameters. The form of least squares [Englezos and Kalogerakis, 2000; Johnson and Faont, 1992; Strejc, 1977; van de Geer, 2005; Walter and Pronzato, 1997] is commonly used, which estimates parameter values by minimizing the weighted sum of squares of the errors $\mathbf{e_i}$:

$$J_{LS}(\mathbf{p}, \mathbf{u}) = \sum_{1}^{NL} \mathbf{e}_i^T \mathbf{W}_i \mathbf{e}_i$$

where $\mathbf{p} = [\boldsymbol{\theta}\ \ \boldsymbol{\beta}]^T$; $\mathbf{e}_i$ is the residual vector and $\mathbf{W}_i$ is the weighting matrix. It should be noted that $\mathbf{u}$ variables can be wiped off from the formulation if they are known *a priori*.

In many cases, a series of runs should be conducted in order to make a better estimation and the parameters can be estimated using all data sets simultaneously. In that situation the objective function is expressed as follows:

$$J_{LS}(\mathbf{p}, \mathbf{u}) = \sum_{k=1}^{NS} \left[ \sum_{i=1}^{NL} \mathbf{e}_i^T \mathbf{W}_i \mathbf{e}_i \right]_k \tag{2.3}$$

where NS is the number of data sets.

The weighting matrix $\mathbf{W}_i$ can be chosen by the user in accordance with the criterion of statistical properties of the measurement data. In the simplest case called unweighted least squares, the user can use $\mathbf{W}_i = \mathbf{W}$. With Weighted Least Squares (WLS) one

can use constant weights $\mathbf{W}_i = \mathbf{I}$ for all $i = 1, \cdots, NL$ and keep $\mathbf{W}_i$ the same for all data sets. In generalized least squares estimation (GLS) the user-supplied weighting matrices $\mathbf{W}_i$ are non-constant and differ from measurement point to measurement point. From the statistical point of view, in many fields measurement errors can be assumed to be normally distributed with zero mean and with a known covariance matrix, then $\mathbf{W}_i$ is chosen as the inverse of the covariance matrix:

$$\mathbf{W}_i = [COV(\varsigma_i)]^{-1} = \sum\nolimits_i^{-1}$$

As a consequence, the objective function in maximum likelihood estimation (ML) [Box and Tiao, 1992; Gauss, 1809; Gauss and Stewart, 1995] can be written as:

$$J_{\mathrm{ML}}(\mathbf{p}, \mathbf{u}) = \sum_{k=1}^{NS} \left[ \sum_{i=1}^{NL} \mathbf{e}_i^T \sum\nolimits_i^{-1} \mathbf{e}_i \right]_k$$

However, to known the exact values of covariance matrices $\sum_i$ is unrealistic in practical engineering. Some reasonable assumptions about the statistical properties of $\sum_i$ can be made to define the ML objective function [Box and Tiao, 1992; Johnson and Faont, 1992].

For example, with the assumption that the variance of the measurement errors are constant and differ from measurement point to measurement point, we have

$$\sum\nolimits_i = \begin{bmatrix} \sigma_{e1}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{e2}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{e(n_u+n_{\bar{y}})}^2 \end{bmatrix};\ i = 1, 2, \ldots, NL \tag{2.4}$$

Often in engineering practice $\sigma_{eij}$ can be parameterized as follow [Walter and Pronzato, 1997]:

$$\sigma_{eik}^2(a, b) = a|\mathbf{\emptyset}_k(t_i)|^b$$

where $a > 0$ is unknown scaling factor; and $0 \leq b \leq 2$ . Note that $a = 0$ corresponds to WLS estimation. If $b = 2$, then we have

$$\sigma_{eij}^2 = a|\mathbf{\emptyset}_j(t_i)|^2$$

The weighting matrices $\mathbf{W}_i$ can be written as follow:

$$\mathbf{W}_i = diag(\mathcal{O}_{i,1}^{-2}, \mathcal{O}_{i,2}^{-2}, \ldots, \mathcal{O}_{i,(n_u+n_{\bar{y}})}^{-2})$$

The above chosen weighting matrices $\mathbf{W}_i$ have the disadvantage that the weights are still unknown. In engineering practice, if the magnitude of the measurement errors $\boldsymbol{\varsigma}_i$ is not too exaggerated, one can choose the measurement values $\boldsymbol{\chi}_i^m$ instead of $\boldsymbol{\chi}_i$ with equivalent results, i.e.,

$$\mathbf{W}_i = diag\left((\mathcal{O}_{i,1}^m)^{-2}, (\mathcal{O}_{i,2}^m)^{-2}, \ldots, (\mathcal{O}_{i,(n_u+n_{\bar{y}})}^m)^{-2}\right) \tag{2.5}$$

At the end for this Section, in order to make the objective function to be more general as in a dynamic optimization problem, one can extend the dimension of the weighting matrices $\mathbf{W}_i$ to $(n_u + n_y)$ with the assumption that any unmeasured variable in variables $y$ takes its weighting value equal to zero.

### The constraints

Besides the governing DAEs, the physical system described by a mathematical model is usually accompanied with some physical constraints of the model parameters, states variables and control inputs. These constraints are introduced to ensure physical meaning of the relevant elements and system safety requirements. For example, control valves in chemical processes have the maximum open and minimum close positions. The temperature of a chemical reaction should not be exceed a maximum value. Reaction rate coefficients are always positive. Concentrations of all chemical components must be non-negative. In automobile systems, acceleration and deceleration are associated with limitations of the throttle minimum and maximum displacement as well as the braking capacity, etc. Furthermore the system of interest may has more complicated constraints that reflect the bounds of the relation between its inputs, states and output. Such restrictions are mathematically formulated as inequality constraints. For a DAEs model they can be written in a general way as follows:

$$0 \leq H(\mathbf{z}(t), \mathbf{y}(t), \mathbf{u}(t), \boldsymbol{\theta}(t), \mathbf{p}, t) \tag{2.6}$$

Bound constraints also can be explicitly expressed in the form:

$$\begin{aligned}
\mathbf{z}_L &\leq \mathbf{z}(t) \leq \mathbf{z}_U \\
\mathbf{y}_L &\leq \mathbf{y}(t) \leq \mathbf{y}_U \\
\mathbf{u}_L &\leq \mathbf{u}(t) \leq \mathbf{u}_U \\
\boldsymbol{\theta}_L &\leq \boldsymbol{\theta}(t) \leq \boldsymbol{\theta}_U \\
\mathbf{p}_L &\leq \mathbf{p} \leq \mathbf{p}_U
\end{aligned} \tag{2.7}$$

where $\mathbf{z}_L, \mathbf{y}_L, \mathbf{u}_L, \boldsymbol{\theta}_L, \mathbf{p}_L$ and $\mathbf{z}_U, \mathbf{y}_U, \mathbf{u}_U, \boldsymbol{\theta}_U, \mathbf{p}_U$ are the lower and upper bound of the differential variables, algebraic variables, inputs, time-dependent parameters and time-independent parameters, respectively. These constraints play an important role in the solution of the parameter estimation problem.

## 2.3 Parameter estimation - Optimization of Dynamic Systems

Based on the above definitions, a general parameter estimation problem of DAEs can be formulated as:

$$\begin{aligned}
\min_{\mathbf{u(t)},\mathbf{z(t)},\mathbf{y(t)},\boldsymbol{\theta}(t),\mathbf{p},(\mathbf{z}_0)} \mathbf{F} = \sum_{k=1}^{NS}\sum_{i=1}^{NL} \Big[ &(\mathbf{u}_{k,i}^m - \mathbf{u}_{k,i})^T \mathbf{W}_{k,i}^u (\mathbf{u}_{k,i}^m - \mathbf{u}_{k,i}) \\
&+ (\mathbf{y}_{k,i}^m - \mathbf{y}_{k,i})^T \mathbf{W}_{k,i}^y (\mathbf{y}_{k,i}^m - \mathbf{y}_{k,i}) \Big]
\end{aligned} \tag{2.8a}$$

$$\text{s.t.} \quad \dot{\mathbf{z}}(\mathrm{t}) = F(\mathbf{z}(t), \mathbf{y}(t), \mathbf{u}(t), \boldsymbol{\theta}(t), \mathbf{p}, t), \qquad t \in [t_0, t_f], \tag{2.8b}$$

$$0 = G(\mathbf{z}(t), \mathbf{y}(t), \mathbf{u}(t), \boldsymbol{\theta}(t), \mathbf{p}, t), \qquad t \in [t_0, t_f], \tag{2.8c}$$

$$\mathbf{z}(0) = \mathbf{z}_0, \tag{2.8d}$$

$$0 \leq H(\mathbf{z}(t), \mathbf{y}(t), \mathbf{u}(t), \boldsymbol{\theta}(t), \mathbf{p}) \tag{2.8e}$$

$$\mathbf{p}_L \leq \mathbf{p} \leq \mathbf{p}_U \tag{2.8f}$$

where $\mathbf{W}^u$ and $\mathbf{W}^y$ are the weighting matrices of the inputs and measured outputs, respectively.

In this dynamic optimization problem, the unknown variables are the inputs $\mathbf{u}(t) \in R^{n_u}$, the differential states $\mathbf{z}(t) \in R^{n_z}$, the algebraic variable $\mathbf{y}(t) \in R^{n_y}$, the time-dependent parameters $\boldsymbol{\theta}(t) \in R^{n_\theta}$, as well as the time-independent parameters $\mathbf{p} \in R^{n_p}$ (sometimes $\mathbf{z}_0 \in R^{n_z}$ if it is unknown). In the case that the inputs are known *a priori*, $\mathbf{W}^u$ will be set at zero. $NS$ is the number of measured data sets.

Obviously (2.8) is a constrained DOP with discrete time points in the objective function but continuous time in the constraints. This type of optimization problem appears not only in parameter estimation but also in (open loop) *Optimal control* [Betts, 1998] and *integrated process* design [Banga et al., 2003]. This dynamic optimization problem can be solved by an appropriate method among *dynamic programming* [Bellman, 1957; Luus, 1990], *indirect* [Betts, 2010; Bryson, 1975], and *direct* methods [Biegler, 2010; Binder et al., 2001; Diehl et al., 2006; Hull, 1997; Papamichail and Adjiman, 2002; Rao, 2009]. The first two methods deal with only small problems and have difficulties in solving problems with bound constraints on state variables [Betts, 2010; Biegler and Grossmann, 2004; Chachuat, 2009] and therefore they get less and less attention. Thus in this thesis, only the direct methods are used and more details on them are explained in the following sections.

### 2.3.1 Numerical methods to DOPs

#### Direct Sequential Strategy

In the *direct sequential strategy (DSQ)*, only the inputs of problem (2.8) are discretized by a set of control elements which then be treated as optimization variables of the NLP solver in an optimization layer; the model equations (2.8b-2.8d) are solved by appropriate numerical integration methods in a simulation layer [Barton et al., 1998; Binder et al., 2001; Hicks and Ray, 1971; Logsdon and Biegler, 1992; Morison and Sargent, 1986; Sargent and Sullivan, 1978; Vassiliadis, 1993; Vassiliadis et al., 1994a,b]. This method is known also as direct *single shooting* or *control parameterization* or the sequential method. The dimension of the resulted NLP in the sequential method is the smallest one in comparison with other methods and does not depend on the size of the model equations.

This method is a *feasible path* strategy because the model equations are satisfied at each iteration of the NLP algorithm, i.e., the results can still be usable in the case that the NLP solver cannot converge. In this method, the sensitivities of the state variables with respect to the inputs, required in the NLP layer, are expensive to calculate in the simulation layer by appropriate methods. Therefore, DSQ could be time consuming for large-scale DAEs systems [Diehl, 2001; Jockenhövel, 2004a; Oldenburg et al., 2003]. Besides that, it also inherits some other drawbacks or difficulties in the integration step with arbitrary initial values of the control variables for stiff and unstable problems due to the fact that intermediate inputs in the early time have usually a strong effect on the later parts of the trajectories [Ascher et al., 1995; Biegler, 2000; Diehl, 2001]. Furthermore, state variables path constraints can be only handled approximately in this method [Vassiliadis et al., 1994b; Zavala et al., 2008a].

**Direct Simultaneous Strategy**

In the *direct simultaneous strategy (DSM)*, also referred as direct *collocation approach*, both the input and the state variables of problem (2.8) are discretized at the same time by using various discretization methods [Biegler, 2007; Biegler et al., 2002; Jockenhövel, 2004a]. Then all of these discretized control and state variables are put into the NLP solver as optimization variables. In these methods, DAE model Eqs. (2.8b-2.8d) are transformed into algebraic equations which then are treated as equalities in the resulted NLP problem.

Among other methods, this method leads to the largest dimension of a NLP problem that usually requires a special solution strategies, especially with the case of large-scale DAEs systems [Jockenhövel, 2004a]. In spite of this fact, direct simultaneous strategy (DSM) can easily solve DOPs problems with path constraints on the state variables by imposing these constraints on each discrete points. Due to the fact that the DSM solves the DAEs system only once at the convergence where all the equality constraints are satisfied, this method can avoid solving intermediate instable systems which may have no solution or require much computation effort.

Contrary to the sequential strategy, DSM is an *infeasible path* method meaning that intermediate non-convergence results will be useless. The accuracies of this method depend much on the level of the discretization intervals which can lead to very large and sparse NLP problems. In general, the result of the DSM is assessed as less accurate than that of the sequential strategy. However, the DSM usually is less time consuming

than the sequential strategy [Jockenhövel, 2004a].

**Direct multiple shooting strategy**

Direct multiple shooting strategy can be considered as a hybrid method that was developed by Bock and Plitt [Bock and Plit, 1984; Plitt, 1981]. In this method, the time horizon is divided into time intervals in which control variables are discretized as the same manner in DSQ whereas model Eqs. (2.8b-2.8d) are exactly solved in each time interval by appropriate ODE or DAE solvers. In order to keep continuity between the subintervals, additional interconnections, which enforce the last values of the previous interval to be equal to the initials of the next one, are introduced as equality constraints in the NLP problem. This makes the dimension of the outer NLP problem slightly larger than that of the DSQ but smaller than that of DSM. This method also allows advanced DAE solvers to be applied to calculate the function and derivative values.

Furthermore the DAE solvers are decoupled on separate multiple shooting intervals, therefore, the direct multiple shooting strategy (DMS) method is suitable for parallel computing which can speed up the solution process. This approach also can deal with multistage problems, control and path constraints as well as multi-point boundary conditions. For the optimization of boundary value problems, DMS is considered to be more stable and efficient than the DSQ [Bock et al., 2000]. Beside that, this method also has some drawbacks, e.g., the sensitivity information is too expensive to calculate and the state constraints are difficult to implement, especially ones which are inside each time interval [Jockenhövel, 2004b].

The above three strategies have been widely used in solving DOPs. As we can see, each one has its own *pros and cons* that nearly compensate each other. For this reason, developing new methods to overcome these disadvantages has attracted researchers. Following this trend, two hybrid methods have been developed, which will be presented in the following.

**Quasi-sequential strategy**

In *quasi-sequential strategy (QSQ)*, the DSQ using collocation on finite elements was developed and applied successfully to a large-scale optimal control of dynamic system [Hong et al., 2006]. The QSQ is considered as a hybrid method since it inherits features from both DSM and DSQ methods. In this method, the control variables

are discretized in each time interval as the same manner in DSQ while the state variables are discretized using collocation on finite elements in the same interval as in the DSM method. A simulation layer is used to deal with the nonlinear equation systems resulted from the discretized DAEs model. Using a nonlinear solver, the model equation is solved sequentially from the first interval to the last one to get the state values and appropriate sensitivity information. The continuity of the states between the intervals are forced by using the last collocation point of the previous interval as the initial point of the next interval.

In comparison with the DSM method, the state variables are eliminated from the NLP leading to a smaller problem like in the DSQ method. The inequalities (2.8e) are forced to be held in NLP at each collocation point. This method has advantages of a small NLP problem and easy handling of the sensitivities and inequalities. The method of active-set SQP is usually applied to solve the NLP problem. As mentioned before, the SQP method can be computationally expensive to deal with active constraints in large NLP problems. To overcome this drawback, more efficient NLP solvers need to be developed.

**Combined multiple shooting and collocation strategy**

Combined multiple shooting and collocation strategy can be considered as a combination of the DMS and the DSM strategy [Tamimi and Li, 2010]. This method is based on the DMS strategy with a modification in solving the DAEs. In the CMSC formulation, the model equations in each time interval are discretized using collocation on finite elements to produce a system of nonlinear equations as in the DSM and the QSQ methods. This system of nonlinear equations then is solved by a nonlinear solver. This makes it easier to get state values and the sensitivity information. However, in this method the system constraints (2.8e) are only imposed at the grid points, and thus there can be violations in the time period between grid points. An improvement needs to be considered to surmount this shortcoming.

### 2.3.2 Identifiability analysis

Identifiability analysis is a critical first step in a parameter estimation problem due to the fact that a model is only valid and useful when all of its parameter can be uniquely estimated from the experimental data. In addition, it will be difficult for the

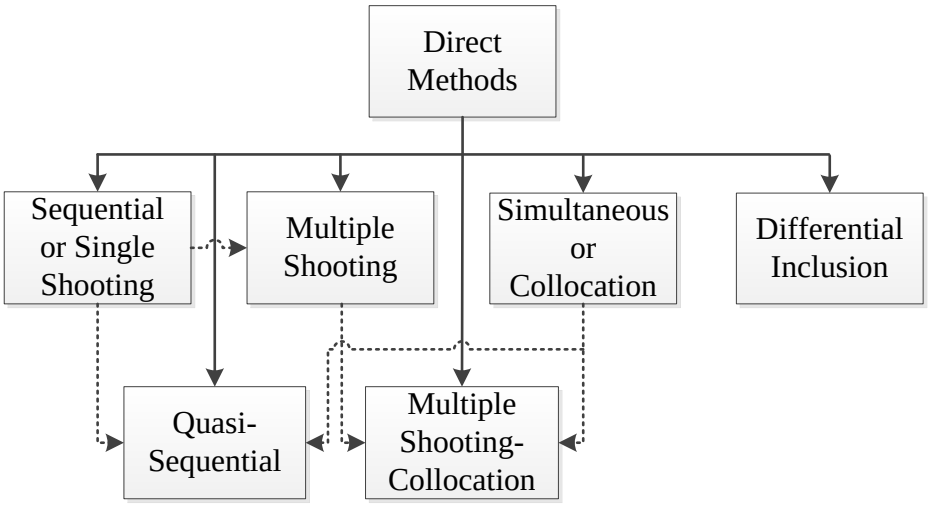



Fig. 2.4 Numerical methods to dynamic optimization problems

optimization solvers to converge if the postulated model is non-identifiable. Despite its importance, theoretical identifiability analysis in modeling studies has been overlooked for a long time [Berthoumieux, 2012; Villaverde and Banga, 2013] since the first study in [Bellman and Åström, 1970]. In recent years, identifiability analysis has gained great interest in the field of chemical process and systems biology, e.g., see [Berthoumieux, 2012; Chis, 2011; McLean and McAuley, 2012; Miao et al., 2011; Raue et al., 2014] and references therein. As a result, a number of specialized commerce and open software packages, which utilize several different approaches, have been built to support the modeler, e.g., DAISY [Bellu et al., 2007; Saccomani et al., 2010], GenSSI [Chis et al., 2011a], PottersWheel [Hengl et al., 2007; Maiwald and Timmer, 2008; Raue et al., 2009].

Identifiability of a model can be classified into Structural (or *a priori*) identifiability [Bellman and Åström, 1970] and Practical (or *a posteriori*) identifiability [Cobelli and DiStefano, 1980]. Structural identifiability means that the model is identifiable with *ideal continuous noise free* observations [Walter and Pronzato, 1997]. Structural identifiability expresses that it is a property of the model itself and this property depends on how the model is constructed. Structural identifiability should be checked before conducting experiments. Obviously, structural identifiability is necessary but not sufficient to affirm an accurate estimation of the model parameters from experimental data. In contrast, practical identifiability associates with *real sparse noisy*

measurements and it thus depends on how the experiments are conducted. Practical identifiability in principle can be solved by means of suitable experimental design.

Methods for checking the structural identifiability of nonlinear models are among power series based approaches including Taylor series approach [Dochain et al., 1995; Petersen et al., 2003; Pohjanpalo, 1978] and generating series approach [Jayasankar et al., 2009; Walter and Lecourtier, 1982] with possible combination of identifiability tableaus [Balsa-Canto et al., 2010; Chis et al., 2011a], the similarity transformation approach [Vajda et al., 1989a], the differential algebra based method [Bellu et al., 2007; Ljung and Glad, 1994b; Saccomani et al., 2010], the direct test method [Denis-Vidal et al., 2001; Walter et al., 2004], the implicit function theorem approach [Xia and Moog, 2003] and the profile likelihood approach using simulated data [Flassig et al., 2015; Hengl et al., 2007; Kreutz et al., 2012; Maiwald and Timmer, 2008; Raue et al., 2010, 2009; Schaber, 2012].

The Taylor series approach requires high order derivatives of the system outputs with respect to time by using the Taylor series expansion of the outputs in the vicinity of the initial state. The Taylor series coefficients are calculated to form a system of nonlinear algebraic equations in the parameters. The uniqueness of the solution of this resulting system guarantees the structural identifiability of the original system. This method is conceptually simple but the number of required derivatives is generally unknown. Moreover, the resulting system of algebraic equations may be too complicated to solve. Thus this method is not popularly used in practice [Chis, 2011; Miao et al., 2011].

The generating series approach uses the same concept as the Taylor series approach. This method also uses the expansion of the outputs of the postulated system with respect to inputs and time. The exhaustive summary that contains the coefficients of the output functions and Lie derivatives is then used to examine the structural identifiability of the original system as the same manner as in the Taylor series approach. Again, the generating series approach presents a challenge due to the unknownness of the minimum number of the required Lie derivatives and the solution of the resulting system of the algebraic equations [Chis, 2011; Miao et al., 2011]. Although the difficulty in handling the resulting system of algebraic equations can be partially solved by using the identifiability tableaus in [Balsa-Canto et al., 2010; Chis et al., 2011a], power series based approaches may be not able to assess the identifiability of the parameters for some particular cases [Chis, 2011].

The similarity transformation approach, which is based on the local state isomorphism theorem, is only suitable to solve the identifiability problem for single-input single-output (SISO) models. With multi-input multi-output (MIMO) models, it has difficulties in solving partial differential equations and checking the prerequisite for the controllability and observability conditions [McLean and McAuley, 2012].

In the differential algebra based method, the non-observable differential states are eliminated to produce differential relations between inputs, outputs and parameters. The exhaustive summary then are obtained and can be solved by algebraic methods. The solution of the resulting algebraic equations can precisely give the identifiability information of the parameters. The disadvantage of this method is that it needs large computational efforts when dealing with complex models [Chis, 2011].

The direct test method uses directly the identifiability definition to analytically or numerically assess the parameter identifiability. This method can be only applied to uncontrolled and autonomous system model and not suitable for large-scale problems [Miao et al., 2011].

The implicit function theorem approach tries to eliminates the unobservable states by computing the derivatives of the observable outputs with respect to time of a differential system, which depends on known inputs, outputs and parameters. The partial derivatives of these differential equations with respect to parameters then are computed to define an identification matrix. If this matrix is not singular then the original system is identifiable. This method also requires high-order derivatives and then the identification matrix may be very complicated to verify its singularity [Chis, 2011; Miao et al., 2011; Xia and Moog, 2003].

In summary, many approaches have been developed for identifiability analysis of parameters in nonlinear dynamic models. Approaches for assessing global identifiability are usually difficult to implement and restricted to moderate dimension systems. It can be concluded that further studies still need to be done in order to find new methods that can easily address the identifiability problem of complex dynamic models, especially mathematically to figure out the type of the functional relations among the parameters in a non-identifiability model.

# Chapter 3

# Fundamentals of Direct Methods to Dynamic Optimization Problems

The main idea of direct methods to dynamic optimization is to transform original infinite DOPs into finite *nonlinear programming problems* (*NLPs*). Thanks to the increasing efficiency of numerical methods for solving large-scale *initial value problems (IVPs)* in the form of differential equations, the development of methods for discretizing the input and/or the state space with a finite dimension, and the advent of powerful computation techniques for solving large-scale NLP problems in accordance with high performance computation, the direct methods that employ finite dimension approximations of infinite DOPs have gained significant achievements and widely applied in technique as well as non-technique areas [Conway, 2012; Maurer and Pesch, 2008; Rao, 2009]. Direct methods transform the continuous DOPs into parameterized optimization problems by discretizing the inputs (independent or control variables) and/or state (dependent) variables. The resulting problems then can be solved by an appropriate existing NLP solver.

Based on the methods used for discretization, direct methods can be divided into three main approaches and two other derivations as depicted in Fig 2.4. The method termed as differential inclusion, in which only the state variables are discretized, is limited in some optimal control problems, therefore it will not be discussed in this Chapter. Readers with interest can refer to [Aseev, 2001; Conway and Larson, 1998; Lobo Pereira and Borges de Sousa, 1992; Loewen and Rockafellar, 1994; Seywald, 1993] for more information. The major aspects of direct methods lie in the numerical methods to

solve NLP problems as well as methods to solve (discretized) IVPs for DAEs systems. In this Chapter, the fundamental elements of direct methods to dynamic optimization will be presented.

## 3.1 Discretization of Independent Variables

In direct methods for solving DOP problems (2.8), the optimization horizon $[t_0, t_f]$ will be subdivided into $NL \geq 1$ time intervals,

$$t_0 < t_1 < t_2 < \cdots < t_{i-1} < t_i < t_{i+1} < \cdots < t_{NL} = t_f$$

On each interval the independent (control) variables including inputs and/or time-dependent parameters $\mathbf{v} = [\mathbf{u}, \boldsymbol{\theta}]^T$ will be parameterized by using $M$-order polynomials:

$$\mathrm{v}(t) = \mathcal{V}^i(t, \tilde{v}^i), \;\; t_{i-1} \leq t \leq t_i \tag{3.1}$$

where $\tilde{v}^i \in R^{(n_u+n_\theta)M}$. Certainly, different variables in different intervals can share the same or even different order $M$. In this Chapter, the same order $M$ is assumed to be used for every variable in every time interval for the a simple discreption. A polynomial representation of Eq. (3.1) of $q^{th}$ independent variable in time interval $i$ is given as:

$$\mathrm{v}_q(t) = \mathcal{V}_q^i(t, \tilde{v}^i) = \sum_{l=0}^{M} v_{q,l}^i \phi_l^{(M)}(\tau^{(i)}), \quad t_{i-1} \leq t \leq t_i \tag{3.2}$$

where

$$\begin{aligned} &\tau^{(i)} = \frac{t - t_{i-1}}{\Delta t_i} \in [0,1] \text{ is normalized time in time interval } \; i \\ \text{and} \quad &\Delta t_i = t_i - t_{i-1} \end{aligned} \tag{3.3}$$

The $M$-order Lagrange polynomial, which is particularly useful to approximate the independent variable schemes, $\phi_l^{(M)}(.)$ is expressed as:

$$\phi_l^{(M)}(\tau) = \begin{cases} 1, & \text{if } \; M = 0 \qquad (3.4a) \\ \displaystyle\prod_{\substack{j=0 \\ j \neq l}}^{M} \frac{\tau - \tau_j}{\tau_l - \tau_j}, & \text{if } \; M \geq 1 \qquad (3.4b) \end{cases}$$

with points $0 \leq \tau_0 < \tau_1 < \cdots < \tau_M \leq 1$ called *collocation* points. The term of *collocation* means that the approximate profile $\mathcal{V}_q^i(t, \tilde{v}^i)$ is collocated or accommodated so as to exactly fit the true profile at a set of points $\tau_k,\ k = 0, 1, \ldots, M$. From Eq. (3.4b) it is the property of Lagrange polynomials that

$$\phi_l^{(M)}(\tau_k) = \prod_{\substack{j=0 \\ j \neq l}}^{M} \frac{\tau - \tau_j}{\tau_l - \tau_j} = \begin{cases} 1, & \text{if } k = l \quad (3.5a) \\ 0, & \text{if } k \neq l \quad (3.5b) \end{cases}$$

then, from Eqs. (3.2), (3.3) and (3.5), at each collocation point $\tau_k,\ k = 0, 1, \ldots, M$ of time interval $i$ we get:

$$\mathrm{v}_q(t) = \mathrm{v}_q(t_{i-1} + \tau_k \Delta t) = v_{q,k}^i$$

With $M = 0$ we have a special case of piecewise constant parameterization which is usually used; $M = 1$ corresponds to apiecewise linear case, etc. The bound constraints of variables in this parameterization can be enforced as:

$$\mathrm{v}_q^L \leq v_{q,k}^i \leq \mathrm{v}_q^U,\ \ k = 0, 1, \ldots, NC$$

If the continuity of the independent variables is desired, we can set the following linear equality constraint:

$$v_{q,0}^{i+1} = v_{q,NC}^i$$

Different degrees of parameterization of independent variables are depicted as in Fig. 3.1.

## 3.2 Numerical methods for solving DAEs Systems

Numerical methods of direct approaches to DOPs require calculating the values of variables of model equations described by an initial value DAE as in Eqs. (2.8b)-(2.8d) in each time interval $[t_i, t_{i+1}]$. Furthermore, these dependent variables depend on the independent variables such as control inputs $u$, parameters $\theta$ and $p$. In addition, there exists requirements to compute their sensitivities with respect to such variables. Hereafter some numerical methods for semi-explicit index-1 DAE (3.6) system shall

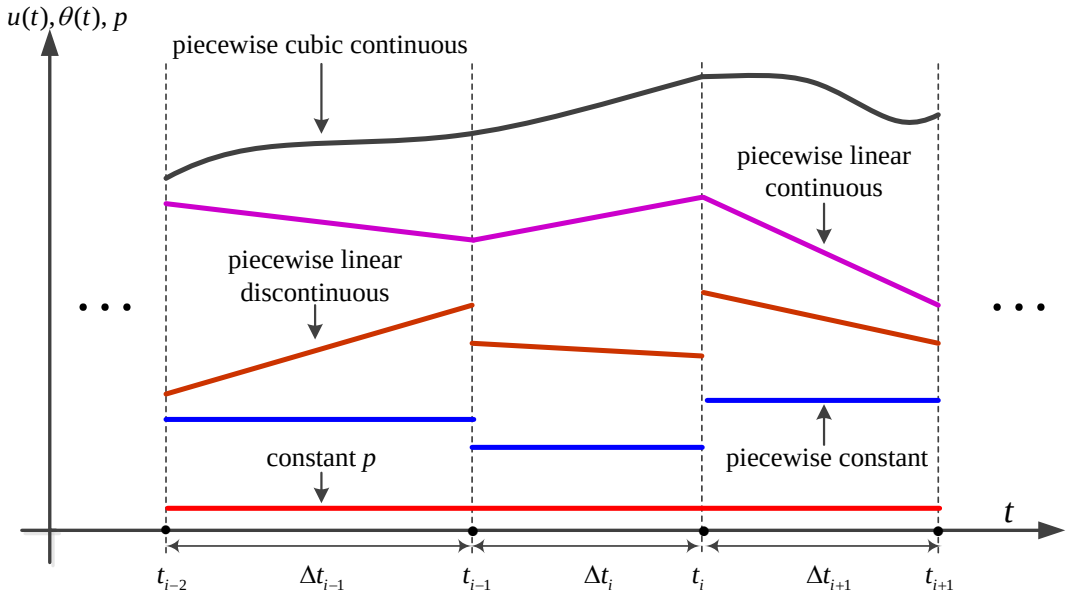



Fig. 3.1 Parameterization of independent variable methods

be discussed:

$$\dot{\mathbf{z}}_i(t) = F(\mathbf{z}_i(t), \mathbf{y}_i(t), \mathbf{u}_i, \boldsymbol{\theta}_i, \mathbf{p}, t), \qquad t \in [t_i, t_{i+1}], \tag{3.6a}$$

$$0 = G(\mathbf{z}_i(t), \mathbf{y}_i(t), \mathbf{u}_i, \boldsymbol{\theta}_i, \mathbf{p}, t), \qquad t \in [t_i, t_{i+1}], \tag{3.6b}$$

$$\mathbf{z}(t_i) = \mathbf{z}_{i,0} \tag{3.6c}$$

where $\dfrac{\partial G}{\partial y}$ is nonsingular in a neighborhood of the exact solution. Theoretically, by the implicit function theorem, from Eq. (3.6b) we have:

$$\frac{\partial G}{dt} + \frac{\partial G}{dz}\frac{\partial z}{dt} + \frac{\partial G}{dy}\frac{\partial y}{dt} = 0 \;\Rightarrow\; \frac{\partial y}{dt} = -\left(\frac{\partial G}{dy}\right)^{-1}\frac{\partial G}{dz}\left(\frac{\partial z}{dt}\right) - \left(\frac{\partial G}{dy}\right)^{-1}\frac{\partial G}{dt}$$

One differentiation step yields the differential equation:

$$\frac{\partial y}{dt} = -\left(\frac{\partial G}{dy}\right)^{-1}\frac{\partial G}{dz}F(z, y, u, \theta, p, t) - \left(\frac{\partial G}{dy}\right)^{-1}\frac{\partial G}{dt} \tag{3.7a}$$

$$= \hat{G}(z_i(t), y_i(t), u_i(t), \theta_i(t), p, t) \tag{3.7b}$$

The values of independent variables $u_i(t), \theta_i(t), p$ are given in this case. For a consistent representation we can couple the variables:

$$x_i(t) = \begin{pmatrix} z_i(t) \\ y_i(t) \end{pmatrix} \; ; \; x_i(t) \in R^{n_x}; \;\; n_x = (n_z + n_y)$$

then Eq. (3.6) becomes:

$$\begin{aligned} &\dot{x} = \hat{F}(x_i(t), u_i(t), \theta_i(t), p, t), \\ &x_i(t_{i.0}) = x_{i,0}, \\ &\text{where} \quad t_{i,0} = t_i, \;\; x_{i,0} \in R^{n_x} \end{aligned} \tag{3.8}$$

The ODE form of Eq. (3.8) can be solved by various methods which can be found in rich of literatures written in mature field of numerical solution for ODEs, and therefore will not be discussed any further in this thesis.
The initial value DAEs are not easy to transform and underlying ODE may be stiff, and thus direct discretization methods are usually preferred. Collocation methods and backward differentiation formula (BDF) are most widely used for numerical solution of explicit index-1 DAEs [Ascher and Petzold, 1998; Brenan et al., 1995].

### 3.2.1 Backward Differentiation Formulas Methods

For a consistent representation, we consider the following initial value DAE problem:

$$\dot{z} = f(t, z, y, \bar{p}) \tag{3.9a}$$

$$0 = g(t, z, y, \bar{p}) \tag{3.9b}$$

$$z(t_0) = z_0 \tag{3.9c}$$

where $\bar{p} = (u, \theta, p)^T \in R^{n_{\bar{p}}}, \; n_{\bar{p}} = n_u + n_\theta + n_p$ are known values of the independent variables, $z \in R^{n_z}, y \in R^{n_y}$, $t \in [t_0, t_f]$.
Given $z_i = z(t_i)$ and $y_i = y(t_i)$ and let $h = t_{i+1} - t_i$ be the step-size, the basic idea of BDF methods is simply to approximate $z$ and $\dot{z}$ by a discretization formula like linear multi-step methods:

$$\dot{z}_{i+1} = (\text{linear} \quad \text{combination} \quad \text{of} \quad z_{i+1}, z_i, z_{i-1}, \cdots, z_{i-m+1}),$$

in the mean time simultaneously compute $y_{i+1} = y(t_{i+1})$. There exists a unique $m-th$ degree polynomial $P$ that interpolates the $m-1$ points:

$$(t_{i+1}, z_{i+1}), (t_i, z_i), (t_{i-1}, z_{i-1}), \ldots, (t_{i-m+1}, z_{i-m+1})$$

In general, polynomial $P$ can be written as:

$$P(t) = \sum_{j=0}^{m} z_{i-j+1} L_j(t) \tag{3.10}$$

with the Lagrange polynomial $L_j(t)$ is expressed as:

$$L_j(t) = \prod_{\substack{l=0 \\ l \neq j}}^{m} \left( \frac{t - t_{i-l+1}}{t_{i-j+1} - t_{i-l+1}} \right), \; j = 0, 1, \ldots, m. \tag{3.11}$$

One can observe that the Lagrange polynomial (3.11) has the property of:

$$P(t_{i-j+1}) = z_{i-j+1}, \quad j = 0, 1, \ldots, m.$$

Replacing $\dot{z}_{i+1}$ by $\dot{P}(t_{i+1})$ we obtain:

$$\dot{P}(t_{i+1}) = f(t_{i+1}, z_{i+1}) \tag{3.12}$$

From Eq. (3.10) at time $t = t_{i+1}$ we have:

$$\dot{P}(t_{i+1}) = \sum_{j=0}^{m} z_{i-j+1} \dot{L}_j(t_{i+1}) = z_{i+1} \dot{L}_0(t_{i+1}) + \sum_{j=1}^{m} z_{i-j+1} \dot{L}_j(t) \tag{3.13}$$

From Eq. (3.13) and Eq. (3.12) with some transformations we have:

$$z_{i+1} = -\sum_{j=1}^{m} z_{i-j+1} \frac{\dot{L}_j(t_{i+1})}{\dot{L}_0(t_{i+1})} + \frac{1}{\dot{L}_0(t_{i+1})} f(t_{i+1}, z_{i+1}) \tag{3.14}$$

We can define:

$$a_j = \frac{\dot{L}_j(t_{i+1})}{\dot{L}_0(t_{i+1})}, \; j = 1, \ldots, m$$

$$b_m = \frac{1}{h \dot{L}_0(t_{i+1})}$$

These values $a_j$ and $b_m$ can be read from lookup tables which are usually calculated by applying quadrature rules [Ascher and Petzold, 1998]. Then for a DAE in explicit form, the m-step BDF algorithm (BDF m) gives $z_{i+1}$ and $y_{i+1}$ as the solution of:

$$z_{i+1} = -\sum_{j=1}^{m} a_j z_{i-j+1} + b_m h f(t_{i+1}, z_{i+1}, y_{i+1}) \tag{3.15a}$$

$$0 = g(t_{i+1}, z_{i+1}, y_{i+1}) \tag{3.15b}$$

In BDF methods, Eq. (3.15) leads to a system of nonlinear equations that needs to be solved by iteration methods, e.g., (modified) Newton, and thus it requires the well-conditioned property of the Jacobian $\dfrac{\partial g}{\partial \omega}$ , where $\omega = (t, z, y)$. The algebraic variables $y(t)$ are determined by the consistent conditions including initials. Given $z_0 = z(t_0)$, BDF requires consistent initial values that can be obtained by solving:

$$g(t_0, z_0, y_0) = 0$$

to determine $y_0 = y(t_0)$.
This m-step BDF algorithm converges if $m \leq 6$, i.e.,

$$z_i - z(t_i) \leq O(h^m),\ y_i - y(t_i) \leq O(h^m)$$

for consistent initial conditions [Ascher and Petzold, 1998; Brenan et al., 1995].
At the end of this Section, it would be of importance to mention some software tools that utilize BDF for DAEs. In the MATLAB environment, *ode*15*i* uses the BDF method to solve ODEs and index-1 DAEs in a fully implicit form. Beside that, *ode*15*s* is a solver based on the numerical differentiation formulas (NDFs) which is a variant of the BDF method. Optionally it uses the exact BDF method [Shampine et al., 1999]. There are also some open source codes, such as DASSL [Brenan et al., 1995], IDA (under Sundials), ODEPACK, etc.

### 3.2.2 Collocation on Finite Elements

$z(t)$ and $y(t)$ in Eq. (3.9) are continuous functions (of time $t$) with unknown properties. The idea of collocation methods is that we collocate the state functions $x(t) = [z(t) \quad y(t)]^T$ through simpler functions, e.g., polynomials $P(t)$, in order to

capture the properties of $x(t)$ by using $P(t)$. This can be done by applying the Weierstrass approximation theorem [Antia, 2002]:

**Theorem 3.2.1 (Weierstrass Approximation Theorem (1885))**
*Consider a real value space $C[a,b]$ as the set of all continuous functions over a real interval $[a,b]$, together with $\infty-$ norm defined on it as $\|x(t)\| = \max_{t\in[a,b]} |x(t)|$*
*If $x(t)$ is a continuous real-valued function on the real interval $t \in [a,b]$, then for any given $\varepsilon > 0$, for every $x(t) \in C[a,b]$ there exists a polynomial function $p(t)$ such that for all $t \in [a,b]$, we have $|x(t) - p(t)| < \varepsilon$, or equivalently, $\|x - p\| < \varepsilon$.*

*Proof.* There are several proofs of this theorem. A constructive proof of this theorem using Bernstein polynomials can be found in [Bartle and Sherbert, 2011].

This fundamental result has been used to form the basis of various numerical techniques. However, this theorem does not specify how to construct the approximating polynomial $p(t)$.
An approximating polynomial $p(t)$ is required to have a given property:

$$p(t_i) = x(t_i) = x_i, \quad i = 1, \ldots, n. \tag{3.16}$$

A polynomial that satisfies the conditions in Eq. (3.16) is called *interpolating polynomial*. The points $t_i$ are called *interpolation points* or *interpolation nodes*. This relating property is known as *interpolatory property* based on the following theorem [Antia, 2002]:

**Theorem 3.2.2 (Uniqueness of the Interpolating Polynomial)**
*Given $n+1$ unequal points $t_0, t_1, t_2, \cdots, t_n$ and arbitrary values $x_0, x_1, x_2, \cdots, x_n$ there exists a unique polynomial $p(t)$ of degree less or equal to $n$ such that*

$$p(t_i) = x_i, \quad i = 0, 1, \ldots, n.$$

*Proof.* The proof of Theorem 3.2.2 can be referred to [Antia, 2002].

To construct the interpolating polynomial, at first a basic $p_0, p_1, \cdots, p_n$ of the space of polynomials of degree less or equal to $n+1$ are given, then we can write:

$$p(t) = a_0 p_0(t) + a_1 p_1(t) + \cdots + a_n p_n(t)$$

where $a_i, \quad i = 0, 1, \ldots, n$ are *coefficients* of polynomials.
From Eq. (3.16) the coefficients $a_i$ can be found by sequent of values:

$$\begin{aligned} p(t_0) &= a_0 p_0(t_0) + a_1 p_1(t_0) + \cdots + a_n p_n(t_0) = x_0 \\ p(t_1) &= a_0 p_0(t_1) + a_1 p_1(t_1) + \cdots + a_n p_n(t_1) = x_1 \\ &\vdots \\ p(t_n) &= a_0 p_0(t_n) + a_1 p_1(t_n) + \cdots + a_n p_n(t_n) = x_n \end{aligned} \tag{3.17}$$

Eq. (3.17) leads to the linear system written in the following way:

$$\begin{pmatrix} p_0(t_0) & p_1(t_0) & \cdots & p_n(t_0) \\ p_0(t_1) & p_1(t_1) & \cdots & p_n(t_1) \\ \vdots & \vdots & & \vdots \\ p_0(t_n) & p_1(t_n) & \cdots & p_n(t_n) \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix} \tag{3.18}$$

with assumption that $t_i \neq t_j$ for $i \neq j$. Thus, we can compute $[a_0 \quad a_1 \quad \cdots \quad a_n]^T$ if we know $[x_0 \quad x_1 \quad \cdots \quad x_n]^T$ and vice-versa. But $x_0, x_1, \cdots, x_n$ are unknown because $x(t)$ is not yet known. This means that both $[a_0 \quad a_1 \quad \cdots \quad a_n]^T$ and $[x_0 \quad x_1 \quad \cdots \quad x_n]^T$ in Eq. (3.18) are unknown. In order to avoid working with these two unknown vectors, we can define the basic polynomials $p_0, p_1, \cdots, p_n$ in a better way that makes Eq. (3.18) can be easily solved. Given $n+1$ unequal points $t_0, t_1, ..., t_n$, the $i^{th}$ Lagrange polynomial is written as:

$$L_i(t) = \prod_{\substack{j=0 \\ j \neq i}}^{n} \frac{t - t_j}{t_i - t_j}.$$

The Lagrange polynomials $L_i$ are polynomials of degree $n$ and has a given property:

$$L(t_k) = \begin{cases} 1, & \text{if } \; k = i \qquad (3.19a) \\ 0, & \text{if } \; k \neq i \qquad (3.19b) \end{cases}$$

With the basis functions $p_i(t) = L_i(t)$, the linear system (3.18) associated with the polynomial interpolation problem become:

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & \vdots & \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{pmatrix}$$

Then the interpolating polynomial is given by:

$$p(x_{|t_0,\ldots,t_n})(t) = \sum_{i=0}^{n} x_i L_i(t) \tag{3.20}$$

The polynomial $p(t)$ in Eq. (3.20) can be made to satisfy the Weierstrass' theorem by taking sufficiently large number of time instants $t_0, t_1, \cdots, t_n$ from $[a, b]$. The most important thing is how to chose these time points. A good idea is to select the same set of values $\tau_1, \tau_2, \ldots, \tau_n$ within the interval $[0, 1]$ and define the $t_i's$ collocation points on various intervals [a, b] as follows:

$$t_i = a + \tau_i \Delta t, \quad i = 1, \ldots, n$$

where $\Delta t = (b - a)$.

Numerical methods for one-dimensional integrals can provide information on how to select $\tau_1, \tau_2, \ldots, \tau_n$. Gauss quadrature rules are one of the best numerical integration methods [Ascher and Petzold, 1998; Brenan et al., 1995].
Suppose that we would like to evaluate an integral:

$$I[f] = \int_0^1 f(\tau) d\tau,$$

for some function $f(\tau)$ on an interval $[0, 1]$ by a basic quadrature rule:

$$Q_n[f] = \sum_{k=1}^{n} \omega_k f(\tau_k)$$

where the integration nodes $\tau_1, \tau_2, \ldots, \tau_n \in [0,1]$ and $\omega_1, \omega_2, \ldots, \omega_n$ are weights that are constructed based on the interval $[0,1]$.
Our aim at this point is to use an approximation:

$$I(f) = \int_0^1 f(\tau)d\tau \approx Q_n[f] = \sum_{k=1}^{n} \omega_k f(\tau_k).$$

Once the quadrature rule $Q_n[\cdot]$ is constructed it can be used to approximate integrals of various functions.

**Definition 3.2.1 (Scalar product)**
*The scalar product of functions $p_j$ and $p_l$ with respect to the interval $[0,1]$ is written as*

$$\langle p_j, p_l \rangle = \int_0^1 p_j(\tau)p_l(\tau)d\tau.$$

**Definition 3.2.2 (Orthogonal functions)**
*Two functions $p_j$ and $p_l$ are orthogonal on the interval $[0,1]$ if*

$$\langle p_j, p_l \rangle = \int_0^1 p_j(\tau)p_l(\tau)d\tau = 0.$$

Orthogonal polynomials on $[0,1]$ (as defined in Def. 3.2.2) often used are known as shifted Lagrange polynomials. The Lagrange polynomials satisfy the Three-term recurrence relation as theorem below [Shen et al., 2011]:

**Theorem 3.2.3 (Three-term recurrence relation)**
*Suppose $p_0, p_1, \ldots$ is the set of shifted Lagrange orthogonal polynomials on $[0,1]$ with degree $deg(p_n) = n$ and leading coefficient equal to 1. The shifted Lagrange polynomials are generated by the relation*

$$p_{n+1}(\tau) = (\tau - a_n)p_n(\tau) - b_n p_{n-1}(\tau)$$

*with $p_0(\tau) = 1$ and $p_{-1}(\tau) = 0$, where the recurrence coefficients are given as*

$$a_n = \frac{1}{2}, \quad n = 0, 1, 2, \ldots$$
$$b_n = \frac{n^2}{4(4n^2 - 1)}, \quad n = 0, 1, 2, \ldots$$

The first few shifted Lagrange orthogonal polynomials are shown in Table 3.1:

Table 3.1 Shifted Lagrange orthogonal polynomials

| Order | $p(\tau)$ |
|---|---|
| 0 | $1$ |
| 1 | $2\tau - 1$ |
| 2 | $6\tau^2 - 6\tau + 1$ |
| 3 | $20\tau^3 - 30\tau^2 + 12\tau - 1$ |
| 4 | $70\tau^4 - 140\tau^3 + 90\tau^2 - 20\tau + 1$ |

Given any set of orthogonal polynomials $p_0, p_1, p_2, \ldots$, their following properties hold true:

- Any finite set of orthogonal polynomials $p_0, p_1, \ldots, p_{N-1}$ is linearly independent.
- The polynomial $P_N$ is orthogonal to each of $p_0, p_1, \ldots, p_{N-1}$.
- Any non-zero polynomial $q$ with degree $deg(q) \leq N - 1$ can be written as a linear combination:

  $$q(\tau) = c_0 p_0(\tau) + c_1 p_1(\tau) + \cdots + c_{N-1} p_{N-1}(\tau)$$

  where at least one of the scalars $c_0, c_1, \ldots, c_{N-1}$ is non-zero.

A quadrature rule here is required to be exact for polynomials up to degree $2N - 1$. Let $P(\tau)$ be any polynomial of degree $2N - 1$ then:

$$I(P) = Q_N[P] \Rightarrow \int_0^1 P(\tau) d\tau = \sum_1^N \omega_i P(\tau_i) \tag{3.21}$$

Given the $N - th$ degree orthogonal polynomial $p_N$, the polynomial $P$ can be written as:

$$P(\tau) = p_N(\tau) q(\tau) + r(\tau)$$

where $q(\tau)$ and $r(\tau)$ are polynomials such that $0 < deg(r) \leq N-1$ and $deg(q) = N-1$ since $2N-1 = deg(P) = deg(p_N) + deg(q) = N + deg(q)$. From (3.21) it follows that:

$$\int_a^b (p_N(\tau)q(\tau) + r(\tau))d\tau = \sum_{i=1}^{N} \omega_i (p_N(\tau_i)q(\tau_i) + r(\tau_i))$$

$$\Rightarrow \int_a^b p_N(\tau)q(\tau)d\tau + \int_a^b r(\tau)d\tau = \sum_{i=1}^{N} \omega_i p_N(\tau_i)q(\tau_i) + \sum_{i=1}^{N} \omega_i r(\tau_i) \tag{3.22}$$

Orthogonality implies:

$$\int_a^b p_N(\tau)q(\tau)d\tau = 0. \tag{3.23}$$

Using polynomial exactness, we have:

$$\int_a^b r(\tau)d\tau = \sum_{i=1}^{N} \omega_i r(\tau_i). \tag{3.24}$$

From Eqs. (3.22)-(3.24), it follows that:

$$\sum_{i=1}^{N} \omega_i p_N(\tau_i)q(\tau_i) = 0. \tag{3.25}$$

It can be seen that if the quadrature nodes $\tau_1, \tau_2, \ldots, \tau_N$ are zeros of the $N-th$ degree shifted Lagrange polynomial $p_N(\tau)$, then:

- all the roots $\tau_1, \tau_2, \ldots, \tau_N$ lie inside $(0, 1)$;
- the quadrature weights are determined from:

$$\omega_i = \int_0^1 L_i(\tau)d\tau,$$

  where $L_i(\tau), \quad i = 1, \ldots, N$ is the Lagrange function defined by using $\tau_1, \tau_2, \ldots, \tau_N$ and $\omega_i > 0, \quad i = 1, \ldots, N$.

