

# Tailored Indirect Algorithms for Efficient On-line Optimization of Batch and Semi-Batch Processes

**Dissertation**

zur Erlangung des akademischen Grades

**Doktoringenieur**

**(Dr.-Ing.)**

von: M.Sc. Erdal Aydin  
geb. am: 6. Januar 1990  
in: Istanbul, Türkei

genehmigt durch die Fakultät für Verfahrens- und Systemtechnik der Otto-von-Guericke-Universität Magdeburg

Betreuer: Prof. Dr. Ing. Kai Sundmacher  
Gutachter: Prof. Dr. Dominique Bonvin  
Prof. Dr. rer. nat. Sebastian Sager

eingereicht am: 27. März 2018  
Promotionkolloquium am: 29. Mai 2018



*dedicated to my parents H. İsmet - Selma,*

*aunt Servet,*

*uncle İsmail...*



MAX-PLANCK-GESELLSCHAFT

# ABSTRACT

The trend toward high-quality, low-volume and high-added value production has put more emphasis on batch and semi-batch processing due to its increased flexibility of operations. Dynamic optimization plays an important role toward improving the operation of batch and semi-batch processes. In addition, nonlinear model predictive control (NMPC) is an important tool for the real-time optimization of batch and semi-batch processes under uncertainty. However, the fact that the transient behaviour as well as the flexibility decrease with respect to time leads to very challenging optimization problems.

The preferred strategy to solve constrained nonlinear dynamic optimization problems is usually to use a so-called direct method. Nevertheless, based on the problem type at hand and the solution algorithm used, direct methods may lead to computational complexity. In particular, the large prediction horizons required in the NMPC of batch and semi-batch processes increase the real-time computational effort because of expensive matrix factorizations in the solution steps, especially at the beginning of the batch. The computational delay associated with advanced control methods is usually underestimated in theoretical studies. However, this delay may contribute to suboptimal or, worse, infeasible operation in real-life applications.

Alternatively, indirect methods based on Pontryagin's Minimum Principle (PMP) could efficiently deal with the optimization of batch and semi-batch processes. In fact, the interplay between states and co-states in the context of PMP might turn out to be computationally quite efficient. The main indirect solution technique is the shooting method, which however often leads to convergence problems and instabilities caused by the integration of the co-state equations forward in time. It has been extensively argued that indirect methods are usually non-convergent and inefficient for constrained problems. This study proposes an alternative, convergent and effective indirect solution technique. Instead of integrating the states and co-states simultaneously forward in time, the proposed algorithm parameterizes the inputs and integrates the state equations forward in time and the co-state equations backward in time, thereby leading to a gradient-based optimization approach. Constraints are handled by indirect adjoining to the Hamiltonian function, which allows meeting the active constraints explicitly at every iteration step. The performance of the solution strategy is compared to direct methods through three different case studies. The results show that the proposed PMP-based quasi-

Newton strategy is effective in dealing with complicated constraints and is quite competitive computationally.

In addition, this work suggests using the proposed indirect solution technique in the context of shrinking-horizon NMPC under uncertainty. Uncertainties can be handled by the introduction of time-varying backoff terms for the path constraints. The resulting NMPC algorithm is applied to a two-phase semi-batch reactor for the hydroformylation of 1-dodecene in the presence of uncertainty, and its performance is compared to that of NMPC that uses a direct simultaneous optimization method. The results show that the proposed algorithm (i) results in feasible operation for different uncertainty realizations both within batch or from batch to batch, and (ii) is much faster than direct simultaneous NMPC, especially at the beginning of the batch. In addition, a modification of the PMP-based NMPC scheme is proposed to enforce the active constraints to reduce the real-time computational effort further.

This thesis also details the combination of an indirect solution scheme together with an alternative parameterization scheme. The idea is to parameterize the sensitivity-seeking input arcs in a parsimonious way so as to decrease the computational load of constrained nonlinear dynamic optimization problems. The proposed method is tested on the simulated examples of a batch binary distillation column with terminal purity constraints and a two-phase semi-batch hydroformylation reactor with a complex path constraint. The performance of the proposed indirect parsimonious solution scheme is compared with those of a fully parameterized PMP-based and a direct simultaneous solution approaches. It is observed that the combination of the indirect approach with parsimonious input parameterization can result in significant reduction in computational time. Finally, in this work, the application of parsimonious input parameterization to the shrinking-horizon NMPC is suggested in order to minimize the computational delay in feedback. The proposed approach is illustrated on two case studies in the presence of uncertainty. The results show that the suggested parsimonious shrinking-horizon NMPC performs very similarly to the standard shrinking-horizon NMPC in terms of cost, is computationally much faster than the standard shrinking-horizon NMPC especially at the beginning of the batch and is robust to plant-model mismatch.

# ZUSAMMENFASSUNG

Der Trend zu einer qualitativ hochwertigen, kleinvolumigen und wertschöpfungsintensiven Produktion hat Semi-Batch-Prozesse aufgrund ihrer höheren Flexibilität der Betriebsweise stärker in den Vordergrund gerückt. Dynamische Optimierung spielt eine wichtige Rolle bei der Verbesserung der Fahrweise von Batch und Semi-Batch-Prozessen. Darüber hinaus ist die nichtlineare modellprädiktive Regelung (NMPC) ein wichtiges Werkzeug zur Echtzeit-Optimierung von Batch und Semi-Batch-Prozessen unter Unsicherheit. Das transiente Verhalten sowie die zeitlich abnehmende Flexibilität führen jedoch zu sehr anspruchsvollen Optimierungsproblemen.

Die bevorzugte Strategie zur Lösung von Problemen im Zusammenhang mit der nichtlinearen dynamischen Optimierung ist normalerweise die Verwendung einer sogenannten direkten Methode. Dennoch können direkte Methoden, basierend auf dem vorliegenden Problemtyp und dem verwendeten Lösungsalgorithmus, zu einer Komplexität der Berechnungen führen. Insbesondere die großen Prognosehorizonte, die im NMPC von Semi-Batch-Prozessen benötigt werden, erhöhen den Echtzeit-Rechenaufwand durch teure Matrixfaktorisierungen in den Lösungsschritten, insbesondere zu Prozessbeginn. Die Verzögerung aufgrund der Berechnungszeit, die mit modernen Regelungsmethoden verbunden ist, wird in der Regel in theoretischen Studien unterschätzt. Diese Verzögerung kann jedoch zu einem suboptimalen oder sogar nicht realisierbaren Betrieb in der Praxis führen.

Alternativ könnten indirekte Methoden, die auf Pontryagin's Minimum Prinzip (PMP) basieren, effizient mit der Optimierung von Batch und Semi-Batch-Prozessen umgehen. Tatsächlich könnte sich das Zusammenspiel von Zuständen und Ko-Zuständen im Rahmen von PMP rechnerisch als effizient erweisen. Die wichtigste indirekte Lösungsmethode ist das Schiess-Verfahren, das jedoch häufig zu Konvergenzproblemen und Instabilitäten führt, die durch die Vorwärts-Integration der Gleichungen der Ko-Zustände verursacht werden. Generell wurde ausgiebig argumentiert, dass indirekte Methoden in der Regel nicht konvergent und ineffizient für beschränkte Probleme sind. Diese Arbeit schlägt jedoch eine alternative, konvergente und effektive indirekte Lösungstechnik vor. Anstatt die Zustände und Ko-Zustände gleichzeitig vorwärts in der Zeit zu integrieren, parametrisiert der vorgeschlagene Algorithmus die Eingänge und integriert die Zustandsgleichungen vorwärts in der Zeit und die Ko-Zustandsgleichungen rückwärts in der Zeit, was zu einem Gradienten-

basierten Optimierungsansatz führt. Nebenbedingungen werden durch die indirekte Anbindung an die Hamilton-Funktion behandelt, die es erlaubt, die aktiven Nebenbedingungen bei jedem Iterationsschritt explizit zu erfüllen. Die Leistungsfähigkeit der Lösungsstrategie wird anhand von drei verschiedenen Fallstudien mit direkten Methoden verglichen. Die Ergebnisse zeigen, dass die vorgeschlagene PMP-basierte Quasi-Newton-Strategie effektiv im Umgang mit komplizierten Nebenbedingungen ist und rechnerisch durchaus konkurrenzfähig ist.