- the quadrature rule $Q_N[\cdot]$ integrates polynomials degree up to $2N-1$ exactly.

Hence, equation (3.25) holds true if $p_N(\tau_1) = p_N(\tau_2) = \cdots = p_N(\tau_N) = 0$. Therefore, the quadrature nodes $\tau_1, \tau_2, \ldots, \tau_N$ are chosen as the zeros of the $N-th$ degree shifted Lagrange orthogonal polynomial.

Table 3.2 Orthogonal collocation points at the roots of shifted Lagrange polynomials

| Order | Roots |
|---|---|
| 1 | 0.5000000 |
| 2 | 0.2113249, 0.7886751 |
| 3 | 0.1127017, 0.5000000, 0.8872983 |
| 4 | 0.0694318, 0.3300095, 0.6699905, 0.9305682 |
| 5 | 0.0469101, 0.2307653, 0.5000000, 0.7692347, 0.9530899 |

Table 3.2 lists values of $\tau_i$ for shifted Lagrange polynomials for orders up to 5. Using these values we can construct collocation equations by first determining the collocation points $t_0, t_1, t_2, \ldots, t_N \in [a, b]$. Note that in Table 3.2 we have $0 < \tau_i < 1; i = 1, 2, \ldots, N$. In order to do transform the time interval from $[a, b]$ into $[0, 1]$ with a requirement that $t_0 = a$ and $t_N = b$ , we add $\tau_0 = 0$ and form the following relation:

$$t_i = a + \frac{\tau_i}{\tau_N}\Delta t \tag{3.26}$$

where $\Delta t = b - a; \quad i = 0, 1, \ldots, N.$
Then corresponding to each differential and algebraic variable in Eq. (3.9) we define the collocation polynomials:

$$z_k(t) = \sum_{i=0}^{N} z_i^{(k)} L_i(t), \qquad k = 1, \ldots, n_z; \tag{3.27}$$

$$y_j(t) = \sum_{i=0}^{N} y_i^{(j)} L_i(t), \qquad j = 1, \ldots, n_y; \tag{3.28}$$

where

$$L_i(t) = \prod_{\substack{l=0 \\ l \neq i}}^{N} \frac{t - t_l}{t_i - t_l}$$

By discretizing the system above using $t_0, t_1, \ldots, t_N$ we obtain:

$$\dot{z}_k(t_l) = \sum_{i=0}^{N} \frac{dL_i(t_l)}{dt} z_i^{(k)}, k = 1, \ldots, n_z; l = 0, 1, \ldots, N; \tag{3.29}$$

Here we represent the state in the given element $\Delta t$ as:

$$\sum_{i=0}^{N} \frac{dL_i(\tau_l)}{d\tau} z_i^{(k)} = \frac{\Delta t}{\tau_N} f_k \left( t_l, (z_l^{(1)}, z_l^{(2)}, \ldots, z_l^{(n_z)}), (y_l^{(1)}, y_l^{(2)}, \ldots, y_l^{(n_y)}), \bar{\mathrm{p}} \right); \tag{3.30a}$$

$$0 = g_j \left( t_l, (z_l^{(1)}, z_l^{(2)}, \ldots, z_l^{(n_z)}), (y_l^{(1)}, y_l^{(2)}, \ldots, y_l^{(n_y)}), \bar{\mathrm{p}} \right); \tag{3.30b}$$

$$k = 1, \ldots, n_z; \quad j = 1, \ldots, n_y; \quad l = 0, 1, \ldots, N;$$

where

$$L_i(\tau) = \prod_{\substack{l=0 \\ l \neq i}}^{N} \frac{\tau - \tau_l}{\tau_i - \tau_l}$$

We consider the case at $\tau_0 = 0$ (when $l = 0$). It can be seen that Eq. (3.30a) becomes trivial and thus can be neglected. Eq. (3.30b) can be explicitly written as:

$$0 = g_j \left( t_0, (z_0^{(1)}, z_0^{(2)}, \ldots, z_0^{(n_z)}), (y_0^{(1)}, y_0^{(2)}, \ldots, y_0^{(n_y)}), \bar{\mathrm{p}} \right); \quad j = 1, \ldots, n_y; \tag{3.31}$$

These $n_y$ equations can be solved to get initial values $y_0^{(j)}$ of algebraic states in order to meet the consistent requirement of DAEs Eqs. (3.9).
Let $x = [z \;\; y]^T \in R^{n_x}$; $n_x = n_z + n_y$, Eqs. (3.30) can be rewritten in the compact form as:

$$\begin{gathered} F_l^{(k)} \left( t_l, (x_l^{(1)}, x_l^{(2)}, \ldots, x_l^{(n_x)}), z_0^{(1)}, z_0^{(2)}, \ldots, z_0^{(n_z)}, \Delta t, \bar{\mathrm{p}} \right) = 0; \\ k = 1, 2, \ldots, n_x; \quad l = 1, 2, \ldots, N; \end{gathered} \tag{3.32}$$

where: $x_l^{(1)}, x_l^{(2)}, \ldots, x_l^{(n_x)}$ are unknown variables;
$z_0^{(1)}, z_0^{(2)}, \ldots, z_0^{(n_z)}$ are known initial values of the differential states;
$\Delta t$ is the length of time interval,
$\bar{\mathrm{p}}$ are parameters.

These $n_x \times N$ (nonlinear) equations can be solved by the Newton-Raphson method and its variants [Kelley, 2003]. Figure 3.2 shows the collocation method with the

number of collocation points $N = 3$. There are some others collocation methods that use different quadrature rules, such as Radau and Lobatto collocation [Ascher and Petzold, 1998; Biegler, 2010; Brenan et al., 1995]. Collocation methods have some advantages that they are efficient for both initial value as well as boundary value DAEs, numerically accurate, and can be used for higher index DAEs. Beside that, they also inherit some disadvantages, e.g., they are computationally expensive, the approximating polynomials may display oscillatory properties. In this dissertation, collocation on finite elements using shifted Lagrange polynomials is used due to the fact that the error, $|x_{exact}(t) - x(t)|$, absolutely being evenly distributed in the whole (time) $t$ domain [Gupta, 1995].

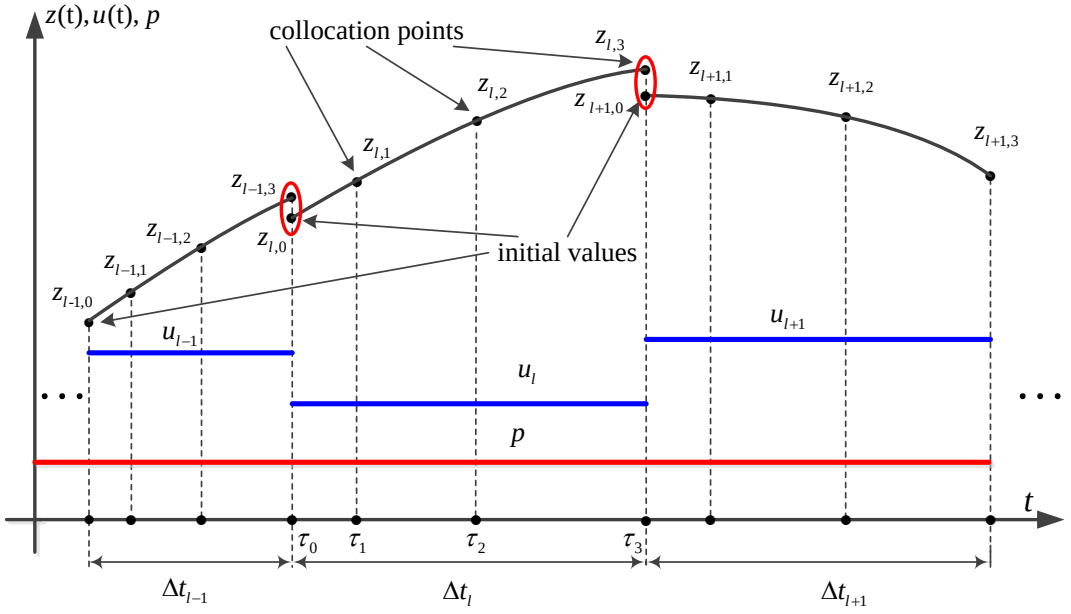



Fig. 3.2 Collocation on finite elements (NC=3)

### 3.2.3 Sensitivity Calculations

Accurate and efficient gradient calculations are needed for the NLP solver. To carry out this task, the derivatives of the dependent variables $z, y$ with respect to the independent variables $\bar{p}$ need to be computed implicitly and simultaneously with the

solution of the DAEs.
Sensitivity analysis of DAE models can be done using three methods: perturbation, direct sensitivity, and adjoint sensitivity [Biegler et al., 2012]. Perturbation is straightforward but inefficient and unreliable due to its truncation and roundoff errors. Therefore this method is not commonly used. The last two approaches which can provide sensitivity information with the desired level of accuracy are briefly described below.

#### 3.2.3.1 Direct Sensitivity Computation

We take into account a DAE model (3.9) in one time interval $t \in [0, t_f]$ with an assumption that $f(\cdot)$ and $g(\cdot)$ are continuously differentiable with respect to all their arguments. Sensitivity analysis requires finding the derivative of the model (3.9) with respect to each independent variable so that an additional $n_s = n_{\bar{p}} \times (n_z + n_y)$ sensitivity equations are yielded:

$$\begin{aligned} \dot{s}_i(t) &= \frac{\partial f}{\partial z} s_i(t) + \frac{\partial f}{\partial y} r_i(t) + \frac{\partial f}{\partial \bar{p}_i}, \quad s_i(0) = \frac{dz_0}{d\bar{p}_i} \\ 0 &= \frac{\partial g}{\partial z} s_i(t) + \frac{\partial g}{\partial y} r_i(t) + \frac{\partial g}{\partial \bar{p}_i}, \quad i = 1, \ldots, n_{\bar{p}} \end{aligned} \tag{3.33}$$

where $s_i(t) = \frac{dz}{d\bar{p}_i}$ and $r_i(t) = \frac{dy}{d\bar{p}_i}$.

Eqs. (3.33) are index-1 DAEs with the linearity characteristic and require the values of state variables in their right-hand sides. The DAEs (3.9) and sensitivity systems (3.33) can be solved by *staggered direct*, *simultaneous corrector*, and *staggered corrector* methods [Biegler et al., 2012; Feehery et al., 1997; Li and Petzold, 2000; Li et al., 2000]. BDF methods in Section 3.2.1 are often used to solve the combined state-sensitivity system due to their desirable stability properties. For these methods, several *Automatic Differentiation* (AD) tools, such as ADIFOR [Bischof et al., 1992], ADOL-C [Griewank et al., 1996; Walther and Griewank, 2012], are available to calculate the partial derivatives.
Considering large-scale DAE systems with a large number of state variables and sensitivities with respect to a large number of parameters, the forward sensitivity approach can become intractable. Therefore, these problems can often be more efficiently solved by the method of *Adjoint Sensitivity Analysis* [Cao et al., 2002, 2003].
There are a number of good IVP, ODE and DAE codes, including SUNDIALS [Hind-

marsh et al., 2005] in that CVODES for ODEs [Hindmarsh and Serban, 2002] and IDAS for DAEs with *Direct Sensitivity Analysis* and *Adjoint Sensitivity Analysis* capabilities [Radu Serban and Hindmarsh, 2012], etc.

#### 3.2.3.2 Collocation-based Sensitivity Computation

Using collocation methods, at the convergence of solving Eq. (3.32), the implicit function theorem gives:

$$\frac{d\mathbf{x}}{d\bar{\mathbf{p}}} = -\left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{F}}{\partial \bar{\mathbf{p}}} \tag{3.34}$$

where

$$\frac{d\mathbf{x}}{d\bar{\mathbf{p}}} = \begin{bmatrix} \frac{dx_1}{d\bar{p}_1} & \frac{dx_1}{d\bar{p}_2} & \cdots & \frac{dx_1}{d\bar{p}_{n_{\bar{p}}}} \\ \frac{dx_2}{d\bar{p}_1} & \frac{dx_2}{d\bar{p}_2} & \cdots & \frac{dx_2}{d\bar{p}_{n_{\bar{p}}}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{dx_{(N\times n_x)}}{d\bar{p}_1} & \frac{dx_{(N\times n_x)}}{d\bar{p}_2} & \cdots & \frac{dx_{(N\times n_x)}}{d\bar{p}_{n_{\bar{p}}}} \end{bmatrix}_{(N\times n_x)\times n_{\bar{p}}} \tag{3.35}$$

$$\frac{\partial \mathbf{F}}{\partial \bar{\mathbf{p}}} = \begin{bmatrix} \frac{dF_1}{d\bar{p}_1} & \frac{dF_1}{d\bar{p}_2} & \cdots & \frac{dF_1}{d\bar{p}_{n_{\bar{p}}}} \\ \frac{dF_2}{d\bar{p}_1} & \frac{dF_2}{d\bar{p}_2} & \cdots & \frac{dF_2}{d\bar{p}_{n_{\bar{p}}}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{dF_{(N\times n_x)}}{d\bar{p}_1} & \frac{dF_{(N\times n_x)}}{d\bar{p}_2} & \cdots & \frac{dF_{(N\times n_x)}}{d\bar{p}_{n_{\bar{p}}}} \end{bmatrix}_{(N\times n_x)\times n_{\bar{p}}} \tag{3.36}$$

$$\frac{\partial \mathbf{F}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{dF_1}{dx_1} & \frac{dF_1}{dx_2} & \cdots & \frac{dF_1}{dx_{(N\times n_x)}} \\ \frac{dF_2}{dx_1} & \frac{dF_2}{dx_2} & \cdots & \frac{dF_2}{dx_{(N\times n_x)}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{dF_{(N\times n_x)}}{dx_1} & \frac{dF_{(N\times n_x)}}{dx_2} & \cdots & \frac{dF_{(N\times n_x)}}{dx_{(N\times n_x)}} \end{bmatrix}_{(N\times n_x)\times(N\times n_x)} \tag{3.37}$$

The partial derivatives in Eqs. (3.36) and (3.37) can be computed by analytical methods for simple problems or by *automatic differentiation* (AD) tools, such as ADIFOR [Bischof et al., 1992], ADOL-C [Griewank et al., 1996; Walther and Griewank, 2012] and CppAD [Bell, 2012] for large complicated problems. In this thesis, CppAD code

is used due to its much smaller run times as compared to ADOL-C, which might be due to the fact that ADOL-C uses the hard disk to store its temporary values with large problems [Bell and Burke, 2008].
A similar way can be applied to compute the other sensitivity values:

$$\frac{d\mathbf{x}}{d\mathbf{x}_0} = -\left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{F}}{\partial \mathbf{x}_0} \tag{3.38}$$

$$\frac{d\mathbf{x}}{d\Delta t} = -\left(\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right)^{-1} \frac{\partial \mathbf{F}}{\partial \Delta t} \tag{3.39}$$

## 3.3 Methods for Solving Nonlinear Optimization Problems

The NLP problems resulted in direct methods to DOPs can be expressed as follows:

$$\min_{x \in R^n} \quad f(x) \tag{3.40a}$$

$$\text{s.t.} \quad h(x) = 0 \tag{3.40b}$$

$$g(x) \leq 0 \tag{3.40c}$$

where $f : R^n \to R$ is the objective function, $h : R^n \to R^m$ and $g : R^n \to R^p$ are the equality and inequality constraints, respectively. $f(x)$, $h(x)$ and $g(x)$ are assumed to be twice continuously differentiable. Some basic definitions and theorems, which are extracted from [Nocedal and Wright, 2006] in which all the proofs and details can be found, are given in the following.

### 3.3.1 Basic definitions and theorems

**Definition 3.3.1 (Feasible set)**
*The set of points that satisfy the equality* (3.40b) *and inequality* (3.40c) *constraints of the NLP problem* (3.40)*, i.e.,*

$$\mathcal{S} = \{x \in R^n | h(x) = 0, \quad g(x) \leq 0\} \tag{3.41}$$

*is called the* ***feasible set*** *of the NLP* (3.40)*. Each element of this set is referred to as* ***feasible point****.*

**Definition 3.3.2 (Local minimizers)**
*A point $x^* \in R^n$ is called a local minimizer of the NLP problem* (3.40) *if $x^* \in \mathcal{S}$ and there exits a neighborhood $\Omega(x^*) \subset R^n$ such that $f(x) \geq f(x^*)$ for all $x \in \Omega(x^*) \cap \mathcal{S}$.*

A Lagrangian function $\mathcal{L} : R^n \times R^m \times R^p \to R$ associated with the NLP (3.40) to investigate its local optimality is defined as:

$$\mathcal{L}(x, \lambda, \mu) = f(x) + \lambda^T h(x) + \mu^T g(x) \tag{3.42}$$

**Definition 3.3.3 (Active set)**
*Consider $x \in \mathcal{S}$, then the set of indexes:*

$$\mathcal{I}_{ac}(x) = \{1 \leq i \leq p \mid g_i(x) = 0\} \tag{3.43}$$

*is the set of active inequality constraints at $x$.*
*Its complement $\mathcal{I}_{ic}(x) = \{1, \ldots, p\} \backslash \mathcal{I}_{ac}(x)$ is referred to as the set of inactive inequality constraints.*

The vectors $\nabla f(x)$ are called the gradients of the objective function $f(x)$, i.e,

$$\nabla f(x) = \left( \frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_1}, \cdots, \frac{\partial f(x)}{\partial x_n} \right)^T$$

The vectors $\nabla^2_{xx} f(x)$ are referred as the Hessian of the objective function $f(x)$, i.e,

$$\nabla^2_{xx} f(x) = \begin{pmatrix} \frac{\partial^2 f(x)}{\partial x_1^2} & \frac{\partial^2 f(x)}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f(x)}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f(x)}{\partial x_2 \partial x_1} & \frac{\partial^2 f(x)}{\partial x_2^2} & \cdots & \vdots \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial^2 f(x)}{\partial x_n \partial x_1} & \cdots & \cdots & \frac{\partial^2 f(x)}{\partial x_n^2} \end{pmatrix}$$

The vectors $\nabla h_i(x), i = 1, \ldots, m$, and $\nabla g_j(x), j = 1, \ldots, p$ are called the gradients of the equalities $h_i(x)$ and inequalities $g_j(x)$ at $x$.
The vector $B(x)$ called Jacobian matrix denotes the group of gradients of constraints, i.e.,

$$B(x) = (\nabla h_i(x), i = 1, \ldots, m; \nabla g_j(x), j = 1, \ldots, p)^T$$

**Definition 3.3.4 (Linear Independence Constraint Qualification)**
*Consider the set of active inequality constraints $\mathcal{I}_{ac}(x^*), x^* \in \mathcal{S}$, the Linear Independence Constraint Qualification (LICQ) is satisfied at $x^*$ if the active constraint gradients*

$$G(x) = \{\nabla h_i(x^*), i = 1, \ldots, m; \nabla g_j(x^*), j \in \mathcal{I}_{ac}(x^*)\} \tag{3.44}$$

*are linearly independent.*

**Theorem 3.3.1 (Karush-Kuhn-Tucker conditions)**
*Assume that $x^* \in S$ is a local solution of the NLP problem* (3.40) *and that the LICQ is satisfied at $x^*$. Then, there exist Lagrange multipliers $\lambda^* \in R^m$ and $\mu^* \in R^p_+$ such that the following conditions hold true at the triple $(x^*, \lambda^*, \mu^*)$:*

$$\nabla_x \mathcal{L}(x^*, \lambda^*, \mu^*) = 0 \tag{3.45a}$$
$$h_i(x^*) = 0, \quad i = 1, \ldots, m \tag{3.45b}$$
$$g_j(x^*) \leq 0, \quad j = 1, \ldots, p \tag{3.45c}$$
$$\mu_k^* \geq 0, \quad k = 1, \ldots, p \tag{3.45d}$$
$$\mu_l^* g_l(x^*) = 0, \quad l = 1, \ldots, p \tag{3.45e}$$

where Eq. (3.45a) is *Stationarity condition*, Eqs. (3.45b) and (3.45c) are *Primal feasibility* conditions, Eq. (3.45d) is *Dual feasibility condition* and Eq. (3.45e) is *Complementary condition*. The triple $(x^*, \lambda^*, \mu^*)$ is called a Karush-Kuhn-Tucker (KKT)

point.
The first order optimality conditions (KKT-conditions) are only necessary conditions. They give information about the behavior of the first derivatives of the objective function $f(x)$ and the constraints $h(x)$ and $g(x)$ at a local solution $x^* \in \mathcal{S}$ of the NLP problem (3.40).

**Definition 3.3.5 (Strict complementarity)**
*Consider a local solution $x^* \in \mathcal{S}$ of the NLP problem* (3.40) *and Lagrange multipliers $\lambda^*, \mu^*$ that satisfy the KKT conditions Eq.* (3.45)*, then strict complementarity holds true if*

$$\mu_i^* > 0, \quad i \in \mathcal{I}_{ac}(x^*) \tag{3.46}$$

**Theorem 3.3.2 (Second order sufficient optimality conditions)**
*Assume that $x^* \in S$ is a feasible point and there exist Lagrange multipliers $\lambda^* \in R^m$ and $\mu^* \in R^p$ are satisfying the KKT conditions* (3.45)*. Furthermore, suppose that the strict complementarity holds at $x^*$ and that the Hessian of the Lagrangian with respect to $x$ is positive definite on the null space of active constraint gradients $G(x^*)^T$, i.e.,*

$$\omega^T \nabla^2_{xx} \mathcal{L}(x^*, \lambda^*, \mu^*)\omega > 0 \ \text{ for all } \ \omega \neq 0 \ \text{ such that } \ G(x^*)^T\omega = 0 \tag{3.47}$$

*Then, $x^*$ is a strict local solution of NLP problem* (3.40)*.*

### 3.3.2 Quadratic Programming

An important case of a NLP problem (3.40) arises when the objective function is quadratic and the constraints are linear. This type of NLP problems is called *quadratic programming* (QP) problem [Nocedal and Wright, 2006] and is usually treated as a subproblem in a sequence method for general constrained NLP. Its general form can be expressed as follows:

$$\min_{x \in R^n} \quad f(x) = \frac{1}{2}x^T A x - x^T a \tag{3.48a}$$

$$s.t. \quad B_1 x - c = 0 \tag{3.48b}$$

$$B_2 x - d \leq 0 \tag{3.48c}$$

where $A \in R^{n\times n}$ is symmetric, $B_1 \in R^{m\times n}$, $B_2 \in R^{p\times n}$ and logically, $a \in R^n$, $c \in R^m$, $d \in R^p$. The QP problem (3.48) is usually solved iteratively by *active set* or *interior point* strategies in that an equality constrained QP problem is executed. Therefore, in the next Section, methods for a QP problem with only equality constraints are discussed.

#### 3.3.2.1 Equality constrained quadratic programming

Consider a QP problem (3.48) with only equality constraints:

$$\min_{x\in R^n} \quad f(x) = \frac{1}{2}x^T Ax - x^T a \tag{3.49a}$$

$$s.t. \quad Bx - c = 0 \tag{3.49b}$$

where $B \in R^{m\times n}$ is assumed to have full row rank $m$, $m \leq n$.
The KKT conditions for problem (3.49) at the solution $x^* \in R^n$ can be derived as:

$$\begin{aligned} Ax^* + B^T\lambda^* - a = 0 \\ Bx^* - c = 0 \end{aligned} \tag{3.50}$$

where $\lambda^* \in R^m$ is the associated Lagrange multiplier.
Eq. (3.50) can be rewritten in matrix form as:

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} x^* \\ \lambda^* \end{pmatrix} = \begin{pmatrix} a \\ c \end{pmatrix} \tag{3.51}$$

Set $K = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}$ and define here a matrix $Z$ for which the columns span ker $A$, and consequently for which $AZ = 0$, then matrix $K$ is call the *KKT matrix* and the matrix $Z^TAZ$ is called the *reduced Hessian.*

**Lemma 3.3.3 (Existence, uniqueness and global of solution of KKT system)** *Suppose that $B \in R^{m\times n}$ has full row rank $m \leq n$ and that the reduced Hessian $Z^TAZ$ is positive definite. Then, consequently the $KKT$ matrix $K$ is nonsingular. Therefore, the $KKT$ system* (3.51) *provides the unique and global solution $(x^*, \lambda^*)$. $x^*$ is the global minimizer of the QP problem* (3.49).

Then the solution of the QP equality problem (3.49) can be achieved by solving the KKT system (3.51). There exits several methods to solve this linear KKT system, such as *Direct methods* including symmetric factorization, the range-space and null-space approach, or *Iterative methods* which can be further referred to [Nocedal and Wright, 2006].

#### 3.3.2.2 Inequality constrained quadratic programming

We now consider the QP problem (3.48) with more strict assumption that $A \in R^{n\times n}$ is symmetric positive (semi)definite so accordingly that problem (3.48) now becomes a convex QP problem. Two efficient groups of methods are widely used to solve convex QP problems. The first group are called *Active-set* methods which are effective for small and medium sized problems. The second group are *Interior-point* methods that are well suited for large-scale problems. There exits a special type of active-set methods, which is called a *gradient projection* method. This third one is most effective when the constraints in the problem are only simple bounds on the variables [Nocedal and Wright, 2006]. In the following, the first group of methods are described briefly.

**Active set strategies**

Active-set methods for inequality constrained QP can be divided into three sub-method: *primal, dual, and primal-dual.* In the following, primal method will be discussed.

Primal active-set method generates iterations that are kept feasible with respect to the primal problem (3.48) and simultaneously the objective function $f(x)$ is steadily decreased . For the ease of analysis, we rewrite the matrices $B_1$ and $B_2$ in the following form

$$B_1 = \begin{bmatrix} b_{11} \\ \vdots \\ b_{1m} \end{bmatrix}, \quad b_{1i} \in R^n, \quad B_2 = \begin{bmatrix} b_{21} \\ \vdots \\ b_{2p} \end{bmatrix}, \quad b_{2i} \in R^n, \tag{3.52}$$

Then the inequality constraints (3.48c) can be rewritten as

$$b_{2i}^T x \leq d_i, \quad 1 \leq i \leq p \tag{3.53}$$

The primal active-set method executes an iterative procedure:
Given a feasible iterate $x(\nu), \nu \geq 0$, an active-set is determined

$$\mathcal{I}_{ac}(x^{(\nu)}) \subset \{1, \ldots, p\} \tag{3.54}$$

and the corresponding constraints are considered as equality constraints, while the remaining inequality constraints are disregarded. Defining

$$q = x^{(\nu)} - x, \quad a^{(\nu)} = Ax^{(\nu)} - a, \tag{3.55}$$

and substituting (3.55) into the objective function (3.48a) we get

$$\begin{aligned} f(x) = f(x^{(\nu)} - q) &= \frac{1}{2} q^T A q - (a^{(\nu)})^T q + \frac{1}{2}(x^{(\nu)})^T A x^{(\nu)} - a^T x^{(\nu)} \\ &= \frac{1}{2} q^T A q - (a^{(\nu)})^T q + g_{cons} \end{aligned} \tag{3.56}$$

where $g_{cons} = \frac{1}{2}(x^{(\nu)})^T A x^{(\nu)} - a^T x^{(\nu)}$ is not dependent on $q$ so that it can be dropped out of the objective function (3.56).
The new equality constrained QP problem to be solved at the $(\nu + 1) - st$ iteration step is then written as:

$$\min_{q \in R^n} \quad \frac{1}{2} q^T A q - (a^{(\nu)})^T q \tag{3.57a}$$

$$s.t. \quad B_1 q = 0 \tag{3.57b}$$

$$b_{2i}^T q = 0, \quad i \in I_{ac}(x^{(\nu)}) \tag{3.57c}$$

After solving the QP problem (3.57) by the methods described in Section 3.3.2.1 we get $q^{(\nu)}$. Then the new iteration $x^{(\nu+1)}$ can be obtained by setting

$$x^{(\nu+1)} = x^{(\nu)} - \alpha_\nu q^{(\nu)}, \quad \alpha_\nu \in [0, 1] \tag{3.58}$$

where $\alpha_\nu$ is chosen such a way that $x^{(\nu+1)}$ stays feasible.
Suppose that $q^{(\nu)} = 0$ is the solution of the QP problem (3.57). According to the KKT conditions associated with that QP problem, there exist Lagrange multipliers

$\lambda^{(\nu)} \in R^m$ and $\mu_i^{(\nu)}, i \in \mathcal{I}_{ac}(x^{(\nu)})$, such that *Stationarity condition* is satisfied

$$a^{(\nu)} = Ax^{(\nu)} - a = \sum_{i=1}^{m} \lambda_i^{(\nu)} b_{1i} + \sum_{i \in I_{ac}(x^{(\nu)})} \mu_i^{(\nu)} b_{2i} \tag{3.59}$$

If we set

$$\mu_i^{(\nu)} = 0, \quad i \in \{1, \ldots, p\} \setminus \mathcal{I}_{ac}(x^{(\nu)})$$

it is obvious that the triple $(x^{(\nu)}, \lambda^{(\nu)}, \mu^{(\nu)})$ satisfy the first KKT condition with respect to the original QP problem (3.48). Also the second and the third KKT conditions hold true because $x^{(\nu)}$ is kept feasible. Moreover, the fourth KKT condition holds true if:

$$\mu_i^{(\nu)} \geq 0, \quad i \in \mathcal{I}_{ac}(x^{(\nu)})$$

Recalling that $A$ is symmetric positive semidefinite matrix, the four KKT conditions hold means $x^{(\nu)}$ is a global solution of the QP problem (3.48). In the other case that $\mu_j^{(\nu)} < 0, \quad j \in I_{ac}(x^{(\nu)})$, the removal of such constraints from the active-set needs to carry out. For more details, one can refer to [Nocedal and Wright, 2006].

### 3.3.3 Active-Set Sequential Quadratic Programming Methods

Sequential Quadratic Programming is considered as one of the most successful methods for the solution of general NLP problems owing to their reliability and efficiency [Gould and Toint, 2000; Nocedal and Wright, 2006]. SQP methods have been particularly used successfully to solve associated NLPs arising from optimal trajectory problems [Hargraves and Paris, 1987].

SQP methods are iterative procedures which obtain search directions by a sequence of sub-QP problems. In these methods, at each current iteration $x^k$ of the original problem (3.40), the constraints of each sub-QP problem are approximated by linearizing the constraints in the original problem, while the objective function of the sub-QP problem is approximated by a local quadratic function of the Lagrangian $\mathcal{L}(x, \lambda, \mu)$ (3.42).

To illustrate the sub-QP problem, the special case of problem (3.40) without inequality constraints is taken into account. A local minimizer of this problem is calculated

by solving the KKT system:

$$\Psi(x,\lambda) = \begin{pmatrix} \nabla_x \mathcal{L}(x,\lambda) \\ h(x) \end{pmatrix} = \begin{pmatrix} \nabla f(x) + B(x)^T \lambda \\ h(x) \end{pmatrix} = 0 \tag{3.60}$$

where $B(x)$ is the Jacobian of the equality constraints (3.40b).
Then the Jacobian that couples $\nabla\Psi_x(x,\lambda)$ and $\nabla\Psi_\lambda(x,\lambda)$ is given as:

$$\Psi'(x,\lambda) = \begin{pmatrix} \nabla^2_{xx} L(x,\lambda) & B(x)^T \\ B(x) & 0 \end{pmatrix} \tag{3.61}$$

Assuming that the Newton method is applied to solve the nonlinear system (3.60), from the current point $(x_k, \lambda_k)$ a search direction $(d_x, d_\lambda)$ is obtained by solving:

$$\begin{pmatrix} \nabla^2_{xx} \mathcal{L}(x_k,\lambda_k) & B(x_k)^T \\ B(x_k) & 0 \end{pmatrix} \begin{pmatrix} d_x \\ d_\lambda \end{pmatrix} = - \begin{pmatrix} \nabla f(x_k) + B(x_k)^T \lambda_k \\ h(x_k) \end{pmatrix} \tag{3.62}$$

In each Newton step at the current point $(x_k, \lambda_k)$ the following update is made:

$$\begin{aligned} x_{k+1} &= x_k + d_x \\ \lambda_{k+1} &= \lambda_k + d_\lambda \end{aligned} \tag{3.63}$$

By substituting Eq. (3.63) into Eq. (3.62) and doing elimination of $(B(x_k)^T \lambda_k)$ term we get:

$$\begin{pmatrix} A(x_k) & B(x_k)^T \\ B(x_k) & 0 \end{pmatrix} \begin{pmatrix} d_x \\ \lambda_{k+1} \end{pmatrix} = \begin{pmatrix} -\nabla f(x_k) \\ -h(x_k) \end{pmatrix} \tag{3.64}$$

where $A(x_k) = \nabla^2_{xx} \mathcal{L}(x_k, \lambda_k)$.
Explicitly rewrite Eq. (3.64) we have:

$$\begin{aligned} A(x_k) d_x + B(x_k)^T \lambda_{k+1} + \nabla f(x_k) &= 0 \\ B(x_k) d_x + h(x_k) &= 0 \end{aligned} \tag{3.65}$$

By comparing Eq. (3.64) to Eq. (3.50) and Eq. (3.65) to Eq. (3.51) together with considering Lemma (3.3.3) with its assumptions, we go to the conclusion that Eq. (3.65) is exactly the necessary condition associated with the following QP subproblem:

$$\begin{aligned} \min_{d_x} \quad & f(d_x) \quad = \tfrac{1}{2} d_x^T A(x_k) d_x + \nabla f(x_k)^T d_x \\ s.t. \quad & \qquad\qquad B(x_k) d_x + h(x_k) = 0 \end{aligned} \tag{3.66}$$

Therefore, instead of solving the KKT system (3.60) by the Newton method, we can solve a sequence of QP subproblems (3.66). This strategy motivates the SQP methods for solving NLP general problem (3.40).
For general NLP problems (3.40), we do linearization of both equality and inequality constraints to get:

$$\begin{aligned} \min_{d_x} \quad & \tfrac{1}{2} d_x^T \nabla^2_{xx} \mathcal{L}(x_k, \lambda_k, \mu_k) d_k + \nabla f(x_k)^T d_k \\ s.t. \quad & \nabla h(x_k) d_k + h(x_k) = 0 \\ & \nabla g(x_k) d_k + g(x_k) \leq 0 \end{aligned} \tag{3.67}$$

This QP subproblem can be solved by using methods for general QP problems, such as *active-set* methods, which are briefly described above. The solution of this QP subproblem $(d_k, \lambda_{k+1}, \mu_{k+1})$ forms the new iteration value $(x_k + d_k, \lambda_{k+1}, \mu_{k+1})$.
A simplest form of the active SQP method is stated in the following *Algorithm* (3.3.1).

**Algorithm 3.3.1:** a basic local SQP Algorithm for solving Eq. (3.40)

Choose an initial triple $(x_0, \lambda_0, \mu_0)$, set $k \leftarrow 0$;
**repeat**`/* Outer iteration */`
1. Evaluate $\nabla f(x_k), \nabla^2_{xx} \mathcal{L}(x_k, \lambda_k, \mu_k), h(x_k), g(x_k), \nabla h(x_k), \nabla g(x_k)$ ;
2. Solve QP problem (3.67) to get $d_k, \lambda_{k+1}, \mu_{k+1}$ `/* Inner iteration */` ;
3. Set $x_{k+1} \leftarrow x_k + d_k$ together with $\lambda_{k+1}, \mu_{k+1}$; $k \leftarrow k + 1$;

**until** *convergence criteria are satisfied*;

When the sub-QP problem in Algorithm (3.3.1) is solved by an *active-set* strategy, the active-set $\mathcal{I}_{ac}(x_k)$ of this problem at the solution will be used as a guess of the active set $\mathcal{I}_{ac}(x^*)$ at the optimal solution of the NLP (3.40) according to the following Theorem that was first stated in [Robinson, 1974] and then was rewritten in [Nocedal and Wright, 2006]:

**Theorem 3.3.4**

*Suppose that $x^*$ is a local solution of* (3.40) *at which the KKT conditions are satisfied for some $(\lambda^*, \mu^*)$. Suppose, too, that the linear independence constraint qualification* (*LICQ*) (*Definition* (3.3.4)), *the strict complementarity condition* (*Definition* (3.3.5)),

*and the second order sufficient conditions* (*Theorem* (3.3.2)) *hold at* $(x^*, \lambda^*)$. *Then if* $(x_k, \lambda_k, \mu_k)$ *is sufficiently close to* $(x^*, \lambda^*, \mu^*)$, *there is a local solution of the subproblem* (3.67) *whose active-set* $\mathcal{I}_{ac}(x_k)$ *is the same as the active-set* $\mathcal{I}_{ac}(x^*)$ *of the nonlinear program* (3.40) *at* $x^*$.

The main disadvantage of this inequality constrained QP (IQP) method arises from the expense of solving each QP subproblem in that the active-set $\mathcal{I}_{ac}(x_k)$ needs to be recalculated. This concern can be raised when the size of NLP problem increases.
To avoid this disadvantage, an equality QP (EQP) method can be used. In the EQP strategy, a working set that contains only a subset of the constraints at each outer iteration is selected such that all the constraints in this working set are imposed as equalities and the rest of the constraint space are ignored. The sub-QP problem is then solved within this working set as an equality QP problem (3.49). This working set is updated at every outer iteration by some rules [Nocedal and Wright, 2006]. The main advantage of this EQP method in the case of large-scale NLP problem is that the equality sub-QP problems (3.49) are less expensive to be solved than the general QP problems (3.67).
A very brief discreption to the principles of SQP methods is described above, while there still remain many important issues to put the methods into practice. More details on these important topics can be referred to [Biegler, 2010; Nocedal and Wright, 2006]. Research on (active-set) SQP methods was declined in between the late 1980s and early 1990s due to some reasons: the rise of interior point (IP) methods (which will be discussed in the next Section), the rapid development of computer architecture and the development of automatic differentiation (AD) methods which can provide first and second derivatives automatically. However, SQP methods attract the interest of researchers again with more development in ODE and PDE-based optimization with mesh refinement and mixed-integer nonlinear programming [Gill and Wong, 2012; Morales et al., 2011].

### 3.3.4 Interior-Point Methods

Together with above active-set SQP methods, interior-point (IP)(also known as Barrier) methods have been used successfully to solve nonlinear optimization problems. These two groups of methods with their variants form the most powerful algorithms

for large.scale NLPs [Nocedal and Wright, 2006]. In this Section, some basic knowledge of IP methods is described. The general NLP problem (3.40) can be rewritten in the following form:

$$\min_{x \in R^n} \quad f(x) \tag{3.68a}$$

$$\text{s.t.} \quad h(x) = 0 \tag{3.68b}$$

$$g(x) + s = 0 \tag{3.68c}$$

$$s \geq 0 \tag{3.68d}$$

where $f : R^n \to R$ is the objective function, $h : R^n \to R^m$ and $g : R^n \to R^p$ are the equality and inequality constraints, respectively. $f(x)$, $h(x)$ and $g(x)$ are assumed to be twice continuously differentiable. The inequality constraints $g(x)$ are transformed into equalities by adding a vector of positive slack variables $s$ (3.68d). In order to keep the slack variables $s$ positive and prevent them from being too close to zero, a positive parameter $\upsilon$ and a logarithm function $ln(\cdot)$ are introduced to form a barrier problem:

$$\min_{x \in R^n} \quad f(x) - \upsilon \sum_{i=1}^{p} \ln(s_i) \tag{3.69a}$$

$$\text{s.t.} \quad h(x) = 0 \tag{3.69b}$$

$$g(x) + s = 0 \tag{3.69c}$$

The solution of the barrier problem (3.69) with a sequence of parameter $\upsilon_k$ that is monotonically decreased to zero approaches that of problem (3.68) as well as problem (3.40).
The Lagrangian function for problem (3.69) is written as:

$$\mathcal{L}(x, s, \lambda, \mu) = f(x) - \upsilon \sum_{i=1}^{p} \ln(s_i) + \lambda^T h(x) + \mu^T (g(x) + s) \tag{3.70}$$

The KKT conditions for (3.69) can be written as follows:

$$\nabla f(x) + B_h^T(x)\lambda + B_g^T(x)\mu = 0 \tag{3.71a}$$
$$-\upsilon S^{-1}e + \mu = 0 \tag{3.71b}$$
$$h(x) = 0 \tag{3.71c}$$
$$g(x) + s = 0 \tag{3.71d}$$

where $B_h(x)$ and $B_g(x)$ are the Jacobian matrices of the functions $h(x)$ and $g(x)$, respectively, and $\lambda$ and $\mu$ are their Lagrange multipliers. $S$ and $M$ are defined to be the diagonal matrices whose diagonal entries are given by the vectors $s$ and $\mu$, respectively, and $e = (1, 1, \ldots, 1)^T$. Because the diagonal elements $s$ of $S$ are all positive, we can multiply equation (3.71b) by $S$ to yield:

$$\nabla f(x) + B_h^T(x)\lambda + B_g^T(x)\mu = 0 \tag{3.72a}$$
$$S\mu - \upsilon e = 0 \tag{3.72b}$$
$$h(x) = 0 \tag{3.72c}$$
$$g(x) + s = 0 \tag{3.72d}$$

In order to solve the KKT system (3.72) with the variables $x, s, \lambda, \mu$, the Newton method is applied to obtain:

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & 0 & B_h^T(x) & B_g^T(x) \\ 0 & M & 0 & S \\ B_h(x) & 0 & 0 & 0 \\ B_g(x) & I & 0 & 0 \end{bmatrix} \begin{bmatrix} d_x \\ d_s \\ d_\lambda \\ d_\mu \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + B_h^T(x)\lambda + B_g^T(x)\mu \\ S\mu - \upsilon e \\ h(x) \\ g(x) + s \end{bmatrix} \tag{3.73}$$

After having computed the step $d^k = [d_x^k, d_s^k, d_\lambda^k, d_\mu^k]^T$ of the current iterate $k$, the new iterate values $[x^{k+1}, s^{k+1}, \lambda^{k+1}, \mu^{k+1}]^T$ can be computed as:

$$\begin{aligned} x^{k+1} &= x^k + \alpha_x^{\max} d_x^k \\ s^{k+1} &= s^k + \alpha_s^{\max} d_s^k \\ \lambda^{k+1} &= \lambda^k + \alpha_\lambda^{\max} d_\lambda^k \\ \mu^{k+1} &= \mu^k + \alpha_\mu^{\max} d_\mu^k \end{aligned} \tag{3.74}$$

where $\alpha \in (0,1]$ is the step length that can be obtained by the *line search* or *trust region* methods, e.g.,

$$\begin{aligned} \alpha_s^{\max} &= \max\{\alpha \in (0,1] : s + \alpha d_s \geq (1-\tau)s\} \\ \alpha_\mu^{\max} &= \max\{\alpha \in (0,1] : \mu + \alpha d_\mu \geq (1-\tau)\mu\} \end{aligned} \tag{3.75}$$

with $\tau \in (0,1)$, typically $\tau = 0.995$.
Another important task in the IP method is to choose the sequence to decrease the value of the barrier parameter $\upsilon_k$. The $\upsilon_k$ can be held constant for a number of iterations until the Newton method (3.73) for the KKT conditions (3.72) converges to a certain tolerance. Another strategy can also be applied to update $\upsilon_k$ value in each iteration.

To build a basic algorithm for the IP method, an error function is introduced to set the convergence criteria for solution of KKT conditions (3.72):

$$E(x,s,\lambda,\mu;\upsilon) = max\left\{\left\|\nabla f(x) + B_h^T(x)\lambda + B_g^T(x)\mu\right\|, \|S\mu - \upsilon e\|, \|h(x)\|, \|g(x)+s\|\right\} \tag{3.76}$$

The basic algorithm for the IP method is stated by the following *Algorithm* (3.3.2):

**Algorithm 3.3.2:** Basic Interior-Point Algorithm for solving NLP problem Eq. (3.40) [Nocedal and Wright, 2006]

Choose initial values $x_0$ and $s_0 > 0$, compute initial values $f(\lambda_0, \mu_0)$. Select a value for initial barrier parameter $\upsilon_0 > 0$ and parameters $\sigma, \tau \in (0,1)$. Set $k \leftarrow 0$;
**repeat**/* Outer iteration */
  **repeat**/* Inner iteration */
    1. Solve (3.73) to obtain the search direction $d = (d_x, d_s, d_\lambda, d_\mu)$;
    2. Compute $(\alpha_s^{\max}, \alpha_\mu^{\max})$ using (3.75);
    3. Compute $(x^{k+1}, s^{k+1}, \lambda^{k+1}, \mu^{k+1})$ using (3.74);
    3. Set $\upsilon_{k+1} \leftarrow \upsilon_k$ and $k \leftarrow k+1$;
  **until** $E(x,s,\lambda,\mu;\upsilon) \leq \upsilon_k$;
  Choose $\upsilon_k \in (0, \sigma\upsilon_k)$;
**until** *convergence criteria are satisfied*;

For more details on IP methods, one can see [Biegler et al., 2012; Byrd et al., 2000,

1999, 1998, 2006; Nocedal and Wright, 2006]. There exists several software packages that utilize IP methods. KNITRO [Byrd et al., 2000, 1999, 1998, 2006; Nocedal and Wright, 2006] is a commercial package that is written in FORTRAN77 for solving general NLP problems. KNITRO combines two IP methods together with an AS-SQP methods to provides flexible solvers. Its interfaces include C/C++, FORTRAN, JAVA, AMPL, GAMS, Mathematica, MATLAB and even in MS EXCEL environment. IPOPT [Wächter and Biegler, 2006] is an open source package that is written in both FORTRAN (old version) and C/C++ for large, sparse and many degrees of freedom NLP problems [Poku et al., 2004]. The NLP problems can be directly coded in FORTRAN, C/C++. Beside that, IPOPT has been integrated into various modeling environment, such as AMPL, GAMS, MATLAB. For other IP codes, readers can refer to [Biegler et al., 2012; Nocedal and Wright, 2006].

### 3.3.5 Summary of Numerical Method for NLPs

In the sections above, two group of methods for solving NLP problems, active-set SQP and interior-point, are briefly introduced. Each group has its advantages and disadvantages and is only suitable for a certain class of NLP problems. As for the size of NLP problems, IP methods are shown to be more efficient in dealing with large and very large problems, while AS-SQP methods are suitable for small and medium problems. Problems with a number of linear constraints should be better solved by AS-SQP methods because IP methods do not distinguish between linear and nonlinear constraints. In the case of a good initial vector available that is feasible and near the solution, AS-SQP methods will work better due to their fast convergence, while IP methods can deal with any initials, even with infeasible ones. IP methods are fast in dealing with one-off problems in that the structure of the KKT matrix does not change at each iteration so that they can inherit advantage of sparsity of the Hessian, Jacobian in (3.73). Problems with many inequality constraints can be solved faster with IP methods due to the avoidance of the combinatorial problem in selecting the active-set, which happens in AS-SQP methods [Jockenhövel, 2004a].
For the aim of solving dynamic optimization problems, the characteristics of the resulted NLP problem depend on the method of discretization that will be discussed in the next Section. It means that each of discretization methods produces a different type of NLP problem. Therefore one should carefully chose an appropriate solver for

solving resulted NLP problem.

## 3.4 Sequential approach

In a *sequential approach*, also known as direct single shooting or control parameterization method, only the inputs $u(t)$ of the problem (2.8) are discretized as piecewise polynomials or piecewise constants and their coefficients $\hat{u}$ become optimization variables of the NLP solver in the optimization layer. The model equations (2.8b-2.8d) will be solved by appropriate numerical integration methods in a simulation layer to get the values $\tilde{x} = (z, y)^T$ and their sensitivities with respect to $\hat{u}$ and $p$ [Barton et al., 1998; Binder et al., 2001; Logsdon and Biegler, 1992; Vassiliadis, 1993; Vassiliadis et al., 1994a]. All of this information then is used to compute the value of the objective function $f$ and its gradients with respect to $\hat{u}$ and $p$. The general constraints $c$ excluding the model equations are then calculated and transfered to the optimization layer. In the control and parameter space, the NLP solver solves the problem with updated control values $\hat{u}$ and $p$ and gives the results back to the simulation layer. The dimension of the resulting NLP in a sequential method is the smallest one in comparison with other methods and does not depend on the size of the model equations. This method is a *feasible path* strategy because the model equations are satisfied at each iteration of the NLP algorithm, which can be still usable in the case that the NLP solver cannot converge to the final solution. Sequential methods could be time consuming for large-scale DAE systems. Besides that, it also inherits other disadvantages of difficulties in the integration step with stiff and unstable problems due to the fact that intermediate inputs in early time may usually have a strong effect on the later parts of the trajectories. Furthermore, state path constraints can be only handled approximately in this method [Zavala et al., 2008a].

## 3.5 Simultaneous approach

In the *simultaneous approach*, both input and state variables of the problem (2.8) are discretized by using a discretization method at the same time (e.g., methods described in Section 3.1 for the inputs and Section 3.2.2 for the states variables). Then all of these discretized control and state variables are put into the NLP solver [Biegler, 2007;

Biegler et al., 2002; Jockenhövel, 2004a]. In this methods, DAE model Eqs. (2.8b-2.8d) are reformulated into algebraic equations that then are treated as equalities in the resulted NLP problem. Without loss of generality, here controls $u$ are assumed to be discretized in the same manner as states $z$ and $y$, and $p$ are assumed to be constant all over the time. The DOP problem (2.8) then can be formulated as follows:

$$\min_{\hat{z}_{l,j},\hat{y}_{l,j},\hat{u}_{l,j},p} f(\hat{z}_{l,j},\hat{y}_{l,j},\hat{u}_{l,j},p) \tag{3.77a}$$

$$\text{s.t.} \quad \dot{L}_0(\tau_j)\hat{z}_{(l-1),NC} + \sum_{k=1}^{NC} \dot{L}_k(\tau_j)\hat{z}_{l,k} - \frac{\Delta t_l}{\tau_{NC}} F(\hat{z}_{l,j},\hat{y}_{l,j},\hat{u}_{l,j},p) = 0 \tag{3.77b}$$

$$G(\hat{z}_{l,j},\hat{y}_{l,j},\hat{u}_{l,j},p) = 0, \tag{3.77c}$$

$$\hat{z}_L \le \hat{z}_{l,j} \le \hat{z}_U, y_L \le \hat{y}_{l,j} \le y_U, u_L \le \hat{u}_{l,j} \le u_U, p_L \le p \le p_U, \tag{3.77d}$$

$$l = 1,\ldots,NL;\ j = 1,\ldots,NC \tag{3.77e}$$

$$z_{0,NC} = z(t_0) = z_0 \tag{3.77f}$$

where $NL$ and $NC$ are the number of time intervals and inner collocation points, respectively. The continuity of the differential states is enforced in Eq. (3.77b), in that the initials of the next interval are treated as the last values of the previous interval.