Darüber hinaus schlägt diese Arbeit die Verwendung der vorgeschlagenen indirekten Lösungstechnik im Rahmen der shrinking horizon NMPC unter Unsicherheit vor. Unsicherheiten können durch die Einführung von zeitvariablen Backoffs für die Pfadbeschränkungen beseitigt werden. Der resultierende NMPC-Algorithmus wird auf einen zweiphasigen Semi-Batch-Reaktor für die Hydroformylierung von 1-dodecen in Gegenwart von Unsicherheit angewendet und seine Leistung mit einem NMPC-Algorithmus verglichen, der eine direkte simultane Optimierungsmethode verwendet. Die Ergebnisse zeigen, dass der vorgeschlagene Algorithmus Ergebnisse im machbarem Betrieb für verschiedene Unsicherheitsrealisierungen erzielt und zwar sowohl innerhalb des Batch oder von Batch zu Batch. Er ist viel schneller als der direkte simultane NMPC-Algorithmus, vor allem zu Beginn des Batch. Darüber hinaus wird eine Modifikation des PMP-basierten NMPC-Schemas vorgeschlagen, um die aktiven Nebenbedingungen durchzusetzen und den Echtzeit-Berechnungsaufwand weiter zu reduzieren.

Diese Arbeit beschreibt auch die Kombination eines indirekten Lösungsschemas mit einem alternativen Parametrierungsschema. Die Idee ist es, die sensitivitätssuchenden Eingänge in einer sparsamen Art zu parametrisieren, um die rechnerische Belastung von nichtlinearen dynamischen Optimierungsproblemen unter Nebenbedingungen zu verringern. Die vorgeschlagene Methode wird an simulierten Beispielen einer Batch-Binärdestillationskolonne mit endständigen Reinheitsanforderungen und einem zweiphasigen Semi-Batch-Hydroformylierungsreaktor mit komplexer Pfadbeschränkung getestet. Die Leistung des vorgeschlagenen indirekten sparsamen Lösungsschemas wird mit der eines vollständig parametrisierten PMP-basierten und eines direkten simultanen Lösungsansatzes verglichen. Es wird gezeigt, dass die Kombination des indirekten Ansatzes mit sparsamer Eingabeparametrierung zu einer signifikanten Verkürzung der Rechenzeit führen kann.

Schließlich wird in dieser Arbeit die Kombination von einfachen Lösungsmodellen mit sparsamer Eingangsparmetrierung im Rahmen der shrinking horizon NMPC vorgeschlagen, um die rechnerische Verzögerung der Rückkopplung zu minimieren. Lösungsmodelle nutzen die nominale optimale Lösung, um sparsame Parametrierungen (insbesondere für sensitiv-suchende Lösungs-bereiche) vorzuschlagen, die zu einer schnellen Optimierung führen. Der vorgeschlagene Ansatz wird anhand von zwei Fallstudien in Gegenwart von Unsicherheit veranschaulicht.

## ACKNOWLEDGEMENTS

First of all, it gives me great pleasure to express my gratitude to my ‘*Doktorvater*’, Prof. Kai Sundmacher, for the scientific position he offered me, for the interesting scientific subject and the support he provided throughout my years at the Max Planck Institute in Magdeburg. He was the only one who believed in me in the very beginning. He gave me the total scientific freedom and always stood by my decisions. Besides that, he showed me the importance of being able to be independent in research. In addition, he always encouraged me for visiting international conferences and supported me financially. Thank you very much for everything, but most importantly for your kindness and infinite support Prof. Sundmacher!

Secondly, I would like to gratefully thank Prof. Dominique Bonvin, my ‘*unofficial second supervisor*’ and scientific collaborator. He has always been there for scientific discussions and sharing ideas. This thesis would not be possible without him. I will never forget his professionalism, exactness, support, kindness and patience. It has been always an honor and pleasure for me to have the chance to collaborate with you Prof. Bonvin. Thank you very much!

I would like to thank Prof. Sebastian Sager for accepting to be in the defense committee and for the scientific critiques and suggestions.

I would like to express my gratitude to my former advisor Prof. Yaman Arkun for everything he has done for me. His door was always open for discussions and sharing ideas. I learned quite a lot from his side. In addition, in his group, I had the chance to meet with my ‘*unofficial mentor*’, but more importantly a very close friend, Dr. Hasan Şıldır, who has always been there for me for any kind of support! Thank you Hasan!