For ease of use, NLP problem (3.77) can be rewritten in compact form as following:

$$\min_{\hat{z}_{l,j},\hat{y}_{l,j},\hat{u}_{l,j},p} f(\hat{z}_{l,j},\hat{y}_{l,j},\hat{u}_{l,j},p) \tag{3.78a}$$

$$\text{s.t.} \quad 0 = \hat{G}(\hat{z}_{l,j},\hat{y}_{l,j},\hat{u}_{l,j},p), \tag{3.78b}$$

$$0 \le \hat{H}(\hat{z}_{l,j},\hat{y}_{l,j},\hat{u}_{l,j},p), \tag{3.78c}$$

$$l = 1,\ldots,NL;\ j = 1,\ldots,NC \tag{3.78d}$$

$$z_{0,NC} = z(t_0) = z_0 \tag{3.78e}$$

In order to rewrite the NLP problem (3.77) into a compact form, let $\hat{z}$, $\hat{y}$ and $\hat{u}$ denote the elements of the corresponding discretized variables $\hat{z}_{l,j}$, $\hat{y}_{l,j}$ and $\hat{u}_{l,j}$, respectively. The equalities (3.77b) and (3.77c) can be combined as vector functions denoted as $h$. The NLP problem (3.77) then can be reformulated as:

$$\min_{x \in R^n} f(x) \tag{3.79a}$$

$$\text{s.t.} \quad g(x) = 0 \tag{3.79b}$$

$$h(x) \geq 0 \tag{3.79c}$$

where $x = (\hat{z}, \hat{y}, \hat{u}, p)^T$; $\hat{z} \in R^{n_z \times NL \times NC}$; $\hat{y} \in R^{n_y \times NL \times NC}$; $\hat{u} \in R^{n_u \times NL \times NC}$; $p \in R^{n_p}$; and $x \in R^n$ with $n = ((n_z + n_y + n_u) \times NL \times NC + n_p)$.

Due to the discretization of both state and control variables, this method leads to the largest dimension for the NLP problem that usually requires special solution strategies for the NLP problem (3.79), especially in the case of large-scale DAE systems [Jockenhövel, 2004a]. In spite of this fact, DSM can easily solve DOPs problems with path constraints on state variables by imposing each discretized variable at each collocation point, see Eq. (3.77d). Due to the fact that the DSM solves the DAEs system only once at the convergence point where all the equality constraints are satisfied, this method can avoid solving intermediate instable systems that may have no solution or require much computation effort. Contrary to the sequential strategy, DSM is an *infeasible path* method, meaning that intermediate non-convergence results will be useless. The accuracy of this method depends heavily on the fine level of the discretization intervals $\Delta t_l$ which can lead to very large and sparse NLP problems. In general, the result of the DSM is assessed as less accurate than that of the sequential strategy. However, the DSM usually is less time consuming than the sequential strategy [Jockenhövel, 2004a]. The article [Biegler, 2007] and references therein gives more details about this method.

## 3.6 Quasi-sequential approach

In *quasi-sequential strategy (QSQ)*, the DSQ using the collocation method was developed and applied successfully to optimal control of a large-scale dynamic system [Hong et al., 2006]. The QSQ is considered as a hybrid method that inherits features from both DSM and DSQ methods. In this method, the control variables are discretized in each time interval in the same manner as DSQ while the state variables are dis-

cretized using collocation on finite elements in the same interval as in DSM method. A simulation layer is used to deal with the nonlinear equation systems resulted from discretized DAE model. Using a nonlinear solver, the model equation is solved sequentially from the first interval to the last to get the state values and appropriate sensitivity information. The continuity of the states between the intervals are forced by using the last collocation point of the previous interval as the initial of the next interval. In comparison to the DSM method, the state variables are eliminated from the NLP leading to a smaller NLP problem like in the DSQ method. The inequalities (2.8e) are forced to become inequality constraints of the NLP at each collocation point of the model equations. For the ease in discussion, let $\hat{x} = (\hat{z}, \hat{y})^T$ and $c$ denote the model Eqs. (2.8b-2.8d). The DOP problem (2.8) then can be formulated as follows:

$$\min_{\hat{x},\hat{u},p} \quad f(\hat{x},\hat{u},p) \tag{3.80a}$$

$$\text{s.t.} \quad c(\hat{x},\hat{u},p) = 0 \tag{3.80b}$$

$$\hat{x}_L \leq \hat{x} \leq \hat{x}_U \tag{3.80c}$$

$$\hat{u}_L \leq \hat{u} \leq \hat{u}_U \tag{3.80d}$$

$$p_L \leq p \leq p_U \tag{3.80e}$$

At iteration $k$ in the optimization layer, the Newton-Raphson method is used to solve Eq. (3.80b) for $\hat{x}$, i.e., there exists an implicit relation:

$$\hat{x} = \hat{x}(\hat{u}, p) \quad \text{or} \quad \hat{x} = \hat{x}(\tilde{u}) \quad \text{with} \quad \tilde{u} = (\hat{u}, p)^T \tag{3.81}$$

The problem (3.80) is then reduced to the form:

$$\min_{\tilde{u}} \quad f(\hat{x}(\tilde{u}),\tilde{u}) \tag{3.82a}$$

$$\text{s.t.} \quad 0 \leq \hat{H}(\tilde{u}) \tag{3.82b}$$

$$\tilde{u}_L \leq \tilde{u} \leq \tilde{u}_U \tag{3.82c}$$

In this method, the model Eqs. (3.80b) are solved successively from element to element at each NLP iteration, so that sensitivities of states with respect to controls can be computed simultaneously. Here by considering the discretized model Eqs. (3.80b) in the $k^{th}$ iteration of NLP problem with the $l^{th}$ element, the model equations at the collocation points can be written as:

$$c_l\left(\hat{z}_{l,0}^k, \tilde{u}_l^k, \hat{z}_l^k\right) = 0; \qquad l = 1, ..., NL \tag{3.83}$$

where $\hat{z}_{l,0}^k$ is the initial value of $\hat{z}_l^k$ of the element, $NL$ is the number of elements, respectively. According to the theorem of implicit functions, one obtains:

$$\frac{d\hat{z}_l}{d\tilde{u}_l} = -\nabla_{\hat{z}_l} c_l^{-T} \nabla_{\tilde{u}_l} c_l \qquad \text{and} \qquad \frac{d\hat{z}_l}{d\hat{z}_{l,0}} = -\nabla_{\hat{z}_l} c_l^{-T} \nabla_{\hat{z}_{l,0}} c_l \tag{3.84}$$

For the state profile continuity, the last collocation point of element $l$ is taken as the initial point of the next element $l+1$, so that the sensitivities can be transferred by the chain rule. The Jacobian of the NLP problem (3.82) has the following structure:

$$\frac{d\hat{z}}{d\tilde{u}} = \begin{bmatrix} a_1 & & & & & & \hat{z}_1, \text{interval } 1 \\ a_{2,1} & a_2 & & & 0 & & \hat{z}_2, \text{interval } 2 \\ \vdots & \ddots & \ddots & & & & \\ a_{2,l} & \ddots & \ddots & a_l & & & \hat{z}_l, \text{interval } l \\ \vdots & \ddots & a_{i,j} & \ddots & \ddots & & \\ a_{NL,1} & a_{NL,2} & \cdots & a_{NL,l} & \cdots & a_{NL} & \hat{z}_{NL}, \text{interval NL} \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix} \tag{3.85}$$

$$\begin{matrix} \tilde{u}_1 & \tilde{u}_2 & \tilde{u}_l & \tilde{u}_{NL} \\ \text{interval 1} & & \text{interval } l & \text{interval NL} \end{matrix}$$

where intervals mean the time elements of discretization of model Eqs. (3.80b) and the elements of the Jacobian of NLP problem (3.82) can be calculated as:

$$\begin{aligned} & a_l = \left(\frac{d\hat{z}_l^k}{d\tilde{u}_l^k}\right)^T l = 1, \ldots, NL \\ & a_{i,j} = \left(\frac{d\hat{z}_i^k}{d\tilde{u}_j^k}\right)^T = \left(\frac{d\hat{z}_i^k}{d\hat{z}_{0,i}^k}\right)^T \left(\frac{d\hat{z}_{i-1}^k}{d\tilde{u}_{0,i-1}^k}\right)^T \cdots \left(\frac{d\hat{z}_j^k}{d\tilde{u}_j^k}\right)^T \\ & i, j = 1, ..., NL; \qquad i > j \end{aligned} \tag{3.86}$$

This problem consists only of control variables and parameters as well as inequality constraints which are enforced at each collocation point. This method has advantages of a small NLP problem and easy handling of sensitivity computation and of inequalities. The active-set SQP was applied to solve the NLP problems (3.82). As pointed out, the SQP method can be computationally expensive to deal with many active constraints in a NLP problem.

**Algorithm 3.6.1:** Algorithm of active-set SQP quasi-sequential approach

Given a number of elements, discretize the control variables, and state variables with the collocation method, provide initial values of control variables and bounds for all variables, set $k = 0$;

**repeat**`/* Outer iteration */`

1. **for** $j = 1$ *to* $NL$ **do**
   1. Solve nonlinear model Eqs. (3.80b) using the (variation of) Newton-Raphson method;
   2. Evaluate the sensitivities $\dfrac{d\hat{x}}{d\tilde{u}^k}$, $\dfrac{d\hat{x}}{d\tilde{x}_0^k}$;

**end**

; `/* Simulation layer */`

2. Evaluate the values of the object function and constraints: $f(\hat{x}(\tilde{u}^k), \tilde{u}^k), \hat{H}(\tilde{u}^k)$, as well as their gradients with respect to $\tilde{u}^k$: $\nabla_{\tilde{u}} f(\hat{x}(\tilde{u}^k), \tilde{u}^k), \nabla_{\tilde{u}} \hat{H}(\tilde{u}^k)$ as following:

$$\nabla_{\tilde{u}} f = \frac{\partial f}{\partial x^k}\frac{\partial \hat{x}^k}{\partial \tilde{u}^k} + \frac{\partial f}{\partial \tilde{u}^k}$$

$$\text{and} \quad \nabla_{\tilde{u}} \hat{H} = \frac{\partial \hat{H}}{\partial \hat{x}^k}\frac{\partial \hat{x}^k}{\partial \tilde{u}^k} + \frac{\partial \hat{H}}{\partial \tilde{u}^k}$$

3. Call the active-set SQP solver to solve the NLP to get the search direction $d\tilde{u}_k$;
4. Set $\tilde{u}_{k+1} \leftarrow \tilde{u}_k + d\tilde{u}_k; k \leftarrow k + 1$;

**until** *convergence criteria are satisfied*;

## 3.7 Multiple shooting

Direct multiple shooting can be considered as a hybrid method that was developed by Bock and Plit [1984]; Plitt [1981]. In this method, the time horizon was divided into discrete time intervals in which control variables are discretized in the same manner

as DSQ. The initial conditions of the states are separately parameterized in each subinterval $l$.

$$z_{l,0}(t_{l,0}) = z_{l,0} \tag{3.87a}$$
$$y_{l,0}(t_{l,0}) = y_{l,0} \tag{3.87b}$$

The discretized values $\hat{u}$, $p$ and the initials $z_{l,0}, y_{l,0}$ become optimization variables of the NLP solver in the optimization layer. Model Eqs. (2.8b-2.8d) are solved on each time interval by appropriate ODE or DAE solvers to get all the information in the same manner as DSQ method. In order to keep the differential state continuity between the subintervals, additional interconnections, which enforce the last values of previous interval to be equal to the initials of the next one, are introduced as equality constraints in the optimization NLP problem. Together with the algebraic states, the consistency initials must be satisfied at the grid-points.

$$z_{(l+1),0} = z_l(z_{l,0}, y_{l,0}, \hat{u}_l, p, t_l) \tag{3.88a}$$
$$0 = G(z_{l,0}, y_{l,0}, \hat{u}_l, p, t_l) \tag{3.88b}$$

This fact makes the dimension of the NLP problem slightly larger than that of the DSQ but smaller than that of DSM. This method also allows advanced DAE solvers to be applied to calculate the function and derivative values. Furthermore the DAE solvers are decoupled on separate multiple shooting intervals, and therefore the DMS method is suitable for parallel computation. This approach also can deal with multistage problems, control and path constraints as well as multi-point boundary conditions. For the optimization of boundary value problems, DMS is considered to be more stable and efficient than the DSQ [Bock et al., 2000]. Beside that, this method also has some inherent drawbacks, e.g., the sensitivity information is too expensive to achieve and state constraints are difficult to implement, especially those which are inside each time interval [Jockenhövel, 2004b].

## 3.8 Combined multiple shooting and collocation strategy

The combined multiple shooting and collocation strategy can be considered as a combination of the DMS and the DSM strategy [Tamimi and Li, 2010]. This method is based on the DMS strategy with a modification in solving the DAEs model equations. In the CMSC formulation, the DAEs model equations in each time interval are discretized using collocation on finite elements to formulate a system of nonlinear equations as in the DSM and the QSQ methods. The inequalities (2.8e) are forced to become inequality constraints of the NLP at the grid points of the model equation. Let $\hat{x} = (\hat{z}, \hat{y})^T$ and $c$ denote the model Eqs. (2.8b-2.8d). The DOP problem (2.8) then can be formulated as follows:

$$\min_{z_{l,0},\hat{u}_l,p} \quad f(\hat{x}_l, \hat{u}_l, p) \tag{3.89a}$$

$$\text{s.t.} \quad c_l(\hat{x}_{l,i}, \hat{u}_l, z_{l,0}, p) = 0 \tag{3.89b}$$

$$z_{l+1,0} = \hat{z}_{l,NC} \tag{3.89c}$$

$$\hat{x}_L \leq \hat{x}_{l,NC} \leq \hat{x}_U \tag{3.89d}$$

$$\hat{u}_L \leq \hat{u}_l \leq \hat{u}_U \tag{3.89e}$$

$$p_L \leq p \leq p_U \tag{3.89f}$$

$$l = 1, \ldots, NL; i = 1, \ldots, NC \tag{3.89g}$$

At iteration $k$ in the optimization layer, the Newton-Raphson method is used to solve Eq. (3.89b) for $\hat{x}$, i.e., there exists an implicit relation:

$$\hat{x}_l = \hat{x}(\hat{u}_l, p, z_{l,0}) \quad \text{or} \quad \hat{x}_l = \hat{x}(\tilde{u}_l, z_{l,0}) \quad \text{with} \quad \tilde{u} = (\hat{u}, p)^T \tag{3.90}$$

The problem (3.80) is then reduced to the following form:

$$\min_{\tilde{u}_l, z_{l,0}} \quad f(\hat{x}(\tilde{u}_l, z_{l,0}), \tilde{u}_l) \tag{3.91a}$$

$$\text{s.t.} \quad z_{l+1,0} = \hat{z}_{l,NC} \tag{3.91b}$$

$$0 \leq \hat{H}(\hat{x}_{l,NC}, \tilde{u}, z_{l,0}) \tag{3.91c}$$

$$\tilde{u}_L \leq \tilde{u}_l \leq \tilde{u}_U \tag{3.91d}$$

$$l = 1, \ldots, NL \tag{3.91e}$$

This replacement makes it easier to get model state values and the sensitivities. However, in this method the system constraints (2.8e) are only imposed at the grid points ($i = NC$) in (3.91c), and thus there can be violations inside the time intervals between the grid points.

**Algorithm 3.8.1:** Algorithm of combined multiple shooting and collocation strategy

Given a number of elements, discretize the control variables, and state variables with the collocation method, provide initial values of control variables and bounds for all variables, set $k = 0$;

**repeat**`/* Outer iteration */`

1. **for** $l = 1$ *to* $NL$ **do**
   1. Solve nonlinear model Eqs. (3.89b) using the (variation of) Newton-Raphson method;
   2. Evaluate the sensitivities $\dfrac{d\hat{x}}{d\tilde{u}^k}$, $\dfrac{d\hat{x}}{d\tilde{z}_0^k}$;

**end**

; `/* Simulation layer */`

2. Evaluate the values of the object function and constraints: $f(\hat{x}(\tilde{u}^k, z_0^k), \tilde{u}^k), \hat{H}(\hat{x}_{NC}(\tilde{u}^k, z_0^k), \tilde{u}^k, z_0^k)$, as well as their gradients with respect to $\tilde{u}^k$ and $z_0^k$: $\nabla_{\tilde{u}} f(\hat{x}(\tilde{u}^k, z_0^k), \tilde{u}^k), \nabla_{z_0^k} f(\hat{x}(\tilde{u}^k, z_0^k), \tilde{u}^k), \nabla_{\tilde{u}} \hat{H}(\hat{x}_{NC}(\tilde{u}^k, z_0^k), \tilde{u}^k, z_0^k)$, $\nabla_{z_0^k} \hat{H}(\hat{x}_{NC}(\tilde{u}^k, z_0^k), \tilde{u}^k, z_0^k)$ as following:

$$\nabla_{\tilde{u}} f = \frac{\partial f}{\partial x^k}\frac{\partial \hat{x}^k}{\partial \tilde{u}^k} + \frac{\partial f}{\partial \tilde{u}^k}$$

$$\nabla_{z_0^k} f = \frac{\partial f}{\partial x^k}\frac{\partial \hat{x}^k}{\partial z_0^{k}} + \frac{\partial f}{\partial z_0^{k}}$$

$$\nabla_{\hat{u}} \hat{H} = \frac{\partial \hat{H}}{\partial \hat{x}_{NC}^k}\frac{\partial \hat{x}_{NC}^k}{\partial \tilde{u}^k} + \frac{\partial \hat{H}}{\partial \tilde{u}^k}$$

$$\nabla_{z_0^k} \hat{H} = \frac{\partial \hat{H}}{\partial \hat{x}_{NC}^k}\frac{\partial \hat{x}_{NC}^k}{\partial z_0^k} + \frac{\partial \hat{H}}{\partial z_0^k}$$

3. Call the active-set SQP solver to solve the NLP to get the search direction $d\tilde{u}_k$;
4. Set $\tilde{u}_{k+1} \leftarrow \tilde{u}_k + d\tilde{u}_k; k \leftarrow k + 1$;

**until** *convergence criteria are satisfied*;

# Chapter 4

# Improved Approaches to Dynamic Optimization

In the previous Chapter, direct methods are introduced to transform the infinite DOPs into finite NLPs. Each discretization strategy leads to a different type of resulting NLP problem with particular characteristics. Then each of these NLP problems need to be solved by a suitable NLP solver. In [Hong et al., 2006] the active-set SQP solver was applied to solve the NLP problem in the quasi-sequential approach. In the quasi-sequential approach, at each collocation point, state variables are treated as implicit-functional inequality constraints. This fact makes it difficult for a active-set SQP solver to solve the resulting NLP problem when there are a large number of active constraints. To over come this drawback, in this Chapter, an interior point based method are developed in Section 4.1 to solve the NLP problem resulted in the quasi-sequential approach in a full space form. Then in Section 4.2, a reduced-space interior point method are proposed to further improve the computational results.

Each discretization strategy applied to the DOP has its pros and cons and each is suitable for a particular class of DOPs. In [Tamimi and Li, 2010] the CMSC method was developed in a manner that the constraints of state variables were only imposed at grid points of each collocation interval. In Section 4.4 of this Chapter an improvement is made to the CMSC method to impose the state constraints at all collocation points in order to prevent possible constraint violations.

Section 4.3 illustrates a new strategy to deal with the PE problem with multiple data

sets in the error-in-variables (EIV) formulation in which the inputs of PE problem are assumed to be unknown and need to be estimated based on measurement data. Parallel strategies are developed and implemented in Section 4.3 to speed up the computation for PE with multiple data sets as well as in improved CMSC strategy.

## 4.1 Interior Point Quasi-sequential approach

In this Section, the quasi-sequential approach in Section 3.6 is extended by using the interior-point method, see Section 3.3.4, to handle the NLP problem inside the solution framework. This is due to the fact that the IP method can efficiently handle a large number of inequality constraints. In this new approach, termed as interior-point quasi-sequential approach, both the state and control variables are first discretized completely as in the simultaneous approach with collocation on finite elements, so as to overcome the difficulty in the sequential approach with satisfying state path constraints. Second, model equations and state variables are eliminated in the same manner as in the sequential approach by a simulation step, so that the optimization problem is reduced to a small NLP only with inequality constraints and control variables. Finally, the primal-dual interior-point method is employed to handle the inequality constraints. In this way the final optimization problem to be solved becomes a NLP with equality constraints by adding slack variables to the inequalities. Mathematical derivations and computation schemes for the interior-point quasi-sequential approach will be developed.

Taking a two-dimensional equality constrained optimization problem as an example the resulting approach is compared with the simultaneous and the quasi-sequential approach in terms of the path solution with a graphical interpretation. A highly nonlinear reactor optimal control problem is also considered to demonstrate the effectiveness of this approach. It can be concluded from the results that the interior-point quasi-sequential approach is more efficient than the quasi-sequential approach for solving highly nonlinear large-scale dynamic optimization problems with multiple inequality path constraints. Therefore it provides a promising solution to parameter estimation of large-scale nonlinear dynamic models and real-time dynamic optimization of engineering processes.

### 4.1.1 The nonlinear programing problem formulation

In a framework of interior-point methods to solve dynamic optimization problems, the inequality constrained functions will be reformulated through the addition of slack variables with lower bounds of zero. These slack variables are included to the state variable vector. Correspondingly, the reformulated inequality constraints which are now equalities are added to the system of DAEs. The quasi-sequential approach [Hong et al., 2006] possesses advantages of both the simultaneous and the sequential approach. It is based on a completely discretization of the state and control variables with collocation on finite elements like the simultaneous approach, and also it eliminates the state variables and the DAEs in the manner as the sequential approach. Only control variables and inequalities will be handled in the NLP solver. By using the collocation method in Section 3.2.2, the differential equations of the DAE system are converted to a set of algebraic equations. The DAE model is solved successively from element to element at each NLP iterate, and the sensitivities of the state variables with respect to control variables are computed in parallel with the DAE solution. The interior-point quasi-sequential approach is derived in more detail as follows. After discretization with the collocation method, problem (2.8) can be represented as the following NLP problem:

$$\begin{aligned} \min_{\hat{z}\in R^m, \hat{u}\in R^{n-m}} & f(\hat{z},\hat{u}) \\ \text{s.t.} \quad & c(\hat{z},\hat{u}) = 0 \\ & z_L \le \hat{z} \le z_U \\ & u_L \le \hat{u} \le u_U \end{aligned} \tag{4.1}$$

After the solution of the model equations in the quasi-sequential framework, the problem is reduced to the form:

$$\min_{\hat{u}\in R^{n-m}} f(\hat{z}(\hat{u}),\hat{u}) \tag{4.2a}$$

$$\text{s.t.} \quad z_L \le \hat{z}(\hat{u}) \le z_U \tag{4.2b}$$

$$u_L \le \hat{u} \le u_U \tag{4.2c}$$

This problem consists only of control variables as well as inequality constraints which are enforced at each collocation point. Without loss of generality, the problem formu-

lated in Eq. (4.2) can be rewritten as following:

$$\begin{aligned} \min_{\hat{u}\in R^{n-m}} & f(\hat{z}(\hat{u}),\hat{u}) \\ \text{s.t.} \quad & 0 \le \hat{z}(\hat{u}) \\ & 0 \le \hat{u} \end{aligned} \tag{4.3}$$

Now the inequalities are replaced by a logarithmic barrier term that is added to the objective function to yield:

$$\min_{\hat{u}\in R^{n-m}} \phi_\mu(\hat{u}) = f(\hat{z}(\hat{u}),\hat{u}) - \mu\left(\sum_{i=1}^{m}\ln(\hat{z}(\hat{u})^{(i)}) - \sum_{i=1}^{n-m}\ln(\hat{u}^{(i)})\right) \tag{4.4}$$

with the barrier parameter $\mu > 0$. Now the problem 4.3 is converted to an unconstrained NLP problem. For a given $\mu$, multiplier estimates, $\nu_1$ and $\nu_2$, are introduced and the optimality conditions for NLP (4.4) can be written in the primal-dual form:

$$\begin{aligned} \nabla f - (\frac{d\hat{z}}{d\hat{u}})^T\nu_1 - \nu_2 = 0 \\ V_1 Ze - \mu e = 0 \\ V_2 Ue - \mu e = 0 \end{aligned} \tag{4.5}$$

where $e = [1,\ldots,1]^T$, $\nu_1 = \mu Ze$ and $\nu_2 = \mu Ue$, $Z$ and $U$ are diagonal matrices with $\hat{z}$ and $\hat{u}$ on their diagonals, respectively. The Newton method can be used to solve the system of nonlinear Eqs. (4.5). Then the search direction $(d_k^{\hat{u}}, d_k^{\nu_1}, d_k^{\nu_2})$ at iterate $k$ $(\hat{u}_k, v_{1,k}, v_{2,k})$ can be obtained as a solution of the linearization of (4.5), that is:

$$\begin{bmatrix} W_k & (\frac{d\hat{z}}{d\hat{u}})_k^T & -I \\ V_{1k}(\frac{d\hat{z}}{d\hat{u}})_k & Z_k e & 0 \\ V_{2k}e & 0 & U_k e \end{bmatrix} \begin{pmatrix} d_k^{\hat{u}} \\ d_k^{v_1} \\ d_k^{v_2} \end{pmatrix} = - \begin{pmatrix} \nabla f_k - (\frac{d\hat{z}}{d\hat{u}})_k^T v_{1k} - v_{2k} \\ V_{1k}Z_k e - \mu_k e \\ V_{2k}U_k e - \mu_k e \end{pmatrix} \tag{4.6}$$

where $W_k = \nabla^2_{\hat{u}\hat{u}} L(\hat{u}_k, \nu_{1,k}, \nu_{2,k})$ . Eliminating the $d_k^{\nu_1}$ and $d_k^{\nu_2}$ in Eq. 4.6 yields the search direction:

$$(W_k + \Sigma_k) d_k^{\hat{u}} = -\nabla_\mu \phi(\hat{u}, v_1, v_2) \tag{4.7}$$

where:

$$\sum_k = (\tfrac{d\hat{z}}{d\hat{u}})_k^T V_{1k} Z_k^{-1} (\tfrac{d\hat{z}}{d\hat{u}})_k + V_{2k} U_k^{-1}$$

$$d_k^{v_1} = -v_{1k} + Z_k^{-1}\mu_k e - Z_k^{-1}V_{1k}(\tfrac{d\hat{z}}{d\hat{u}})_k \; d_k^{\hat{u}}$$
$$d_k^{v_2} = -v_{2k} + U_k^{-1}\mu_k e - U_k^{-1}V_{2k} \; d_k^{\hat{u}}$$

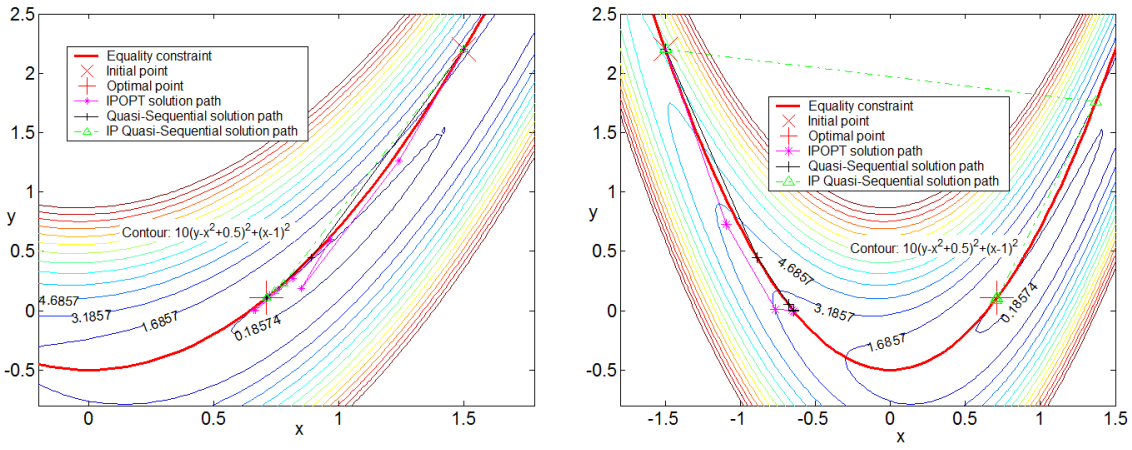
### 4.1.2 Case studies

#### 4.1.2.1 The Rosenbrock two-dimensional optimization problem

The solution path of the quasi-sequential, the interior-point quasi-sequential and the simultaneous approach is illustrated with the following example. The Rosenbrock two-dimensional constrained optimization problem is written as:

$$\begin{aligned} \min_{x,y} \quad & 10(y - x^2 + 0.5)^2 + (x-1)^2 \\ \text{s.t.} \quad & y = 1.2x^2 - 0.5 \\ & 0 \le y \le 5 \end{aligned} \tag{4.8}$$

In the sequential approach, the variable $x$ is regarded as the control variable. The NLP problem is solved by the simultaneous, the quasi-sequential approach, and interior-point quasi-sequential approach, respectively. The initial point is set at (1.5, 2.2) and (-1.5, 2.2). The optimal point should be at (0.7117, 0.1078) with the minimum objective function value 0.18574. The solution paths of the three approaches are presented in Fig. 4.1 where the interior-point quasi-sequential approach is called IP quasi-sequential for short. And the program package IPOPT [Wächter, 2002] is used for the simultaneous approach. As shown in Fig. 4.1a, when the initial point is set at (1.5, 2.2), all three approaches can converge to the optimal point. Moreover, the solution paths are almost the same. But if the initial point is changed to (-1.5, 2.2) as shown in Fig. 4.1b, only the IP quasi-sequential approach can converge to the optimal point. The IPOPT and quasi-sequential approaches fail, and terminate at the bound of $y = 0$. The IP quasi-sequential approach gives a different solution path and a better convergence performance.




(a) the initial point from (1.5, 2.2) (b) the initial point from (-1.5, 2.2)

Fig. 4.1 Solution path with respect to different initial point

#### 4.1.2.2 Optimal control of a CSTR

A continuous stirred tank reactor (CSTR), in which an exothermic, irreversible, first order reaction A $\rightarrow$ B occurs in the liquid phase and the temperature is regulated with external cooling, is considered for optimal control. This example is taken from [Henson and Seborg, 1997; Pannocchia and Rawlings, 2003] with the consideration that the liquid level is also a state variable. The mass and energy balances lead to the following highly nonlinear model equations:

$$\begin{aligned}
\frac{dh}{dt} &= \frac{F_0 - F}{\pi r^2} \\
\frac{dc}{dt} &= \frac{F_0(c_0 - c)}{\pi r^2 h} - k_0 c \exp(-\frac{E}{RT}) \\
\frac{dT}{dt} &= \frac{F_0(T_0 - T)}{\pi r^2 h} + \frac{-\Delta H}{\rho C_p} k_0 c \exp(-\frac{E}{RT}) + \frac{2U}{r\rho C_p}(T_c - T)
\end{aligned} \tag{4.9}$$

The state (controlled) variables are the level of the tank, $h$, and the mole fraction $c$. The third state variable is the reactor temperature, $T$. The control (manipulated) variables are the outlet flow rate $F$ and the coolant liquid temperature $T_c$. Moreover, it is considered that the inlet flow acts as a disturbance.

An operation situation is considered in that, at time $t = 10min$ a disturbance enters the plant, which is an increment of 10% on the inlet flow $F_0$. Here the objective is defined as to minimize the offset caused by the disturbance by controlling the outlet

flow rate $F$ and the coolant liquid temperature $T_c$ over a time horizon of $t_f = 50min$. Thus the optimal control problem is formulated as follows:

$$\min \quad \int_0^{t_f} \left[(h-h^s)^2 + 100(c-c^s)^2 + 0.1(F-F^s)^2 + 0.1(T_c - T_c^s)^2\right] dt \tag{4.10a}$$

$$\text{s.t.} \quad \text{DAEs model} \quad (4.9) \tag{4.10b}$$

$$85 \leq F \leq 115\,(l/\text{min}) \tag{4.10c}$$

$$299 \leq T_c \leq 301\,(K) \tag{4.10d}$$

In order to make it easier to treat the integral (4.10a), a transformation is applied here, let:

$$\frac{dx^{temp}}{dt} = \left[(h-h^s)^2 + 100(c-c^s)^2 + 0.1(F-F^s)^2 + 0.1(T_c - T_c^s)^2\right] \tag{4.11}$$

with $x^{temp}(0) = 0$. Then the DOP (4.10) can be rewritten as:

$$\min \quad x^{temp}(t_f) \tag{4.12a}$$

$$\text{s.t.} \quad \frac{dh}{dt} = \frac{F_0 - F}{\pi r^2} \tag{4.12b}$$

$$\frac{dc}{dt} = \frac{F_0(c_0 - c)}{\pi r^2 h} - k_0 c \exp(-\frac{E}{RT}) \tag{4.12c}$$

$$\frac{dT}{dt} = \frac{F_0(T_0 - T)}{\pi r^2 h} + \frac{-\Delta H}{\rho C_p} k_0 c \exp(-\frac{E}{RT}) + \frac{2U}{r\rho C_p}(T_c - T) \tag{4.12d}$$

$$\frac{dx_{temp}}{dt} = \left[(h-h^s)^2 + 100(c-c^s)^2 + 0.1(F-F^s)^2 + 0.1(T_c - T_c^s)^2\right] \tag{4.12e}$$

$$85 \leq F \leq 115\,(l/\text{min}) \tag{4.12f}$$

$$299 \leq T_c \leq 301\,(K) \tag{4.12g}$$

$$[h^s, c^s, F^s, T_c^s]^T = [0.659, 0.877, 100.0, 300.0]^T$$

$$[h_0, c_0, T_0, x_0^{temp}]^T = [0.659, 0.877, 324.5, 0.0]^T$$

where $h^s, c^s, F^s, T_c^s$ are the values at the desired steady-state operating point. This

leads to a typical constrained dynamic optimization problem. For the numerical solution, the time horizon $t_f$ is divided into 50 subintervals, and the problem is discretized with the 3-point-collocation in each time interval. The two control variables are represented as piecewise constant. Consequently, there are $2 \times 50 = 100$ control variables and $3 \times 3 \times 50 = 450$ state variables after the discretization. In implementing the three approaches for comparison, the bounds are imposed only on control constraints. All three approaches, the simultaneous, quasi-sequential and IP quasi-sequential, are coded in FORTRAN (the simultaneous approach directly using the IPOPT Fortran package). At the optimal solution, the three approaches provide the same profiles for control variables as shown in Fig. 4.2. The iterations and CPU time are given in Table 4.1. The results show that the IP quasi-sequential approach requires less iteration numbers, but much computational cost compared with the quasi-sequential approach. It should be pointed out, however, that the comparison is affected by a number of implementation details. In the quasi-sequential approach, the SQP subroutine DNCONG in the IMSL Library is used as the NLP solver. The IP quasi-sequential approach is coded ourself, where a lot of implementation parameters should be refined. Moreover, the IPOPT used here is an old version; it is being developed or renewed constantly.

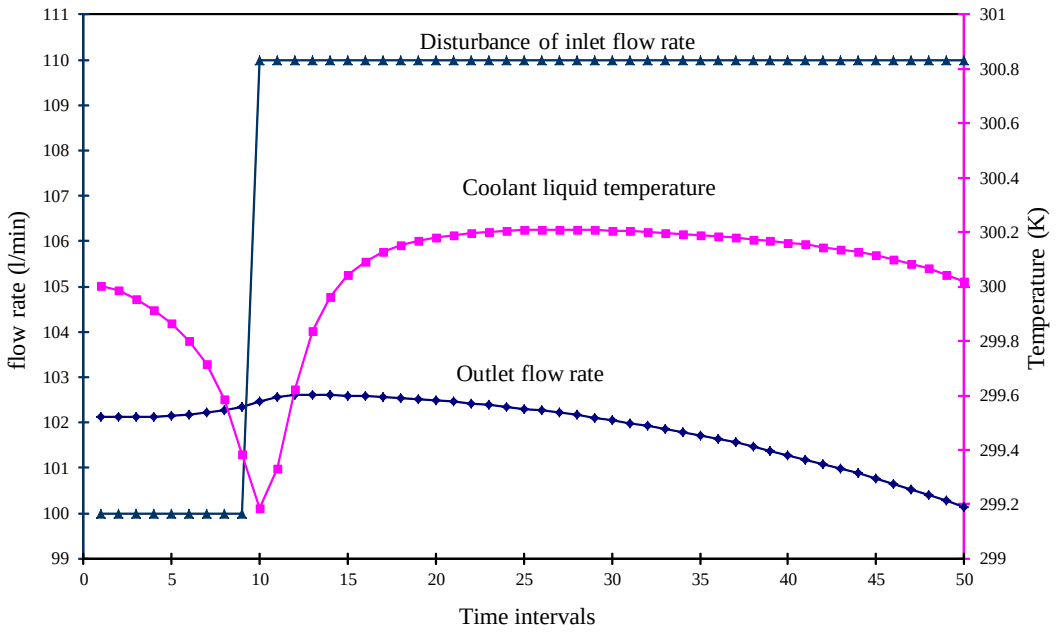



Fig. 4.2 Optimal trajectories of control variables

### 4.1.3 Conclusions

The active-set SQP quasi-sequential approach would be not very effective for problems with many active constraints, like the most active-set methods for inequality constrained optimization. Here an improved quasi-sequential approach, the IP quasi-sequential approach, is presented and analyzed to utilize the ability of IP methods for efficient handling a large number of inequalities. The results of the taken examples show that the new approach has advantages of the solution path and a higher convergence rate compared with the active-set SQP quasi-sequential approach. But its implementation parameters, such as the update of barrier parameters, should be improved. Finally, the IP quasi-sequential approach to handle very large optimization problems with more inequality constraints needs to be studied in the further research.

Table 4.1 Comparison of different approaches to the CSTR problem

| Disturbance F0 (l/min) | IP Quasi-Sequential (Iter/CPU(s)) | Active-set SQP Quasi-Sequential (Iter/CPU(s)) | IPOPT (Iter/CPU(s)) |
|---|---|---|---|
| 100-110 | 10/0.7031 | 17/0.5513 | 44/1.4219 |

## 4.2 Reduced-Space Interior Point Quasi-sequential approach

In this Section, a reduced-space IP approach is proposed to DOPs with general inequality constraints. It is an extension of the IP quasi-sequential approach to dynamic optimization of large-scale systems. Inequality constraints are formed by adding slack variables to an equality constrained barrier IP problem which is solved by a range space step and a null space step in every iteration. Mathematical derivations and computation schemes are presented. A highly nonlinear parameter estimation problem is taken as an example to demonstrate the effectiveness of this approach. The result is compared with the full space approach in terms of overall CPU time and the number of iterations.

### 4.2.1 NLP Problem formulation

Back to Section (4.1), Eqs. (4.2) contains only controls and inequality constraints which are enforced at each collocation point. Adding slack variables $s$ to inequalities (4.2b), one gets:

$$\begin{array}{lll} \min\limits_{\hat{u}\in R^{n-m}} & f\left(\hat{z}(\hat{u}),\hat{u}\right) & f: R^n \to R \\ \text{s.t.} & \hat{z}(\hat{u}) - s = 0 & \\ & z_L \leq s \leq z_U & s \in R^m \\ & u_L \leq \hat{u} \leq u_U & \hat{u} \in R^{n-m} \end{array} \tag{4.13}$$

Without loss of generality, problem (4.13) can be rewritten as:

$$\begin{array}{lll} \min\limits_{x\in R^{n-m}} & f(x) & f: R^n \to R \\ \text{s.t.} & h(x) = 0 & h: R^n \to R^m \\ & 0 \leq x & s \in R^m \end{array} \tag{4.14}$$

where

$$x = (s \quad \hat{u})^T, \qquad h(x) = \hat{z}(\hat{u}) - s \tag{4.15}$$

### 4.2.2 Interior-point approach

The inequalities now can be replaced by a logarithmic barrier term added to the objective function to yield:

$$\begin{array}{ll} \min\limits_{x\in R^n} & \varphi_\mu(x) = f(x) - \mu \sum\limits_{i=1}^{n} \ln\left(x^i\right) \\ \text{s.t.} & h(x) = 0 \end{array} \tag{4.16}$$

where the barrier parameter $\mu \geq 0$. The Lagrangian of problem (4.16) will be:

$$\mathcal{L}(x,\lambda) = f(x) - \mu \sum_{i=1}^{n} \ln\left(x^i\right) + h(x)^T \lambda \tag{4.17}$$

For a given $\mu$, multiplier estimates, $\lambda$, are introduced and the optimality conditions for (4.16) can be written in the primal-dual form:

$$\begin{aligned} \nabla_x \mathcal{L} = \nabla_x f(x) + \nabla h(x)\lambda - \nu = 0 \\ XVe - \mu e = 0 \\ h(x) = 0 \end{aligned} \tag{4.18}$$

where $V = diag(v),\ X = diag(x),\ e = (1, 1, \ldots, 1)^T$. The search direction $\left(d_k^x, d_k^\lambda, d_k^\nu\right)$ at iterate $k\,(x_k, \lambda_k, v_k)$ can be obtained as a solution of the linearization of (4.18), that is:

$$\begin{aligned} W_k d_k^x + \nabla h\,(x_k)\, d_k^\lambda - d_k^v = -\nabla \mathcal{L}\,(x_k) \\ \nabla h\,(x_k)\, d_k^x = -h\,(x_k) \\ V_k d_k^x + X_k d_k^v = -\left(X_k V e - \mu e\right) \end{aligned} \tag{4.19}$$

Eq. (4.19) is rewritten into matrix form:

$$\begin{bmatrix} W_k & A_k^T & -I \\ A_k & 0 & 0 \\ V_k & 0 & X_k \end{bmatrix} \begin{pmatrix} d_k^x \\ d_k^\lambda \\ d_k^v \end{pmatrix} = - \begin{pmatrix} \nabla f\,(x_k) + A_k - v_k \\ h\,(x_k) \\ X_k V_k e - \mu e \end{pmatrix} \tag{4.20}$$

where $W_k = \nabla^2_{xx}\mathcal{L}(x_k, \lambda_k, \nu_k)$ and $A_k = \nabla h(x_k)^T$. A solution to (4.20) can be obtained by first solving:

$$\begin{bmatrix} H_k & A_k \\ A_k^T & 0 \end{bmatrix} \begin{pmatrix} d_k^x \\ \lambda_k^+ \end{pmatrix} = - \begin{pmatrix} \nabla \varphi_\mu\,(x_k) \\ h\,(x_k) \end{pmatrix} \tag{4.21}$$

where $H_k = W_k + X_k^{-1} V_k$ and then computing $d_k^\lambda$ and $d_k^v$ as follows:

$$\begin{aligned} d_k^\lambda &= \lambda_k^+ - \lambda_k \\ d_k^v &= \mu X_k^{-1} e - v_k - \Sigma_k d_k^x \end{aligned} \tag{4.22}$$

### 4.2.3 Reduced-space interior-point approach formulation

The search direction $d_k^x$ obtained from the linear system (4.21) comes from the solution of the following QP problem:

$$\begin{aligned} \min_{d \in R^n} \quad & \nabla \varphi_\mu(x_k)^T d + \frac{1}{2} d^T H_k d \\ \text{s.t.} \quad & A_k^T d + h\left(x_k\right) = 0 \end{aligned} \tag{4.23}$$

The matrix $H_k$ is assumed to be positive definite in the null space of $A_k^T$. Here the coordinate decomposition applied for the IP method can be used to partition the primal variables $x = \begin{pmatrix} x^I & x^D \end{pmatrix}^T = \begin{pmatrix} s & \hat{u} \end{pmatrix}^T$ into dependent variables $x^D \in R^m$ that occupies the whole slack variables space $s$ and independent variables $x^I \in R^{n-m}$ that occupies the whole control variables space $\hat{u}$ (see (4.15) above). As the consequence of this decomposition, the constraint Jacobian is formed as:

$$A^T = \left[ C\left(x^I\right) \;\; N\left(x^D\right) \right] = \left[ C\left(s\right) \;\; N\left(\hat{u}\right) \right] \tag{4.24}$$

Obviously, one gets $C\left(s\right) = -I_m$. Then the basis matrix can be defined as:

$$R_k = \begin{bmatrix} I_m \\ 0 \end{bmatrix} \quad \text{and} \quad Q_k = \begin{bmatrix} -C_k^{-1} N_k \\ I_{n-m} \end{bmatrix} = \begin{bmatrix} N_k \\ I_{n-m} \end{bmatrix} \tag{4.25}$$

where $C_k = C\left(x_k^D\right) = C\left(s_k\right) = -I_m$ and $N_k = N\left(x_k^I\right) = N\left(\hat{u}_k\right)$. The primal search direction $d_k^x$ then can be written as:

$$d_k^x = R_k d_R + Q_k d_Q \tag{4.26}$$

The range-space direction can be computed as:

$$d_R = -\left[ A_k^T R_k \right]^{-1} h_k = -h_k \tag{4.27}$$

The null-space direction $d_Q$ is the solution of the following reduced QP sub-problem:

$$\min_{d_Q \in R^{n-m}} \left( Q_k^T \nabla \varphi_\mu\left(x_k\right) + \zeta_k \omega_k \right)^T d_Q + \frac{1}{2} d_Q^T Q_k^T H_k Q_k d_Q \tag{4.28}$$

It can be obtained by directly solving the dense symmetric linear system:

$$d_Q = -\tilde{B}_k^{-1}\left(Q_k^T \nabla \varphi_\mu\left(x_k\right) + \zeta_k \omega_k\right) \tag{4.29}$$

where $\tilde{B}_k$ is the overall reduced Hessian; $\zeta_k$ is damping parameter $\zeta_k \in (0, 1]$ and $\omega_k$ is the cross term that can be approximated as $\omega_k = Q_k^T \Sigma_k d_R$, respectively.

### 4.2.4 Jacobian computation

Back to (4.24), the next task is to calculate the Jacobian part $N(\hat{u})$ of the constraint with respect to the independent variables $x^I = \hat{u} \in R^{n-m}$. In our method, the DAEs are solved successively from element to element at each NLP iteration, so that sensitivities of the states with respect to controls can be computed simultaneously. Here by considering the discretized DAE system in the $k^{th}$ iteration of NLP with the $l^{th}$ element, the model equations at the collocation points can be written as:

$$c_\ell\left(\hat{z}_{\ell,0}^k, \hat{u}_\ell^k, \hat{z}_\ell^k\right) = 0; \qquad \ell = 1, ..., NL \tag{4.30}$$

where $\hat{z}_{\ell,0}^k$ is the initial value of $\hat{z}_\ell^k$ of the element, $NL$ is the number of elements, respectively. According to the theorem of implicit functions, one obtains:

$$\frac{d\hat{z}_\ell}{d\hat{u}_\ell} = -\nabla_{\hat{z}_\ell} c_\ell^{-T} \nabla_{\hat{u}_\ell} c_\ell \tag{4.31a}$$

$$\frac{d\hat{z}_\ell}{d\hat{z}_{\ell,0}} = -\nabla_{\hat{z}_\ell} c_\ell^{-T} \nabla_{\hat{z}_{\ell,0}} c_\ell \tag{4.31b}$$

For the state profile continuity, the last collocation point of element $\ell$ is taken as the initial point of the next element $\ell + 1$, so that the sensitivities can be transferred by

the chain rule. The Jacobian of part $N(\hat{u})$ has the following structure:

$$N(\hat{u}) = \begin{bmatrix} a_1 & & & & & \\ a_{2,1} & a_2 & & & 0 & \\ \vdots & \ddots & \ddots & & & \\ a_{2,\ell} & \ddots & \ddots & a_\ell & & \\ \vdots & \ddots & a_{i,j} & \ddots & \ddots & \\ a_{NL,1} & a_{NL,2} & \cdots & a_{NL,\ell} & \cdots & a_{NL} \end{bmatrix} \begin{matrix} \hat{z}_1, \text{interval 1} \\ \hat{z}_2, \text{interval 2} \\ \\ \hat{z}_\ell, \text{interval } \ell \\ \\ \hat{z}_{NL}, \text{interval NL} \end{matrix} \tag{4.32}$$

$$\begin{matrix} \hat{u}_1 & & \hat{u}_2 & & \hat{u}_\ell & & \hat{u}_{NL} \\ \text{interval 1} & & & & \text{interval } \ell & & \text{interval NL} \end{matrix}$$

where the intervals mean the time elements of discretization of DAEs and the elements of (4.32) can be calculated as:

$$a_\ell = \left(\frac{d\hat{z}_\ell^k}{d\hat{u}_\ell^k}\right)^T ; \ell = 1, ..., NL \tag{4.33a}$$

$$a_{i,j} = \left(\frac{d\hat{z}_i^k}{d\hat{u}_j^k}\right)^T = \left(\frac{d\hat{z}_i^k}{d\hat{z}_{0,i}^k}\right)^T \left(\frac{d\hat{z}_{i-1}^k}{d\hat{u}_{0,i-1}^k}\right)^T \cdots \left(\frac{d\hat{z}_j^k}{d\hat{u}_j^k}\right)^T \tag{4.33b}$$

$$i, j = 1, \ldots, NL; \qquad i > j$$

## 4.3 Parameter Estimation Problems framework with Multiple Datasets

Mathematical models are increasingly used in process control and optimization. Based on chemical, physical and thermodynamic principles, mathematical models are usually described with a large number of nonlinear DAEs in which model parameters must be identified based on measured data-sets. The quality of the parameter identification plays an essential role in the on-line model utilization, such as model predictive control (MPC). Thus, multiple data-sets from a series of dynamic curves are needed to improve the accuracy of the parameter identification. For dynamic systems there may be time-dependent parameters to be estimated from available plant data. Therefore, it is desirable to develop efficient estimation strategies and numerical algorithms which

should be able to solve such challenging estimation problems, including multiple data profiles and large parameter sets.

The approach proposed in this work is an extension of the work by Faber et al. [2003] to address parameter estimation problems for dynamic systems. A novel three-stage computation framework, whose core is based on a quasi-sequential interior-point method, takes advantages of space decomposition in the interior-point method and parallel computing in order to reduce the computation time.

First, the dynamic parameter estimation problem is transformed into a large NLP problem using collocation on finite elements. Then the model parameters to be estimated are treated in the upper stage by solving a NLP problem with only boundary on variables. The middle stage consists of multiple parallel nested NLPs which represents the data reconciliation step for each data set. By dividing the discretized variables space into a dependent (state) and an independent (control) space, only independent variables are treated by the IP solver. In the lower stage (or simulation layer), the model equations are solved by using the Newton method to compute the dependent variables. The sensitivities of the dependent variables with respect to the independent variables as well as estimated parameters of each time interval, which are required from the middle and upper stage, will be transmitted through the continuity relation from element to element according to the chain rule. Mathematical derivations and computation schemes are introduced together with an example of parameter estimation of a CSTR to demonstrate the effectiveness of the proposed approach.

### 4.3.1 Error-In-Variables formulation of parameter estimation problem

Over the past years, many efforts have been made to solve different parameter estimation problems, and a number of decomposition algorithms have been proposed to improve the performance of estimation problems with EIV formulation. By employing a two-level strategy for estimation and simulation, Doví and Paladino [1989] presented a constrained variation approach to decouple the parameter estimation problem where the dependent variables are eliminated by solving model equations through a simulation step. Kim et al. [1990] developed a slightly different approach using a two-stage NLP procedure to address data reconciliation and parameter estimation

step separately. Tjoa and Biegler [1991] proposed a similar approach based on SQP for parameter estimation problem with implicit models.

Due to the differential constraints usually present in the dynamic process model, nonlinear dynamic estimation problems with EIV formulation are more challenging. Albuquerque and Biegler [1995] proposed an estimation approach based on an efficient decomposition strategy to estimate the process states and parameters simultaneously. In [Arora and Biegler, 2004] a nonlinear trust-region SQP approach using a full discretization method was developed to the parameter estimation for a polymerization reactor. Zavala et al. [2008b] extended this simultaneous approach to solve the multiple data, large-scale, DAE constrained parameter estimation problems. Further development of this approach was presented in [Zavala and Biegler, 2006], in which the associated large-scale parameter estimation problem is solved using IP algorithm (IPOPT) and parallel computing strategy. Due to the nonlinear nature of the process models, the resulting parameter estimation optimization problem is non-convex and may contain multiple local optima. To obtain global optimum of the parameter estimation problems, a global optimization procedure based on the deterministic branch and bound global optimization algorithm ($\alpha$BB) was presented to solve the EIV formulation [Esposito and Floudas, 2000]. More recently, some heuristic optimization methods, such as genetic algorithm (GA) [Wongrat et al., 2005] and particle swarm optimization (PSO) [Prata et al., 2010], have been used to solve complex parameter estimation and data reconciliation problems.

Although there has been considerable interest in parameter estimation in the process industry, very few studies have been made on exploiting the structure of the EIV formulation and multiple data profiles. All the aforementioned approaches need large mathematical manipulations to obtain the second-order derivatives of the model equations, which will become extremely expensive for the solution of large-scale DAE constrained parameter estimation problems and, therefore, have not been easily implemented with standard NLP software. Making use of the optimality condition of the sub-NLP problem, Faber et al. [2003] proposed a three-stage framework for the estimation of nonlinear steady-state systems with multiple data-sets. In this study the method of [Faber et al., 2003] is extended to the PE of dynamic systems described by DAEs and derive a quasi-sequential algorithm for dynamic parameter estimation problems. It means that in this study a dynamic three-stage estimation framework is developed. Due to the decomposition of the optimization variables, the proposed

approach can also solve time-dependent parameter estimation problems with multiple data profiles by a standard NLP solver. Further, case studies are used to demonstrate the performance of this approach.

### 4.3.2 Three-layer Quasi-Sequential Approach

Using the collocation method, a polynomial approximation of state variables is applied to the differential Eq. (2.8b). The input variables and parameters are assumed to be piecewise constants in each element. The optimization problem (2.8) is then reformulated as the following large NLP problem:

$$\min_{\hat{p}_l,\hat{u}_{j,l},\hat{z}_{j,l,i},\hat{y}_{j,l,i}} F = \sum_{j=1}^{NS} f_j = \sum_{j=1}^{NS}\sum_{l=1}^{NL} \Big[ \left(\hat{y}_{j,l,NC} - y^M_{j,l}\right)^T V_y^{-1} \left(\hat{y}_{j,l,NC} - y^M_{j,l}\right) + \left(\hat{u}_{j,l} - u^M_{j,l}\right)^T V_u^{-1} \left(\hat{u}_{j,l} - u^M_{j,l}\right) \Big] \tag{4.34a}$$

s.t.

$$0 = \hat{G}_j(\hat{z}_{j,l,i}, \hat{y}_{j,l,i}, \hat{u}_{j,l}, \hat{p}_l), \tag{4.34b}$$

$$0 \leq \hat{H}_j(\hat{z}_{j,l,i}, \hat{y}_{j,l,i}, \hat{u}_{j,l}, \hat{p}_l), \tag{4.34c}$$

$$p_L \leq \hat{p}_l \leq p_U \tag{4.34d}$$

where $\hat{G}_j \equiv \begin{bmatrix} F_j & G_j \end{bmatrix}^T$ and $i = 1, \ldots, NC$; $NC$ is number of collocation point (in each time interval). The number of (discretized) variables of this NLP problem is $NS \times NL \times (NC \times (n_z + n_y) + n_u) + NL \times n_p$.

For the parameter estimation of nonlinear steady-state models, Faber et al. [2003] proposed a sequential approach to solving large scale parameter estimation problems with multiple data-sets, where a nested three-stage computation was presented to decompose the problem. This idea is extended in this thesis to develop a new decomposition approach for dynamic parameter estimation problems. Based on the collocation method and the quasi-sequential dynamic optimization approach, a novel three-stage

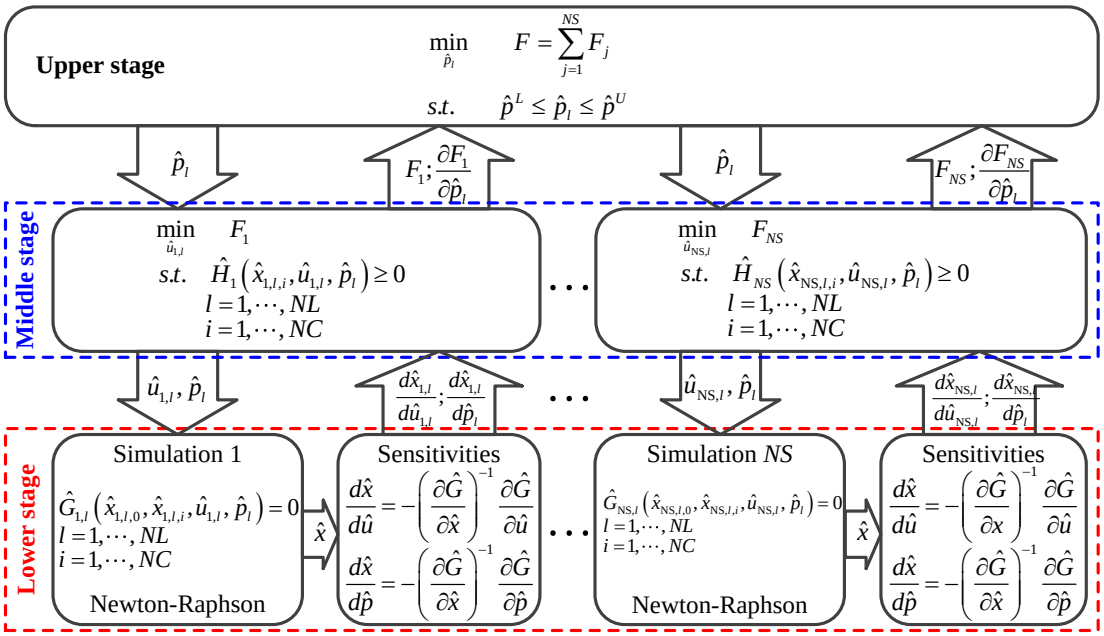



Fig. 4.3 Diagram of the three-stage computation framework

computation framework for parameter estimation problem to dynamic systems is derived in this Section. In particular, estimation problems with time-dependent parameters can be addressed with this method. The scheme of the proposed estimation framework is shown in Fig. 4.3.

In this work, it is assumed that the measurement points coincide with the collocation grids. For large-scale DAE systems, problem (4.34) may be too large for a standard NLP solver to deal with. Wächter and Biegler [2006] developed the IPOPT source code based on the IP method coupled with the simultaneous approach to NLP problems. Hong et al. [2006] proposed a quasi-sequential approach to large-scale dynamic optimization problems. In this study, in order to take advantages from both of them, the quasi-sequential approach is coupled with the IP method to solve the parameter identification problem (4.34).

#### 4.3.2.1 The upper stage

The upper stage as shown in Fig. 4.3 solves the parameter estimation problem in which the variables $u$ and $y$ are considered as functions of $p$:

$$\min_{\hat{p}_l} \quad F = \sum_{j=1}^{NS} f_j = \sum_{j=1}^{NS} \sum_{l=1}^{NL} \left[ \left(\hat{y}_{j,l} - y_{j,l}^m\right)^T V_y^{-1} \left(\hat{y}_{j,l} - y_{j,l}^m\right) + \left(\hat{u}_{j,l} - u_{j,l}^m\right)^T V_u^{-1} \left(\hat{u}_{j,l} - u_{j,l}^m\right) \right] \tag{4.35a}$$

s.t.

$$p_L \leq \hat{p}_l \leq p_U \tag{4.35b}$$

Since only $\hat{p}_l$ are treated as optimization variables with only bound constraints (4.35b), the size of this optimization problem is $NL \times n_p$. The objective function value (4.35a) and its gradient will be supplied from *middle stage*.

#### 4.3.2.2 The middle stage

The middle stage consists of multiple NLPs nested in the upper stage, representing a data reconciliation step for each data profile, which can be considered as a dynamic optimization problem. With given values of the parameters from the upper stage, the sub-NLP problem for each data profile has the following form:

$$\min_{\hat{u}_{j,l}, \hat{z}_{j,l,i}, \hat{y}_{j,l,i}} f_j = \sum_{l=1}^{NL} \left[ \left(\hat{y}_{j,l,NC} - y_{j,l}^m\right)^T V_y^{-1} \left(\hat{y}_{j,l,NC} - y_{j,l}^m\right) + \left(\hat{u}_{j,l} - u_{j,l}^m\right)^T V_u^{-1} \left(\hat{u}_{j,l} - u_{j,l}^m\right) \right] \tag{4.36a}$$

s. t.

$$0 = \hat{G}_j(\hat{z}_{j,l,i}, \hat{y}_{j,l,i}, \hat{u}_{j,l}, \hat{p}_l), \tag{4.36b}$$

$$0 \leq \hat{H}_j(\hat{z}_{j,l,i}, \hat{y}_{j,l,i}, \hat{u}_{j,l}, \hat{p}_l), \tag{4.36c}$$

The dimension of this problem is $(NL \times NC \times (n_z + n_y) + NL \times n_u)$. This number may be still too large for a standard NLP solver to deal with. To solve this large-scale NLP problem with available software, the efficient quasi-sequential optimization method proposed in Section 3.6 is used. The quasi-sequential approach uses a two-

layer optimization strategy for solving dynamic optimization problems, as shown in Fig. 4.5.

Let $\hat{x} = [\hat{z}, \hat{y}]^T$, in the quasi-sequential method, (4.36) can be rewritten as follows:

$$\min_{\hat{u}_{j,l}} f_j = \sum_{l=1}^{NL} \left[ \left(\hat{y}_{j,l,NC} - y_{j,l}^m\right)^T V_y^{-1} \left(\hat{y}_{j,l,NC} - y_{j,l}^m\right) + \left(\hat{u}_{j,l} - u_{j,l}^m\right)^T V_u^{-1} \left(\hat{u}_{j,l} - u_{j,l}^m\right) \right] \tag{4.37a}$$

s. t.

$$0 = \hat{G}_j(\hat{x}_{j,l,i}, \hat{u}_{j,l}, \hat{p}_l^*) \tag{4.37b}$$

$$0 \leq \hat{H}_j(\hat{x}_{j,l,i}, \hat{u}_{j,l}, \hat{p}_l^*) \tag{4.37c}$$

#### 4.3.2.3 The lower stage

In this stage, the following nonlinear equation system Eq. (4.37b) at the temporary point $\hat{p}_l^*$ will be solved:

$$0 = \hat{G}_{j,l}(\hat{z}_{j,l,0}, \hat{x}_{j,l,i}, \hat{u}_{j,l}, \hat{p}_l^*) \tag{4.38}$$

where $j = 1, \ldots, NS; l = 1, \ldots, NL; i = 1, \ldots, NC$; and $k$ denotes the $k^{th}$ iteration of NLP. The Newton steps are generated by:

$$\Delta\hat{x}_{kk} = -\left(\frac{\partial \hat{G}_j}{\partial \hat{x}}\right)_{\tilde{z}_{kk}}^{-1} \hat{G}_j\left(\hat{x}_{kk}, \hat{u}_k, \hat{p}^*\right) = -C^{-1}\left(\hat{x}_{kk}, \hat{u}_k, \hat{p}^*\right) \hat{G}_j\left(\hat{x}_{kk}, \hat{u}_k, \hat{p}^*\right) \tag{4.39a}$$

$$\hat{x}_{kk+1} = \hat{x}_{kk} + \Delta\hat{x}_{kk} \tag{4.39b}$$

where $kk$ denotes the Newton iterations.

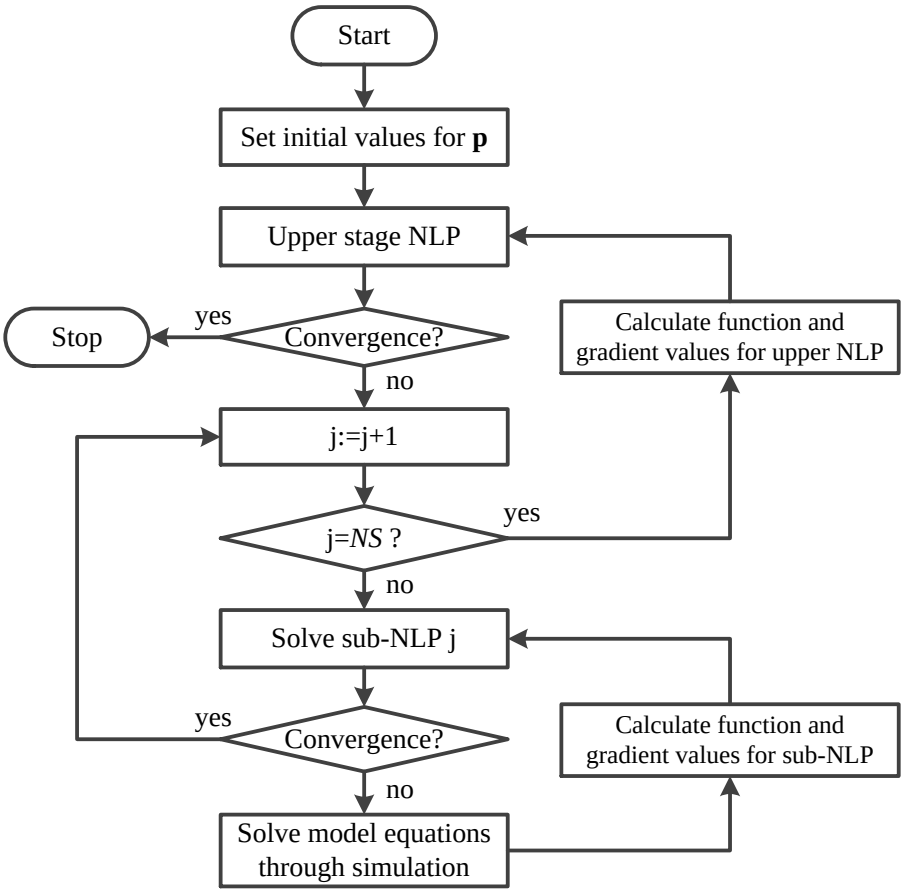



Fig. 4.4 Flowchart of the three-stage serial computation framework

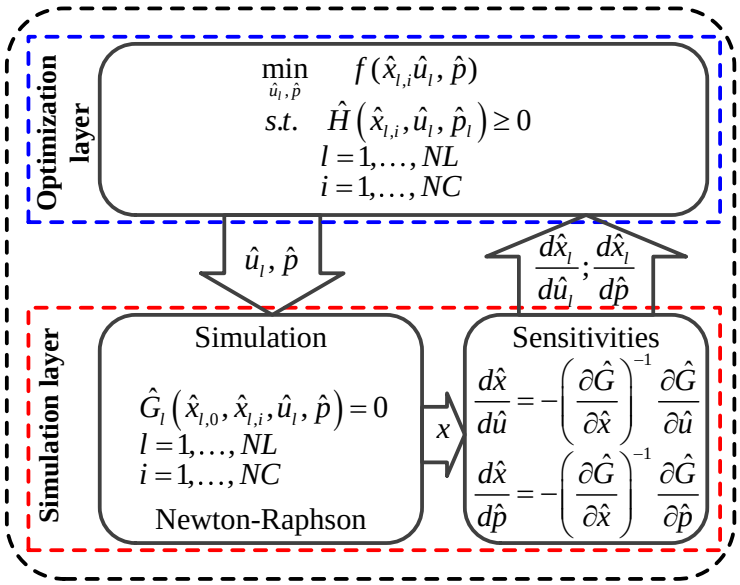



Fig. 4.5 Structure of two-layer optimization

#### 4.3.2.4 Calculation of the gradient and sensitivity matrix

The gradient of the objective function in the upper stage can be formulated as:

$$\frac{dF}{d\hat{p}} = \sum_{j=1}^{NS}\left[\frac{\partial f_j}{\partial \hat{u}_j}\frac{\partial \hat{u}_j}{\partial \hat{p}} + \frac{\partial f_j}{\partial \hat{y}_j}\left(\frac{\partial \hat{y}_j}{\partial \hat{u}_j}\frac{\partial \hat{u}_j}{\partial \hat{p}} + \frac{\partial \hat{y}_j}{\partial \hat{p}}\right)\right] = \sum_{j=1}^{NS}\left[\left(\frac{\partial f_j}{\partial \hat{u}_j} + \frac{\partial F_j}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial \hat{u}_j}\right)\frac{\partial \hat{u}_j}{\partial \hat{p}} + \frac{\partial f_j}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial \hat{p}}\right] \tag{4.40}$$

According to the optimality condition of the sub-NLP problem in the middle stage, one gets:

$$\Phi_j\left(\hat{u}_j^*, \hat{p}\right) = \frac{\partial f_j}{\partial \hat{u}_j} + \frac{\partial f_j}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial \hat{u}_j} = 0 \tag{4.41}$$

Then, the required gradient for the upper stage will be:

$$\frac{dF}{d\hat{p}} = \sum_{j=1}^{NS}\left[\left(\frac{\partial f_j}{\partial \hat{u}_j} + \frac{\partial f_j}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial \hat{u}_j}\right)\frac{\partial \hat{u}_j}{\partial \hat{p}} + \frac{\partial f_j}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial \hat{p}}\right] \Rightarrow \frac{dF}{d\hat{p}} = \sum_{j=1}^{NS}\left[\frac{\partial f_j}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial \hat{p}}\right] \tag{4.42}$$

Therefore, the values of $\dfrac{\partial \hat{u}_j}{\partial \hat{p}}$, which are related to second derivatives [Faber et al., 2003], are not required. In this way the computation expense will be significantly reduced. For the middle stage, the method IP quasi-sequential optimization in Section 3.6 is applied.

For the middle stage, considering the model equation system at the convergence of the sub-NLPs, the sensitivity matrix can be obtained by linearization of model equations Eq. (4.38) as:

$$\nabla_{\hat{z}_{l,0}}\hat{G}_{j,l}^T\Delta\hat{z}_{j,l,0} + \nabla_{\hat{x}_{j,l}}\hat{G}_{j,l}^T\Delta\hat{x}_{j,l} + \nabla_{\hat{u}_{j,l}}\hat{G}_{j,l}^T\Delta\hat{u}_{j,l} = 0 \tag{4.43}$$

For the sensitivity of the state variables with respect to the input variables as well as the estimated parameters, according to the theorem of implicit functions, one obtains:

$$\frac{d\hat{x}_{j,l}}{d\hat{u}_{j,l}} = -\left(\frac{\partial \hat{G}_{j,l}}{\partial \hat{x}_{j,l}}\right)^{-1}\frac{\partial \hat{G}_{j,l}}{\partial \hat{u}_{j,l}} \tag{4.44a}$$

$$\frac{d\hat{x}_{j,l}}{d\hat{z}_{j,l,0}} = -\left(\frac{\partial \hat{G}_{j,l}}{\partial \hat{x}_{j,l}}\right)^{-1}\frac{\partial \hat{G}_{j,l}}{\partial \hat{z}_{j,l,0}} \tag{4.44b}$$

**Algorithm 4.3.1:** Algorithm of three-layer IP quasi-sequential approach

Provide initial values $p_0$ and bounds of parameter variables, set $q = 0$;

**repeat**// `Outer iteration`

1. Input $p$ values into sub-NLP problems;
2. **for** $j = 1$ *to* $NS$ **do**
   3. Given a number of elements, discretize the control variables, and state variables with the collocation method, provide initial values of control variables and bounds for all variables, set $k = 0$ for the $j^{th}$ sub-NLP;
   4. **repeat**// `Inner iteration`
      5. Solve nonlinear model Eqs. (4.34b) using Newton-Raphson method in the **Simulation layer**;
      6. Compute the sensitivities $\frac{d\hat{x}_{j,l}}{d\hat{u}_{j,l}}$ and $\frac{d\hat{x}_{j,l}}{d\hat{z}_{j,l,0}}$ by Eq. (4.44);
      7. Compute the values of the object function and constraints: $f_j(\cdot), \hat{H}_j(\cdot)$, as well as their gradients with respect to $\hat{u}^k$: $\nabla_{\hat{u}} f_j(\cdot)$ and $\nabla_{\hat{u}} H_j(\cdot)$ as following:

$$\nabla_{\hat{u}} f = \frac{\partial f}{\partial \hat{y}^k}\frac{\partial \hat{y}^k}{\partial \hat{u}^k} + \frac{\partial f}{\partial \hat{u}^k}$$

      and $\nabla_{\hat{u}} \hat{H}(\hat{z}) = \frac{\partial \hat{H}}{\partial \hat{z}^k}\frac{\partial \hat{z}^k}{\partial \hat{u}^k} + \frac{\partial \hat{H}}{\partial \hat{u}^k}$, $\nabla_{\hat{u}} \hat{H}(\hat{y}) = \frac{\partial \hat{H}}{\partial \hat{y}^k}\frac{\partial \hat{y}^k}{\partial \hat{u}^k} + \frac{\partial \hat{H}}{\partial \hat{u}^k}$;
      8. Call an (IP) solver to get the search direction $d\hat{u}_k$;
      9. Set $\hat{u}_{k+1} \leftarrow \hat{u}_k + d\hat{u}_k$; $k \leftarrow k + 1$;
   **until** *convergence criteria the inner NLP are satisfied*;
   10. Compute the sensitivities $\frac{d\hat{x}_{j,l}}{dp_l} = -\left(\frac{\partial \hat{G}_{j,l}}{\partial \hat{x}_{j,l}}\right)^{-1}\frac{\partial \hat{G}_{j,l}}{\partial p_l^*}$;
**end**
11. Compute the objective function value $F = \sum_{j}^{NS} f_j$, and the gradient $\frac{dF}{d\hat{p}} = \sum_{j=1}^{NS}\left[\frac{\partial f_j}{\partial \hat{y}_j}\frac{\partial \hat{y}_j}{\partial \hat{p}}\right]$;
12. Call an (IP) solver to get the search direction $dp_q$;
13. Set $p_{q+1} \leftarrow p_q + dp_q$; $q \leftarrow q + 1$;

**until** *convergence criteria of the outer NLP are satisfied*;

### 4.3.3 Parallel computing

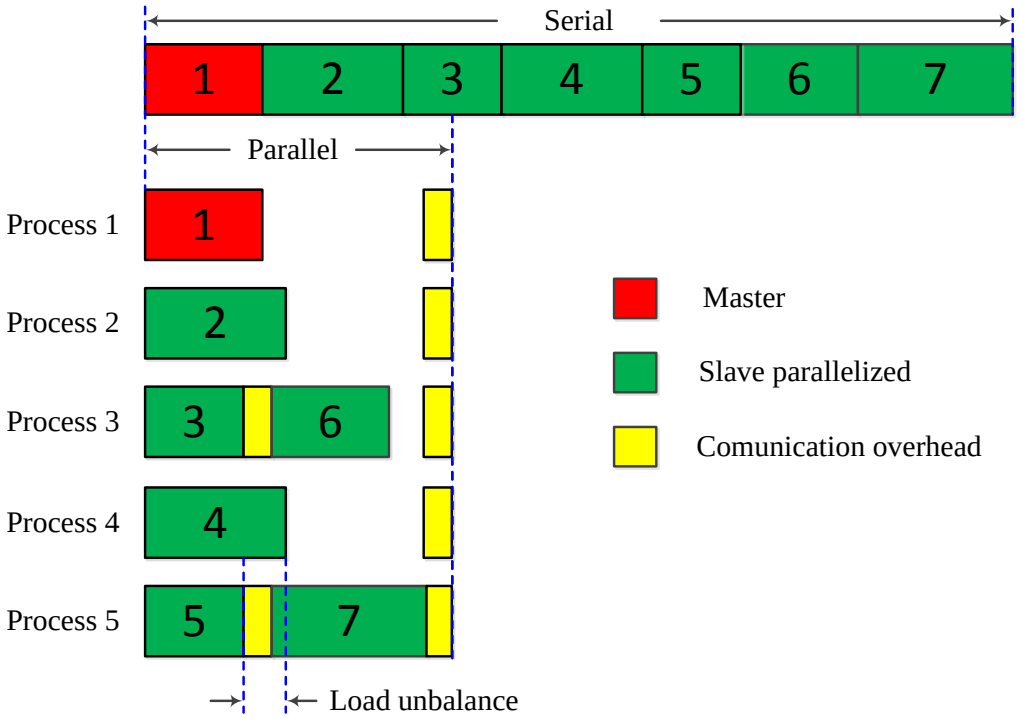



Fig. 4.6 Parallel timing diagram

Thanks to the significant development of multi-core architectures, high performance computing (HPC) currently enables parallel processing for the solution of large-scale and complex problems. Two structures are usually used for parallel programing: message passing interface (MPI) and Open Multi-Processing (OpenMP).

Open Multi-Processing (OpenMP) [Chapman et al., 2007] uses a set of compiler directives to influence the runtime behavior of threads. It can simply inserts compiler directives into a serial program to make it be a parallel one. These directives can be incrementally added, so that the existing serial program can be parallelized one portion after another. This means that there is no need to dramatically change the code. OpenMP can be treated as comments if sequential compilers are used, so that a unified code may be used for both serial and parallel applications. OpenMP is considered to be easier to program and debug in comparison to MPI. The most important drawback of the OpenMP is that it can only run in shared memory computers that

are usually more expensive than distributed memory ones. OpenMP is mostly used for paralleling loop operations. It also needs a compiler that can support OpenMP structure [Chandra et al., 2000; Chapman et al., 2007].

MPI uses method of message passing to send and receive data as well as revoke the actual code in different processes to run [Gropp et al., 2014a,b]. MPI can runs on both shared and distributed memory architectures and exploits both task parallelism and data parallelism. These facts make it be widely used in a vast range of problems. In MPI routines it is much easier to maintain the values of variables because each process controls its own local variables. MPI is preferred to distributed memory architectures which are much cheaper than shared memory architectures. Since MPI does the real parallel task, it requires more programming efforts to change from a serial to a parallel version. Therefore MPI can be harder to debug. More details are given in the literature, e.g., [Chapman et al., 2007] for OpenMP and [Gropp et al., 2014a,b; Keller et al., 2010] for MPI. Hybrid methods that couple OpenMP and MPI can be found in [Drosinos and Koziris, 2004; Jost et al., 2003; Nakajima, 2012; Rabenseifner et al., 2009; Weiss, 2012].

### 4.3.4 A case study: parameter estimation of the CSTR model

#### 4.3.4.1 The interior point quasi-sequential approach

The CSTR model Eqs. (4.9) is used here as an example of the PE problem. Following [Kim et al., 1990], to avoid the collinearity between parameters, a parameter transformation is used, resulting in the following rate equations:

$$k_1 = p_1 \exp\left[-p_2\left(\frac{T_r}{T} - 1\right)\right] \tag{4.45a}$$

$$p_1 = k_0 \exp\left(\frac{-E}{RT_r}\right); \tag{4.45b}$$

$$p_2 = \left(\frac{E}{RT_r}\right); \tag{4.45c}$$

where $T_r = 350(K)$ is reference temperature.

where $k_1$ is the Arrhenius rate expression and $k_0$ and $E$ are the Arrhenius constants. These parameters will be estimated based on measurement data profiles.

Here, two parameters $p_1$ and $p_2$ need to be estimated. The state variables are the level of the tank $h$, the molar concentration $c$, and the reactor temperature $T$. Control variables are the outlet flow rate $F$ and the coolant liquid temperature $T_c$. $F_0, c_0, T_0$ are the known constant inlet flow, inlet concentration and coolant liquid temperature. Data sets are generated by adding Gaussian distributed error of 5% to 10% noises to the signals of the model. In addition, step changes to input and control variables $F_0, c_0, T_0, F, T_c$ are introduced to the system and 10 data sets are produced in the case that parameter $p_1$ is assumed to start to decrease from $p_{1,0} = 1.0(min^{-l})$ to $0.8(min^{-l})$ at $t = 15min$ with a first-order filter (time constant $15min.$) due to some strong disturbances in the operation condition and $p_2$ keeps constant at the value of $p_2 = 25.0$ over the time horizon of $t_f = 50min$. The data sets are generated at the sampled time of 1 minute so that there are 50 subintervals $NC = 50$ (see Fig. 4.2). The PE problem is formulated as follows:

$$\min_{p_l, u_{j,l}, x_{j,l}, y_{j,l}} F = \sum_{j=1}^{10} F_j = \sum_{j=1}^{10} \sum_{l=1}^{50} \Big[ \left(y_{j,l} - y_{j,l}^M\right)^T V_y^{-1} \left(y_{j,l} - y_{j,l}^M\right) + \left(u_{j,l} - u_{j,l}^M\right)^T V_u^{-1} \left(u_{j,l} - u_{j,l}^M\right) \Big] \tag{4.46a}$$

s.t.

$$\frac{dh}{dt} = \frac{F_0 - F}{\pi r^2} \tag{4.46b}$$

$$\frac{dc}{dt} = \frac{F_0(c_0 - c)}{\pi r^2 h} - p_1 \exp\left[-p_2\left(\frac{T_r}{T} - 1\right)\right] c \tag{4.46c}$$

$$\frac{dT}{dt} = \frac{F_0(T_0 - T)}{\pi r^2 h} + \frac{-\Delta H}{\rho C_p} p_1 \exp\left[-p_2\left(\frac{T_r}{T} - 1\right)\right] c + \frac{2U}{r\rho C_p}(T_c - T) \tag{4.46d}$$

$$0.5 \le h \le 2.5 \quad (m) \tag{4.46e}$$

$$0.87 \le c \le 1.0 \quad (mol/l) \tag{4.46f}$$

$$290 \le T \le 350 \quad (K) \tag{4.46g}$$

$$85 \le F \le 115 \quad (l/min) \tag{4.46h}$$

$$290 \le Tc \le 310 \quad (K) \tag{4.46i}$$

$$p^L \le p \le p^U \tag{4.46j}$$

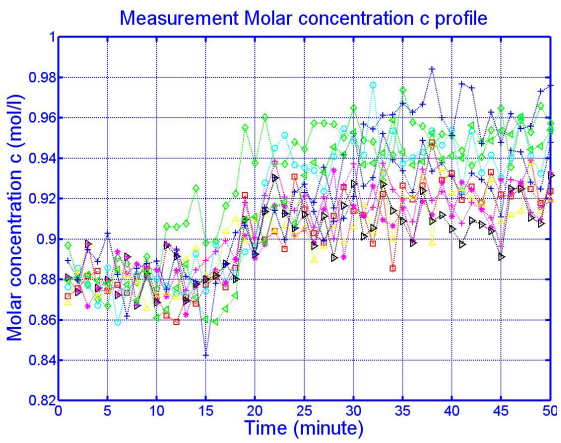
where : $p = [p_1, p_2]^T;\ u = [F, T_c]^T;\ y = [h, c, T]^T;\ x = \emptyset$.

The middle layer sub-NLP problems are solved by the IPOPT Fortran coded by A. Wächter [Wächter, 2002] in two manners: the reduced-space and the full space type in three cases, the $1^{st}$ case (start value $[p_{1,0}, p_{2,0}] = [0.9, 24.0]$) and $2^{nd}$ case (start value $[p_{1,0}, p_{2,0}] = [0.8, 26.0]$) with the assumption that both $p_1$ and $p_2$ are free variables, and the $3^{rd}$ case (start value $p_{1,0} = 0.8$) with the assumption that $p_1$ is free and $p_2$ is kept constant at the true value $p_2 = 25.0$. Results computed by a desktop CPU Intel D830 3.0GHz and 1GB RAM are shown in Table 4.2, Fig. 4.7 to Fig. 4.11.

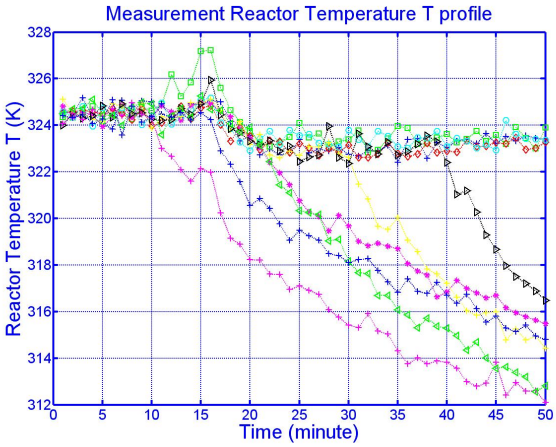
As shown clearly in Table 4.2, the reduced-space method with the basic matrix formed by slack variables is generally 2.38 times to 4.58 times faster than the full space method whereas the number of iterations are nearly the same. Fig. 4.8a and Fig. 4.8b show that, in the $1^{st}$ and $2^{nd}$ case, both $p_1$ and $p_2$ are kept at their shapes but cannot converge to their real values. This result may be due to the strong correlations between the two estimated parameters $p_1$ and $p_2$ because the results in Fig. 4.10a and Fig. 4.10b show that the state variables in the two cases nearly fit their originals (with different values of $p_1$ and $p_2$). In the $3^{rd}$ case, we keep the parameter $p_2$ constant as its real value. Then we obtain the result as shown in Fig. 4.9a. It can be seen that $p_1$ fits its real value in an acceptable manner. This means there is an identifiability problem. This is maybe due to the fact that the sensitivities of parameter $p_2$ to the output variables are too small. Fig. 4.9a also shows that when we increase the number of measurement data sets, the parameter estimation will have better results.

As for the data reconciliation, Figs. 4.10 and 4.11 show that the EIV method gives acceptable results. Fig. 4.8 also shows that if the number of measurement data sets are increased from 1 to 5 and 10, the parameter estimation will provide better results.

Moreover, in order to test the case in that the measure of all the state and control variables is sparse and incomplete (that is usually the real situation), it is assumed that the concentration cannot be measured during the experiment. It means that the $y$ variables in Eq. (4.46) now changed to $y = [h,\ T]^T$ and $x = [c]^T$ . Results shown in Table 4.2, Figs. 4.8 to 4.11 lead to the conclusion that in this situation, the proposed method still can work well despite of the fact that it takes larger CPU time to solve the optimization problem.

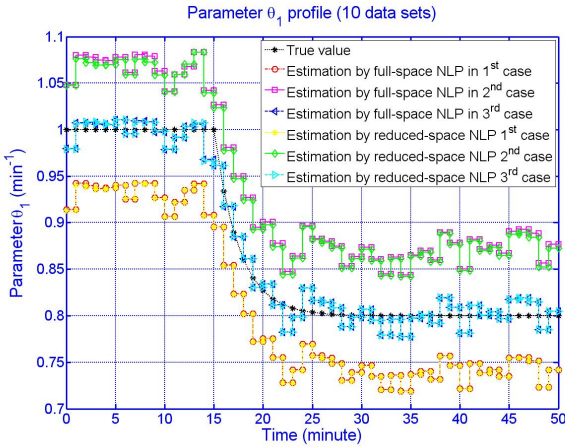



(a) Measurement of molar concentration

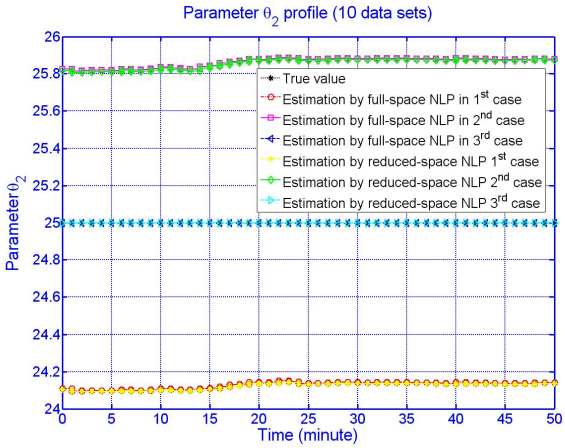



(b) Measurement of reactor temperature

Fig. 4.7 Measurement of 10 data sets




(a) Parameter $p_1$ profile

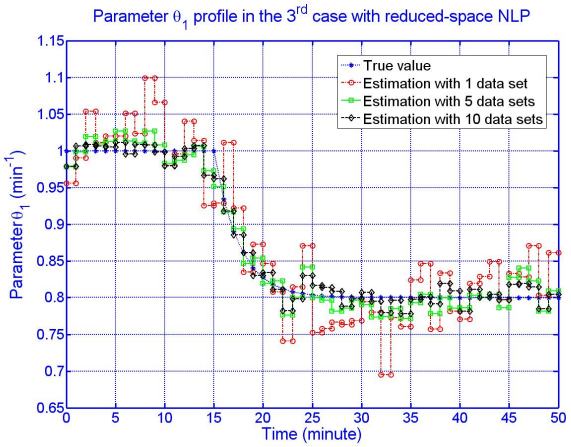



(b) Parameter $p_2$ profile

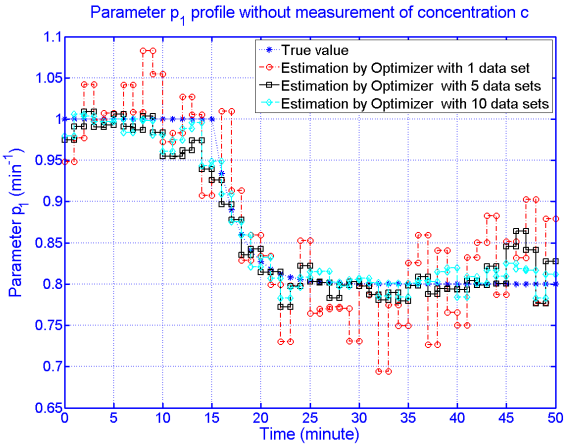
Fig. 4.8 Parameter identification results with 10 data sets in three cases

Table 4.2 CPU time and number of iterations

| **Description** | | **1st case** | | | **2nd case** | | | **3rd case** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Num. of data sets** | | **1** | **5** | **10** | **1** | **5** | **10** | **1** | **5** | **10** |
| CPU time (s) | rIP | 4.4 | 12.1 | 33.5 | 4.2 | 27.6 | 71.4 | 2.4 | 22.4 | 71.5 |
| | Full | 10.5 | 55.4 | 82.8 | 7.1 | 43.5 | 163.8 | 8.1 | 56.8 | 218.5 |
| Iterations | rIP | 6 | 5 | 7 | 7 | 8 | 8 | 5 | 6 | 6 |
| | Full | 6 | 6 | 7 | 7 | 8 | 7 | 5 | 6 | 6 |

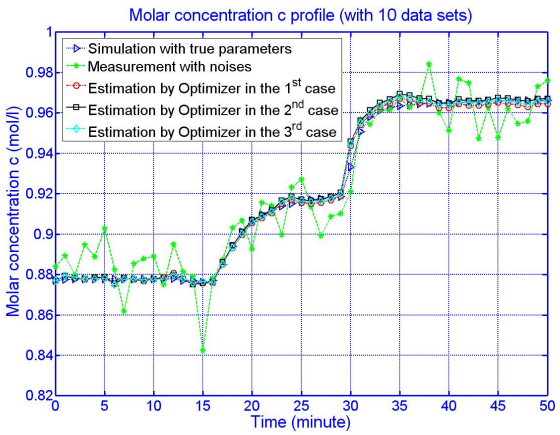



(a) Parameter $p_1$ with full measurement

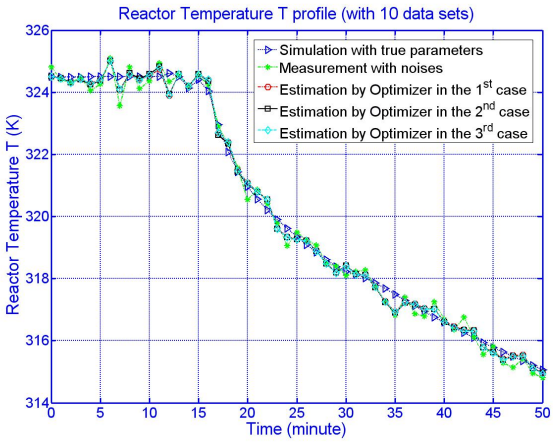



(b) Parameter $p_1$ without measurement of c

Fig. 4.9 Parameter $p_1$ identification results with full and lack of measurement of variables

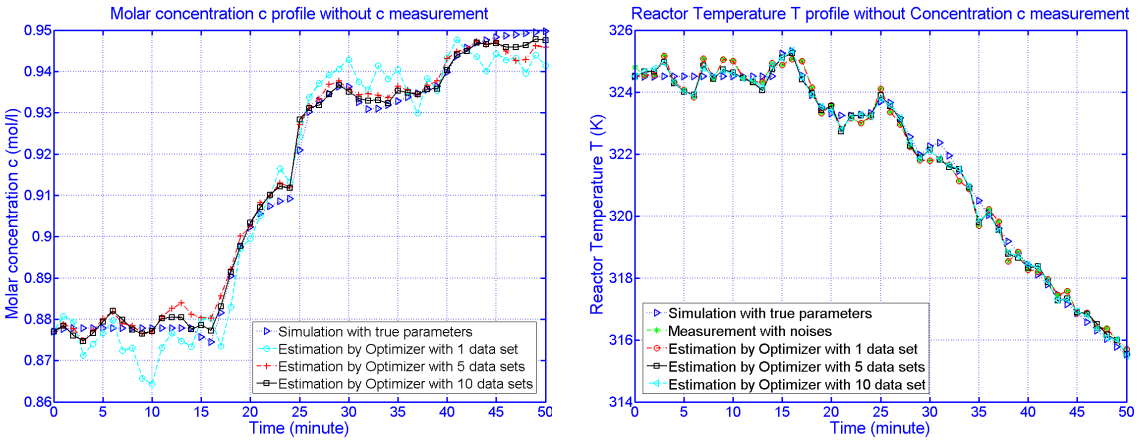



(a) Estimation of $c$ with full measurement




(b) Estimation of $T$ with full measurement

Fig. 4.10 Estimation of variables $c$ and $T$ with full measurement of variables




(a) Estimation of $c$ (b) Estimation of $T$

Fig. 4.11 Estimation of variables $c$ and $T$ with lack of measurement of $c$ variable

#### 4.3.4.2 The parallel computation approach

In the three layer quasi-sequential approach, the sub-NLPs are independent of each other. Therefore this method is suitable to be impelemented for parallel computation. In this study, a parallel strategy using the MPI structure (in the MPICH2 environment) is applied as in the following algorithm 4.3.2 and flowchart 4.12:

**Algorithm 4.3.2:** Algorithm of three-layer IP quasi-sequential approach in parallel mode

1. MPI Init;
2. pid ← id of process;
3. **if** *pid = 0* **then**
   `master()` ; `/* Procedure` (4.3.1) `*/`
   **else**
   `slave()` ; `/* Procedure` (4.3.2) `*/`
   **end**
4. MPI_Finalize();

In the MPICH2 environment, the number of nodes $N_{MPI}$ is depicted in the first column of Table 4.3. The reduction of the total CPU time $t^T$ in the parallel computation framework of the three-stage EIV problem depends much on the time of the function evaluation $t^F$ because this stage is to solve the sub-NLPs. The IPOPT CPU time $t^{IPOPT}$ of the upper layer NLP takes a smaller portion in the total CPU time $t^T$ and does not change much. The CPU time of the function evaluation $t^F$ relates to the

number of the data sets (associated with the sub-NLPs) in that the sub-NLPs are solved separately by each node in the MPI parallel mode. As we can see in Table 4.3 and Fig. 4.14, $t^F$ is reduced when the number of nodes are increased from 2 to 6. At the number of node $N_{MPI} = 6$, the total CPU time $t^F$ is reduced approximately 3 times in comparison with the case of 2 nodes (meaning there is no parallel node at all). In fact, the time reduction in the parallel mode depends on many factors as shown in Fig. 4.6, e.g., the hard level of tasks, the communication time between nodes, the load unbalance between tasks, and so on. When the number of nodes $N_{MPI}$ is increased from 6 to 10 nodes, the total CPU time does not change. This maybe is due to the hard task of some sub-NLPs that take most of the time $t^F$. When the $N_{MPI}$ is increased to 11, the total CPU time is increased too. This fact can be explained by the computation capacity of the computer system. The PC used in this study has only 4 physical cores that can handle only 8 threads. As shown in Fig. 4.13, when the number of nodes $N_{MPI}$ is larger than 10, the CPU of the dedicated PC is alway consumed at 100% capacity.

### 4.3.5 Summary

Parameter estimation for nonlinear dynamic systems remains as a challenging task both methodologically and computationally. In this Section, a three-stage computation framework is developed for solving parameter estimation problems for dynamic systems based on multiple data profiles. First, the dynamic parameter estimation problem is transformed to an NLP problem by using collocation on finite elements. The model parameters to be estimated are treated in the upper stage by solving an NLP problem. The middle stage consists of multiple NLPs nested in the upper stage, representing the data reconciliation step for each data profile. The dynamic optimization problems in the middle stage are solved by an efficient IP QSQ dynamic optimization method. Since the second-order derivatives of the model equations are avoided, the computation expense is significantly reduced. In addition, the decoupled sub-NLPs allow parallel computing that can make use of the HPC to improve CPU time consumption. The computational results obtained from parameter estimation for the case study demonstrate the effectiveness of the proposed approach.

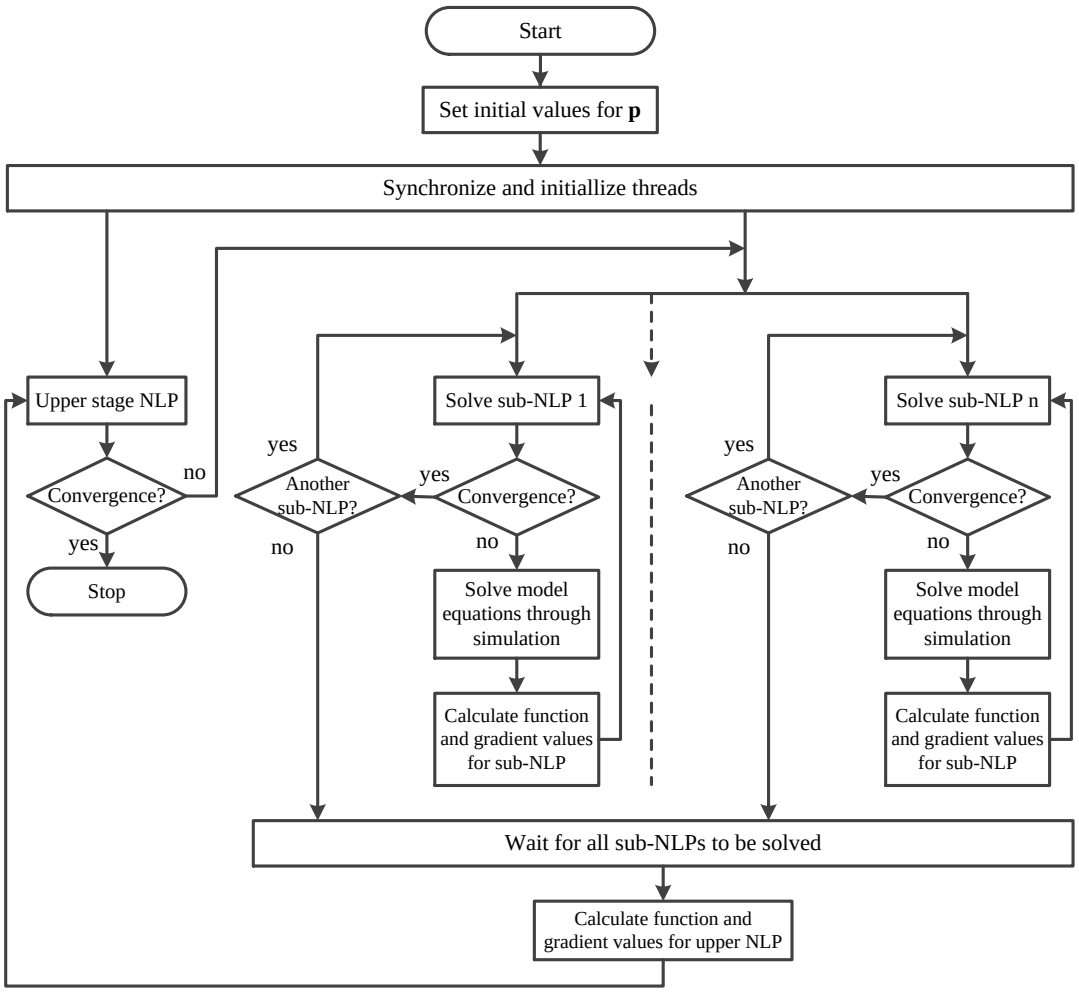



Fig. 4.12 Flowchart of the three-stage parallel computation framework

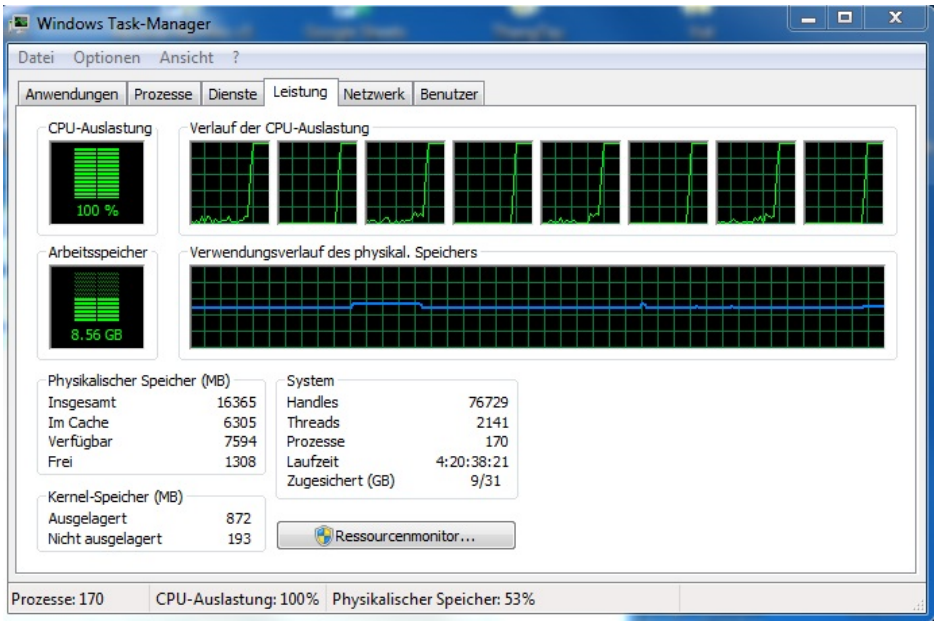



Fig. 4.13 CPU performance of the CSTR PE problem in parallel mode

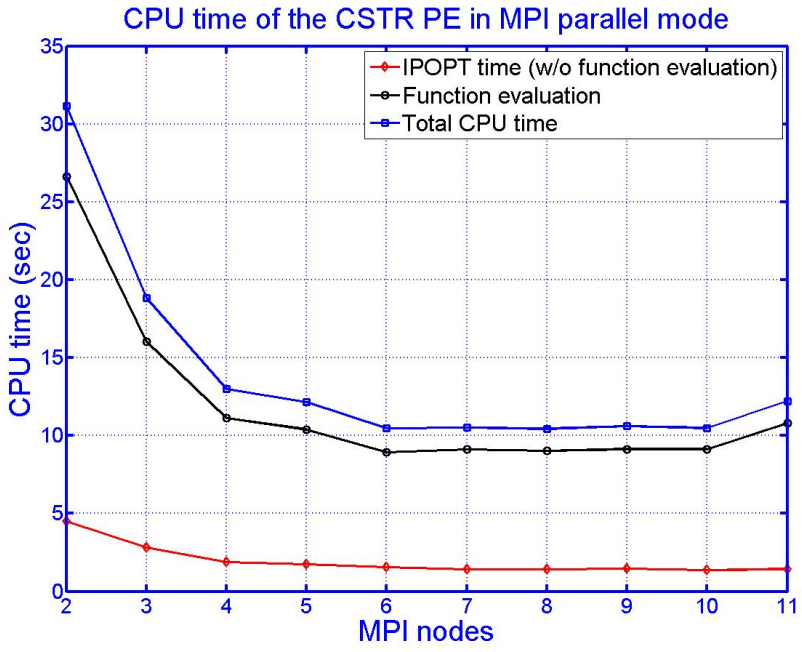



Fig. 4.14 Parallel computation of the CSTR parameter estimation

**Procedure 4.3.1:** master for the three-stage parallel computation framework

Provide initial values $p_0$ and bounds of parameter variables, set $q = 0$;
**repeat**// `Outer iteration`
1. ntasks ← number of process;
2. work ← NS ; `/* NS is number of data sets,` $ntasks \leq NS$ `*/`
3. **for** $j = 1$ *to ntasks* **do**
   4. Send a Message containing kind of work to every slave;
**end**
5. **for** $j = 1$ *to ntasks* **do**
   6. Send data (values of $p$) to every slave;
**end**
7. **while** *work != NULL* **do**
   8. Receive the results (values of objective function $f$ and the sensitivities $\frac{d\hat{y}_j}{d\hat{p}_j^q}$) from the slave that has finished its work;
   9. Send the free slave a new data;
   10. work ← work-1;
**end**
; `/* There is no more data set available */`
11. **for** $j = 1$ *to ntasks* **do**
   12. Receive all the outstanding results from the slaves;
**end**
13. Compute the objective function value $F = \sum_{j}^{NS} f_j$, the gradient

$$\frac{dF}{d\hat{p}} = \sum_{j=1}^{NS} \left[ \frac{\partial f_j}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial \hat{p}} \right] ;$$

14. Call an (IP) solver to get the search direction $dp_q$;
15. Set $\hat{p}_{q+1} \leftarrow \hat{p}_q + d\hat{p}_q$; $q \leftarrow q + 1$;

**until** *convergence criteria of the outer NLP are satisfied*;

**Procedure 4.3.2:** slave for the three-stage parallel computation framework

1. **while** *True* **do**
   2. Receive Message from the Master ; `/* kind of work */`
   3. **if** *Message = Problem solved* **then**
      break ; `/* get out of current loop! */`
   **else**
      4. **while** *True* **do**
         5. Receive Message containing $p$ value from the Master;
         6. **if** *Message = No more data set available* **then**
            break ; `/* get out of current loop! */`
         **else**
            7. Given a number of elements, discretize the control variables, and state variables with the collocation method, provide initial values of control variables and bounds for all variables, set $k = 0$ for the $j^{th}$ sub-NLP;
            8. **repeat// Inner iteration**
               9. Solve nonlinear model Eqs. (4.34b) using Newton-Raphson method in the **Simulation layer**;
               10. Compute the sensitivities $\frac{d\hat{z}_j}{d\hat{u}_j^k}$ and $\frac{d\hat{y}_j}{d\hat{u}_j^k}$;
               11. Compute the values of the object function and constraints: $f_j(\cdot), \hat{H}_j(\cdot)$, as well as their gradients with respect to $\hat{u}^k$: $\nabla_{\hat{u}} f_j(\cdot)$ and $\nabla_{\hat{u}} H_j(\cdot)$ as following:

$$\nabla_{\hat{u}} f = \frac{\partial f}{\partial \hat{y}^k}\frac{\partial \hat{y}^k}{\partial \hat{u}^k} + \frac{\partial f}{\partial \hat{u}^k}$$

               and $\nabla_{\hat{u}} \hat{H}(\hat{z}) = \frac{\partial \hat{H}}{\partial \hat{z}^k}\frac{\partial \hat{z}^k}{\partial \hat{u}^k} + \frac{\partial \hat{H}}{\partial \hat{u}^k}$, $\nabla_{\hat{u}} \hat{H}(\hat{y}) = \frac{\partial \hat{H}}{\partial \hat{y}^k}\frac{\partial \hat{y}^k}{\partial \hat{u}^k} + \frac{\partial \hat{H}}{\partial \hat{u}^k}$;
               12. Call an (IP) solver to get the search direction $d\hat{u}_k$;
               13. Set $\hat{u}_{k+1} \leftarrow \hat{u}_k + d\hat{u}_k; k \leftarrow k + 1$;
            **until** *convergence criteria the inner NLP are satisfied*;
            14. Compute the sensitivities $\frac{d\hat{y}_j}{d\hat{p}_j^q}$;
            15. Send the objective function value $f_j$ and the sensitivities $\frac{d\hat{y}_j}{d\hat{p}_j^q}$ to the Master;
         **end**
      **end**
   **end**
**end**

Table 4.3 Parallel computation of the CSTR parameter estimation

| Number of Nodes $N_{MPI}$ | IPOPT time (w/o function) $t^I$ | Function evaluation $t^F$ | Total CPU time $t^T$ |
|---|---|---|---|
| 2 | 4.524 | 26.624 | 31.148 |
| 3 | 2.814 | 16.011 | 18.825 |
| 4 | 1.868 | 11.125 | 12.993 |
| 5 | 1.744 | 10.396 | 12.14 |
| 6 | 1.542 | 8.916 | 10.458 |
| 7 | 1.409 | 9.114 | 10.523 |
| 8 | 1.404 | 9.029 | 10.433 |
| 9 | 1.466 | 9.14 | 10.606 |
| 10 | 1.355 | 9.135 | 10.49 |
| 11 | 1.429 | 10.789 | 12.218 |

## 4.4 An improved Multiple-Shooting Approach

In Section 3.8 of Chapter 3, the CMSC approach is described. In that development [Tamimi and Li, 2010] the inequalities (2.8e) are forced to become inequality constraints of the NLP only at the grid points of the model equation. This fact can let the state variables to violate the constraints (2.8e) at the collocation points inside the time intervals between the grid points. In order to overcome this drawback, an improvement is applied to the previous CMSC approach by applying constraints to all collocation points in each time interval. In the improvement approach, the NLP problem (3.91) can be rewritten as:

$$\min_{\tilde{u}_l, z_{l,0}} \quad f(\hat{x}(\tilde{u}_l, z_{l,0}), \tilde{u}_l) \tag{4.47a}$$

$$\text{s.t.} \quad z_{l+1,0} = \hat{z}_{l,NC} \tag{4.47b}$$

$$0 \leq \hat{H}(\hat{x}_{l,i}, \tilde{u}, z_{l,0}) \tag{4.47c}$$

$$\tilde{u}_L \leq \tilde{u}_l \leq \tilde{u}_U \tag{4.47d}$$

$$l = 1, \ldots, NL; i = 1, \ldots, NC \tag{4.47e}$$

together with the model equation discretized by the collocation method:

$$c_l(\hat{x}_{l,i}, \hat{u}_l, z_{l,0}, p) = 0 \tag{4.48}$$

### 4.4.1 Sequential simulation layer

As mentioned in 3.8, in the simulation layer, a (variation of) Newton-Raphson method is applied to solved the nonlinear systems (4.48) formed by the collocation method. A normal sequential solution of these nonlinear systems was applied in [Tamimi and Li, 2010] as depicted in the following Algorithm 4.4.1:

### 4.4.2 Parallel simulation layer

In the multiple shooting method, the initial value problem (IVP) solutions and derivative computations in all time intervals are independent of each other. Therefore this method is also suitable for parallel computation. In this Section, a parallel strategy using MPI environment is applied as in the following algorithm 4.4.2:

**Algorithm 4.4.2:** Algorithm of CMSC approach in parallel mode

1. MPI Init;
2. pid ← id of process;
3. **if** *pid = 0* **then**
   `master()` ; `/* Procedure` (4.3.1) `*/`
   **else**
   `slave()` ; `/* Procedure` (4.3.2) `*/`
   **end**
4. MPI_Finalize();

**Algorithm 4.4.1:** Algorithm of the improved combined multiple shooting and collocation strategy with sequential simulation layer

Given a number of elements, discretize the control variables, and state variables with the collocation method, provide initial values of control variables and bounds for all variables, set $k = 0$;

**repeat**`/* Outer iteration */`

1. **for** $l = 1$ *to* $NL$ **do**
   1. Solve nonlinear model Eqs. (3.89b) using the (variation of) Newton-Raphson method;
   2. Evaluate the sensitivities $\frac{d\hat{x}}{d\tilde{u}^k}$, $\frac{d\hat{x}}{dz_0^k}$;

**end**

; `/* Sequential simulation layer */`

2. Evaluate the values of the object function and constraints: $f(\hat{x}(\tilde{u}^k, z_0^k), \tilde{u}^k), \hat{H}(\hat{x}(\tilde{u}^k, z_0^k), \tilde{u}^k, z_0^k)$, as well as their gradients with respect to $\tilde{u}^k$ and $z_0^k$: $\nabla_{\tilde{u}} f(\hat{x}(\tilde{u}^k, z_0^k), \tilde{u}^k), \nabla_{z_0^k} f(\hat{x}(\tilde{u}^k, z_0^k), \tilde{u}^k), \nabla_{\tilde{u}} \hat{H}(\hat{x}_i(\tilde{u}^k, z_0^k), \tilde{u}^k, z_0^k)$, $\nabla_{z_0^k} \hat{H}(\hat{x}_i(\tilde{u}^k, z_0^k), \tilde{u}^k, z_0^k)$ (with $i = 1, \ldots, NC$) as following :

$$\nabla_{\tilde{u}} f = \frac{\partial f}{\partial x^k}\frac{\partial \hat{x}^k}{\partial \tilde{u}^k} + \frac{\partial f}{\partial \tilde{u}^k}$$

$$\nabla_{z_0^k} f = \frac{\partial f}{\partial x^k}\frac{\partial \hat{x}^k}{\partial z_0^k} + \frac{\partial f}{\partial z_0^k}$$

$$\nabla_{\hat{u}} \hat{H} = \frac{\partial \hat{H}}{\partial \hat{x}_i^k}\frac{\partial \hat{x}_i^k}{\partial \tilde{u}^k} + \frac{\partial \hat{H}}{\partial \tilde{u}^k}$$

$$\nabla_{z_0^k} \hat{H} = \frac{\partial \hat{H}}{\partial \hat{x}_i^k}\frac{\partial \hat{x}_i^k}{\partial z_0^k} + \frac{\partial \hat{H}}{\partial z_0^k}$$

3. Call the active-set SQP solver to solve the NLP to get the search direction $d\tilde{u}_k$;
4. Set $\tilde{u}_{k+1} \leftarrow \tilde{u}_k + d\tilde{u}_k; k \leftarrow k + 1$;

**until** *convergence criteria are satisfied*;

**Procedure 4.4.3:** Master for the CMSC

Provide initial values $u_{l,0}, p_0, z_{l,0}$ and bounds of all variables, set $q = 0$;

**repeat**// `Outer iteration`

1. ntasks ← number of process;
2. work ← NL ; `/* NL is number of time intervals,` $ntasks \leq NL$ `*/`
3. **for** $j = 1$ *to ntasks* **do**
   4. Send a Message containing kind of work to every slave;

**end**

4. **for** $j = 1$ *to ntasks* **do**
   5. Send data (values of $p$) to every slave;

**end**

6. **while** *work != NULL* **do**
   7. Receive the results (values of $\hat{x}_j$ and the sensitivities $\frac{d\hat{x}_j}{d\tilde{u}_j^q}$, $\frac{d\hat{x}_j}{d\hat{z}_0^q}$) from the slave that has finished its work;
   2. Send the free slave a new data;
   3. work ← work-1;

**end**

; `/* There is no more time interval available */`

6. **for** $j = 1$ *to ntasks* **do**
   1. Receive all the outstanding results from the slaves;

**end**

2. Evaluate the values of the object function and constraints: $f(\hat{x}(\tilde{u}^k, z_0^k), \tilde{u}^k), \hat{H}(\hat{x}(\tilde{u}^k, z_0^k), \tilde{u}^k, z_0^k)$, as well as their gradients with respect to $\tilde{u}^k$ and $z_0^k$: $\nabla_{\tilde{u}} f(\hat{x}(\tilde{u}^k, z_0^k), \tilde{u}^k), \nabla_{z_0^k} f(\hat{x}(\tilde{u}^k, z_0^k), \tilde{u}^k), \nabla_{\tilde{u}} \hat{H}(\hat{x}_i(\tilde{u}^k, z_0^k), \tilde{u}^k, z_0^k)$, $\nabla_{z_0^k} \hat{H}(\hat{x}_i(\tilde{u}^k, z_0^k), \tilde{u}^k, z_0^k)$ (with $i = 1, \ldots, NC$) as following :

$$\nabla_{\tilde{u}} f = \frac{\partial f}{\partial x^k}\frac{\partial \hat{x}^k}{\partial \tilde{u}^k} + \frac{\partial f}{\partial \tilde{u}^k}$$

$$\nabla_{z_0^k} f = \frac{\partial f}{\partial x^k}\frac{\partial \hat{x}^k}{\partial z_0^k} + \frac{\partial f}{\partial z_0^k}$$

$$\nabla_{\hat{u}} \hat{H} = \frac{\partial \hat{H}}{\partial \hat{x}_i^k}\frac{\partial \hat{x}_i^k}{\partial \tilde{u}^k} + \frac{\partial \hat{H}}{\partial \tilde{u}^k}$$

$$\nabla_{z_0^k} \hat{H} = \frac{\partial \hat{H}}{\partial \hat{x}_i^k}\frac{\partial \hat{x}_i^k}{\partial z_0^k} + \frac{\partial \hat{H}}{\partial z_0^k}$$

8. Call an (IP) solver to get the search direction $dp_q$;
9. Set $\tilde{u}_{q+1} \leftarrow \tilde{u}_q + d\tilde{u}_q, p_{q+1} \leftarrow p_q + dp_q$; $q \leftarrow q + 1$;

**until** *convergence criteria of the outer NLP are satisfied*;

**Procedure 4.4.4:** Slave for the CMSC

1. **while** *True* **do**
   2. Receive Message from the Master ; `/* kind of work */`
   3. **if** *Message = Do the computing* **then**
      4. **while** *True* **do**
         5. Receive Message containing $z_0, u, p$ value from the Master;
         6. **if** *Message = No more work* **then**
            goto step 2 ; `/* get out of current loop! */`
         **else**
            7. Solve the nonlinear system Eqs. (4.48);
            8. Compute the sensitivities $\frac{d\hat{x}_j}{d\tilde{u}_j^q}$, $\frac{d\hat{x}_j}{d\hat{z}_0^q}$;
            9. Send the objective function value $\hat{x}_j$ and the sensitivities to the Master;
         **end**
      **end**
   **else**
      break ; `/* get out of current loop! */`
   **end**

**end**

### 4.4.3 Case studies

#### 4.4.3.1 Control of a van der Pol oscillator

This van der Pol model is taken from [Canto et al., 2002] with a modification that constrains the first state with a lower boundary (4.49e). The modified problem is expressed as:

$$\min_{u} \quad x_3(t_f) \tag{4.49a}$$

$$s.t. \quad \dot{x}_1 = (1 - x_2^2)x_1 - x_2 + u \tag{4.49b}$$

$$\dot{x}_2 = x_1 \tag{4.49c}$$

$$\dot{x}_3 = x_1^2 + x_2^2 + u^2 \tag{4.49d}$$

$$-0.5 \leq x_1(t) \tag{4.49e}$$

$$-0.3 \leq u(t) \leq 1.0 \tag{4.49f}$$

$$x(0) = [0, 1, 0]^T \tag{4.49g}$$

$$t_f = 5.0 \tag{4.49h}$$

This parameter problem is now solved using three methods: IP QSQ, pure CMSC and modified CMSC with the number of time interval NL=[25,50,100,200,500] and random initials within the constraints (4.49f) with and without the constraint (4.49e) applied to the state $x_1$. The IP code IPOPT [Wächter and Biegler, 2006] is used to solve the NLP problem. The resulting nonlinear equation systems are solved by using subroutines in the NAG library [NAG, 2012]. The whole problem is coded in C/C++ and run on a one core of a desktop PC Core i7-2600 CPU 3.4GHz and 16GB RAM.

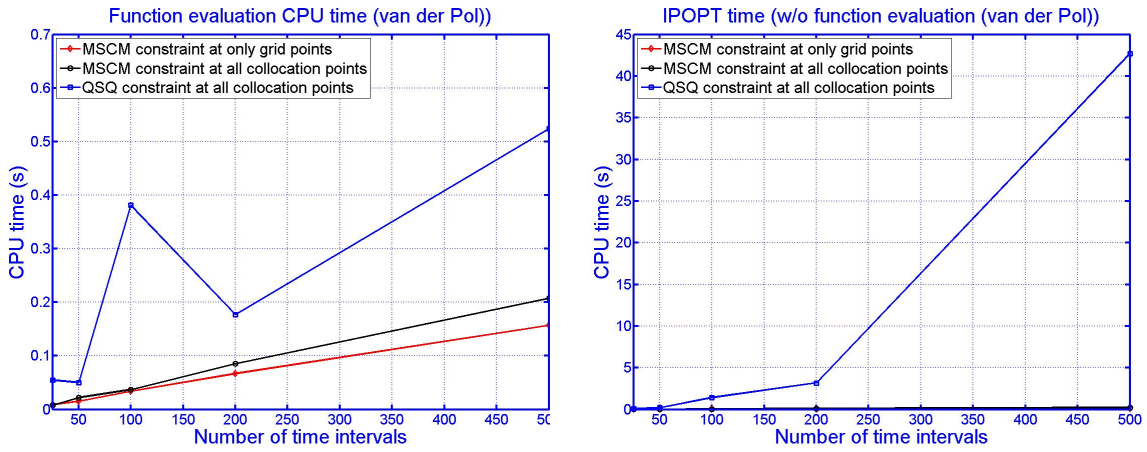
Results with CPU time including the time for solving the NLP without (w/o) function evaluation (IPOPT time), the time for solving model Eqs. (4.49b)-(4.49d) (function evaluation) and the total time, and the objective function value (OBJ) are shown in Table4.4.

In comparison to the function evaluation CPU time reported in Table 4.4 and Fig. 4.15a in the three cases, i.e., the IP QSQ with constraints at all collocation points, the pure CMSC with constraints at only grid points, the modified CMSC with constraints at all collocation points. It can be seen that IP QSQ takes the longest time which increases fast when the number of time intervals is increased. There are also instabilities in the function evaluation, which may be due to the sequential solution of the equation system Eqs. (4.49b)-(4.49d).

The CPU time of the two approaches, the pure CMSC and the modified CMSC, increases linearly with the number of time intervals. The CPU time of the modified CMSC is little bit higher than that one of the pure CMSC due to the increasing

number of constraints.

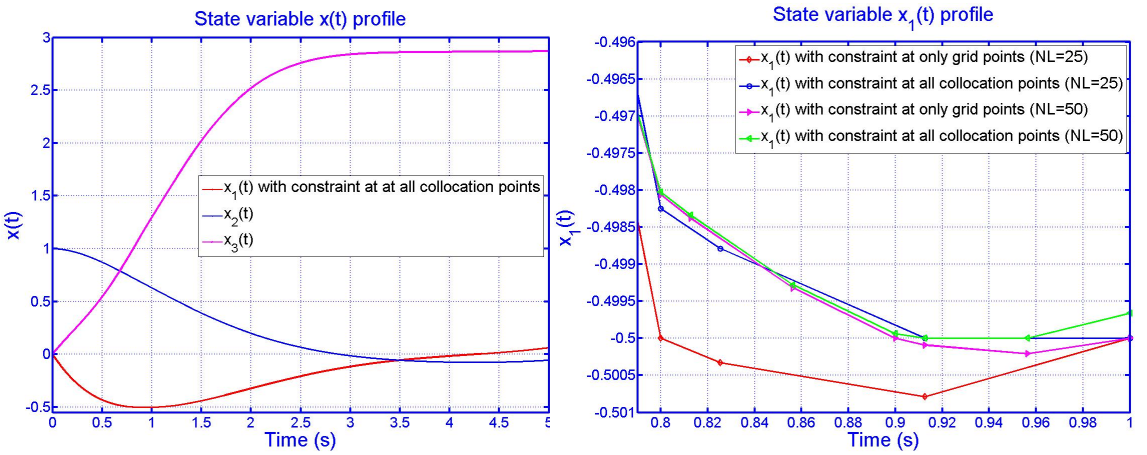
The most important advantage of the modified CMSC over the pure CMSC is shown in Fig. 4.16b. The pure CMSC lets the state $x_1$ variable violate the constraint conditions inside the time interval whereas the modified CMSC force the state $x_1$ to meet the constraint at all collocation points ($-0.5 \leq x_1(t)$). The different control $u(t)$ profiles that make the different $x(t)$ profiles are shown in Fig. 4.17.




(a) Function evaluation CPU time

(b) IPOPT CPU time

Fig. 4.15 CPU time of the van der Pol control problem with $x_1$ constraint



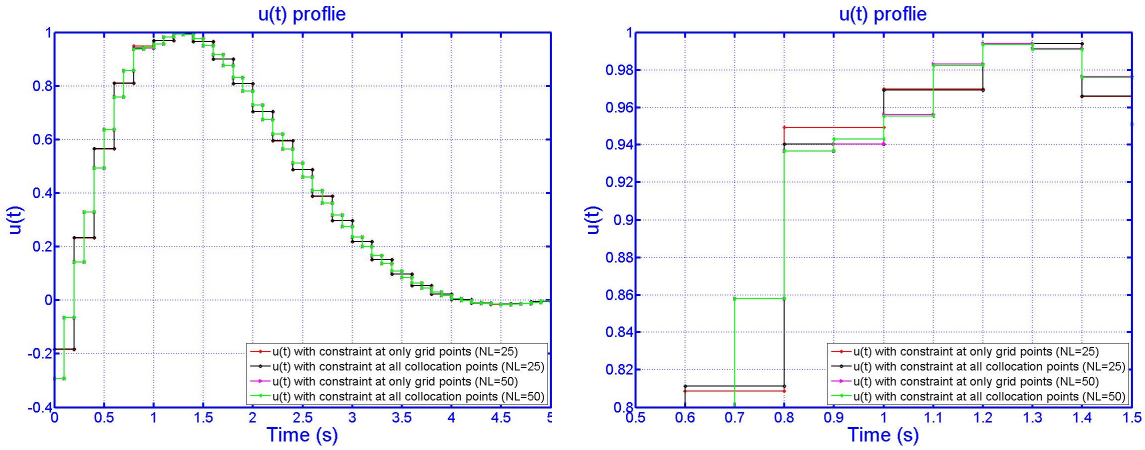

(a) x(t) profile with $x_1$ constraint ($-0.5 \leq x_1(t)$)

(b) State variable $x_1(t)$ profile with different constraint levels

Fig. 4.16 x(t) profile with $x_1$ constraint of the van der Pol control problem

Table 4.4 Results of the van der Pol Control problem

| Number of time intervals | IPOPT time (w/o function) (s) | Function evaluation (s) | Total CPU time (s) | OBJ value | $x_1$ constraint |
|---|---|---|---|---|---|
| quasi-sequential | | | | | |
| 25 | 0.057 | 0.019 | 0.076 | 2.876324 | no |
| 50 | 0.12 | 0.044 | 0.164 | 2.869367 | no |
| 100 | 0.835 | 0.107 | 0.942 | 2.867781 | no |
| 200 | 4.762 | 0.205 | 4.967 | 2.867391 | no |
| 500 | 36.632 | 0.394 | 37.026 | 2.867282 | no |
| 25 | 0.097 | 0.055 | 0.152 | 2.877997 | yes all |
| 50 | 0.186 | 0.05 | 0.236 | 2.870763 | yes all |
| 100 | 1.42 | 0.381 | 1.801 | 2.869039 | yes all |
| 200 | 3.18 | 0.177 | 4.95 | 2.868635 | yes all |
| 500 | 42.686 | 0.524 | 43.21 | 2.868522 | yes all |
| multiple shooting and collocation | | | | | |
| 25 | 0.015 | 0.01 | 0.025 | 2.876324 | no |
| 50 | 0.02 | 0.018 | 0.038 | 2.869367 | no |
| 100 | 0.027 | 0.031 | 0.058 | 2.867781 | no |
| 200 | 0.038 | 0.063 | 0.101 | 2.867391 | no |
| 500 | 0.085 | 0.157 | 0.242 | 2.867282 | no |
| 25 | 0.018 | 0.009 | 0.027 | 2.877546 | yes grid |
| 50 | 0.018 | 0.015 | 0.033 | 2.870631 | yes grid |
| 100 | 0.033 | 0.034 | 0.067 | 2.869019 | yes grid |
| 200 | 0.039 | 0.067 | 0.106 | 2.868627 | yes grid |
| 500 | 0.084 | 0.157 | 0.241 | 2.868517 | yes grid |
| 25 | 0.026 | 0.008 | 0.034 | 2.877637 | yes all |
| 50 | 0.029 | 0.022 | 0.051 | 2.870657 | yes all |
| 100 | 0.048 | 0.037 | 0.085 | 2.869027 | yes all |
| 200 | 0.106 | 0.085 | 0.191 | 2.868629 | yes all |
| 500 | 0.226 | 0.208 | 0.434 | 2.868517 | yes all |




(a) u(t) profile (b) u(t) profile in the time between [0.5 1.5] s

Fig. 4.17 u(t) profile with $x_1$ constraint of the van der Pol control problem

#### 4.4.3.2 Control of the nonlinear CSTR system

The DOP problem (4.12) is reconsidered here. This problem is solved by using three strategies, i.e., the IP QSQ, the pure CMSC and the modified CMSC with the number of time interval NL=50 and random initials of control variables within the constraints (4.12f)-(4.12g). The whole problem is coded in C/C++ with the MPI parallel environment MPICH2 and run on a desktop PC Core i7-2600 4-cores (8 threads) CPU 3.4GHz and 16GB RAM.

Results are reported in Table 4.5 and Fig. 4.18. As we can see, IP QSQ and the pure CMSC can converge to the objective function value of $x^{temp_*} = 0.901581$ after $653ms$ and $133ms$ CPU time, respectively. The modified CMSC is used here with parallel computation. In the MPICH2 environment, the number of nodes $N_{MPI}$ is depicted in the first column of Table 4.5. The reduction of the total CPU time $t^T$ in the parallel computation mode depends much on the time of the function evaluation $t^F$ because the IPOPT CPU time $t^{IPOPT}$ does not change. The CPU time of function evaluation, which relates to the number of time intervals in which the model equations are solved separately by each node in the MPI parallel mode, is reduced when the number of nodes are increased from 2 to 5. At the number of node $N_{MPI} = 5$, the total CPU time $t^T$ is reduced approximately 2.46 times in comparison to the pure CMSC method. In fact, the time reduction in the parallel mode in comparison to the serial mode depends on many factors as depicted in Fig. 4.6, e.g., the hard level

of tasks, the communication time between nodes, the load unbalance between tasks, and so on. When the number of nodes $N_{MPI}$ is increased to over 5 nodes, the total CPU time is increased significantly. This issue can be explained by the computation capacity of the computer system. The PC used in this study has only 4 physical cores with 8 threads. As shown in Fig. 4.19, when the number of nodes $N_{MPI}$ is larger than 5, the CPU of the dedicated PC is alway consumed at 100%.

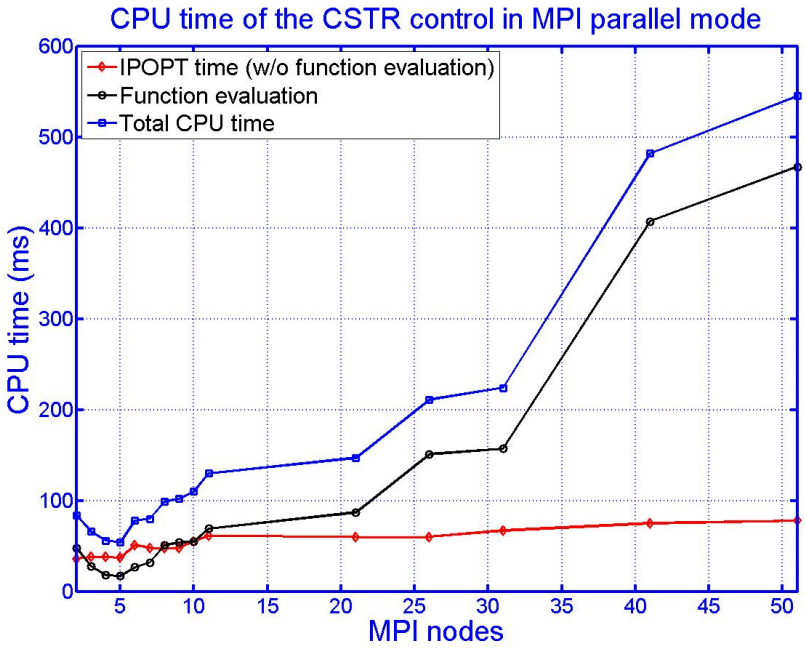



Fig. 4.18 Parallel computation performance of the CSTR control in MPI mode

#### 4.4.3.3 Parameter estimation of a three-step pathway model

A three-step pathway [Mendes, 2001; Moles et al., 2003; Rodriguez-Fernandez et al., 2006] is modeled by 8 nonlinear ODEs that describe 8 metabolic concentrations (state variables) with 36 parameters, as given in Eq. (4.50). The $P$ and $S$ values in the ODEs are considered as two control inputs specified by experimental design. To do the PE problem, it is assumed that all the states can be measured with different inputs that is used to produced 16 data sets. Each state is sampled with 20 measurement points during the time horizon $t_f = 120(s)$.

Table 4.5 CPU time (second) and OBJ value of the CSTR control problem

| Number of Nodes $N_{MPI}$ | IPOPT time (w/o function) $t^{IPOPT}(s)$ | Function evaluation $t^F(s)$ | Total CPU time $t^T(s)$ | OBJ value $x_*^{temp}$ |
|---|---|---|---|---|
| IP Quasi-sequential | | | | |
| 1 | 0.555 | 0.098 | 0.653 | 0.901581 |
| Multiple shooting with collocation [Tamimi and Li, 2010] | | | | |
| 1 | 0.081 | 0.052 | 0.133 | 0.901581 |
| MPI parallel | | | | |
| 2 | 0.036 | 0.048 | 0.084 | 0.901581 |
| 3 | 0.038 | 0.028 | 0.066 | 0.901581 |
| 4 | 0.038 | 0.018 | 0.056 | 0.901581 |
| 5 | 0.037 | 0.017 | 0.054 | 0.901581 |
| 6 | 0.051 | 0.027 | 0.078 | 0.901581 |
| 7 | 0.048 | 0.032 | 0.08 | 0.901581 |
| 8 | 0.048 | 0.051 | 0.099 | 0.901581 |
| 9 | 0.048 | 0.054 | 0.102 | 0.901581 |
| 10 | 0.055 | 0.055 | 0.11 | 0.901581 |
| 11 | 0.061 | 0.069 | 0.13 | 0.901581 |
| 21 | 0.06 | 0.087 | 0.147 | 0.901581 |
| 26 | 0.06 | 0.151 | 0.211 | 0.901581 |
| 31 | 0.067 | 0.157 | 0.224 | 0.901581 |
| 41 | 0.075 | 0.407 | 0.482 | 0.901581 |
| 51 | 0.078 | 0.467 | 0.545 | 0.901581 |

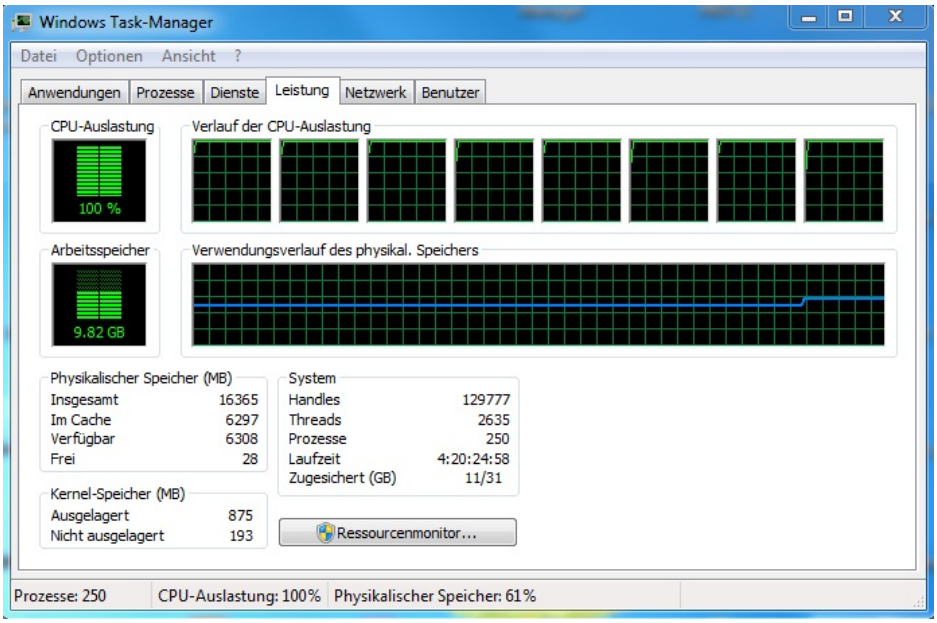



Fig. 4.19 Performance of the PC in parallel computation with over 5 nodes

$$\dot{G}_1 = \frac{V_1}{1 + \left(\frac{P}{Ki_1}\right)^{ni_1} + \left(\frac{Ka_1}{S}\right)^{na_1}} - k_1 G_1 \tag{4.50a}$$

$$\dot{G}_2 = \frac{V_2}{1 + \left(\frac{P}{Ki_2}\right)^{ni_2} + \left(\frac{Ka_2}{M_1}\right)^{na_2}} - k_2 G_2 \tag{4.50b}$$

$$\dot{G}_3 = \frac{V_3}{1 + \left(\frac{P}{Ki_3}\right)^{ni_3} + \left(\frac{Ka_3}{M_2}\right)^{na3}} - k_3 G_3 \tag{4.50c}$$

$$\dot{E}_1 = \frac{V_4 G_1}{K_4 + G_1} - k_4 E_1 \tag{4.50d}$$

$$\dot{E}_2 = \frac{V_5 G_2}{K_5 + G_2} - k_5 E_2 \tag{4.50e}$$

$$\dot{E}_3 = \frac{V_6 G_3}{K_5 + G_3} - k_6 E_3 \tag{4.50f}$$

$$\dot{M}_1 = \frac{kcat_1 E_1 (\frac{1}{Km_1})(S - M_1)}{1 + \frac{S}{Km_1} + \frac{M_1}{Km_2}} - \frac{kcat_2 E_2 (\frac{1}{Km_3})(M_1 - M_2)}{1 + \frac{M_1}{Km_3} + \frac{M_2}{Km_4}} \tag{4.50g}$$

$$\dot{M}_2 = \frac{kcat_2 E_2 (\frac{1}{Km_3})(M_1 - M_2)}{1 + \frac{M_1}{Km_3} + \frac{M_2}{Km_4}} - \frac{kcat_3 E_3 (\frac{1}{Km_5})(M_2 - P)}{1 + \frac{M_2}{Km_5} + \frac{P}{Km_6}} \tag{4.50h}$$

The PE problem can be briefly expressed as:

$$\min_{p_l, y_{j,l}} F = \sum_{j=1}^{NS} F_j = \sum_{j=1}^{NS} \sum_{l=1}^{NL} \left[ \left( y_{j,l} - y_{j,l}^M \right)^T V_y^{-1} \left( y_{j,l} - y_{j,l}^M \right) \right] \tag{4.51a}$$

$$s.t. \quad \text{the model Eqs.(4.50)} \tag{4.51b}$$

$$y_L \leq y \leq y_U \tag{4.51c}$$

$$p_L \leq p \leq p_U \tag{4.51d}$$

$$y(0) = y_0 \tag{4.51e}$$

$$t_f = 120.0(s)$$

where $y = [G_1, G_2, G_3, E_1, E_2, E_3, M_1, M_2]^T, NS = 16, NL = 20$, $p$ is a vector of 36 parameters named in the first column of Tables 4.7 and 4.8.

This DOP problem is formulate in two approaches, i.e., the sequential simulation layer (serial) with the algorithm 4.4.1 and the parallel simulation layer with the algorithm 4.4.2 coupled with Procedures 4.4.3 and 4.4.4. The program is coded in C/C++ and run on a PC with Intel Core i7-980, 6 physical cores at 3.33GHz (12 threads), 4GB RAM, Windows XP. The IP code IPOPT [Wächter and Biegler, 2006] is used to solve the NLP problem. The resulting nonlinear equation systems are solved by using subroutines in the NAG library [NAG, 2012]. The MPICH2 environment is used to utilize parallel computing.

In order to make a comparison, the Advanced Model Identification using Global Optimization (AMIGO) package [Balsa-Canto et al., 2010; Balsa-Canto and Banga, 2011; E Balsa-Canto, 2010a,b] is used as a competitor. All the required data for the PE problem, i.e., boundaries for unknown parameters $p_L, p_U$, initial values $p_0$, experimental data sets, are suggested by AMIGO. This problem is solved in two extreme cases with two different initial values of the parameter, one with small values near the lower boundaries and the other with large values near the upper boundaries, as depicted in Tables 4.7 and 4.8. Results are reported in Table 4.6 (for CPU time consumption), and Tables 4.7 and 4.8 (for the estimated parameters values). Figs. 4.20-4.23 show the state profile associated with estimated parameter values in the high initials case. As depicted in Tables 4.7 and 4.8, in these two extreme cases, AMIGO cannot converge to the true values of the parameters $p$ whereas the proposed approach (modified CMSC) does. Table 4.6 shows the CPU time in which the modified CMSC in the parallel mode is nearly 3.6 and 2.48 times faster than using the serial mode in the low and

high initials case, respectively. In the parallel mode, the modified CMSC is also 5.23 and 1.85 times faster than using the AMIGO package, respectively.

Table 4.6 CPU time comparison between AMIGO and modified CMSC

| CPU time (s) | | | | | | |
|---|---|---|---|---|---|---|
| AMIGO | | modified CMSC | | | | |
| Low | High | Low | | High | | |
| | | Serial | Parallel | Serial | Parallel | |
| | | 20.9 | 19.3 | 51.6 | 44.5 | IPOPT |
| | | 48.7 | 8.8 | 196.8 | 24.7 | Func. |
| 147 | 128 | 69.6 | 28.1 | 248.4 | 69.2 | Total |

Table 4.7 Result of parameter estimation problem of three path-way

| Para | LB | UB | Initials | | True | AMIGO Results | | mod. CMSC results | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Low | High | | Low | High (with confidence interval) | Low | High |
| $V_1$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 2.72e+02 | 3.5427e+0 +− 1.9137e+2 | 7.09e−01 | 7.09e−01 |
| $Ki_1$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 1.93e+02 | 1.0049e+0 +− 3.4310e−1 | 9.97e−01 | 9.97e−01 |
| $ni_1$ | 1.00e-1 | 1.00e+1 | 1.00e-1 | 3.00e+0 | 2.00e+0 | 7.36e+00 | 1.9968e+0 +− 1.9592e+0 | 2.00e+00 | 2.00e+00 |
| $Ka_1$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 4.17e+02 | 9.9758e−1 +− 4.1384e−1 | 9.97e−01 | 9.97e−01 |
| $na_1$ | 1.00e-1 | 1.00e+1 | 1.00e-1 | 3.00e+0 | 2.00e+0 | 4.75e−01 | 1.9950e+0 +− 7.9737e−1 | 2.03e+00 | 2.03e+00 |
| $k_1$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 4.62e+01 | 3.5470e+0 +− 1.9162e+2 | 7.10e−01 | 7.10e−01 |
| $V_2$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 3.17e+02 | 1.0906e+0 +− 9.9678e+0 | 9.04e−01 | 9.04e−01 |
| $Ki_2$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 7.47e+01 | 1.0055e+0 +− 3.7916e−1 | 9.99e−01 | 9.99e−01 |
| $ni_2$ | 1.00e-1 | 1.00e+1 | 1.00e-1 | 3.00e+0 | 2.00e+0 | 8.40e+00 | 2.0358e+0 +− 2.6171e+0 | 2.00e+00 | 2.00e+00 |
| $Ka_2$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 2.40e+02 | 9.9402e−1 +− 4.1269e−1 | 1.00e+00 | 1.00e+00 |
| $na_2$ | 1.00e-1 | 1.00e+1 | 1.00e-1 | 3.00e+0 | 2.00e+0 | 4.60e-01 | 1.9927e+0 +− 9.2647e−1 | 2.00e+00 | 2.00e+00 |
| $k_2$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 6.29e+01 | 1.0954e+0 +− 1.0024e+1 | 9.04e−01 | 9.04e−01 |
| $V_3$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 4.51e+02 | 1.4751e+0 +− 1.7430e+1 | 9.58e−01 | 9.57e−01 |
| $Ki_3$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 4.47e+02 | 7.7397e−1 +− 1.5451e+0 | 9.99e−01 | 9.99e−01 |
| $ni_3$ | 1.00e-1 | 1.00e+1 | 1.00e-1 | 3.00e+0 | 2.00e+0 | 1.74e+00 | 1.4950e+0 +− 3.1014e+0 | 2.00e+00 | 2.00e+00 |
| $Ka_3$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 5.23e+01 | 1.2700e+0 +− 2.2474e+0 | 9.99e−01 | 9.99e−01 |
| $na_3$ | 1.00e-1 | 1.00e+1 | 1.00e-1 | 3.00e+0 | 2.00e+0 | 6.87e−01 | 1.8456e+0 +− 1.5074e+0 | 2.00e+00 | 2.00e+00 |
| $k_3$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 7.98e+01 | 1.1228e+0 +− 1.3148e+1 | 9.58e−01 | 9.57e−01 |

Table 4.8 Result of parameter estimation problem of three path-way (continued)

| Para | LB | UB | Initials | | True | AMIGO Results | | mod. CMSC results | |
|---|---|---|---|---|---|---|---|---|---|
| | | | Low | High | | Low | High (with confidence interval) | Low | High |
| $V_4$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 3.00e−1 | 1.00e−1 | 1.92e+02 | 9.2291e−2 +- 1.6956e−1 | 1.01e−01 | 1.01e−01 |
| $K_4$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 5.34e−01 | 1.0521e+0 +- 2.1861e+0 | 9.63e−01 | 9.63e−01 |
| $k_4$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 3.00e−1 | 1.00e−1 | 2.71e+02 | 8.9625e−2 +- 1.3832e−1 | 1.04e−01 | 1.04e−01 |
| $V_5$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 3.00e−1 | 1.00e−1 | 3.24e+02 | 1.0281e−1 +- 2.0309e−1 | 1.00e−01 | 1.00e−01 |
| $K_5$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 5.54e+00 | 1.0502e+0 +- 2.4784e+0 | 9.98e−01 | 9.98e−01 |
| $k_5$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 3.00e−1 | 1.00e−1 | 8.10e+01 | 9.9780e−2 +- 1.8010e−1 | 1.01e−01 | 1.01e−01 |
| $V_6$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 3.00e−1 | 1.00e−1 | 4.23e+02 | 9.1005e−2 +- 2.3789e−1 | 1.00e−01 | 1.00e−01 |
| $K_6$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 1.46e+00 | 7.9651e−1 +- 2.2426e+0 | 9.99e−01 | 9.99e−01 |
| $k_6$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 3.00e−1 | 1.00e−1 | 3.13e+02 | 1.0598e−1 +- 2.4081e−1 | 1.00e−01 | 1.00e−01 |
| $kcat_1$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 1.28e+02 | 9.2281e−1 +- 4.7910e−1 | 1.01e+00 | 1.01e+00 |
| $Km_1$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 2.32e+02 | 1.1794e+0 +- 2.2996e+0 | 1.00e+00 | 1.00e+00 |
| $Km_2$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 4.94e+02 | 2.2439e+0 +- 1.9350e+1 | 1.08e+00 | 1.08e+00 |
| $kcat_2$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 2.13e+02 | 9.4993e−1 +- 1.5644e+0 | 9.88e−01 | 9.88e−01 |
| $Km_3$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 1.95e+02 | 1.0784e+0 +- 2.8363e+0 | 1.01e+00 | 1.01e+00 |
| $Km_4$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 3.55e+02 | 1.8148e+0 +- 1.6904e+1 | 1.00e+00 | 1.00e+00 |
| $kcat_3$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 2.28e+02 | 1.2107e+0 +- 2.9047e+0 | 9.86e−01 | 9.86e−01 |
| $Km_5$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 6.84e+01 | 1.3336e+0 +- 4.6697e+0 | 1.01e+00 | 1.01e+00 |
| $Km_6$ | 1.00e-6 | 5.00e+2 | 1.00e-2 | 2.00e+0 | 1.00e+0 | 6.94e−01 | 1.1271e+0 +- 2.5395e+0 | 1.01e+00 | 1.01e+00 |

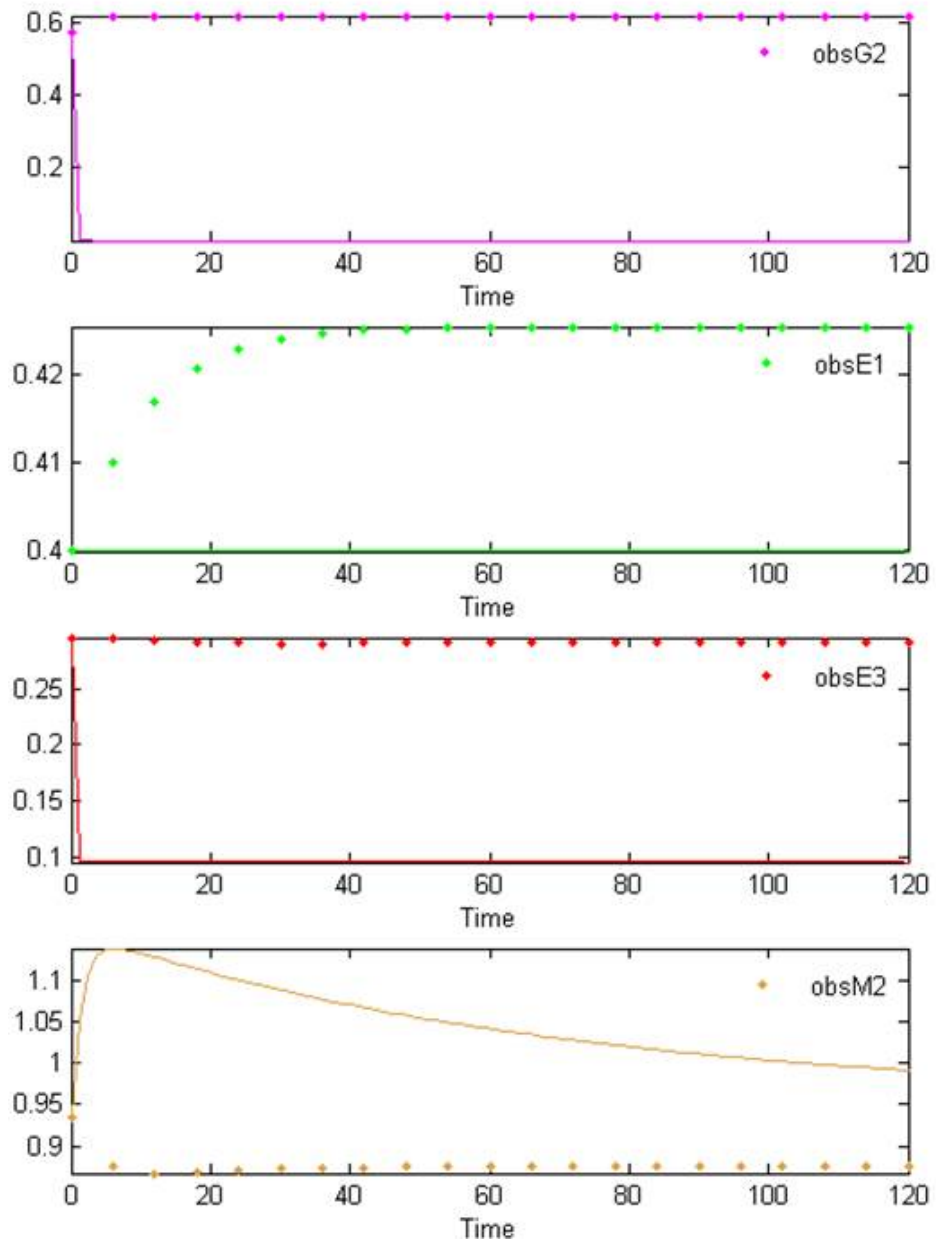



Fig. 4.20 State profiles of the AMIGO package in the high initials case

### 4.4.4 Summary

The pure CMSC has been shown to be efficient to solve DOPs. However, it has a drawback of chance of letting the state variables violate the constraint conditions inside each time interval of the discretization grids. The modified CMSC is proposed

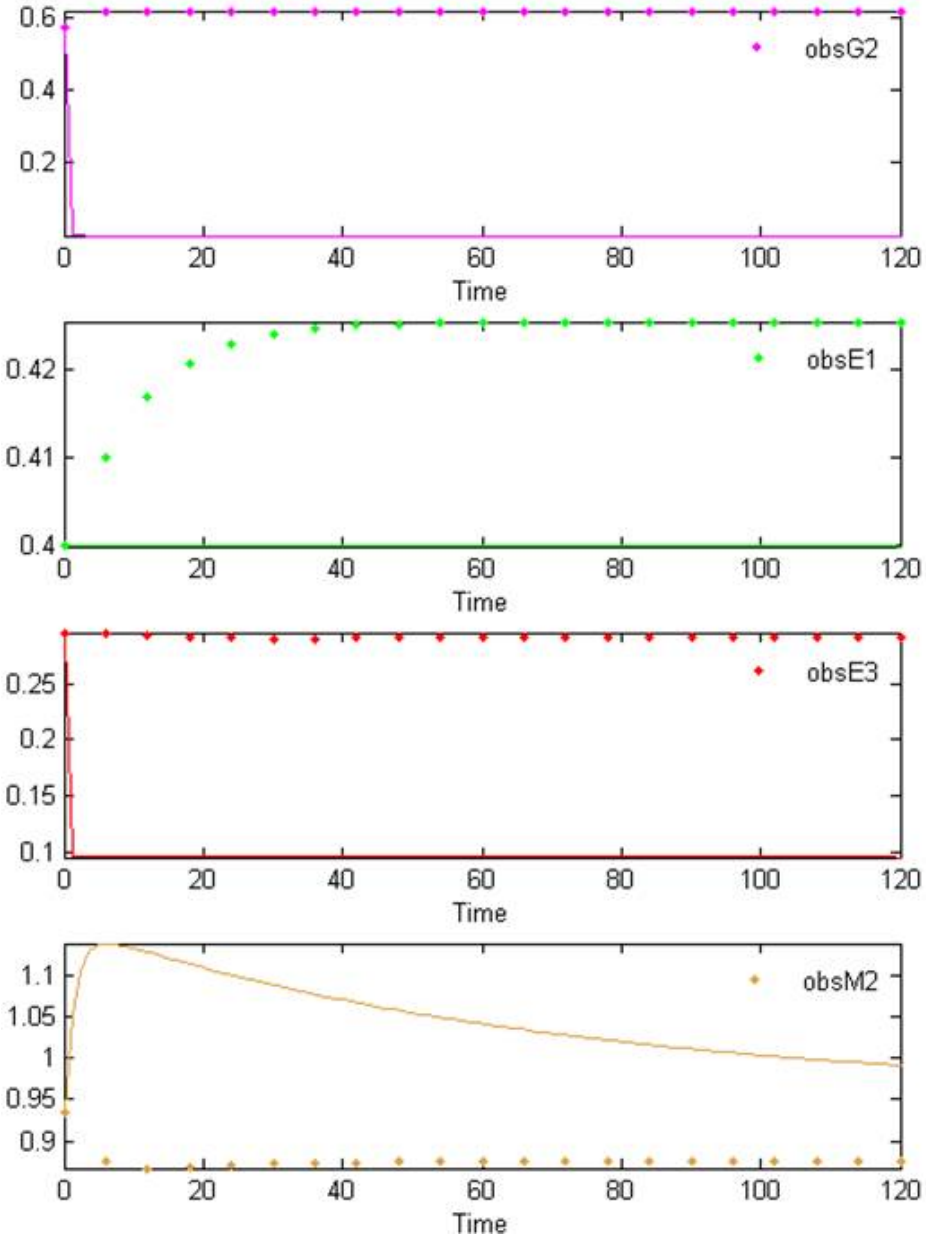



Fig. 4.21 State profiles of the AMIGO package in the high initials case (continued)

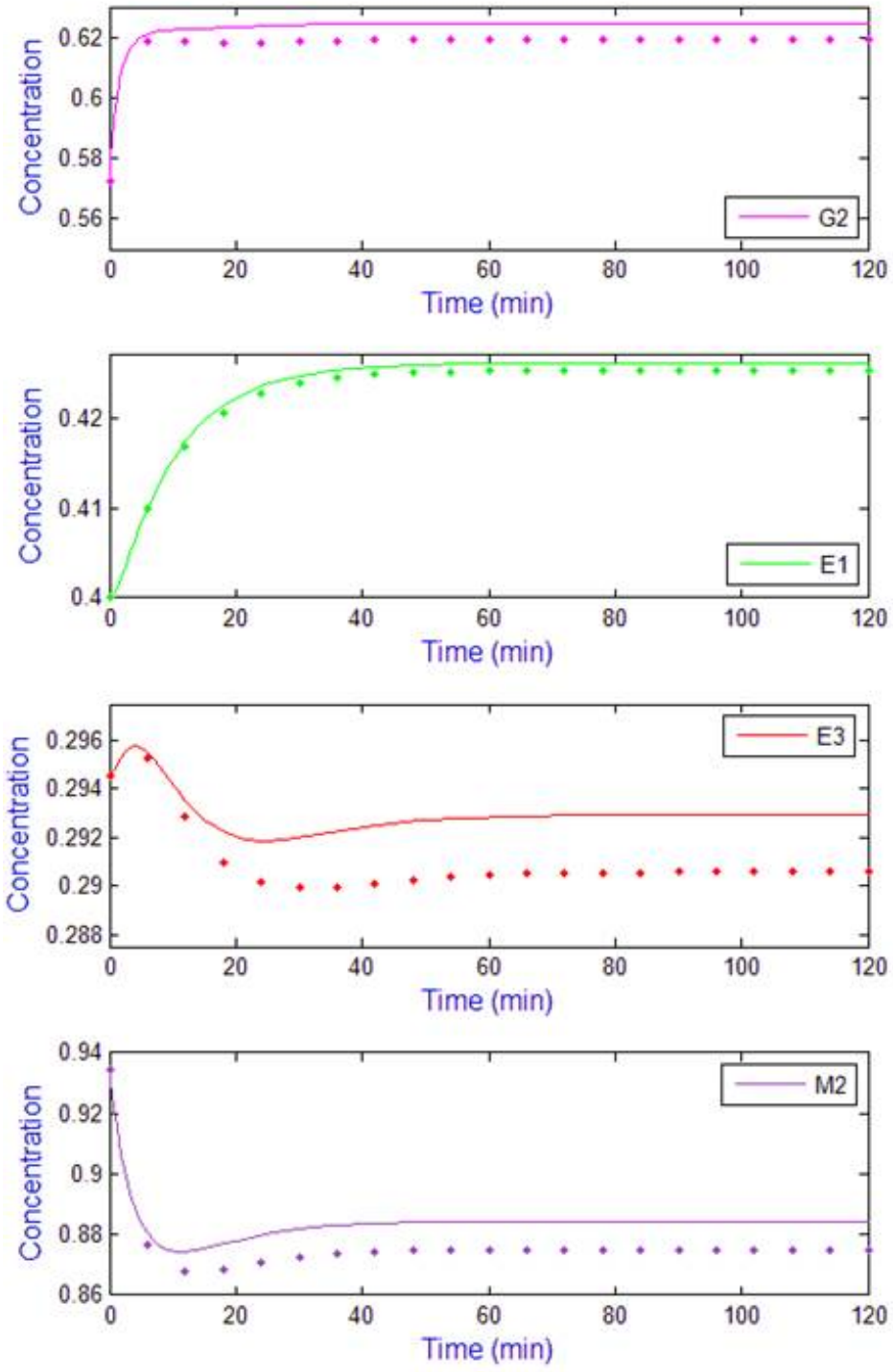



Fig. 4.22 State profiles of the modified CMSC in the high initials case

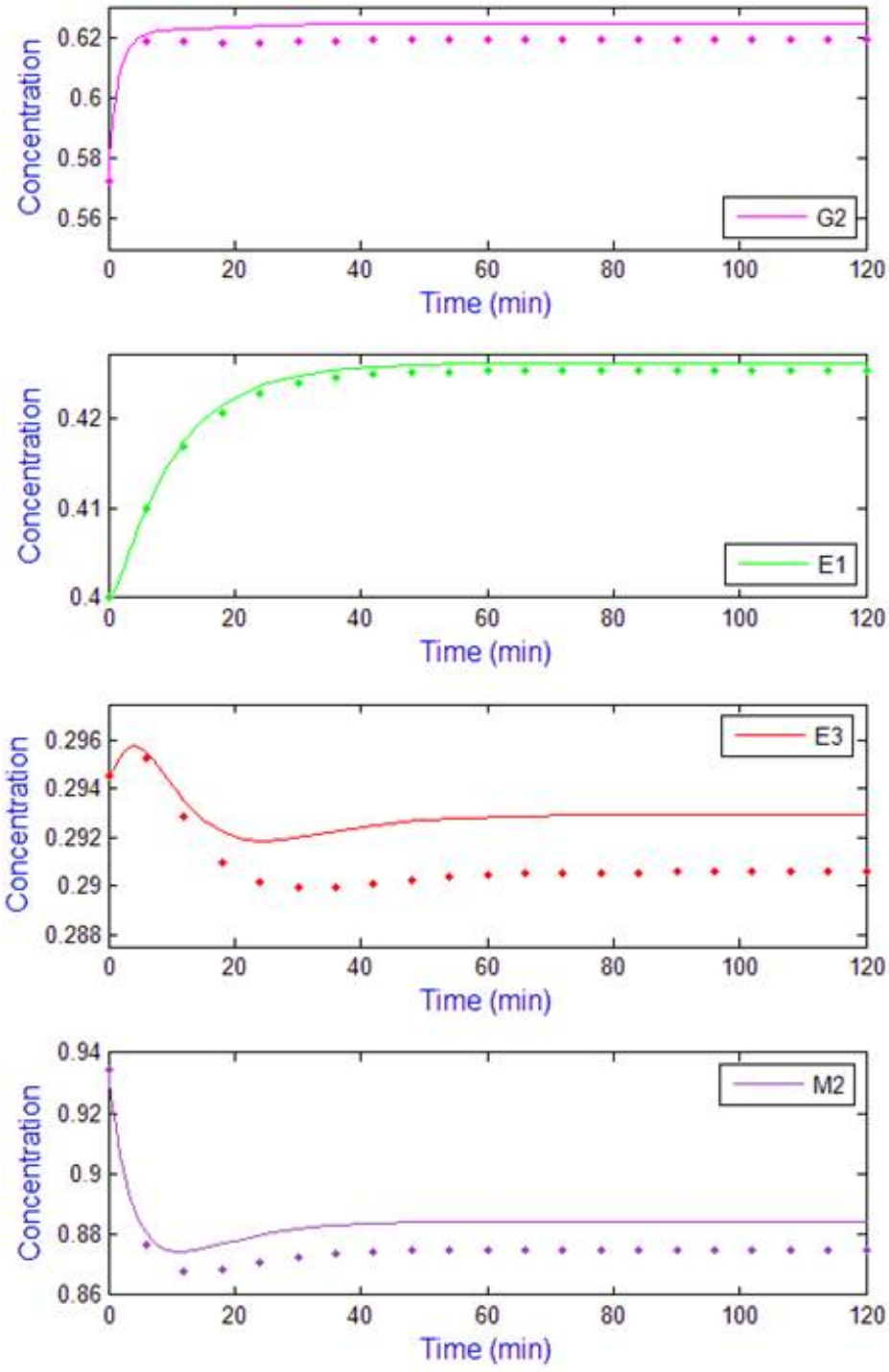



Fig. 4.23 State profiles of the modified CMSC in the high initials case (continued)

here to overcome this issue. The results of the two illustration examples show the efficiency of the modified method. In combination with the algorithm to parallelize the solution of the model equations, the proposed method has advantage of less CPU time taken to solve the DOPs.

# Chapter 5

# Identifiability analysis based on identification of parameter correlations

One of the challenging tasks in the grey-box mathematical modeling of nonlinear dynamic models is parameter estimation. A nonlinear dynamic model usually contains a large number of parameters among which there may exist implicit functional relations (meaning *parameter correlations*) that lead to non-identifiability problems. Although many approaches have been developed to address both structural and practical non-identifiability problems, very few studies have been made to systematically investigate parameter correlations.

In this Chapter a new approach that is able to identify both pairwise parameter correlations and higher order interrelationships among parameters in nonlinear dynamic models is presented. Based on the correlation information obtained in this way both structural and practical non-identifiability can be clarified. Moreover, it can be concluded from the correlation analysis that a minimum number of data sets, which corresponds to the maximum number of correlated parameters among the correlation groups, with different inputs for experimental design are needed to relieve the parameter correlations. The information of pairwise and higher order interrelationships among parameters in nonlinear dynamic models gives a deeper insight into the cause of non-identifiability problems. The result of this correlation analysis provides a neces-

sary condition for experimental design in order to acquire suitable measurement data for unique parameter estimation.

## 5.1 Introduction

In the grey-box modeling of nonlinear dynamic models, a question naturally arises, whether unknown parameters of the chosen model structure can be uniquely estimated from the given data? If not, what is the reason, due to the model structure itself or the experimental data? To answer these questions, modelers need to do the *identifiability analysis* step. An aim of identifiability analysis is to determine if the parameters of a model are identifiable or not, i.e., whether its parameters can be uniquely estimated. Therefore, identifiability analysis is a critical first step for parameter estimation due to the fact that a model is only valid and useful when all of its parameter can be uniquely estimated from the experimental data. In addition, it will be difficult for the optimization solvers to converge if the postulated model is non-identifiable.

Identifiability of a model can be classified into structural (or *a priori*) identifiability [Bellman and Åström, 1970] and practical (or *a posteriori*) identifiability [Cobelli and DiStefano, 1980]. Structural identifiability studies the model with *ideal continuous noise free* observations of the input and output variables [Walter and Pronzato, 1997]. Structural identifiability is a property of the model itself and this property depends on how the model is constructed. Structural identifiability should be checked before conducting experiments. Obviously, structural identifiability is necessary but not sufficient to affirm an accurate estimation of the model parameters from experimental data. In contrast, practical identifiability associates with *real sparse noisy* measurements and it thus depends on how the experiments are conducted. Practical identifiability in principle can be solved by means of suitable experimental design.

## 5.2 Definitions

A general nonlinear dynamic system can be described as follows:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}) \tag{5.1a}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{q}) \tag{5.1b}$$

$$\mathbf{x}(\mathbf{t_0}) = \mathbf{x_0}$$

where $\mathbf{x}(t) \in R^n$ is the state vector, $\mathbf{u}(t) \in R^m$ the control vector, and $\mathbf{y}(t) \in R^r$ the output vector, respectively. In this study, two different sets of parameters, i.e., $\mathbf{p} \in R^{NP}$ in the state equations and $\mathbf{q} \in R^{NQ}$ in the output equations, are considered. In most cases the number of parameters in the state equations is much larger than that in the output equations.

Let $\boldsymbol{\theta} = [\mathbf{p}, \mathbf{q}]^T, \boldsymbol{\theta} \in \boldsymbol{\Omega}$, where $\boldsymbol{\Omega} \in R^{N\boldsymbol{\theta}}, N\boldsymbol{\theta} = NP + NQ$, some essential definitions from [Miao et al., 2011] are expressed as follows:

**Definition 5.2.1 (Identifiability)**
*The dynamic system described by Eq.* (5.1) *is identifiable if the parameter* $\boldsymbol{\theta}$ *can be uniquely determined from the given system input* $\mathbf{u}(t)$ *and the measurable system output* $\mathbf{y}(t)$*; otherwise it is said to be unidentifiable.*

As a results, the parameter $\boldsymbol{\theta}$ is also said to be identifiable and unidentifiable, respectively.

**Definition 5.2.2 (Global identifiability)**
*A system structure is said to be globally identifiable if for any admissible input* $\mathbf{u}(t)$ *and any two parameters vector* $\boldsymbol{\theta}_1$ *and* $\boldsymbol{\theta}_2$ *in the parameter space* $\boldsymbol{\Omega}$*,* $\mathbf{y}(\mathbf{u}, \boldsymbol{\theta}_1) = \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}_2)$ *holds if and only if* $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$*.*

**Definition 5.2.3 (Local identifiability)**
*A system structure is said to be locally identifiable if for any* $\boldsymbol{\theta}$ *within an open neighborhood of some point* $\boldsymbol{\theta}_*$ *in the parameter space,* $\mathbf{y}(\mathbf{u}, \boldsymbol{\theta}_1) = \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}_2)$ *holds if and only if* $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$*.*

**Definition 5.2.4 (Local strong identifiability ($x_0$-identifiability))**
*For an admissible input* $\mathbf{u}(t)$ *in the time range of interest* $[t_0, t_1]$ *and a given initial state* $\mathbf{x}_0 = \mathbf{x}(t_0)$*, which is independent of* $\boldsymbol{\theta}$ *and not an equilibrium point, if there exists*

*an open set* $\mathbf{\Omega}^0$ *within the parameter space* $\mathbf{\Omega}$ *such that, for any two different parameter vectors* $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathbf{\Omega}^0$*, the solutions* $\mathbf{x}(t, \boldsymbol{\theta}, \mathbf{u})$ *exist on* $[t_0, t_1 + \varepsilon](t_0 < \varepsilon \leq t_1 - t_0)$ *for both* $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$*, and* $\mathbf{y}(\mathbf{u}(t), \mathbf{x}_0, \boldsymbol{\theta}_1) = \mathbf{y}(\mathbf{u}(t), \mathbf{x}_0, \boldsymbol{\theta}_2)$ *on* $[t_0, t_0 + \varepsilon]$*, the system structure is said to be locally strongly identifiable (*$\mathbf{x}_0$*-identifiability).*

**Definition 5.2.5 (Structural identifiability)**
*Let* $C_u^N[t_0, t_1]$ *denote the function space expanded by all input functions on* $[t_0, t_1]$ *which are differentiable up to the order N, and let M denote an open set of initial system states. The system structure is said to be structurally identifiable if there exist open and dense subsets* $M^0 \subset M, \mathbf{\Omega}^0 \subset \mathbf{\Omega}$ *and* $U^0 \subset C_u^N[t_0, t_1]$ *such that the system is locally strongly identifiable at* $\boldsymbol{\theta}$ *given* $\mathbf{u}$ *for any* $\mathbf{x}_0 \in M^0, \boldsymbol{\theta} \in \mathbf{\Omega}^0$ *and* $\mathbf{u} \in U^0$*.*

**Definition 5.2.6 (Algebraic identifiability)**
*Based on algebraic equations of system state, input, and output, if a meromorphic function*

$$\Phi = \Phi(\boldsymbol{\theta}, \mathbf{u}, \dot{\mathbf{u}}, \cdots, \mathbf{u}^{(\mathbf{k})}, \mathbf{y}, \dot{\mathbf{y}}, \cdots, \mathbf{y}^{(\mathbf{k})}), \Phi \in R^m,$$

*can be constructed after a finite number of steps of algebraic calculation or differentiation such that* $\Phi = 0$ *and* $\det \dfrac{\partial \Phi}{\partial \boldsymbol{\theta}} \neq 0$ *hold in the time range of interest* $[t_0, t_1]$ *for any* $(\boldsymbol{\theta}, \mathbf{x_0}, \mathbf{u})$ *in an open and dense subset of* $\mathbf{\Omega} \times M \times C_u^N[t_0, t_1]$*, where k is a positive integer,* $\dot{\mathbf{u}}, \ldots, \mathbf{u}^{(\mathbf{k})}$ *the derivatives of* $\mathbf{u}$*, and* $\dot{\mathbf{y}}, \ldots, \mathbf{y}^{(\mathbf{k})}$ *the derivatives of* $\mathbf{y}$*, the system structure is said to be algebraically identifiable.*

The Definition 5.2.6 is used directly in system identifiability analysis techniques called *implicit function theorem* in [Ljung and Glad, 1994b; Xia, 2003; Xia and Moog, 2003].

## 5.3 Structural identifiability analysis

Structural identifiability can be determined by *a priori* methods that use the perfect data set without noise and continuous in time. [Chappell et al., 1990; Chis et al., 2011b; Meshkat et al., 2009]. Methods for checking the structural identifiability of nonlinear models are among power series based approaches including Taylor series approach [Dochain et al., 1995; Petersen et al., 2003; Pohjanpalo, 1978] and generating series approach [Jayasankar et al., 2009; Walter and Lecourtier, 1982] with possible combination of identifiability tableaus [Balsa-Canto et al., 2010; Chis et al., 2011a], the similarity transformation approach [Vajda et al., 1989a], the differential algebra

based method [Bellu et al., 2007; Ljung and Glad, 1994b; Saccomani et al., 2010], the direct test method [Denis-Vidal et al., 2001; Walter et al., 2004], the implicit function theorem approach [Ljung and Glad, 1994b; Xia, 2003; Xia and Moog, 2003] and the profile likelihood approach using simulated data [Flassig et al., 2015; Hengl et al., 2007; Kreutz et al., 2012; Maiwald and Timmer, 2008; Raue et al., 2010, 2009; Schaber, 2012].

The Taylor series approach requires high order derivatives of the system outputs with respect to time by using the Taylor series expansion of the outputs in the vicinity of the initial state. The Taylor series coefficients are calculated to form a system of nonlinear algebraic equations in the parameters. The uniqueness of the solution of this resulting system guarantees the structural identifiability of the original system. This method is conceptually simple but the number of required derivatives is generally unknown. Moreover, the resulting system of algebraic equations may be too complicated to solve. Thus this method is not popularly used in practice [Chis, 2011; Miao et al., 2011].

The generating series approach uses the same concept as the Taylor series approach. This method also uses the expansion of the outputs of the postulated system with respect to inputs and time. The exhaustive summary that contains the coefficients of the output functions and Lie derivatives is then used to examine the structural identifiability of the original system as the same manner as in the Taylor series approach. Again, the generating series approach presents a challenge due to the unknownness of the minimum number of the required Lie derivatives and the solution of the resulting system of the algebraic equations [Chis, 2011; Miao et al., 2011]. Although the difficulty in handling the resulting system of algebraic equations can be partially solved by using the identifiability tableaus in [Balsa-Canto et al., 2010; Chis et al., 2011a], power series based approaches may be not able to assess the identifiability of the parameters for some particular cases [Chis, 2011]. The generating series approach has been applied in the GenSSI code [Chis et al., 2011a] to test the structural identifiability of biological models.

The similarity transformation approach, which is based on the local state isomorphism theorem, is only suitable to solve the identifiability problem for single-input single-output (SISO) models. For multi-input multi-output (MIMO) models, it has difficulties in solving partial differential equations and checking the prerequisite for the controllability and observability conditions [McLean and McAuley, 2012].

In the differential algebra based method, the non-observable differential states are eliminated to produce differential relations between inputs, outputs and parameters. The exhaustive summary then is obtained and can be solved by algebraic methods. The solution of the resulting algebraic equations can precisely give the identifiability information of the parameters. The disadvantage of this method is that it needs large computational efforts when dealing with complex models [Chis, 2011]. The DAISY program that uses the differential algebra based method has been built[Bellu et al., 2007; Saccomani et al., 2010] to support the modelers in biological and physiological areas.

The implicit function theorem approach tries to eliminates the unobservable states by computing the derivatives of the observable outputs with respect to time of a differential system, which depends on known inputs, outputs and parameters. The partial derivatives of these differential equations with respect to parameters then are computed to define an identification matrix. If this matrix is not singular then the original system is identifiable. This method also requires high-order derivatives and then the identification matrix may be very complicated to verify its singularity [Chis, 2011; Miao et al., 2011; Xia and Moog, 2003].

The direct test method uses directly the identifiability definition 5.2.2 or 5.2.3 to analytically or numerically assess the parameter identifiability. This method utilizes the sufficient condition of those definitions to check whether or not a system model is identifiable. If a general dynamical system is identifiable then one gets:

$$\mathbf{y}(\mathbf{u}, \boldsymbol{\theta}_1) = \mathbf{y}(\mathbf{u}, \boldsymbol{\theta}_2) \Rightarrow \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 \tag{5.2}$$

In [Denis-Vidal and Joly-Blanchard, 2000; Denis-Vidal et al., 2001], this method was applied to uncontrolled and autonomous system models, that is:

$$\mathbf{f}(\mathbf{x}, \boldsymbol{\theta}_1) = \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}_2) \Rightarrow \boldsymbol{\theta}_1 = \boldsymbol{\theta}_2 \tag{5.3}$$

This method is not suitable for large-scale problems [Miao et al., 2011].

Recently, Raue et al. [2009] proposed to use profile likelihood to detect non-identifiability for partially observable models. The parameter space is explored for each parameter by repeatedly fitting the model to a given data set, which then provides a likelihood-based confidence region for each parameter. Results from this method show that the number

of practically non-identifiable parameters will decrease when more data sets are used [Steiert et al., 2012]. The profile likelihood approach can also offer information on the correlated relations among the parameters [Bachmann et al., 2011; Hengl et al., 2007; Raue et al., 2009; Steiert et al., 2012]. The information on parameter correlations (e.g., correlated groups, correlated forms in a group, etc.,) is important for experimental design, so that a series of experimental runs with determined conditions can be carried out to acquire proper measurement data sets for improving the quality of parameter estimation. If the non-identifiability does not change for any data, these parameters are called structurally non-identifiable. On the contrary, if the non-identifiability can be remedied by data improvement, they are practically non-identifiable [Raue et al., 2009, 2011]. The disadvantage of the profile likelihood approach is that it cannot show the real type of the functional relation between correlated parameters. The MATLAB toolbox PottersWheel has been developed to utilize the profile likelihood approach in both structural and practical identifiability analysis [Hengl et al., 2007; Maiwald and Timmer, 2008; Raue et al., 2009].

In summary, in recent years, structural identifiability analysis has gained great interest in the field of chemical process and systems biology, e.g., see [Berthoumieux, 2012; Chis, 2011; McLean and McAuley, 2012; Miao et al., 2011; Raue et al., 2014] and references therein. Approaches for assessing global identifiability are usually difficult to implement and restricted to moderate dimensional systems. It can be concluded that further studies still need to be done in order to find new methods that can easily address the identifiability problem of complex dynamic models, especially mathematically to figure out the type of the functional relations among the parameters in a non-identifiability model.

## 5.4 Practical identifiability analysis

The *a posteriori* methods reveal practical identifiability properties based on results from fitting parameters to available data sets. In most previous studies, correlations are detected by analyzing the sensitivity matrix and the Fisher information matrix (FIM) [Ashyraliyev et al., 2009; Chu et al., 2007; Cintrón-Arias et al., 2009; McLean and McAuley, 2012; Vajda et al., 1989b; Yao et al., 2003], from which local confidence regions of parameter solutions can be obtained. Sensitivity analysis is well suitable

to linear models but will have limitations for highly nonlinear models [Dobre et al., 2012; Raue et al., 2011]. The code DecTrees has been written in $C++$ to assess the practical identifiability of biochemical models [Elsheikh, 2013].

In general, the quality of estimation results depends on the quality of data acquisition, the quality of the fitting method, and the quality of the model. Good experiment design can lead to highly informative data which will enhance the accuracy and identifiability of model parameters. Therefore, the task of parameter estimation demands an interactive endeavor of experiment and computation [Jacquez, 1998; Kreutz and Timmer, 2009].

## 5.5 A new approach to detect parameter correlations

Correlated parameters are non-identifiable. Very few studies have been jet made to investigate parameter correlations. Yao et al. [2003] used the rank of the sensitivity matrix to determine the number of estimable parameters. However, the subsets of correlated parameters cannot be identified based on this result. Chu and Hahn [2007] proposed to check the parallel columns in the sensitivity matrix to determine parameter subsets in which the parameters are pairwise correlated. Quaiser and Mönnigmann [2009] proposed a method to rank the parameters from least estimable to most estimable. These methods, however, cannot identify parameter groups in which more than two parameters are correlated together but not in pairwise, i.e., the corresponding columns in the sensitivity matrix are linearly dependent but not parallel. Such correlations are called higher order interrelationships among parameters [McLean and McAuley, 2012].

In system biology, the correlation phenomenon can be explained by the biological background, e.g., genes or proteins which are expressed in a correlated manner, correlations of expression levels between cells. As a consequence, certain regions in the parameter space correspond to good model predictions. It means that the residual value (quadratic error) remains low even if the parameters vary in certain regions. By testing 17 biological models, Gutenkunst et al. [2007] concluded that collective fits to large amounts of ideal time-series data lead to the fact that some eigenvectors are

orders of magnitudes better constrained than others.

### 5.5.1 Identification of parameter correlations

The equation of the state sensitivity matrix can be derived by taking the first order partial derivative of Eq. (5.1a) with respect to parameters $\mathbf{p}$:

$$\dot{\mathbf{S}} = \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)\mathbf{S} + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right) \tag{5.4}$$

where $\mathbf{S} = \dfrac{\partial \mathbf{x}}{\partial \mathbf{p}}$ is the state sensitivity matrix. By solving this equation (see Section A.1 of Appendix A for details) the state sensitivity matrix can be written as:

$$\mathbf{S} = \int_{t_0}^{t} \left(\mathbf{V}(\tau)\left(\frac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)\right) d\tau \tag{5.5}$$

where $\mathbf{V}(\tau)$ is a matrix computed at time $\tau$ . It means that $\mathbf{S}$ has a linear integral relation with the matrix $\left(\dfrac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)$ from $t_0$ to $t$ . If at any time $\left(\dfrac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)$ has the same linearly dependent columns, the corresponding columns in $\mathbf{S}$ will also be linearly dependent, i.e., the corresponding parameters are correlated. Therefore, parameter correlations can be identified by checking the linear dependences of the column in the matrix $\left(\dfrac{\partial \mathbf{f}}{\partial \mathbf{p}}\right)$ which is composed of the first order partial derivatives of the right-hand side of the ODEs. Based on Eq. (5.1b), the output sensitivity matrices are, respectively, given by

$$\frac{\partial \mathbf{y}}{\partial \mathbf{p}} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\frac{\partial \mathbf{x}}{\partial \mathbf{p}} = -\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right)^{-1}\frac{\partial \mathbf{f}}{\partial \mathbf{p}} \tag{5.6a}$$

$$\frac{\partial \mathbf{y}}{\partial \mathbf{q}} = \frac{\partial \mathbf{h}}{\partial \mathbf{q}} \tag{5.6b}$$

To ensure unique estimation of the parameters (i.e., all parameters to be identifiable), based on the measured data of $\mathbf{y}$, it is necessary that the columns in the output sensitivity matrices $\dfrac{\partial \mathbf{y}}{\partial \mathbf{p}}$, $\dfrac{\partial \mathbf{y}}{\partial \mathbf{q}}$ are linearly independent. From Eq. (5.6b), relations of the columns in $\dfrac{\partial \mathbf{y}}{\partial \mathbf{q}}$ can be easily detected. The difficulty comes from Eq. (5.6a), since

the sensitivity functions in $\frac{\partial \mathbf{y}}{\partial \mathbf{p}}$ cannot be analytically expressed. However, from Eq. (5.6a), the output sensitivity matrix is a linear transformation of $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$ . Consequently, there will be linearly dependent columns in $\frac{\partial \mathbf{y}}{\partial \mathbf{p}}$, if there are linearly dependent columns in $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$. It means the necessary condition for unique estimation of $\mathbf{p}$ is that, at least, the matrix $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$ must have a full rank. Based on Eq. (5.1a), $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$ is expressed as vectors of the first order partial derivative functions:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{p}} = \left[ \frac{\partial \mathbf{f}}{\partial p_1}, \frac{\partial \mathbf{f}}{\partial p_2}, \cdots, \frac{\partial \mathbf{f}}{\partial p_{NP}} \right] \tag{5.7}$$

Now relations between the partial derivative functions in Eq. (5.7) are analyzed. If there is no correlation among the parameters, the columns in Eq. (5.7) will be linearly independent, i.e., if

$$\alpha_1 \frac{\partial \mathbf{f}}{\partial p_1} + \alpha_2 \frac{\partial \mathbf{f}}{\partial p_2} + \cdots + \alpha_{NP} \frac{\partial \mathbf{f}}{\partial p_{NP}} = 0 \tag{5.8}$$

there must be $\alpha_i = 0,\ i = 1, \ldots, NP$. Otherwise, there will be some groups of vectors in $\frac{\partial \mathbf{f}}{\partial \mathbf{p}}$ which lead to the following cases of linear dependences due to parameter correlations. Let us consider a subset of the parameters with $k$ correlated parameters denoted as $\mathbf{p}_{sub} = [p_{s+1}, p_{s+2}, \cdots, p_{s+k}]^T$ with $s + k \leq NP$.

**Case 1**

$$\alpha_1 \frac{\partial \mathbf{f}}{\partial p_{s+1}} = \alpha_2 \frac{\partial \mathbf{f}}{\partial p_{s+2}} = \cdots = \alpha_k \frac{\partial \mathbf{f}}{\partial p_{s+k}} \tag{5.9}$$

where $\alpha_i \neq 0,\ i = 1, \ldots, k$. Notice that the coefficient $\alpha_i$ may be a function of the parameters (i.e., $\alpha_i(\mathbf{p})$) and/or of control inputs (i.e., $\alpha_i(\mathbf{u}(t), \mathbf{p})$). It should be also noted that the control inputs $\mathbf{u}(t)$ are considered as constants in these coefficients, since they will be specified in experimental design. The linear dependences described by Eq. (5.9) lead to pairwise correlations among the $k$ parameters, i.e., any pair of the parameters in $\mathbf{p}_{sub}$ are correlated. Moreover, the correlations mean a functional relationship between the parameters, i.e., the relationship between the parameters can

be expressed by an algebraic equation:

$$\phi_{sub}(\gamma(p_{s+1}, p_{s+2}, \cdots, p_{s+k})) = 0 \tag{5.10}$$

where $\gamma(p_{s+1}, p_{s+2}, \cdots, p_{s+k})$ denotes a function of the parameters with one set of pairwise correlated parameters. The parameters in this function are compensated each other in an algebraic relationship, e.g., $\gamma(p_{s+1}+p_{s+2}+\cdots+p_{s+k})$ or $\gamma(p_{s+1}p_{s+2}\cdots p_{s+k})$. Eq. (5.10) describes the functional relationship between the parameters, e.g., $\phi_{sub}(\gamma(\cdot)) = 1+\gamma(\cdot)-(\gamma(\cdot))^2 = 0$. Due to the complexity of biological models, an explicit expression of this equation is not available in most cases. If the coefficients in Eq. (5.9) are functions of only the parameters, i.e., $\alpha_i(\mathbf{p})$, the parameters are *structurally* non-identifiable. In this case, the correlation relations in Eq. (5.9) will remain unchanged by specifying *any* values of control inputs. It means that the non-identifiability cannot be remedied through experimental design. If the coefficients in Eq. (5.9) are functions of both the parameters and control inputs, i.e., $\alpha_i(\mathbf{u}, \mathbf{p})$, the parameters are practically non-identifiable. Different values for $\mathbf{u}$ can be specified which lead to different $\alpha_i(\mathbf{u}, \mathbf{p})$, such that Eq. (5.9) will not hold and therefore the parameter correlations will be relieved. Since $k$ parameters are correlated, $k$ different values of the control inputs $\mathbf{u}^{(j)}, (j = 1, \ldots, k)$ are required, such that the matrix:

$$\frac{\partial \mathbf{f}}{\partial \mathbf{p}_{sub}} = \begin{bmatrix} \frac{\partial \mathbf{f}^{(1)}}{\partial p_{s+1}} & \frac{\partial \mathbf{f}^{(1)}}{\partial p_{s+2}} & \cdots & \frac{\partial \mathbf{f}^{(1)}}{\partial p_{s+k}} \\ \frac{\partial \mathbf{f}^{(2)}}{\partial p_{s+1}} & \frac{\partial \mathbf{f}^{(2)}}{\partial p_{s+2}} & \cdots & \frac{\partial \mathbf{f}^{(2)}}{\partial p_{s+k}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial \mathbf{f}^{(k)}}{\partial p_{s+1}} & \frac{\partial \mathbf{f}^{(k)}}{\partial p_{s+2}} & \cdots & \frac{\partial \mathbf{f}^{(k)}}{\partial p_{s+k}} \end{bmatrix} \tag{5.11}$$

has a full rank. Notice that the columns in Eq. (5.11) are only linearly dependent for the same input, but the columns of the whole matrix are linearly independent. In this way, the non-identifiability is remedied. Moreover, a suggestion for experimental design is provided for the specification of $\mathbf{u}^{(j)}, (j = 1, \ldots, k)$ to obtain k distinct data sets which will be used for parameter estimation.

If all state variables are measurable, according to Eq. (5.5), this subset of parameters can be uniquely estimated based on the $k$ data sets. If the outputs $\mathbf{y}$ are measured and used for the parameter estimation, it can be concluded from Eq. (5.6a) that at least $k$ data sets are required for unique parameter estimation.

**Case 2**

$$\alpha_1 \frac{\partial \mathbf{f}}{\partial p_{s+1}} = \cdots = \alpha_{s+l_1} \frac{\partial \mathbf{f}}{\partial p_{s+l_1}}, \cdots, \alpha_{s+l_{d-1}+1} \frac{\partial \mathbf{f}}{\partial p_{s+l_{d-1}+1}} = \cdots = \alpha_{s+k} \frac{\partial \mathbf{f}}{\partial p_{s+k}} \tag{5.12a}$$

$$\text{and} \quad \alpha_{s+k+1} \frac{\partial \mathbf{f}}{\partial p_{s+1}} + \alpha_{s+k+2} \frac{\partial \mathbf{f}}{\partial p_{s+l_1+1}} + \cdots + \alpha_{s+k+d} \frac{\partial \mathbf{f}}{\partial p_{s+l_{d-1}+1}} = 0$$

$$\text{and} \quad \alpha_{s+k+1} \frac{\partial \mathbf{f}}{\partial p_{s+1}} + \alpha_{s+k+2} \frac{\partial \mathbf{f}}{\partial p_{s+l_1+1}} + \cdots + \alpha_{s+k+d} \frac{\partial \mathbf{f}}{\partial p_{s+l_{d-1}+1}} = 0 \tag{5.12b}$$

where $\alpha_i \neq 0,\ i = 1, \ldots, s+k+d$. Similarly, the coefficients may be functions of the parameters and/or of the control inputs. In this case, there are $d$ sets of pairwise correlated parameters (Eq. (5.12a)). A set is not correlated with another set, but all sets are correlated together (Eq. (5.12b)). The functional relationship in this case can be expressed by:

$$\phi_{sub}(\gamma_1(p_{s+1}, \cdots, p_{s+l_1}), \cdots, \gamma_d(p_{s+l_{d-1}+1}, \cdots, p_{s+k})) = 0 \tag{5.13}$$

Based upon on Eq. (5.12a), the group with the maximum number of parameters $\max(l_1, l_2, \cdots, l_d)$ is of importance for data acquisition. From Eq. (5.12b), in the case of practical non-identifiability, data for at least d different inputs is required. The combination of Eqs. (5.12a) and (5.12b) leads to the conclusion that one needs a number of $\max(l_1, l_2, \cdots, l_d, d)$ data sets with different inputs to eliminate parameter correlations in this case.

**Case 3**

$$\alpha_1 \frac{\partial \mathbf{f}}{\partial p_{s+1}} + \alpha_2 \frac{\partial \mathbf{f}}{\partial p_{s+2}} + \alpha_3 \frac{\partial \mathbf{f}}{\partial p_{s+3}} + \cdots + \alpha_k \frac{\partial \mathbf{f}}{\partial p_{s+k}} = 0 \tag{5.14}$$

where $\alpha_i \neq 0,\ i = 1, \ldots, k$. In this case, all k parameters are not pairwise correlated but they are correlated together in one group. The correlation equation in this case is expressed by:

$$\phi_s(p_{s+1}, p_{s+1}, \cdots, p_{s+k}) = 0 \tag{5.15}$$

which means there is no set of correlated parameters in this case. The approach described above is able to identify pairwise and higher order parameter correlations in

the state equations (Eq. (5.1a)). In the same way, correlations among parameters in q in the output equations (Eq. (5.1b)) can also be detected based on the first order partial derivative functions in Eq. (5.6b).

From the results of this correlation analysis, the maximum number of correlated parameters of the correlation groups can be detected. This corresponds to the minimum number of data sets required for unique estimation of all parameters in the model. Furthermore, it is noted that the initial state of the model has no impact on the parameter correlations, which means that any initial state can be used for the experimental runs for the data acquisition.

For complex models, the correlation equations (Eqs. (5.10),(5.13),(5.15)) cannot be analytically expressed. A numerical method has to be used to illustrate the relationships of correlated parameters of a given model, which is discussed in the next Section.

### 5.5.2 Interpretation of parameter correlations

Here an interpretation of parameter correlations in a biological model is given. Geometrically, for $NP$ parameters, i.e., $\mathbf{p} = [p_1, p_2, \cdots, p_{NP}]^T$, the estimation task can be considered as searching for true parameter values $\mathbf{p}^*$ in the $NP$-dimensional parameter space. To do this, one needs $NP$ linearly independent surfaces in the parameter space which should pass through $\mathbf{p}^*$. Mathematically, such surfaces are described by linearly independent equations with the unknown parameters. Such equations are defined based on the results of fitting model equations (5.1) to a data set ($j$) by minimizing the following cost function:

$$\min_{\mathbf{p}} \; F^{(j)}(\mathbf{p}) = \sum_{l=1}^{M} \sum_{i=1}^{n} w_{i,l} (x_{i,l}^{(j)}(\mathbf{p}) - \hat{x}_{i,l}^{(j)})^2 \tag{5.16}$$

where $M$ is the number of sampling points, $n$ is the number of state variables and denotes the measured data while $\mathbf{x}(\mathbf{p})$ the state variables predicted by the model. $w_{i,l}$ are weighting factors. The fitting results will depend on the data set resulted from the control inputs $\mathbf{u}^{(\mathbf{j})}$, the values of $w_{i,l}$, and the noise level of the measured data. For a geometric interpretation of parameter correlations, idealized measurement data is assumed to be used, i.e., data without any noises. Based on this assumption, the

residual function (5.16) should be zero, when the true parameter set $\mathbf{x}(\mathbf{p})$ is applied, i.e.,

$$x_{i,l}^{(j)}(\mathbf{p}^*) - \hat{x}_{i,l}^{(j)} = 0, \quad i = 1, \ldots, n, \quad l = 1, \ldots, M \tag{5.17}$$

It is noted that Eq. (5.17) is true for any noise-free data set employed for the fitting and independent of $w_{i,q}$. Now a zero residual equation (ZRE) can be defined as

$$\varphi_{i,l}^{(j)}(\mathbf{p}) = x_{i,l}^{(j)}(\mathbf{p}) - \hat{x}_{i,l}^{(j)} = 0 \tag{5.18}$$

This equation contains the parameters as unknowns and corresponds to a zero residual surface passing through the true parameter point $\mathbf{p}^*$. It means that a zero residual surface is built by parameter values which lead to a zero residual value. This suggests that one can find $\mathbf{p}^*$ by solving $NP$ linearly independent ZREs. From the first order Taylor expansion of Eq. (5.18), the linear dependences of ZREs can be detected by the columns in the following matrix

$$\frac{\partial \mathbf{x}^{(j)}}{\partial \mathbf{p}} = \left[\frac{\partial \mathbf{x}^{(j)}}{\partial p_1}, \frac{\partial \mathbf{x}^{(j)}}{\partial p_2}, \cdots, \frac{\partial \mathbf{x}^{(j)}}{\partial p_{NP}}\right] \tag{5.19}$$

where $\mathbf{x}^{(j)} = \left[x_{1,1}^{(j)}, x_{1,2}^{(j)}, \cdots, x_{1,M}^{(j)}, \cdots, x_{n,1}^{(j)}, x_{n,2}^{(j)}, \cdots, x_{n,M}^{(j)}\right]^T$. Eq. (5.19) is exactly the state sensitivity matrix calculated by fitting to the given data set $(j)$. This means, under the idealized data assumption, a zero residual value delivered after the fitting is associated to surfaces passing through the true parameter point. When there are no parameter correlations, the number of linearly independent ZREs will be greater than $NP$ and thus the true parameter point can be found by fitting the current data set.

If there are parameter correlations, the fitting will lead to zero residual surfaces in the subspace of the correlated parameters. For instance, for a group of $k$ correlated parameters, the zero residual surfaces (Eq. (5.18)) will be reduced to a single ZRE represented by Eq. (5.10), Eq. (5.13), or Eq. (5.15). Therefore, in the case of *practical* non-identifiability, $k$ data sets are needed to generate $k$ linearly independent ZREs so that the $k$ parameters can be uniquely estimated. In the case of *structural* non-identifiability, the correlated relations are independent of data sets. It means fitting different data sets will lead to the same ZRE and thus the same surfaces in the parameter subspace. If the measured data are with noises, the fitting results will lead

to a nonzero residual value and nonzero residual surfaces, i.e.,

$$\varphi_{i,l}^{(j)}(\mathbf{p}) = x_{i,l}^{(j)}(\mathbf{p}) - \hat{x}_{i,l}^{(j)} = \varepsilon_{i,l} \tag{5.20}$$

where $\varepsilon_{i,l} \neq 0$. Thus the nonzero residual surfaces will not pass through the true parameter point. However, based on Eq. (5.19) and Eq. (5.20) the first order partial derivatives remain unchanged. It means that parameter correlations do not depend on the quality of the measured data. Moreover, it can be seen from Eq. (5.18) and Eq. (5.20) that the zero residual surfaces and the nonzero residual surfaces will be parallel in the subspace of the correlated parameters.

## 5.6 Case studies

### 5.6.1 A generic branched pathway as S-system

The first model is a generic branched pathway [Gonzalez et al., 2006; Jia et al., 2011; Voit and Almeida, 2004] with 4 state variables and 18 parameters described as following ODEs:

$$\dot{x}_1 = p_1 x_3^{p_2} x_5^{p_3} - p_4 x_1^{p_5}, \qquad x_1(0) = a_1 \tag{5.21a}$$
$$\dot{x}_2 = p_6 x_1^{p_7} - p_8 x_2^{p_9}, \qquad x_2(0) = a_2 \tag{5.21b}$$
$$\dot{x}_3 = p_{10} x_2^{p_{11}} - p_{12} x_3^{p_{13}} x_4^{p_{14}}, \qquad x_3(0) = a_3 \tag{5.21c}$$
$$\dot{x}_4 = p_{15} x_1^{p_{16}} - p_{17} x_4^{p_{18}}, \qquad x_4(0) = a_4 \tag{5.21d}$$
$$x_5 = a_5 \tag{5.21e}$$

By defining the values $a = [a_1, a_2, a_3, a_4, a_5]^T$ (see the footnote of Table 5.1), two datasets are generated in silico for 60 noise-free sampling points. For fitting the model random values for all 18 parameters are used to initialize the computation and all weights in (5.16) are set to 1. The results are taken by a threshold of the total residual value in the order of $10^{-5}$ (Table 5.1). Fig. 5.1a shows the angles between the columns of the sensitivity matrix after fitting to the $1^{st}$ dataset, indicating a

correlation between $p_1$ and $p_3$ by red dashed lines. The other angles shown in Fig. 5.1 not equal to zero mean that the other parameters are uncorrelated. Thus (5.18) reduces to a single ZRE with $p_1$ and $p_3$ as unknowns. The parameter values identified based on the 1st dataset are given in Table 5.1 ($P^{(1)}$), indicating large discrepancies of $p_1$ and $p_3$ to their true values. As expected, the uncorrelated parameters are found almost at their true values.

To plot the correlated relation between $p_1$ and $p_3$, two zero residual surfaces are constructed by fitting the model to the two datasets separately. Repeated fittings are made with fixed values of $p_1$. As shown in Fig. 5.2, as expected, the zero residual surfaces cross the solution point in the parameter space. Since the maximum number of correlated parameters is 2, based on the our approach, 2 datasets are needed to find the parameter solution. The $5^{th}$ column $P^{(1)+(2)}$ in Table 5.1 gives the parameter results by fitting the model to the two datasets together, indicating that $p_1$ and $p_3$ are now correctly identified. And the other parameter values in $P^{(1)+(2)}$ are improved compared with those based on the single dataset ($P^{(1)}$). Fig. 5.1b (red dashed line, lower panel) shows that the angle between the two columns corresponding to $p_1$ and $p_3$ is raised to about 8 degree, i.e., the correlation of $p_1$ and $p_3$ is remedied by fitting to the two datasets together.

The computed results show that $p_1$ and $p_3$ are practically non-identifiable, since different values of a5 in the two datasets are used (see footnote of Table 5.1). In fact, the non-identifiability of $p_1$ and $p_3$ can be easily figured out in the first equation in Eq. (5.18) from which the effect of experimental design can be elucidated. According to Eqs. (5.21) the first function to be partially derived are:

$$f_1 = p_1 x_3^{p_2} x_5^{p_3} - p_4 x_1^{p_5} \tag{5.22}$$

Take the first partially derivative of $f_1$ in Eq. 5.22 with respect to $p_1$ and $p_3$, one gets:

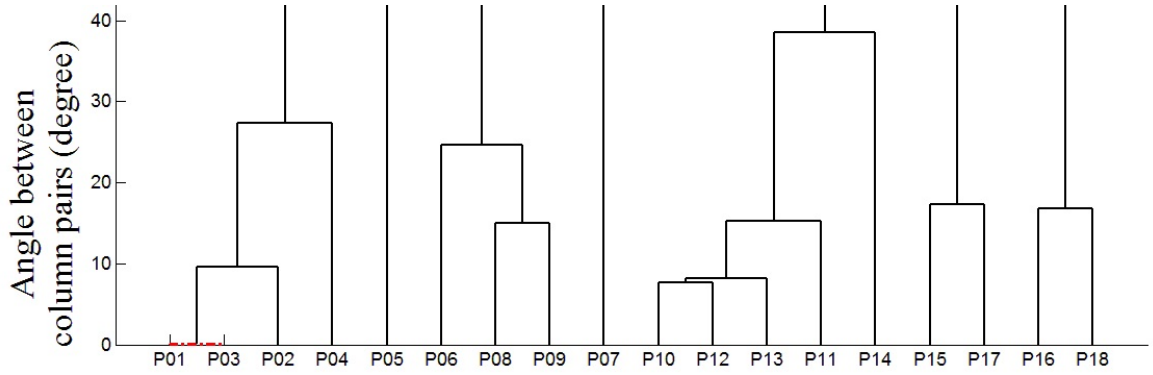
$$\frac{\partial f_1}{\partial p_1} = {x_3}^{p_2} {x_5}^{p_3} \tag{5.23a}$$

$$\frac{\partial f_1}{\partial p_3} = p_1 {x_3}^{p_2} {x_5}^{p_3} ln(x_5) \tag{5.23b}$$

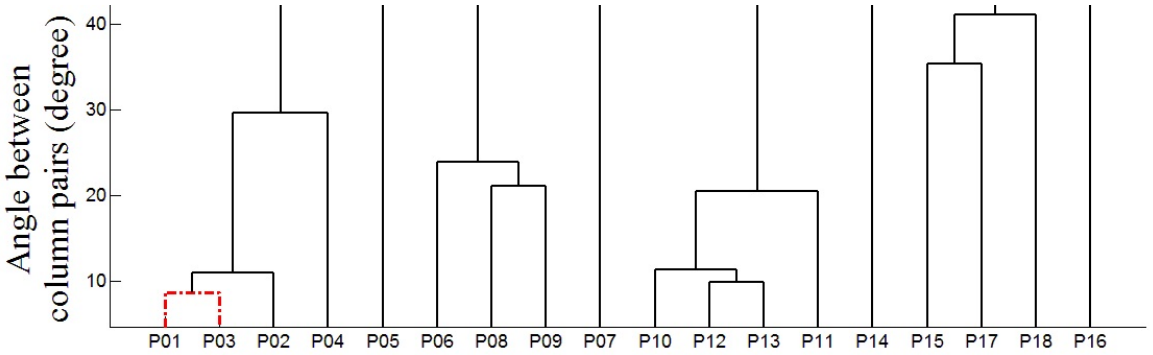
From Eq. (5.23) and (5.21e), it is easy to derive:

$$\frac{\partial f_1}{\partial p_3} = ln(a_5)p_1\left(\frac{\partial f_1}{\partial p_{31}}\right) \tag{5.24}$$

When the concentration of the $5^{th}$ state $a_5$ is not changed from different experimental runs, fitting to the corresponding datasets will lead to a same ZRE. Then $p_1$ and $p_3$ are structurally non-identifiable. If, due to experimental design, $a_5$ is different in different experiments, as shown in Fig. 5.2, they are practically non-identifiable and thus can be remedied.




(a) results from fitting to the $1^{st}$ dataset




(b) results from fitting to the $1^{st}$ and $2^{nd}$ datasets together

Fig. 5.1 Dendrogram of the generic branched pathway

To see the impact of measurement errors on the parameter values, the two datasets are regenerated by introducing 3% Gaussian distributed noises to the simulated data and the computations are carried out once again. The fitting results show residual values in the order of $10^{-1}$ due to the 3% noise level (Table 5.1). It is shown in Table 5.1 ($P^{(1)}$(w) and $P^{(1)+(2)}$(w)) the parameter values identified based on datasets with noises are less precise than those based on datasets without noises ($P^{(1)}$ and $P^{(1)+(2)}$).

Table 5.1 Parameter values of model Eqs. (5.21) based on one and two datasets

| No. | $\mathbf{P}^*$ | $P^{(1)}$ | $P^{(1)}$(w) | $P^{(1)+(2)}$ | $P^{(1)+(2)}$(w) |
|---|---|---|---|---|---|
| 1 (G1) | 20 | 5.84348 | 5.7995 | 19.9941 | 20.1166 |
| 2 | -0.8 | -0.80119 | -0.84939 | -0.80163 | -0.79172 |
| 3 (G1) | 1 | -1.41006 | -1.38822 | 1.0018 | 0.99677 |
| 4 | 10 | 10.0014 | 9.66783 | 9.98661 | 10.1586 |
| 5 | 0.5 | 0.50048 | 0.51568 | 0.50111 | 0.49264 |
| 6 | 8 | 7.99022 | 7.97423 | 7.9984 | 8.0104 |
| 7 | 0.5 | 0.50081 | 0.50077 | 0.5004 | 0.49898 |
| 8 | 3 | 2.99164 | 2.95063 | 2.99661 | 2.99064 |
| 9 | 0.75 | 0.75122 | 0.75425 | 0.75066 | 0.74893 |
| 10 | 3 | 2.99373 | 2.14109 | 3.00075 | 2.15021 |
| 11 | 0.75 | 0.75123 | 0.91648 | 0.74988 | 0.91182 |
| 12 | 5 | 4.99328 | 4.16577 | 5.00094 | 3.96944 |
| 13 | 0.5 | 0.50108 | 0.5689 | 0.49994 | 0.61316 |
| 14 | 0.2 | 0.20042 | 0.2307 | 0.2 | 0.2331 |
| 15 | 2 | 1.99914 | 1.49493 | 2.00129 | 1.94233 |
| 16 | 0.5 | 0.50026 | 0.42521 | 0.49983 | 0.54046 |
| 17 | 6 | 5.99908 | 5.26185 | 6.00172 | 5.89148 |
| 18 | 0.8 | 0.80027 | 0.794 | 0.79984 | 0.81321 |
| Residual | | 1.23E-05 | 3.59E-01 | 1.87E-05 | 6.53E-01 |

$P^*$ are the nominal (true) values, $P^{(1)}$ the values based on the $1^{st}$ data set, $P^{(1)(2)}$ based on the $1^{st}$, $2^{nd}$ data sets together, respectively. (w) means results from 3% noises on the data. The two datasets are generated with $a(1) = [5.6, 3.1, 2.9, 3.1, 0.6]^T$ and $a(2) = [2.9, 3.1, 3.1, 5.6, 0.5]^T$. Correlated parameter group (G1) is highlighted separately.

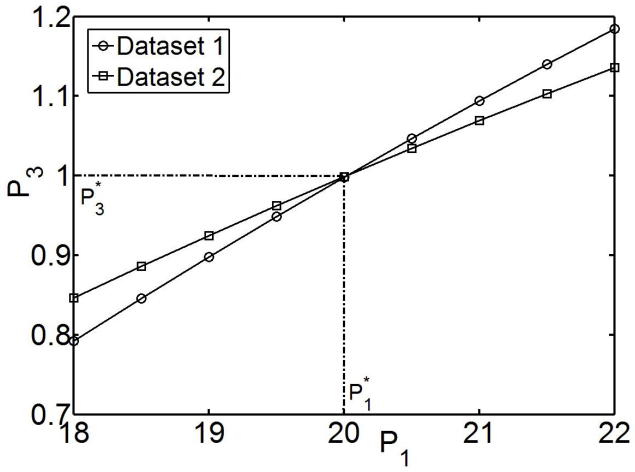



Fig. 5.2 Correlated relations between $p_1$ and $p_3$

The reason is that the parameters are on a contour with nonzero residual values.

The non-identifiability issue was not investigated in previous studies on this pathway model. Six noise-free datasets were used in [Voit and Almeida, 2004] to fit the parameters by using neural network techniques, leading to more imprecise parameter values compared with our results based on two datasets, as given in Table 5.1. Using one noise-free dataset, simulated annealing was employed by Gonzalez *et al.* [Gonzalez et al., 2006] to fit the parameters of this model. Although the total residual value was small (about $10^{-5}$), their results also showed relatively larger deviations to the true parameter values in comparison to those shown in Table 5.1. Very likely, these deviations are due to the inherent non-identifiability of the model. Also large deviations were reported in [Jia et al., 2011] by a two-phase decoupling method.

### 5.6.2 A three-step pathway model

A three-step pathway is considered here consisting of 8 nonlinear ordinary differential equations (ODEs) containing 8 metabolic concentrations (state variables) and 36 parameters [Mendes, 2001; Moles et al., 2003; Rodriguez-Fernandez et al., 2006], as given in Eq. (5.25). The $P$ and $S$ values in the ODEs are considered as two control inputs specified by experimental design. No output equations were considered for this

model in the previous studies.

$$\dot{x}_1 = \frac{p_1}{1 + \left(\frac{P}{p_2}\right)^{p_3} + \left(\frac{p_4}{S}\right)^{p_5}} - p_6 x_1 \tag{5.25a}$$

$$\dot{x}_2 = \frac{p_7}{1 + \left(\frac{P}{p_8}\right)^{p_9} + \left(\frac{p_{10}}{x_7}\right)^{p_{11}}} - p_{12} x_2 \tag{5.25b}$$

$$\dot{x}_3 = \frac{p_{13}}{1 + \left(\frac{P}{p_{14}}\right)^{p_{15}} + \left(\frac{p_{16}}{x_8}\right)^{p_{17}}} - p_{18} x_3 \tag{5.25c}$$

$$\dot{x}_4 = \frac{p_{19} x_1}{p_{20} + x_1} - p_{21} x_4 \tag{5.25d}$$

$$\dot{x}_5 = \frac{p_{22} x_2}{p_{23} + x_2} - p_{24} x_5 \tag{5.25e}$$

$$\dot{x}_6 = \frac{p_{25} x_3}{p_{26} + x_3} - p_{27} x_6 \tag{5.25f}$$

$$\dot{x}_7 = \frac{p_{28} x_4 (S - x_7)}{p_{29}\left(1 + \frac{S}{p_{29}} + \frac{x_7}{p_{30}}\right)} - \frac{p_{31} x_5 (x_7 - x_8)}{p_{32}\left(1 + \frac{x_7}{p_{32}} + \frac{x_8}{p_{33}}\right)} \tag{5.25g}$$

$$\dot{x}_8 = \frac{p_{31} x_5 (x_7 - x_8)}{p_{32}\left(1 + \frac{x_7}{p_{32}} + \frac{x_8}{p_{33}}\right)} - \frac{p_{34} x_6 (x_8 - P)}{p_{35}\left(1 + \frac{x_8}{p_{35}} + \frac{P}{p_{36}}\right)} \tag{5.25h}$$

This pathway model was studied by Moles et al. [2003] using 16 noise-free data sets and Rodriguez-Fernandez et al. [2006] using 16 both noise-free and noisy data sets, respectively. They showed some strong parameter correlations in several groups. Accurate parameter values were identified in [Rodriguez-Fernandez et al., 2006]. However, a clear correlation analysis of the parameters and the relationship between the parameter correlations and the numbers of data sets with different inputs required for the parameter estimation were not given in the previous studies.

#### 5.6.2.1 Identification of correlations

Now parameter correlations in this model are identified using the new approach. Given the model represented by Eq. (5.25), the first order partial derivative functions can be readily derived leading to the following linear dependences (see Section A.2 of Appendix A for detailed derivation).

$$\text{From Eq. (5.25a)}, \qquad \alpha_1 \frac{\partial f_1}{\partial p_1} = \alpha_2 \frac{\partial f_1}{\partial p_2} = \cdots = \alpha_5 \frac{\partial f_1}{\partial p_5} \tag{5.26}$$

$$\text{From Eq. (5.25b)}, \qquad \alpha_6 \frac{\partial f_2}{\partial p_8} = \frac{\partial f_2}{\partial p_9} \text{ and } \alpha_7 \frac{\partial f_2}{\partial p_7} + \alpha_8 \frac{\partial f_2}{\partial p_{10}} = \frac{\partial f_2}{\partial p_8} \tag{5.27}$$

$$\text{From Eq. (5.25c)}, \qquad \alpha_9 \frac{\partial f_3}{\partial p_{14}} = \frac{\partial f_3}{\partial p_{15}} \text{ and } \alpha_{10} \frac{\partial f_3}{\partial p_{13}} + \alpha_{11} \frac{\partial f_3}{\partial p_{16}} = \frac{\partial f_3}{\partial p_{14}} \tag{5.28}$$

$$\text{From Eq. (5.25g)}, \qquad \alpha_{12} \frac{\partial f_7}{\partial p_{28}} + \alpha_{13} \frac{\partial f_7}{\partial p_{29}} = \frac{\partial f_7}{\partial p_{30}} \tag{5.29}$$

$$\text{From Eq. (5.25h)}, \qquad \alpha_{14} \frac{\partial f_8}{\partial p_{35}} = \frac{\partial f_8}{\partial p_{36}} \tag{5.30}$$

The coefficients in Eqs. (5.26 - 5.30), $\alpha_i$, $(i = 1, \ldots, 14)$, are functions of the corresponding parameters and controls in the individual state equations (see Section A.2 of Appendix A). Based on these results, correlated parameters in this model can be described in 5 groups:
Group 1: $G_1(p_1, p_2, p_3, p_4, p_5)$ , among which any pair of parameters are pairwise correlated;
Group 2: $G_2(p_7, p_8, p_9, p_{10})$ , among which $p_8, p_9$ are pairwise correlated and $p_7, p_8, p_{10}$ as well as $p_7, p_9, p_{10}$ are correlated, respectively.
Group 3: $G_3(p_{13}, p_{14}, p_{15}, p_{16})$ , among which $p_{14}, p_{15}$ are pairwise correlated and $p_{13}, p_{14}, p_{16}$ as well as $p_{13}, p_{15}, p_{16}$ are correlated, respectively.
Group 4: $G_4(p_{28}, p_{29}, p_{30})$, the parameters are correlated together but not pairwise.
Group 5: $G_5(p_{35}, p_{36})$, they are pairwise correlated.

Since the coefficients are functions of both of the parameters and the control inputs, these correlated parameters are practically non-identifiable for a single set of data. It is noted that, in $G_2$ and $G_3$, the maximum number of correlated parameters is three. Among the 5 correlated parameter groups the maximum number of correlated parameters is 5 (from $G_1$). It means at least 5 data sets with different control values are required to uniquely estimate the 36 parameters of this model.

#### 5.6.2.2 Verification of the correlations by fitting the model

To verify the proposed approach and check the correlations in this model, numerical experiments are carried out by fitting the parameters to a certain number of simulated data sets with different inputs. The fitting method used is a modified sequential approach suitable for handling multiple data sets [Faber et al., 2003; Zhao et al., 2013].

The nominal parameter values given in [Moles et al., 2003] are used together with initial state values as well as $P$ and $S$ values (see Table 5.2) given in [Rodriguez-Fernandez et al., 2006] to generate 5 noise-free data sets with different inputs containing the time courses of the 8 state variables. For each data set 120 data points were taken with 1 minute as sampling time.

Table 5.2 P and S values for generating 5 datasets

| Dataset | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $P$ | 0.05000 | 0.36840 | 1.00000 | 0.09286 | 0.13572 |
| $S$ | 10.00000 | 2.15440 | 0.10000 | 2.15440 | 2.15440 |

For fitting the parameters random values are used for all 36 parameters to initialize the computation and all weights in Eq. (5.16) were set to 1.0. The results were taken by a threshold of the total residual value in the order of $10^{-9}$ when using noise-free data sets (see Table 5.3).

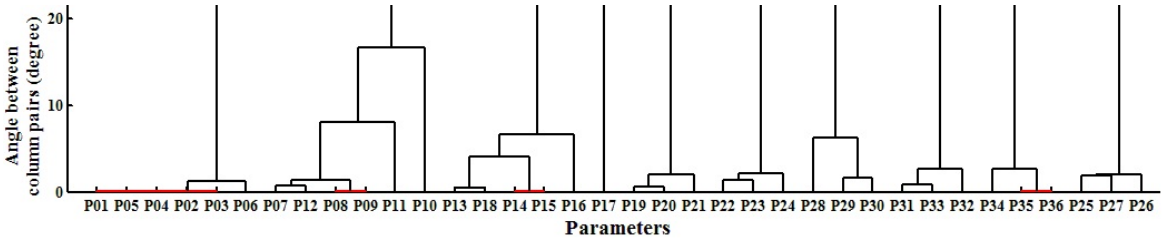
Fig. 5.3a shows the angles between the columns of the state sensitivity matrix by fitting to the $1^{st}$ data set. The zero angles (red lines) mean that the corresponding columns are pairwise parallel. According to Fig. 5.3a, 4 pairwise correlated parameter groups (i.e., $(p_1, p_2, p_3, p_4, p_5), (p_8, p_9), (p_{14}, p_{15}), (p_{35}, p_{36})$) can be detected. However, these are not the same results as identified by the analysis of the model equations. This is because a dendrogram only shows pairwise correlations; it cannot detect higher order interrelationships among the parameters.

To illustrate the geometric interpretation, the group of $G_5(p_{35}, p_{36})$ is first taken as an example to construct ZREs, i.e., to plot the correlated relations between $p_{35}$ and $p_{36}$. This was done by repeatedly fitting the model to the 5 individual data sets with different inputs, respectively, with fixed values of $p_{35}$. The resulting 5 zero residual

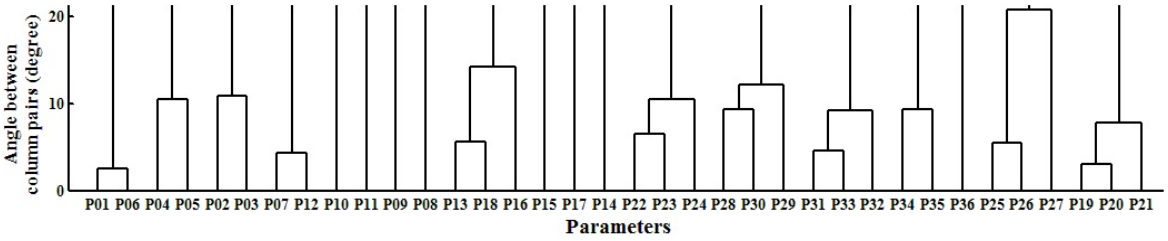
Table 5.3 Fitted parameter values based on different data sets

| No. | $P^*$ | $P^{(1)}$ | $P^{(1)(2)}$ | $P^{(1)(2)(3)}$ | $P^{(1)\cdots(4)}$ | $P^{(1)\cdots(5)}$ | $P^{(1)\cdots(5)}$(w) |
|---|---|---|---|---|---|---|---|
| 1(G1) | 1.0 | 1.06763 | 1.07763 | 1.60486 | 1.73180 | 1.00000 | 0.97145 |
| 2(G1) | 1.0 | 1.40146 | 0.91495 | 0.82116 | 0.75989 | 0.99998 | 1.05917 |
| 3(G1) | 2.0 | 1.47116 | 1.16323 | 2.39189 | 2.00001 | 2.00006 | 1.86755 |
| 4(G1) | 1.0 | 1.55173 | 1.01042 | 2.30123 | 3.19504 | 1.00000 | 0.98664 |
| 5(G1) | 2.0 | 1.40069 | 1.24912 | 0.32136 | 0.25317 | 2.00000 | 2.01339 |
| 6 | 1.0 | 1.00000 | 1.00002 | 1.00000 | 1.00000 | 1.00000 | 0.98154 |
| 7(G2) | 1.0 | 1.00927 | 1.02815 | 1.00000 | 1.00000 | 1.00000 | 0.99124 |
| 8(G2) | 1.0 | 1.32173 | 0.95504 | 1.00000 | 1.00000 | 1.00000 | 0.99919 |
| 9(G2) | 2.0 | 1.34185 | 1.18286 | 2.00000 | 2.00000 | 2.00000 | 1.93527 |
| 10(G2) | 1.0 | 1.00477 | 1.01393 | 1.00000 | 1.00000 | 1.00000 | 0.98693 |
| 11 | 2.0 | 1.99973 | 2.00007 | 2.00000 | 2.00000 | 2.00000 | 2.03582 |
| 12 | 1.0 | 0.99944 | 1.00019 | 1.00000 | 1.00000 | 1.00000 | 1.00435 |
| 13(G3) | 1.0 | 1.00572 | 1.05126 | 1.00001 | 1.00001 | 1.00001 | 1.03448 |
| 14(G3) | 1.0 | 1.39147 | 0.90768 | 1.00000 | 1.00000 | 1.00000 | 0.99558 |
| 15(G3) | 2.0 | 1.45117 | 1.00760 | 2.00003 | 2.00002 | 2.00001 | 1.98699 |
| 16(G3) | 1.0 | 1.00280 | 1.02531 | 1.00001 | 1.00000 | 1.00001 | 0.99786 |
| 17 | 2.0 | 1.99987 | 1.99999 | 1.99999 | 1.99999 | 1.99999 | 1.99586 |
| 18 | 1.0 | 1.00016 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.03924 |
| 19 | 0.1 | 0.10016 | 0.10000 | 0.10000 | 0.10000 | 0.10000 | 0.10000 |
| 20 | 1.0 | 1.00263 | 1.00000 | 1.00000 | 1.00000 | 1.00001 | 0.99469 |
| 21 | 0.1 | 0.10003 | 0.10000 | 0.10000 | 0.10000 | 0.10000 | 0.10007 |
| 22 | 0.1 | 0.10010 | 0.10000 | 0.10000 | 0.10000 | 0.10000 | 0.10000 |
| 23 | 1.0 | 1.00127 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 0.99581 |
| 24 | 0.1 | 0.10003 | 0.10000 | 0.10000 | 0.10000 | 0.10000 | 0.10025 |
| 25 | 0.1 | 0.10003 | 0.10000 | 0.10000 | 0.10000 | 0.10000 | 0.10492 |
| 26 | 1.0 | 1.00023 | 1.00002 | 1.00001 | 1.00000 | 1.00001 | 1.05077 |
| 27 | 0.1 | 0.10001 | 0.10000 | 0.10000 | 0.10000 | 0.10000 | 0.10120 |
| 28(G4) | 1.0 | 0.96519 | 0.99594 | 1.00000 | 1.00000 | 1.00000 | 1.01865 |
| 29(G4) | 1.0 | 1.62390 | 1.04672 | 1.00000 | 1.00000 | 1.00001 | 0.90507 |
| 30(G4) | 1.0 | 1.56817 | 1.04245 | 1.00000 | 0.99999 | 1.00000 | 0.85521 |
| 31 | 1.0 | 0.99997 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.11984 |
| 32 | 1.0 | 1.00110 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 0.97161 |
| 33 | 1.0 | 1.00207 | 0.99998 | 1.00000 | 0.99998 | 0.99998 | 1.33808 |
| 34 | 1.0 | 0.99956 | 1.00000 | 1.00000 | 1.00000 | 1.00000 | 1.01811 |
| 35(G5) | 1.0 | 1.05000 | 1.00001 | 1.00000 | 1.00000 | 1.00000 | 1.05077 |
| 36(G5) | 1.0 | 2.03075 | 0.99999 | 1.00000 | 1.00000 | 1.00000 | 1.20947 |
| Residual value | | 3.62E-9 | 4.26E-9 | 5.31E-9 | 6.49E-9 | 5.35E-9 | 1.12E-0 |

$P^*$ are the nominal (true) values, $P^{(1)}$ the values based on the $1^{st}$ data set, $P^{(1)(2)}$ based on the $1^{st}$, $2^{nd}$ data sets together, $P^{(1)(2)(3)}$ based on the $1^{st}$, $2^{nd}$ and $3^{rd}$ data sets, $P^{(1)\cdots(4)}$ based on the $1^{st}$ to $4^{th}$ data sets, and $P^{(1)\cdots(5)}$ based on the 5 data sets, respectively. (w) means results from 10% noises on the data. Correlated parameter groups are highlighted separately.

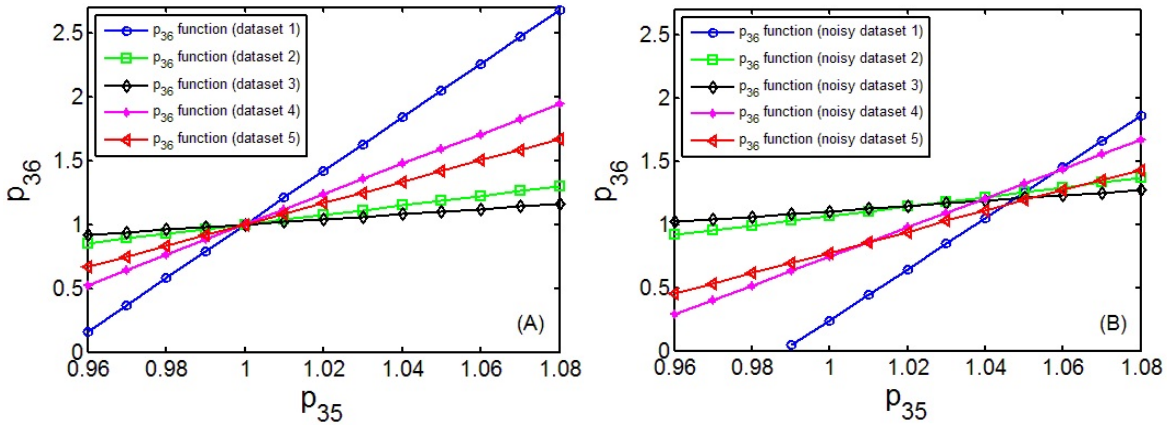



(a) Results from fitting to the $1^{st}$ data set, where pairwise correlations in different groups exist (red dash lines)




(b) Results from fitting to the 5 data sets together, where the pairwise correlations disappear

Fig. 5.3 Dendrogram of the three-step pathway model

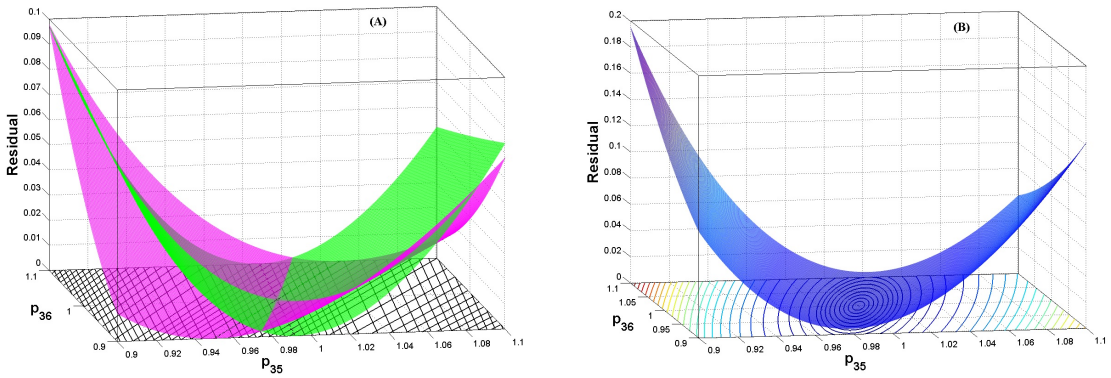



(a) Fitting to noise-free data sets (b) Fitting to the data sets with 10% noise

Fig. 5.4 Correlated relations between $p_{35}$ and $p_{36}$ based on fitting the model to 5 individual data sets with different inputs

surfaces (lines) in the subspace of $p_{35}$ and $p_{36}$ are shown in Fig. 5.4a. The 5 individual zero residual surfaces cross exactly at the true parameter point. It demonstrates that a zero residual surface from any data set will pass through the true parameter point and two data sets will be enough to determine $p_{35}$ and $p_{36}$. As expected, the zero residual surfaces resulted from different data sets cross indeed at the true parameter point in the parameter subspace. Fig. 5.4b shows the relations between $p_{35}$ and $p_{36}$ by fitting the parameters separately to the same 5 data sets on which a Gaussian distributed error of 10% was added. It can be seen that, due to the measurement noises, the crossing points of the nonzero residual surfaces are at different positions but near the true parameter point. Moreover, by comparing the lines in Fig. 5.4a and Fig. 5.4b, it can be seen that the corresponding zero residual surfaces and nonzero residual surfaces are indeed parallel, when fitting the same data set without noises or with noises, respectively.

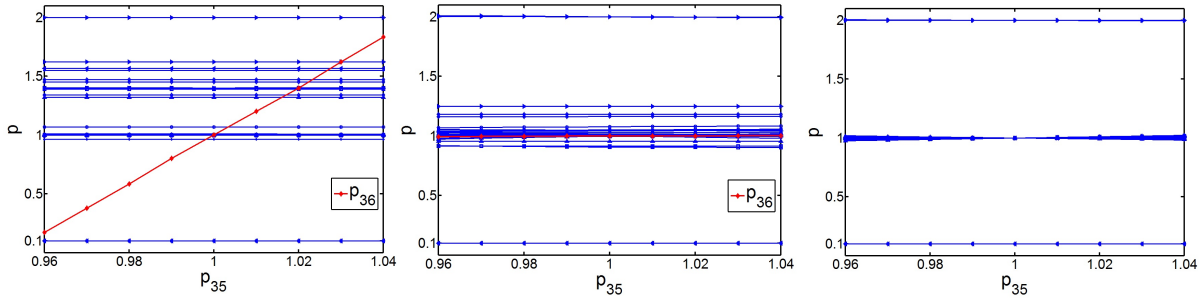
Fig. 5.5 shows the residual surfaces based on fitting to 2 individual noise-free data sets (Fig. 5.5a) and to the same 2 data sets together (Fig. 5.5b). It is shown from Fig. 5.5a that, due to the correlation, two hyperbolic cylinders are built by separately fitting to individual data sets. The bottom minimum lines of the two cylinders corresponding to the zero residual value cross at the true parameter point. Fitting to the two data sets together leads to an elliptic paraboloid (Fig. 5.5b) which has only one minimum point with the zero residual value. This point is the true parameter point, which means the remedy of the correlation between $p_{35}$ and $p_{36}$.




(a) Fitting to 2 individual noise-free data sets (b) Fitting to the data sets with 10% noise

Fig. 5.5 Fitting to the same 2 data sets together

Since the maximum number of parameters among the correlation groups is 5, according to our approach, at least 5 data sets with different inputs are needed to uniquely




(a) Fitting to the $1^{st}$ data set

(b) Fitting to $1^{st}$ and $2^{nd}$ data sets together

(c) Fitting to 5 data sets together

Fig. 5.6 Relationships of $p_{35}$ with other parameters by fitting to different numbers of noise-free data sets with different inputs

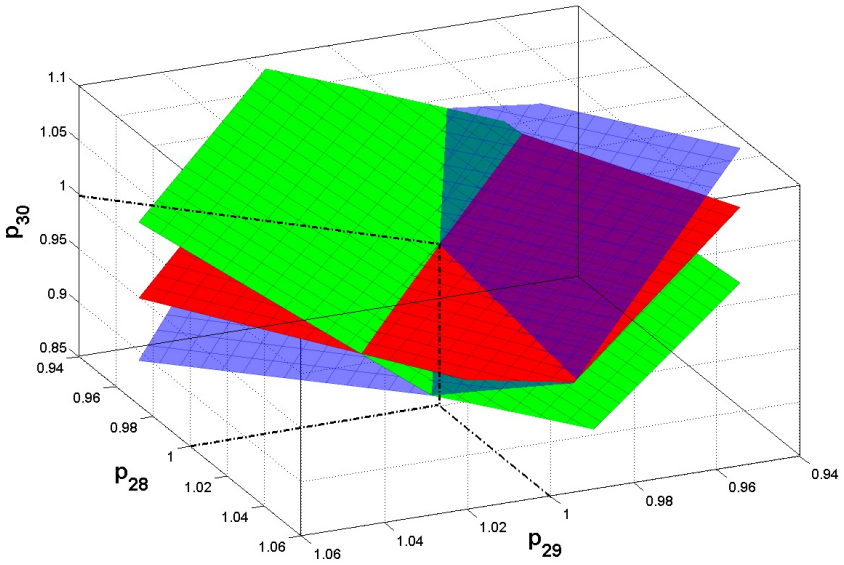



Fig. 5.7 Relations between $p_{28}$, $p_{29}$, and $p_{30}$ based on fitting the model to 3 individual noise-free data sets with different inputs

determine the parameter set. The last column in Table 5.3 ($P^{(1)\cdots(5)}$) shows the parameter values from fitting the model to the 5 data sets together. It can be seen that all of the 36 parameter values fitted are almost at their true values. According our geometric interpretation, this means that the 5 zero residual surfaces expanded by together fitting to the 5 data sets cross at the true parameter point in the parameter subspace. Fig. 5.3b shows these correlated relations indeed disappear based on the results of fitting to the 5 data sets together.

Moreover, it is shown in Table 5.3 ($P^{(1)(2)}$) that the correlation between $p_{35}$ and $p_{36}$ can be remedied by fitting two data sets together. As expected, it can be seen that in ($P^{(1)(2)}$) the parameters in $G_1$ are not well fitted (i.e., 5 correlated parameters cannot be uniquely determined by two data sets). It is also interesting to see in ($P^{(1)(2)}$) the parameter values in $G_2$, $G_3$ and $G_4$ are also not well estimated. This is because the degree of freedom of $G_2(p_7, p_8, p_9, p_{10}), G_3(p_{13}, p_{14}, p_{15}, p_{16})$, and $G_4(p_{28}, p_{29}, p_{30})$ is 3. Indeed, as shown in Table 2 ($P^{(1)(2)(3)}$), these parameters are exactly determined based on fitting the model to 3 data sets together. However, it is shown in Table 2 from the parameter values of ($P^{(1)(2)(3)}$) and ($P^{(1)\cdots(4)}$) that a number of data sets less than 5 is not enough to remedy the correlations of the parameters in $G_1$.

To test the sensitivity of the parameter results to measurement errors, the model is also fitted to the same 5 data sets with different inputs and with 10% noise level together. As shown in the last column in Table 5.3 ($P^{(1)\cdots(5)}$(w)), to some extent, the parameter values identified are deviated from the true values due to an increased residual value. But the overall parameter quality is quite good. It means the crossing points of the 5 nonzero residual surfaces expanded by the 5 noisy data sets are quite close to the true parameter point.

Fig. 5.6 shows profiles of all parameters as a function of $p_{35}$, based on different number of data sets used for fitting. It is seen from Fig. 5.6a that only $p_{36}$ is correlated with $p_{35}$ (red dash line). Moreover, it can be seen that, by fitting to one data set, the other parameters which have higher order interrelationships in other groups cannot be well determined. As shown in Fig. 5.6b, the correlation between $p_{35}$ and $p_{36}$ is remedied by fitting to two data sets together and, moreover, the parameters tend to approach their true values (i.e., 0.1, 1.0 and 2.0, see Table 5.3). Finally, all parameters are uniquely determined (i.e., clearly at the three true values), when 5 data sets were used together for fitting the model, as shown in Fig. 5.6c.

These results clearly demonstrate the scope of our approach to identifying parameter correlations. Moreover, it is clearly seen that adding more data sets with different inputs can remedy the parameter non-identifiability problem in some complex models, but a necessary number of data sets with different inputs (5 for this example) is enough.

To illustrate a higher order interrelationship among parameters, estimations were made by separately fitting the model to 3 individual data sets to plot the relations of the parameters in $G_4(p_{28}, p_{29}, p_{30})$, as shown in Fig. 5.7. The fittings for $p_{30}$ to each data set were made by fixed $p_{28}$ and $p_{29}$ with different values. Three zero residual surfaces are shown: the green plane is based on $1^{st}$ data set, the red plane $2^{nd}$ data set, and the blue plane $3^{rd}$ data set. It can be seen that the three zero residual surfaces (planes) resulted from the three individual data sets cross exactly at the true parameter point in the subspace of the 3 parameters. This demonstrates our geometric interpretation of parameter correlations, i.e., to estimate a group of three correlated parameters at least three distinct data sets with different inputs are needed.

Since parameter correlations determined from the proposed approach are based on the structure of the state equations, our result provides a minimum number of different data sets with different inputs necessary for unique parameter estimation (5 in this example). This is definitely true, if all state variables (8 in this example) are measurable and included in the 5 data sets.

The results shown above are from the solutions of the parameter estimation problem based on the data sets composed of *all* 8 state variables. It is demonstrated that at least 5 data sets with different inputs will be needed to uniquely estimate the 36 parameters. However, our method does not give information on how many state variables which may be fewer than 8 but sufficient to identify the 36 parameters. To achieve this information, an estimation of the parameters based on the generated 5 data sets which include fewer measured state variables (as output variables) is carried out. The identifiability is checked when the 5 data sets consist of data profiles of only a part of the state variables. Computational tests were carried out based on different combinations of the state variables included in the data sets. Table 5.4 shows the minimum sets of state variables which should be included in the data sets so as to achieve a successful fitting. It can be seen, for instance, the 36 parameters can be uniquely estimated in the case that only the first three state variables (i.e., $x_1, x_2, x_3$) are included in the 5 data sets. Moreover, the generated data profiles of $x_7$ and $x_8$

are also enough for identifying the 36 parameters. Due to insufficient data, estimation runs with fewer numbers of the state variables than listed in Table 5.4 could not converge, i.e., the parameters will be non-identifiable.

Table 5.4 Measurable variable sets for a successful fitting

| **No.** | Measured variables |
|---|---|
| $y_1$ | $(x_1, x_2, x_3)$ |
| $y_2$ | $(x_1, x_2, x_6)$ |
| $y_3$ | $(x_1, x_3, x_5)$ |
| $y_4$ | $(x_1, x_5, x_6)$ |
| $y_5$ | $(x_2, x_4, x_6)$ |
| $y_6$ | $(x_4, x_5, x_6)$ |
| $y_7$ | $(x_7, x_8)$ |

## 5.7 Conclusions

In this Chapter, a new approach, which is able to identify both pairwise and higher order parameter correlations, is presented. This approach is based on analysis of linear dependences of the first order partial derivative functions of model equations. In a given model there may be a number of groups with different number of correlated parameters. This method is proposed to identify these groups by analyzing the correlations of the columns of the state sensitivity matrix which can be derived directly from the right-hand side of the ODEs. Therefore, the method proposed in this research is a priori in nature, which means that the parameter correlations considered in this research are not from the results of data-based estimation. A geometric interpretation of parameter correlations is also presented. Using this approach, groups of correlated parameters and the types of correlations can be identified and, hence, the parameter identifiability issue can be addressed. Moreover, the relationship between parameter correlations and the control inputs can be derived. As a result, both structural and practical non-identifiability can be identified by the proposed approach.

In the case of practical non-identifiability, the parameter correlations can be relieved by specifying the values of control inputs for experimental design. Based on the correlation analysis, the maximum number of parameters among the correlation groups

can be determined, which corresponds to the minimum number of data sets with different inputs required for uniquely estimating the parameters of the model under consideration. Numerical results of parameter estimation of a three-step-pathway model clearly demonstrate the efficacy of the proposed approach.

It is well recognized that parameters in many biological models are correlated. Finding the true parameter point remains as a challenge since it is hidden in these correlated relations. In many cases, a direct analysis of parameter correlations based on the output sensitivity matrix depends on experimental design, and the analytical relationship cannot be seen. Instead, this approach presents a new method to analyze parameter correlations based on the matrix of the first order partial derivative functions of state equations which can be analytically derived. In this way, pairwise correlations and higher order interrelationships among the parameters can be detected. The result gives the information about parameter correlations and thus about the identifiability of parameters when all state variables are measurable for fitting the parameters. Since the output sensitivity matrix is a linear transformation of the matrix of first order partial derivative functions, our correlation analysis approach provides a necessary (but not sufficient) condition of parameter identifiability. That is, if there exist parameter correlations, the corresponding parameters are non-identifiable.

In addition, residual surfaces in the parameter subspace are introduced to interpret parameter correlations. Any point on a zero residual surface will result in a zero residual value. The crossing point of multiple zero residual surfaces leads to the true parameter point. Zero residual surfaces correspond to ZREs resulted from noise-free data sets used for fitting the parameters. If the ZREs are linearly independent (i.e., there are no correlations), the model parameters are identifiable, and otherwise they are non-identifiable. If more linearly independent ZREs can be constructed by adding new data sets with different inputs, the parameters are practically non-identifiable, otherwise they are structurally non-identifiable. In the case of practical non-identifiability the true parameter values can be found by together fitting the model to a necessary number of data sets which is the maximum number of parameters among the correlation groups. If the available measured data are from output variables, this should be regarded as the minimum number of data sets with different inputs required for unique parameter estimation. The results of the case study demonstrate the feasibility of our approach.

Moreover, an interesting result of this approach is that parameter correlations are not affected by the initial state. This means that, experimental runs can be conducted with any initial state to obtain the required data sets with different inputs. More interestingly, according to this result, different data sets with different inputs can be gained in one experimental run by changing the values of the control inputs. It is noted that the proposed approach does not address the identifiability issue of the initial states which would be a future research aspect.

The result of identifiable parameters determined by the proposed approach is theoretical. This means that the quality of the available data (the noise level, the length of sampling time, etc.,) has an important impact on the identifiability issue. Parameters which are theoretically identifiable may not be identifiable by an estimator due to low quality of the data. Non-identifiability issues caused by relative data are not considered in this research. In addition, the identification of parameter correlations based on the output equations is not considered in this research.

# Chapter 6

# Conclusions and Future research

This dissertation is set out to deal with the problem of parameter estimation of large-scale dynamical systems that are challenges due to computational intensiveness and numerical difficulties caused by undesirable characteristics of the models. In this final Chapter, the research contributions are briefly reviewed and directions for future research are also discussed.

## 6.1 Conclusions

Parameter estimation of large-scale dynamical systems plays an important role in the development of process models in technical as well as non-technical areas. Although PE problem have currently received substantial attention in the literature, they remain challenging because of the vast increase in the size of systems, undesirable characteristics of the models, such as stiffness, ill-conditioning and correlations among parameters of model equations. These main challenges include: building a well-posed system model that is usually described by a ODE or DAE system, conducting experiments to get measurement data and solving a PE problem that is usually a DOP with a least squares objective function minimizing the difference between estimated outputs and measurement data.

This dissertation is set out to participate in two aspects of above challenges: developing efficient estimation strategies and numerical algorithms which should be able

to efficiently solve such challenging estimation problems, including multiple data profiles and large parameter sets, and developing a method for identifiability analysis to identify the correlations among parameters in complex model equations.

Dealing with the first aspect, this dissertation especially focuses on the QSQ and CMSC that were currently shown to be efficient to solve DOPs. This study couples the interior point method with the QSQ to solve dynamic optimization problems, particularly parameter estimation problems. This IP QSQ method shows some advantages over the active-set QSQ method in solving the PE problem with many constraints on control, state and parameter variables, e.g. advantages of the solution path, better convergence performance and less number of iterations.

Furthermore, an improvement of IP QSQ method is developed to solve parameter estimation problems in that the reduced-space method of IP strategy is used. This reduced-space version is shown to be less time consuming than the full space one.

In further development, an improvement to the existing CMSC is developed. In this improvement, all state constraints are imposed at all collocation points in order to prevent them from violating the constraint rule which maybe happen to the CMSC. A parallel computing algorithm using the MPI method is also applied to parallely solve each portion of the model equation in each time interval. This parallel strategy shows the improvement in reducing the time consumption of the PE problem.

In PE problem, multiple data-sets of measurement data usually are used to improve the quality of the parameter estimation solutions. In this dissertation, an extension to a dynamic three-stage estimation framework is made with a derivation to the quasi-sequential strategy algorithm. Thanks to the decomposition of the optimization variables, the proposed approach can efficiently solve time-dependent parameter estimation problems with multiple data profiles. A parallel computing strategy using the MPI method is also applied successfully to boost computation efficiency.

In the second aspect, this thesis proposes a systematic approach to identify both pairwise parameter correlations and higher order interrelationships among parameters that has not been solved recently. The correlation information obtained in this way clarifies both structural and practical non-identifiability. Moreover, this correlation analysis also shows that a minimum number of data sets, which corresponds to the maximum number of correlated parameters among the correlation groups, with different inputs

for experimental design are needed to remedy the parameter correlations. The result of correlation analysis provides a necessary condition for experimental design in order to collect suitable measurement data for unique parameter estimation.

## 6.2 Future research

Some remaining aspects that arise during the methodology development of this dissertation can be supposed to the future research as follows:

1. In the field of numerical methods to the DOPs, there are still some issues that are worthy to be considered for the future research.
   (a) In the modified CMSC approach, when multiple data sets are used, parallel computation in both level of multiple data sets and collocation intervals can be applied. The three-layer optimization method leads to two levels parallelization, one for optimizing each independent NLP corresponding to each data set, one for solving model equation in each shooting time interval. Hybrid parallel computing that couples both MPI and OpenMP can be used to utilize advantages of each method. Graphics processing unit (GPU) together with multiple-core CPUs are proposed to accelerate the computation of DOPs, especially the PE problems.
   (b) In this dissertation, only local gradient-based methods are used for solving the NLPs. In order to get better results, global methods need to be taken into account.
2. In the field of identifiability analysis, the remaining tasks can be:
   (a) considering stochastic PE problems;
   (b) considering the parameter correlation between rows of the sensitivity matrix;
   (c) considering chance parameter correlations due to the temporary values of parameters that can make the sensitivity matrix be rank deficient; and
   (d) PE problems from *in vitro* to *in vivo*;

# References


Abdellatif, T., Combaz, J., and Sifakis, J. (2013). Rigorous implementation of real-time systems – from theory to application. *Mathematical Structures in Computer Science*, 23:882–914.

Aguilar, O. M. (2013). Parameter estimation in physical models: A comparison between bayesian and frequentist methods. Master's thesis, Colorado State University.

Albuquerque, J. and Biegler, L. (1995). Decomposition algorithms for on-line estimation with nonlinear models. *Computers & Chemical Engineering*, 19(10):1031 – 1039.

Antia, H. (2002). *Numerical Methods for Scientists and Engineers*. Number v. 1 in Numerical Methods for Scientists and Engineers. SPRINGER VERLAG NY.

Arora, N. and Biegler, L. T. (2004). Parameter estimation for a polymerization reactor model with a composite-step trust-region nlp algorithm. *Industrial & Engineering Chemistry Research*, 43(14):3616–3631.

Ascher, U. M., Mattheij, R. M. M., and Russell, R. D. (1995). *Numerical Solution of Boundary Value Problems for Ordinary Differential Equations*. Classics in Applied Mathematics. Society for Industrial and Applied Mathematics.

Ascher, U. M. and Petzold, L. R. (1998). *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1st edition.

Aseev, S. (2001). An optimal control problem for a differential inclusion with state constraints. smooth approximations and necessary optimality conditions. *Journal of Mathematical Sciences*, 103(6):670–685.

Ashyraliyev, M., Fomekong-Nanfack, Y., Kaandorp, J. A., and Blom, J. G. (2009). Systems biology: parameter estimation for biochemical models. *FEBS Journal*, 276(4):886–902.

Bachmann, J., Raue, A., Schilling, M., Böhm, M. E., Kreutz, C., Kaschek, D., Busch, H., Gretz, N., Lehmann, W. D., Timmer, J., and Klingmüller, U. (2011). Division of labor by dual feedback regulators controls jak2/stat5 signaling over broad ligand range. *Molecular Systems Biology*, 7(1).

Balsa-Canto, E., Alonso, A. A., and Banga, J. (2010). An iterative identification procedure for dynamic modeling of biochemical networks. *BMC Systems Biology*, 4(1).

Balsa-Canto, E. and Banga, J. R. (2011). AMIGO, a toolbox for advanced model identification in systems biology using global optimization. *Bioinformatics*, 27(16):2311–2313.

Banga, J. R., Balsa-Canto, E., Moles, C. G., and Alonso, A. A. (2003). Improving food processing using modern optimization methods. *Trends in Food Science & Technology*, 14(4):131 – 144.

Bartle, R. and Sherbert, D. (2011). *Introduction to Real Analysis, 4th Edition*. John Wiley & Sons.

Bartlett, R., Wachter, A., and Biegler, L. (2000). Active set vs. interior point strategies for model predictive control. In *American Control Conference, 2000. Proceedings of the 2000*, volume 6, pages 4229–4233 vol.6.

Barton, P. I., Allgor, R. J., Feehery, W. F., and Galán, S. (1998). Dynamic optimization in a discontinuous world. *Industrial & Engineering Chemistry Research*, 37(3):966–981.

Bell, B. M. (2012). *CppAD: A Package for Differentiation of C++ Algorithms*.

Bell, B. M. and Burke, J. V. (2008). Algorithmic differentiation of implicit functions and optimal values. In Bischof, C. H., Bücker, H. M., Hovland, P. D., Naumann, U., and Utke, J., editors, *Advances in Automatic Differentiation*, pages 67–77. Springer.

Bellman, R. (1957). *Dynamic Programming*. Chemical Industries Series. Princeton University Press.

Bellman, R. and Åström, K. (1970). On structural identifiability. *Mathematical Biosciences*, 7(3-4):329 – 339.

Bellu, G., Saccomani, M. P., Audoly, S., and D'Angiò, L. (2007). Daisy: A new software tool to test global identifiability of biological and physiological systems. *Computer Methods and Programs in Biomedicine*, 88(1):52 – 61.

Berthoumieux, S. (2012). Structural and practical identifiability of approximate metabolic network models. In Kinnaert, M., editor, *16th IFAC Symposium on System Identification*. Elsevier BV.

Betts, J. T. (1998). Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207.

Betts, J. T. (2010). *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*. Society for Industrial and Applied Mathematics, second edition.

Biegler, L. (2000). Efficient solution of dynamic optimization and NMPC problems. In Allgöwer, F. and Zheng, A., editors, *Nonlinear Model Predictive Control*, volume 26 of *Progress in Systems and Control Theory*, pages 219–243. Birkhäuser Basel.

Biegler, L. (2010). *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. MPS-SIAM Series on Optimization. Society for Industrial and Applied Mathematics.

Biegler, L. T. (2007). An overview of simultaneous strategies for dynamic optimization. *Chemical Engineering and Processing: Process Intensification*, 46(11):1043 – 1053. Special Issue on Process Optimization and Control in Chemical Engineering and Processing.

Biegler, L. T., Campbell, S. L., and Mehrmann, V., editors (2012). *Control and Optimization with Differential-Algebraic Constraints*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Biegler, L. T., Cervantes, A. M., and Wächter, A. (2002). Advances in simultaneous strategies for dynamic process optimization. *Chemical Engineering Science*, 57(4):575 – 593.

Biegler, L. T. and Grossmann, I. E. (2004). Retrospective on optimization. *Computers & Chemical Engineering*, 28(8):1169 – 1192.

Binder, T., Blank, L., Bock, H., Bulirsch, R., Dahmen, W., Diehl, M., Kronseder, T., Marquardt, W., Schlöder, J. P., and Stryk, O. (2001). Introduction to model based optimization of chemical processes on moving horizons. In Grötschel, M., Krumke, S., and Rambau, J., editors, *Online Optimization of Large Scale Systems*, pages 295–339. Springer Berlin Heidelberg.

Bischof, C., Carle, A., Corliss, G., Griewank, A., and Hovland, P. (1992). ADIFOR - generating derivative codes from fortran programs. *Scientific Programming*, (1):11–29.

Bock, H., Diehl, M., Leineweber, D., and Schloeder, J. (2000). A direct multiple shooting method for real-time optimization of nonlinear dae processes. In Allgoewer, F. and Zheng, A., editors, *Nonlinear Model Predictive Control*, volume 26 of *Progress in Systems and Control Theory*, pages 245–267. Birkhaoeser Basel.

Bock, H. G. and Plit, K. J. (1984). A multiple shooting algorithm for direct solution of optimal control problems. In *Proc. of the IFAC* $9^{th}$ *World Congress*.

Box, G. and Tiao, G. (1992). *Bayesian inference in statistical analysis*. Wiley classics library. Wiley.

Brenan, K. E., Campbell, S. L., and Petzold, L. R. (1995). *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics.

Bryson, A. (1975). *Applied Optimal Control: Optimization, Estimation and Control*. Halsted Press book'. Taylor & Francis.

Byrd, R. H., Gilbert, J. C., and Nocedal, J. (2000). A trust region method based on interior point techniques for nonlinear programming. *Mathematical Programming*, 89(1):149–185.

Byrd, R. H., Hribar, M. E., and Nocedal, J. (1999). An interior point algorithm for large-scale nonlinear programming. *SIAM J. on Optimization*, 9(4):877–900.

Byrd, R. H., Liu, G., and Nocedal, J. (1998). On the local behavior of an interior point method for nonlinear programming. In *Numerical Analysis 1997*, pages 37–56. Addison Wesley Longman.

Byrd, R. H., Nocedal, J., and Waltz, R. A. (2006). KNITRO: An integrated package for nonlinear optimization. In Di Pillo, G. and Roma, M., editors, *Large-Scale Nonlinear Optimization*, volume 83 of *Nonconvex Optimization and Its Applications*, pages 35–59. Springer US.

Canto, E. B., Banga, J. R., Alonso, A. A., and Vassiliadis, V. S. (2002). Restricted second order information for the solution of optimal control problems using control vector parameterization. *Journal of Process Control*, 12(2):243 – 255.

Cao, Y., Li, S., and Petzold, L. (2002). Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software. *Journal of Computational and Applied Mathematics*, 149(1):171 – 191. Scientific and Engineering Computations for the 21st Century - Me thodologies and Applications Proceedings of the 15th Toyota Conference.

Cao, Y., Li, S., Petzold, L., and Serban, R. (2003). Adjoint sensitivity analysis for differential-algebraic equations: The adjoint dae system and its numerical solution. *SIAM Journal on Scientific Computing*, 24(3):1076–1089.

Cervantes, A. M., Wächter, A., Tütüncü, R. H., and Biegler, L. T. (2000). A reduced space interior point strategy for optimization of differential algebraic systems. *Computers & Chemical Engineering*, 24(1):39 – 51.

Chachuat, B. (2009). *Nonlinear and Dynamic Optimization: From Theory to Practice - IC-32: Spring Term 2009*. Polycopiés de l'EPFL. EPFL.

Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D., and McDonald, J. (2000). *Parallel Programming in OpenMP*. Elsevier Science.

Chapman, B., Jost, G., and Pas, R. v. d. (2007). *Using OpenMP: Portable Shared Memory Parallel Programming (Scientific and Engineering Computation)*. The MIT Press.

Chappell, M. J., Godfrey, K. R., and Vajda, S. (1990). Global identifiability of the parameters of nonlinear systems with specified inputs: A comparison of methods. *Mathematical Biosciences*, 102(1):41 – 73.

Chis, O., Banga, J. R., and Balsa-Canto, E. (2011a). GenSSI: a software toolbox for structural identifiability analysis of biological models. *Bioinformatics*.

Chis, O.-T. (2011). Methods for checking structural identifiability of nonlinear biosystems: A critical comparison. In Bittanti, S., editor, *Proceedings of the 18th IFAC World Congress*. Elsevier BV.

Chis, O.-T., Banga, J. R., and Balsa-Canto, E. (2011b). Structural identifiability of systems biology models: A critical comparison of methods. *PLoS ONE*, 6(11):e27755.

Chu, Y. and Hahn, J. (2007). Parameter set selection for estimation of nonlinear dynamic systems. *AIChE Journal*, 53(11):2858–2870.

Chu, Y., Jayaraman, A., and Hahn, J. (2007). Parameter sensitivity analysis of il-6 signalling pathways. *IET Systems Biology*, 1:342–352(10).

Cintrón-Arias, A., Banks, H. T., Capaldi, A., and Lloyd, A. L. (2009). A sensitivity matrix based methodology for inverse problem formulation. *Journal of Inverse and Ill-posed Problems*, 17(6).

Cobelli, C. and DiStefano, J. J. (1980). Parameter and structural identifiability concepts and ambiguities: a critical review and analysis. *American Journal of Physiology - Regulatory, Integrative and Comparative Physiology*, 239(1):R7–R24.

Conway, B. (2012). A survey of methods available for the numerical optimization of continuous dynamic systems. *Journal of Optimization Theory and Applications*, 152(2):271–306.

Conway, B. A. and Larson, K. M. (1998). Collocation versus differential inclusion in direct optimization. *Journal of Guidance, Control, and Dynamics*, 21(5):780–785.

Denis-Vidal, L. and Joly-Blanchard, G. (2000). An easy to check criterion for (un)indentifiability of uncontrolled systems and its applications. *IEEE Transactions on Automatic Control*, 45(4):768–771.

Denis-Vidal, L., Joly-Blanchard, G., and Noiret, C. (2001). Some effective approaches to check the identifiability of uncontrolled nonlinear systems. *Mathematics and Computers in Simulation*, 57(1â€“2):35 – 44.

Diehl, M. (2001). *Real-time optimization of large-scale nonlinear processes*. PhD thesis, University of Heidelberg, Heidelberg, Germany.

Diehl, M., Bock, H., Diedam, H., and Wieber, P.-B. (2006). Fast direct multiple shooting algorithms for optimal robot control. In Diehl, M. and Mombaur, K., editors, *Fast Motions in Biomechanics and Robotics*, volume 340 of *Lecture Notes in Control and Information Sciences*, pages 65–93. Springer Berlin Heidelberg.

Dobre, S., Bastogne, T., Profeta, C., Barberi-Heyob, M., and Richard, A. (2012). Limits of variance-based sensitivity analysis for non-identifiability testing in high dimensional dynamic models. *Automatica*, 48(11):2740 – 2749.

Dochain, D., Vanrolleghem, P. A., and Daele, M. V. (1995). Structural identifiability of biokinetic models of activated sludge respiration. *Water Research*, 29(11):2571 – 2578.

Doví, V. and Paladino, O. (1989). Fitting of experimental data to implicit models using a constrained variation algorithm. *Computers & Chemical Engineering*, 13(6):731 – 735.

Drosinos, N. and Koziris, N. (2004). Performance comparison of pure MPI vs hybrid MPI-OpenMP parallelization models on SMP clusters. In *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.* Institute of Electrical & Electronics Engineers (IEEE).

E Balsa-Canto, J. B. (2010a). Amigo: A model identification toolbox based on global optimization. In *ICSB 2010, 11th International Conference on Systems Biology.*

E Balsa-Canto, J. B. (2010b). Amigo: A model identification toolbox based on global optimization. In *CAB (11th Computer applications in biotechnology).*

Efron, B. (2012). A 250-year argument: Belief, behavior, and the bootstrap. *Bulletin of the American Mathematical Society*, 50(1):129–146.

Efron, B. (2013). Bayes' theorem in the 21st century. *Science*, 340(6137):1177–1178.

Elsheikh, A. (2013). Assisting identifiability analysis of large-scale dynamical models with decision trees: DecTrees and interactive menus. In *2013 8th EUROSIM Congress on Modelling and Simulation.* Institute of Electrical & Electronics Engineers (IEEE).

Englezos, P. and Kalogerakis, N. (2000). *Applied Parameter Estimation for Chemical Engineers.* Chemical Industries Series. CRC Press INC.

Esposito, W. R. and Floudas, C. A. (2000). Global optimization for the parameter estimation of differential-algebraic systems. *Industrial & Engineering Chemistry Research*, 39(5):1291–1310.

Faber, R., Li, P., and Wozny, G. (2003). Sequential parameter estimation for large-scale systems with multiple data sets. 1. computational framework. *Industrial & Engineering Chemistry Research*, 42(23):5850–5860.

Feehery, W. F., Tolsma, J. E., and Barton, P. I. (1997). Efficient sensitivity analysis of large-scale differential-algebraic systems. *Applied Numerical Mathematics*, 25(1):41 – 54.

Flassig, R. J., Migal, I., van der Zalm, E., Rihko-Struckmann, L., and Sundmacher, K. (2015). Rational selection of experimental readout and intervention sites for reducing uncertainties in computational model predictions. *BMC Bioinformatics*, 16(1).

Gauss, C. (1809). *Theoria motus corporum coelestium in sectionibus conicis solem ambientium.*

Gauss, C. and Stewart, G. (1995). *Theory of the Combination of Observations Least Subject to Errors, Part One, Part Two, Supplement.* Society for Industrial and Applied Mathematics.

Gevers, M. (2006). A personal view of the development of system identification: A 30-year journey through an exciting field. *Control Systems, IEEE*, 26(6):93–105.

Giannakis, G. B. and Serpedin, E. (2001). A bibliography on nonlinear system identification. *Signal Process.*, 81(3):533–580.

Gill, P. E. and Wong, E. (2012). Sequential quadratic programming methods. In Lee, J. and Leyffer, S., editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 147–224. Springer New York.

Girolami, M. (2008). Bayesian inference for differential equations. *Theoretical Computer Science*, 408.

Goh, C. and Teo, K. (1988). Control parametrization: A unified approach to optimal control problems with general constraints. *Automatica*, 24(1):3–18.

Gonzalez, O. R., Kuper, C., Jung, K., Naval, P. C., and Mendoza, E. (2006). Parameter estimation using simulated annealing for s-system models of biochemical networks. *Bioinformatics*, 23(4):480–486.

Gould, N. I. and Toint, P. L. (2000). SQP methods for large-scale nonlinear programming. In Powell, M. and Scholtes, S., editors, *System Modelling and Optimization*, volume 46 of *IFIP - The International Federation for Information Processing*, pages 149–178. Springer US.

Griewank, A., Juedes, D., and Utke, J. (1996). Algorithm 755: ADOL-C: A package for the automatic differentiation of algorithms written in C/C++. *ACM Trans. Math. Softw.*, 22(2):131–167.

Gropp, W., Hoefler, T., Thakur, R., and Lusk, E. (2014a). *Using Advanced MPI: Modern Features of the Message-Passing Interface*. Computer science & intelligent systems. MIT Press.

Gropp, W., Lusk, E., and Skjellum, A. (2014b). *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. The MIT Press.

Gupta, S. (1995). *Numerical Methods for Engineers*. New Age International (P) Limited.

Gutenkunst, R. N., Waterfall, J. J., Casey, F. P., Brown, K. S., Myers, C. R., and Sethna, J. P. (2007). Universally sloppy parameter sensitivities in systems biology models. *PLoS Computational Biology*, 3(10):1871–1878.

Hargraves, C. R. and Paris, S. W. (1987). Direct trajectory optimization using nonlinear programming and collocation. In *(Astrodynamics Conference, Williamsburg, VA, Aug. 18-20, 1986, Technical Papers, p. 3-12) Journal of Guidance, Control, and Dynamics (ISSN 0731-5090), vol. 10, July-Aug. 1987, p. 338-342. Previously cited in issue 23, p. 3418, Accession no. A86-47902.*, volume 10 of *AAS/AIAA Astrodynamics Conference*, pages 3–12.

Hengl, S., Kreutz, C., Timmer, J., and Maiwald, T. (2007). Data-based identifiability analysis of non-linear dynamical models. *Bioinformatics*, 23(19):2612–2618.

Henson, M. and Seborg, D. (1997). *Nonlinear process control*. Prentice Hall PTR.

Hicks, G. A. and Ray, W. H. (1971). Approximation methods for optimal control systems. *Can. J. Chem. Engng.*, (49):522—528.

Hindmarsh, A. C., Brown, P. N., Grant, K. E., Lee, S. L., Serban, R., Shumaker, D. E., and Woodward, C. S. (2005). SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software,*, 31(3):363–396.

Hindmarsh, A. C. and Serban, R. (2002). *User Documentation for CVODES, An ODE Solver with Sensitivity Analysis Capabilities*.

Hong, W., Tan, P., Vu, Q. D., and Li, P. (2009). An improved quasi-sequential approach to large-scale dynamic process optimization. In Rita Maria de Brito Alves, C. A. O. d. N. and Biscaia, E. C., editors, $10^{th}$ *International Symposium on Process Systems Engineering: Part A*, volume 27 of *Computer Aided Chemical Engineering*, pages 249 – 254. Elsevier.

Hong, W., Wang, S., Li, P., Wozny, G., and Biegler, L. T. (2006). A quasi-sequential approach to large-scale dynamic optimization problems. *AIChE Journal*, 52(1):255–268.

Hull, D. G. (1997). Conversion of optimal control problems into parameter optimization problems. *Journal of Guidance, Control, and Dynamics*, 20(1):57–60.

Isermann, R. and Münchhof, M. (2010). *Identification of Dynamic Systems: An Introduction with Applications*. Advanced textbooks in control and signal processing. Springer.

Jacquez, J. (1998). Design of experiments. *Journal of the Franklin Institute*, 335B(2):259 – 279.

Jayasankar, B. R., Ben-Zvi, A., and Huang, B. (2009). Identifiability and estimability study for a dynamic solid oxide fuel cell model. *Computers & Chemical Engineering*, 33(2):484 – 492.

Jia, G., Stephanopoulos, G. N., and Gunawan, R. (2011). Parameter estimation of kinetic models from metabolic profiles: two-phase dynamic decoupling method. *Bioinformatics*, 27(14):1964–1970.

Jockenhövel, T. (2004a). *Dynamic Optimization of Complex Power Plants and Chemical Processes*. Fortschrittberichte VDI / 6: Energietechnik. VDI-Verlag.

Jockenhövel, T. (2004b). *Dynamic Optimization of Complex Power Plants and Chemical Processes*. Fortschrittberichte VDI / 6: Energietechnik. VDI-Verlag.

Johnson, M. L. and Faont, L. M. (1992). Parameter estimation by least-squares methods. *Methods in enzymology*, 210:1–37.

Jost, G., Haoqiang, J., an Mey, D., and Hatay, F. F. (2003). Comparing the openmp, mpi, and hybrid programming paradigms on an SMP cluster. In *Proceedings of the Fifth European Workshop on OpenMP (EWOMP '03)*. Aachen University, Germany.

Juditsky, A., Hjalmarsson, H., Benveniste, A., Delyon, B., Ljung, L., Sjöberg, J., and Zhang, Q. (1995). Nonlinear black-box models in system identification: Mathematical foundations. *Automatica*, 31(12):1725–1750.

Keller, R., Gabriel, E., and Dongarra, J. (2010). *Recent Advances in the Message Passing Interface: 17th European MPI User's Group Meeting, EuroMPI 2010, Stuttgart, Germany, September 12-15, 2010, Proceedings*. LNCS sublibrary: SL 2-programming and software engineering. Springer.

Kelley, C. (2003). *Solving Nonlinear Equations with Newton's Method*. Fundamentals of Algorithms. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

Kim, I.-W., Liebman, M. J., and Edgar, T. F. (1990). Robust error-in-variables estimation using nonlinear programming techniques. *AIChE Journal*, 36(7):985–993.

Klein, V. and Morelli, E. (2006). *Aircraft System Identification: Theory and Practice*. AIAA education series. American Institute of Aeronautics and Astronautics.

Kreutz, C., Raue, A., and Timmer, J. (2012). Likelihood based observability analysis and confidence intervals for predictions of dynamic models. *BMC Syst Biol*, 6(1):120.

Kreutz, C. and Timmer, J. (2009). Systems biology: experimental design. *FEBS Journal*, 276(4):923–942.

Lau, M., Yue, S., Ling, K., and Maciejowski, J. (2009). A comparison of interior point and active set methods for fpga implementation of model predictive control. In *Proc. European Control Conference*, Budapest. European Union Control Association.

Li, P. and Vu, Q. (2013). Identification of parameter correlations for parameter estimation in dynamic biological models. *BMC Systems Biology*, 7(1).

Li, S. and Petzold, L. (2000). Software and algorithms for sensitivity analysis of large-scale differential algebraic systems. *Journal of Computational and Applied Mathematics*, 125(1-2):131 – 145. Numerical Analysis 2000. Vol. VI: Ordinary Differential Equations and Integral Equations.

Li, S., Petzold, L., and Zhu, W. (2000). Sensitivity analysis of differential-algebraic equations: A comparison of methods on a special problem. *Applied Numerical Mathematics*, 32(2):161 – 174.

Ljung, L., editor (1999). *System Identification (2$^{nd}$ Ed.): Theory for the User*. Prentice Hall PTR, Upper Saddle River, NJ, USA.

Ljung, L. (2010). Perspectives on system identification. *Annual Reviews in Control*, 34(1):1 – 12.

Ljung, L. and Glad, T. (1994a). *Modeling of Dynamic Systems*. Prentice Hall Information and System Sciences Series. PTR Prentice Hall.

Ljung, L. and Glad, T. (1994b). On global identifiability for arbitrary model parametrizations. *Automatica*, 30(2):265 – 276.

Lobo Pereira, F. and Borges de Sousa, J. (1992). A differential inclusion algorithm for optimal control problems. In *Decision and Control, 1992., Proceedings of the 31st IEEE Conference on*, pages 1538–1539 vol.2.

Loewen, P. D. and Rockafellar, R. T. (1994). Optimal control of unbounded differential inclusions. *SIAM J. Control Optim.*, 32(2):442–470.

Logsdon, J. and Biegler, L. (1992). Decomposition strategies for large-scale dynamic optimization problems. *Chemical Engineering Science*, 47(4):851–864.

Luus, R. (1990). Application of dynamic programming to high-dimensional non-linear optimal control problems. *International Journal of Control*, 52(1):239–250.

Maine, R., Iliff, K., Aeronautics, U. S. N., Scientific, S. A., and Branch, T. I. (1985). *Identification of Dynamic Systems: Theory and Formulation*. NASA reference publication. National Aeronautics and Space Administration, Scientific and Technical Information Branch.

Maiwald, T. and Timmer, J. (2008). Dynamical modeling and multi-experiment fitting with PottersWheel. *Bioinformatics*, 24(18):2037–2043.

Maurer, H. and Pesch, H. J. (2008). Direct optimization methods for solving a complex state-constrained optimal control problem in microeconomics. *Applied Mathematics and Computation*, 204(2):568 – 579. Special Issue on New Approaches in Dynamic Optimization to Assessment of Economic and Environmental Systems.

McLean, K. A. P. and McAuley, K. B. (2012). Mathematical modelling of chemical processes-obtaining the best model predictions and parameter estimates using identifiability and estimability procedures. *The Canadian Journal of Chemical Engineering*, 90(2):351–366.

Mendes, P. (2001). Modeling large biological systems from functional genomic data: Parameter estimation. *In Foundations of systems biology (ed. H. Kitano)*, pages 163–186.

Meshkat, N., Eisenberg, M., and DiStefano-III, J. J. (2009). An algorithm for finding globally identifiable parameter combinations of nonlinear ode models using gröbner bases. *Mathematical Biosciences*, 222(2):61 – 72.

Miao, H., Xia, X., Perelson, A. S., and Wu, H. (2011). On identifiability of nonlinear ODE models and applications in viral dynamics. *SIAM Rev.*, 53(1):3–39.

Moles, C. G., Mendes, P., and Banga, J. R. (2003). Parameter estimation in biochemical pathways: A comparison of global optimization methods. *Genome Research*, 13(11):2467–2474.

Morales, J. L., Nocedal, J., and Wu, Y. (2011). A sequential quadratic programming algorithm with an additional equality constrained phase. *IMA Journal of Numerical Analysis*.

Morison, K. R. and Sargent, R. W. H. (1986). Optimization of multistage processes described by differential-algebraic equations. In Hennart, J.-P., editor, *Numerical Analysis*, volume 1230 of *Lecture Notes in Mathematics*, pages 86–102. Springer Berlin Heidelberg.

NAG (2012). *NAG Library Manual, Mark 23*. Numerical Algorithms Group, Oxford UK.

Nakajima, K. (2012). OpenMP/MPI hybrid parallel multigrid method on fujitsu FX10 supercomputer system. In *2012 IEEE International Conference on Cluster Computing Workshops*. Institute of Electrical & Electronics Engineers (IEEE).

Nieman, R. E., Fisher, D. G., and Seborg, D. E. (1971). A review of process identification and parameter estimation techniques. *International Journal of Control*, 13(2):209–264.

Nocedal, J. and Wright, S. (2006). *Numerical Optimization*. Springer series in operations research and financial engineering. Springer, 2nd edition.

Oldenburg, J., Marquardt, W., Heinz, D., and Leineweber, D. B. (2003). Mixed-logic dynamic optimization applied to batch distillation process design. *AIChE Journal*, 49(11):2900–2917.

Pannocchia, G. and Rawlings, J. B. (2003). Disturbance models for offset-free model-predictive control. *AIChE Journal*, 49(2):426–437.

Papamichail, I. and Adjiman, C. (2002). A rigorous global optimization algorithm for problems with ordinary differential equations. *Journal of Global Optimization*, 24(1):1–33.

Petersen, B., Gernaey, K., Devisscher, M., Dochain, D., and Vanrolleghem, P. A. (2003). A simplified method to assess structurally identifiable parameters in monod-based activated sludge models. *Water Research*, 37(12):2893 – 2904.

Plitt, K. J. (1981). Ein superlinear konvergentes mehrzielverfahren zur direkten berechnung beschränkter optimaler steuerungen. Master's thesis, University of Bonn.

Pohjanpalo, H. (1978). System identifiability based on the power series expansion of the solution. *Mathematical Biosciences*, 41(1-2):21 – 33.

Poku, M. Y. B., Biegler, L. T., and Kelly, J. D. (2004). Nonlinear optimization with many degrees of freedom in process engineering. *Industrial & Engineering Chemistry Research*, 43(21):6803–6812.

Prata, D. M., Schwaab, M., Lima, E. L., and Pinto, J. C. (2010). Simultaneous robust data reconciliation and gross error detection through particle swarm optimization for an industrial polypropylene reactor. *Chemical Engineering Science*, 65(17):4943 – 4954.

Quaiser, T. and Mönnigmann, M. (2009). Systematic identifiability testing for unambiguous mechanistic modeling –application to jak-stat, map kinase, and nf-$\kappa$b signaling pathway models. *BMC Systems Biology*, 3(50).

Rabenseifner, R., Hager, G., and Jost, G. (2009). Hybrid MPI/OpenMP parallel programming on clusters of multi-core SMP nodes. In *2009 17th Euromicro International Conference on Parallel, Distributed and Network-based Processing*. Institute of Electrical & Electronics Engineers (IEEE).

Radu Serban, C. P. and Hindmarsh, A. C. (2012). *User Documentation for IDAS v1.1.0*.

Rao, A. V. (2009). A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528.

Raue, A., Becker, V., Klingmüler, U., and Timmer, J. (2010). Identifiability and observability analysis for experimental design in nonlinear dynamical models. *Chaos*, 20(4):045105.

Raue, A., Karlsson, J., Saccomani, M. P., Jirstrand, M., and Timmer, J. (2014). Comparison of approaches for parameter identifiability analysis of biological systems. *Bioinformatics*.

Raue, A., Kreutz, C., Maiwald, T., Bachmann, J., Schilling, M., Klingmüller, U., and Timmer, J. (2009). Structural and practical identifiability analysis of partially observed dynamical models by exploiting the profile likelihood. *Bioinformatics*, 25(15):1923–1929.

Raue, A., Kreutz, C., Maiwald, T., Klingmüller, U., and Timmer, J. (2011). Addressing parameter identifiability by model-based experimentation. *Systems Biology, IET*, 5(2):120–130.

Raue, A., Kreutz, C., Theis, F. J., and Timmer, J. (2012). Joining forces of bayesian and frequentist methodology: a study for inference in the presence of non-identifiability. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984):20110544–20110544.

Robinson, S. M. (1974). Perturbed kuhn-tucker points and rates of convergence for a class of nonlinear-programming algorithms. *Mathematical Programming*, 7(1):1–16.

Rodriguez-Fernandez, M., Mendes, P., and Banga, J. R. (2006). A hybrid approach for efficient and robust parameter estimation in biochemical pathways. *Biosystems*, 83(2-3):248 – 265. 5th International Conference on Systems Biology {ICSB} 2004 5th International Conference on Systems Biology.

Saccomani, M. P., Audoly, S., Bellu, G., and D'Angiò, L. (2010). Examples of testing global identifiability of biological and biomedical models with the daisy software. *Comput. Biol. Med.*, 40(4):402–407.

Samaniego, F. (2010). *A Comparison of the Bayesian and Frequentist Approaches to Estimation*. Springer Series in Statistics. Springer.

Sargent, R. W. H. and Sullivan, G. R. (1978). The development of an efficient optimal control package. In Stoer, J., editor, *Optimization Techniques*, volume 7 of *Lecture Notes in Control and Information Sciences*, pages 158–168. Springer Berlin Heidelberg.

Schaber, J. (2012). Easy parameter identifiability analysis with COPASI. *Biosystems*, 110(3):183–185.

Schittkowski, K. (2013). *Numerical Data Fitting in Dynamical Systems: A Practical Introduction with Applications and Software*. Applied Optimization. Springer US.

Seywald, H. (1993). Trajectory optimization based on differential inclusion. *AA(Analytical Mechanics Associates, Inc., Hampton, VA, US)*, pages 649–663.

Shampine, L., Reichelt, M., and Kierzenka, J. (1999). Solving index-1 daes in matlab and simulink. *SIAM Review*, 41(3):538–552.

Shen, J., Tang, T., and Wang, L.-L. (2011). *Spectral Methods*. Springer Berlin Heidelberg.

Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Delyon, B., Glorennec, P.-Y., Hjalmarsson, H., and Juditsky, A. (1995). Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691 – 1724. Trends in System Identification.

Steiert, B., Raue, A., Timmer, J., and Kreutz, C. (2012). Experimental design for parameter estimation of gene regulatory networks. *PLoS ONE*, 7(7):e40052.

Strejc, V. (1977). Least squares in identification theory. *Kybernetika*, 13(2):83–105.

Äström, K. and Eykhoff, P. (1971). System identification - A survey. *Automatica*, 7(2):123 – 162.

Tamimi, J. and Li, P. (2010). A combined approach to nonlinear model predictive control of fast systems. *Journal of Process Control*, 20(9):1092 – 1102. {ADCHEM} 2009 Special Issue.

Tjoa, I. B. and Biegler, L. T. (1991). Simultaneous solution and optimization strategies for parameter estimation of differential-algebraic equation systems. *Industrial & Engineering Chemistry Research*, 30(2):376–385.

Vajda, S., Godfrey, K. R., and Rabitz, H. (1989a). Similarity transformation approach to identifiability analysis of nonlinear compartmental models. *Mathematical Biosciences*, 93(2):217 – 248.

Vajda, S., Rabitz, H., Walter, E., and Lecourtier, Y. (1989b). Qualitative and quantitative identifiability analysis of nonlinear chemical kinetic models. *Chemical Engineering Communications*, 83(1):191–219.

van de Geer, S. A. (2005). *Least Squares Estimation*, volume 2, pages 1041 – 1045. John Wiley & Sons, Ltd.

Vassiliadis, V. (1993). *Computational Solution of Dynamic Optimization Problems with General Differential-algebraic Constraints*. University of London.

Vassiliadis, V. S., Sargent, R. W. H., and Pantelides, C. C. (1994a). Solution of a class of multistage dynamic optimization problems. 1. problems without path constraints. *Industrial & Engineering Chemistry Research*, 33(9):2111–2122.

Vassiliadis, V. S., Sargent, R. W. H., and Pantelides, C. C. (1994b). Solution of a class of multistage dynamic optimization problems. 2. problems with path constraints. *Industrial & Engineering Chemistry Research*, 33(9):2123–2133.

Villaverde, A. F. and Banga, J. R. (2013). Reverse engineering and identification in systems biology: strategies, perspectives and challenges. *Journal of The Royal Society Interface*, 11(91).

Voit, E. O. and Almeida, J. (2004). Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics*, 20(11):1670–1681.

Vu, Q. D. and Li, P. (2010). A reduced-space interior-point quasi-sequential approach to nonlinear optimization of large-scale dynamic systems. In *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2010 IEEE RIVF International Conference on Computing and Communication Technologies*, pages 1–6.

Vu, Q. D., Zhao, C., Li, P., and Su, H. (2010). An efficient parameter identification approach of large-scale dynamic systems by a quasi-sequential interior-point method based on multiple data-sets. In $2^{nd}$ *International Conference on Engineering Optimization EngOpt2010,* published on CD, 8 pages.

Wächter, A. (2002). *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA.

Walter, E., Braems, I., Jaulin, L., and Kieffer, M. (2004). Guaranteed numerical computation as an alternative to computer algebra for testing models for identifiability. In Alt, R., Frommer, A., Kearfott, R., and Luther, W., editors, *Numerical Software with Result Verification*, volume 2991 of *Lecture Notes in Computer Science*, pages 124–131. Springer Berlin Heidelberg.

Walter, E. and Lecourtier, Y. (1982). Global approaches to identifiability testing for linear and nonlinear state space models. *Mathematics and Computers in Simulation*, 24(6):472 – 482.

Walter, É. and Pronzato, L. (1997). *Identification of parametric models from experimental data*. Communications and control engineering. Springer.

Walther, A. and Griewank, A. (2012). Getting started with ADOL-C. In Naumann, U. and Schenk, O., editors, *Combinatorial Scientific Computing*, pages 181–202. Chapman & Hall CRC Computational Science Series.

Wächter, A. and Biegler, L. T. (2005). Line search filter methods for nonlinear programming: Motivation and global convergence. *SIAM J. on Optimization*, 16(1):1–31.

Wächter, A. and Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57.

Weiss, O. (2012). Performance analysis of pure MPI versus MPI+OpenMP for jacobi iteration and a 3D FFT on the Cray XT5. Master's thesis, Iowa State University, Ames, Iowa.

Wiener, N. (1965). *Cybernetics Or Control and Communication in the Animal and the Machine*. M.I.T. paperback series. Wiley.

Wongrat, W., Srinophakun, T., and Srinophakun, P. (2005). Modified genetic algorithm for nonlinear data reconciliation. *Computers & Chemical Engineering*, 29(5):1059 – 1067.

Xia, X. (2003). Estimation of HIV/AIDS parameters. *Automatica*, 39(11):1983–1988.

Xia, X. and Moog, C. (2003). Identifiability of nonlinear systems with application to HIV/AIDS models. *IEEE Transactions on Automatic Control*, 48(2):330–336.

Yao, K. Z., Shaw, B. M., Kou, B., McAuley, K. B., and Bacon, D. W. (2003). Modeling ethylene/butene copolymerization with multi-site catalysts: Parameter estimability and experimental design. *Polymer Reaction Engineering*, 11(3):563–588.

Zavala, V. M. and Biegler, L. T. (2006). Large-scale parameter estimation in low-density polyethylene tubular reactors. *Industrial & engineering chemistry research*, 45(23):7867–7881.

Zavala, V. M., Laird, C. D., and Biegler, L. T. (2008a). Fast implementations and rigorous models: Can both be accommodated in NMPC? *International Journal of Robust and Nonlinear Control*, 18(8):800–815.

Zavala, V. M., Laird, C. D., and Biegler, L. T. (2008b). Interior-point decomposition approaches for parallel solution of large-scale nonlinear parameter estimation problems. *Chemical Engineering Science*, 63(19):4834 – 4845. Model-Based Experimental Analysis.

Zhao, C., Vu, Q., and Li, P. (2013). A quasi-sequential parameter estimation for nonlinear dynamic systems based on multiple data profiles. *Korean Journal of Chemical Engineering*, 30(2):269–277.

# Appendix A

# Supplementary Material

## A.1 The sensitivity matrix derivation

Consider the sensitivity equation

$$\dot{S} = \left(\frac{\partial f}{\partial x}\right) S + \left(\frac{\partial f}{\partial p}\right) \tag{A.1}$$

Using the explicit Euler method at time point with a small time interval, we can write Eq. (A.1) in the discrete form:

$$\frac{S_k - S_{k-1}}{\Delta t} = \left(\frac{\partial f}{\partial x}\right)_{k-1} S_{k-1} + \left(\frac{\partial f}{\partial p}\right)_{k-1} \tag{A.2}$$

It leads to:

$$\begin{aligned} S_k &= S_{k-1} + \Delta t\left(\frac{\partial f}{\partial x}\right)_{k-1} S_{k-1} + \Delta t\left(\frac{\partial f}{\partial p}\right)_{k-1} \\ &= \left(I + \Delta t\left(\frac{\partial f}{\partial x}\right)_{k-1}\right) S_{k-1} + \Delta t\left(\frac{\partial f}{\partial p}\right)_{k-1} \end{aligned} \tag{A.3}$$

where $I$ is a unit matrix. By expanding Eq. (A.3) we get:

$$\begin{aligned} S_k = &\left(\prod_{i=0}^{k-1}\left(I + \Delta t\left(\frac{\partial f}{\partial x}\right)_i\right)\right) S_0 + \Delta t\left(\prod_{i=1}^{k-1}\left(I + \Delta t\left(\frac{\partial f}{\partial x}\right)_i\right)\right)\left(\frac{\partial f}{\partial p}\right)_0 \\ &+ \Delta t\left(\prod_{i=2}^{k-1}\left(I + \Delta t\left(\frac{\partial f}{\partial x}\right)_i\right)\right)\left(\frac{\partial f}{\partial p}\right)_1 + \cdots + \Delta t\left(\frac{\partial f}{\partial p}\right)_{k-1} \end{aligned} \tag{A.4}$$

It can be reformulated as:

$$\begin{aligned} S_k &= \left(\prod_{i=0}^{k-1}\left(I+\Delta t\left(\frac{\partial f}{\partial x}\right)_i\right)\right)S_0 + W_0\left(\frac{\partial f}{\partial p}\right)_0 + W_1\left(\frac{\partial f}{\partial p}\right)_1 + \cdots + W_{k-1}\left(\frac{\partial f}{\partial p}\right)_{k-1} \\ &= \left(\prod_{i=0}^{k-1}\left(I+\Delta t\left(\frac{\partial f}{\partial x}\right)_i\right)\right)S_0 + \sum_{j=0}^{k-1} W_j\left(\frac{\partial f}{\partial p}\right)_j \end{aligned} \tag{A.5}$$

Since $S_0 = \left(\frac{\partial x}{\partial p}\right)_0$ is the sensitivity at the initial state $x(t_0) = x_0$ , there are two possible cases:

#### A.1.0.3 Case 1

$x(t_0) = x_0$ is a steady state. Then:

$$S_0 = \left(\frac{\partial x}{\partial p}\right)_0 = \left(\left(\frac{\partial f}{\partial x}\right)_0\right)^{-1}\left(\frac{\partial f}{\partial p}\right)_0 \tag{A.6}$$

#### A.1.0.4 Case 2

$x(t_0) = x_0$ is not a steady state. Then we can consider that $x(t_0) = x_0$ is evolved from a steady state $x(-l) = x_{-l}$ at time point $t = -l$. According to Eq. (A.5)

$$\begin{aligned} S_0 &= \left(\prod_{i=-l}^{-1}\left(I+\Delta t\left(\frac{\partial f}{\partial x}\right)_i\right)\right)S_{-l} + \sum_{j=-l}^{-1} W_j\left(\frac{\partial f}{\partial p}\right)_j \\ &= \left(\prod_{i=-l}^{-1}\left(I+\Delta t\left(\frac{\partial f}{\partial x}\right)_i\right)\right)\left(\left(\frac{\partial f}{\partial x}\right)_{-l}\right)^{-1}\left(\frac{\partial f}{\partial p}\right)_{-l} + \sum_{j=-l}^{-1} W_j\left(\frac{\partial f}{\partial p}\right)_j \\ &= \tilde{W}_{-l}\left(\frac{\partial f}{\partial p}\right)_{-l} + \sum_{j=-l}^{-1} W_j\left(\frac{\partial f}{\partial p}\right)_j \end{aligned} \tag{A.7}$$

In both cases, $S_0$ has a linear relation with $\left(\frac{\partial f}{\partial p}\right)_j$. Then from Eq. (A.5) there is:

$$S_k = \left(\frac{\partial x}{\partial p}\right)_k = \sum_{j=0}^{k-1} V_j \Delta t\left(\frac{\partial f}{\partial p}\right)_j \tag{A.8}$$

where $V_j$ is a matrix computed at the discrete time point $j$. From Eq. (A.8), for $\Delta t \to 0$, the sensitivity matrix can be expressed as:

$$S = \int_{t_0}^{t} \left( V(\tau) \left( \frac{\partial f}{\partial p} \right) \right) d\tau \tag{A.9}$$

## A.2 The partial derivative functions of the three-step-pathway model

According to Eqs. (5.25) the functions to be partially derived are:

$$f_1 = \frac{p_1}{1 + \left(\frac{P}{p_2}\right)^{p_3} + \left(\frac{p_4}{S}\right)^{p_5}} - p_6 x_1 \tag{A.10a}$$

$$f_2 = \frac{p_7}{1 + \left(\frac{P}{p_8}\right)^{p_9} + \left(\frac{p_{10}}{x_7}\right)^{p_{11}}} - p_{12} x_2 \tag{A.10b}$$

$$f_3 = \frac{p_{13}}{1 + \left(\frac{P}{p_{14}}\right)^{p_{15}} + \left(\frac{p_{16}}{x_8}\right)^{p_{17}}} - p_{18} x_3 \tag{A.10c}$$

$$f_4 = \frac{p_{19} x_1}{p_{20} + x_1} - p_{21} x_4 \tag{A.10d}$$

$$f_5 = \frac{p_{22} x_2}{p_{23} + x_2} - p_{24} x_5 \tag{A.10e}$$

$$f_6 = \frac{p_{25} x_3}{p_{26} + x_3} - p_{27} x_6 \tag{A.10f}$$

$$f_7 = \frac{p_{28} x_4 (S - x_7)}{p_{29} \left(1 + \frac{S}{p_{29}} + \frac{x_7}{p_{30}}\right)} - \frac{p_{31} x_5 (x_7 - x_8)}{p_{32} \left(1 + \frac{x_7}{p_{32}} + \frac{x_8}{p_{33}}\right)} \tag{A.10g}$$

$$f_8 = \frac{p_{31} x_5 (x_7 - x_8)}{p_{32} \left(1 + \frac{x_7}{p_{32}} + \frac{x_8}{p_{33}}\right)} - \frac{p_{34} x_6 (x_8 - P)}{p_{35} \left(1 + \frac{x_8}{p_{35}} + \frac{P}{p_{36}}\right)} \tag{A.10h}$$

From Eq. (A.10a):

$$\frac{\partial f_1}{\partial p_1} = \frac{1 + \left(\frac{P}{p_2}\right)^{p_3} + \left(\frac{p_4}{S}\right)^{p_5}}{\left[1 + \left(\frac{P}{p_2}\right)^{p_3} + \left(\frac{p_4}{S}\right)^{p_5}\right]^2} \tag{A.11a}$$

$$\frac{\partial f_1}{\partial p_2} = \frac{\frac{p_1 p_3}{p_2}\left(\frac{P}{p_2}\right)^{p_3}}{\left[1 + \left(\frac{P}{p_2}\right)^{p_3} + \left(\frac{p_4}{S}\right)^{p_5}\right]^2} \tag{A.11b}$$

$$\frac{\partial f_1}{\partial p_3} = \frac{-p_1\left(\frac{P}{p_2}\right)^{p_3} \ln\left(\frac{P}{p_2}\right)}{\left[1 + \left(\frac{P}{p_2}\right)^{p_3} + \left(\frac{p_4}{S}\right)^{p_5}\right]^2} \tag{A.11c}$$

$$\frac{\partial f_1}{\partial p_4} = \frac{-\frac{p_1 p_5}{p_4}\left(\frac{p_4}{S}\right)^{p_5}}{\left[1 + \left(\frac{P}{p_2}\right)^{p_3} + \left(\frac{p_4}{S}\right)^{p_5}\right]^2} \tag{A.11d}$$

$$\frac{\partial f_1}{\partial p_5} = \frac{-p_1\left(\frac{p_4}{S}\right)^{p_5} \ln\left(\frac{p_4}{S}\right)}{\left[1 + \left(\frac{P}{p_2}\right)^{p_3} + \left(\frac{p_4}{S}\right)^{p_5}\right]^2} \tag{A.11e}$$

$$\frac{\partial f_1}{\partial p_6} = -x_1 \tag{A.11f}$$

It can be clearly seen from Eqs. (A.11a-A.11e) that these partial derivative functions depend only on the parameters and controls. Thus $\frac{\partial f_1}{\partial p_1}, \frac{\partial f_1}{\partial p_2}, \cdots, \frac{\partial f_1}{\partial p_5}$ are pairwise linearly dependent. From Eq. (A.11f), depends on a state variable which will be a time-dependent profile and thus is linearly independent with the other partial derivative functions.

From Eq. (A.10b):

$$\frac{\partial f_2}{\partial p_7} = \frac{1 + \left(\frac{P}{p_8}\right)^{p_9} + \left(\frac{p_{10}}{x_7}\right)^{p_{11}}}{\left[1 + \left(\frac{P}{p_8}\right)^{p_9} + \left(\frac{p_{10}}{x_7}\right)^{p_{11}}\right]^2} \tag{A.12a}$$

$$\frac{\partial f_2}{\partial p_8} = \frac{\frac{p_7 p_9}{p_8}\left(\frac{P}{p_8}\right)^{p_9}}{\left[1 + \left(\frac{P}{p_8}\right)^{p_9} + \left(\frac{p_{10}}{x_7}\right)^{p_{11}}\right]^2} \tag{A.12b}$$

$$\frac{\partial f_2}{\partial p_9} = \frac{-p_7\left(\frac{P}{p_8}\right)^{p_9} \ln\left(\frac{P}{p_8}\right)}{\left[1 + \left(\frac{P}{p_8}\right)^{p_9} + \left(\frac{p_{10}}{x_7}\right)^{p_{11}}\right]^2} \tag{A.12c}$$

$$\frac{\partial f_2}{\partial p_{10}} = \frac{-\frac{p_7 p_{11}}{p_{10}}\left(\frac{p_{10}}{x_7}\right)^{p_{11}}}{\left[1 + \left(\frac{P}{p_8}\right)^{p_9} + \left(\frac{p_{10}}{x_7}\right)^{p_{11}}\right]^2} \tag{A.12d}$$

$$\frac{\partial f_2}{\partial p_{11}} = \frac{-p_7\left(\frac{p_{10}}{x_7}\right)^{p_{11}} \ln\left(\frac{p_{10}}{x_7}\right)}{\left[1 + \left(\frac{P}{p_8}\right)^{p_9} + \left(\frac{p_{10}}{x_7}\right)^{p_{11}}\right]^2} \tag{A.12e}$$

$$\frac{\partial f_2}{\partial p_{12}} = -x_2 \tag{A.12f}$$

Based on Eqs. (A.12b-A.12c), we have:

$$\frac{\partial f_2}{\partial p_8} = \left[\frac{-\frac{p_9}{p_8}}{\ln\left(\frac{P}{p_8}\right)}\right] \frac{\partial f_2}{\partial p_9} \tag{A.13}$$

Since the coefficient in Eq. (A.13) only depends on parameters and a control variable P, $\frac{\partial f_2}{\partial p_8}, \frac{\partial f_2}{\partial p_9}$ are linearly dependent. From Eqs. (A.12a-A.12d) it can be seen that:

$$\frac{\partial f_2}{\partial p_7} - \left[\frac{1 + \left(\frac{P}{p_8}\right)^{p_9}}{\frac{p_7 p_9}{p_8}\left(\frac{P}{p_8}\right)^{p_9}}\right] \frac{\partial f_2}{\partial p_8} + \left(\frac{p_{10} p_{11}}{p_7}\right) \frac{\partial f_2}{\partial p_{10}} = 0 \tag{A.14}$$

$$\frac{\partial f_2}{\partial p_7} + \left[\frac{1 + \left(\frac{P}{p_8}\right)^{p_9}}{p_7\left(\frac{P}{p_8}\right)^{p_9} \ln\left(\frac{P}{p_8}\right)}\right] \frac{\partial f_2}{\partial p_9} + \left(\frac{p_{10} p_{11}}{p_7}\right) \frac{\partial f_2}{\partial p_{10}} = 0 \tag{A.15}$$

Again, the coefficients in Eqs. (A.14-A.15) only depend on the parameters and the

control variable $P$, therefore, two different groups, $\frac{\partial f_2}{\partial p_7}, \frac{\partial f_2}{\partial p_8}, \frac{\partial f_2}{\partial p_{10}}$ and $\frac{\partial f_2}{\partial p_7}, \frac{\partial f_2}{\partial p_9}, \frac{\partial f_2}{\partial p_{10}}$ are linearly dependent, respectively. Similarly, according to Eqs. (A.12e-A.12f), $\frac{\partial f_2}{\partial p_{11}}, \frac{\partial f_2}{\partial p_{12}}$ are different from the other partial derivative functions and thus linearly independent with each other and also with other partial derivative functions.

Similar results can be obtained by comparing the partial derivative functions of Eq. (A.10c), since Eq. (A.10c) has the similar structure as Eq. (A.10b). Therefore, $\frac{\partial f_3}{\partial p_{14}}, \frac{\partial f_3}{\partial p_{15}}$ are linearly dependent in pair, $\frac{\partial f_3}{\partial p_{13}}, \frac{\partial f_3}{\partial p_{14}}, \frac{\partial f_3}{\partial p_{16}}$ and $\frac{\partial f_3}{\partial p_{13}}, \frac{\partial f_3}{\partial p_{15}}, \frac{\partial f_3}{\partial p_{16}}$ are linearly dependent in two groups, respectively.

From Eq. (A.10d):

$$\frac{\partial f_4}{\partial p_{19}} = \frac{x_1}{p_{20} + x_1} \tag{A.16a}$$

$$\frac{\partial f_4}{\partial p_{20}} = \frac{-p_{19} x_1}{(p_{20} + x_1)^2} \tag{A.16b}$$

$$\frac{\partial f_4}{\partial p_{20}} = -x_4 \tag{A.16c}$$

It can be clearly seen that $\frac{\partial f_4}{\partial p_{19}}, \frac{\partial f_4}{\partial p_{20}}, \frac{\partial f_4}{\partial p_{21}}$ are linearly independent. Similarly, according to Eqs. (A.10e-A.10f), there are no linear dependences among $\frac{\partial f_5}{\partial p_{22}}, \frac{\partial f_5}{\partial p_{23}}, \frac{\partial f_5}{\partial p_{24}}$ and $\frac{\partial f_6}{\partial p_{25}}, \frac{\partial f_6}{\partial p_{26}}, \frac{\partial f_6}{\partial p_{27}}$.

From Eq. (A.10g):

$$\frac{\partial f_7}{\partial p_{28}} = \frac{\frac{x_4(S-x_7)}{p_{29}}\left(1+\frac{S}{p_{29}}+\frac{x_7}{p_{30}}\right)}{\left(1+\frac{S}{p_{29}}+\frac{x_7}{p_{30}}\right)^2} \tag{A.17a}$$

$$\frac{\partial f_7}{\partial p_{29}} = \frac{-\frac{p_{28}x_4(S-x_7)}{p_{29}^2}\left(1+\frac{x_7}{p_{30}}\right)}{\left(1+\frac{S}{p_{29}}+\frac{x_7}{p_{30}}\right)^2} \tag{A.17b}$$

$$\frac{\partial f_7}{\partial p_{30}} = \frac{\frac{p_{28}x_4(S-x_7)}{p_{29}p_{30}}\left(\frac{x_7}{p_{30}}\right)}{\left(1+\frac{S}{p_{29}}+\frac{x_7}{p_{30}}\right)^2} \tag{A.17c}$$

$$\frac{\partial f_7}{\partial p_{31}} = \frac{-\frac{x_5(x_7-x_8)}{p_{32}}\left(1+\frac{x_7}{p_{32}}+\frac{x_8}{p_{33}}\right)}{\left(1+\frac{x_7}{p_{32}}+\frac{x_8}{p_{33}}\right)^2} \tag{A.17d}$$

$$\frac{\partial f_7}{\partial p_{32}} = \frac{\frac{p_{31}x_5(x_7-x_8)}{p_{32}^2}\left(1+\frac{x_8}{p_{33}}\right)}{\left(1+\frac{x_7}{p_{32}}+\frac{x_8}{p_{33}}\right)^2} \tag{A.17e}$$

$$\frac{\partial f_7}{\partial p_{33}} = -\frac{\frac{p_{31}x_5(x_7-x_8)}{p_{32}p_{33}}\left(\frac{x_8}{p_{33}}\right)}{\left(1+\frac{x_7}{p_{32}}+\frac{x_8}{p_{33}}\right)^2} \tag{A.17f}$$

From Eqs. (A.17a-A.17c), $\frac{\partial f_7}{\partial p_{28}}, \frac{\partial f_7}{\partial p_{29}}, \frac{\partial f_7}{\partial p_{30}}$ are linearly dependent in one group. But $\frac{\partial f_7}{\partial p_{31}}, \frac{\partial f_7}{\partial p_{32}}, \frac{\partial f_7}{\partial p_{33}}$ are linearly independent, based on Eqs. (A.17d-A.17f). From Eq. (A.10h):

$$\frac{\partial f_8}{\partial p_{34}} = \frac{-\frac{x_6(x_8-P)}{p_{35}}\left(1+\frac{x_8}{p_{35}}+\frac{P}{p_{36}}\right)}{\left(1+\frac{x_8}{p_{35}}+\frac{P}{p_{36}}\right)^2} \tag{A.18a}$$

$$\frac{\partial f_8}{\partial p_{35}} = \frac{\frac{p_{34}x_6(x_8-P)}{p_{35}^2}\left(1+\frac{P}{p_{36}}\right)}{\left(1+\frac{x_8}{p_{35}}+\frac{P}{p_{36}}\right)^2} \tag{A.18b}$$

$$\frac{\partial f_8}{\partial p_{36}} = \frac{-\frac{p_{34}x_6(x_8-P)}{p_{35}p_{36}}\left(\frac{P}{p_{36}}\right)}{\left(1+\frac{x_8}{p_{35}}+\frac{P}{p_{36}}\right)^2} \tag{A.18c}$$

It can be seen from Eqs. (A.18a-A.18a) that $\frac{\partial f_8}{\partial p_{35}}, \frac{\partial f_8}{\partial p_{36}}$ are linearly dependent, but

$\frac{\partial f_8}{\partial p_{34}}$ is linearly independent with $\frac{\partial f_8}{\partial p_{35}}, \frac{\partial f_8}{\partial p_{36}}$.