I would like to acknowledge all my former professors at ODTÜ (METU) Chemical Engineering Department. Special thanks to Prof. Deniz Uner, Prof. Ismail Tosun, Prof. Metin Turkyay and Prof. Burak Erman who always encouraged and supported me to follow an academic path.

My colleagues played a very important role in my Ph.D time. They always supported me, both scientifically and mentally, and they made me feel like being at home. They were more like friends, rather than being only colleagues. Special thanks to Ali elSibai for the immense

support and Jens Bremer for the fruitful scientific discussions, encouragements and suggestions. Besides, I would like to express my gratitude to (in the order of surname): Kristin Czyborra, Robert Flassig, Andreas Himmel, Michael Jokiell, Nicolas M. Kaiser, Georg Liesche, Kevin McBride, Georgios Papakonstantinou, Dennis Pischel, Karsten H.G. Raetze, Cathleen D.-Richter, Dominik Schack, Giulia Slaviero, Antonio Sorrentino and Marcus Wenzel. I also acknowledge the fruitful discussion with Diogo Rodrigues from EPFL.

Last but not the least, I would like to gratefully thank my family: my parents, my aunt and my uncle, who always supported me throughout my life. I feel myself blessed for having them in my life. It is a great pleasure for me to dedicate this thesis to them.

# PREFACE

This work has been carried out during my employment as a scientific member at the Max Planck Institute for Dynamics of Complex Technical Systems, between October 2015 and January 2018. Several publications were prepared in the course of this thesis. The chapters containing parts of these papers are summarized as follows:

- The proposed novel, effective and convergent indirect (PMP-based) dynamic optimization algorithm in **Chapter 3** to be applied to the constrained semi-batch processes has been published in (Aydin et al., 2017a).
- The parsimonious indirect algorithm detailed in **Chapter 4** has been published in (Aydin et al., 2017b).
- The application of the proposed fully parameterized indirect algorithm to the shrinking horizon NMPC (sh-NMPC) discussed in **Chapter 5** has been published in (Aydin et al., 2018b).
- The computationally efficient sh-NMPC algorithm that uses parsimonious input parameterization, which is detailed in **Chapter 6**, has been submitted as (Aydin et al., 2018a).
- Some parts of the detailed comparison of the direct and indirect methods, along with the available solution algorithms and specific tailoring strategies for the indirect methods discussed in **Chapter 2** and **Chapter 4** have been submitted as (Aydin et al., 2018c).

Aydin, E., Bonvin, D., & Sundmacher, K. (2017a). Dynamic optimization of constrained semi-batch processes using Pontryagin's minimum principle—An effective quasi-Newton approach. *Computers & Chemical Engineering*, 99, 135-144. \*

Aydin, E., Bonvin, D., & Sundmacher, K. (2017b). Dynamic Optimization of Constrained Semi-Batch Processes using Pontryagin's Minimum Principle and Parsimonious Parameterization. *Computer Aided Chemical Engineering*, 40, 2041-2046. \*\*

Aydin, E., Bonvin, D., & Sundmacher, K. (2018a). Computationally Efficient NMPC for Batch and Semi-Batch Processes Using Parsimonious Input Parameterization. *Journal of Process Control*, 66, 12-22. \*\*\*

Aydin, E., Bonvin, D., & Sundmacher, K. (2018b). NMPC using Pontryagin's Minimum Principle-Application to a two-phase semi-batch hydroformylation reactor under uncertainty. *Computers & Chemical Engineering*, 108, 47-56. \*\*\*

Aydin, E., Bonvin, D., & Sundmacher, K. (2018c). Toward Fast Dynamic Optimization: An Indirect Algorithm that Uses Parsimonious Input Parameterization. *Industrial & Engineering Chemistry Research*, Submitted. \*\*

# CONTENTS

|  |             |
|--|-------------|
| <b>ABSTRACT</b> .....  | <b>iv</b>   |
| <b>ZUSAMMENFASSUNG</b> .....   | <b>vi</b>   |
| <b>ACKNOWLEDGEMENTS</b> .....  | <b>ix</b>   |
| <b>PREFACE</b> .....   | <b>xi</b>   |
| <b>LIST OF TABLES</b> .....  | <b>xiv</b>  |
| <b>LIST OF FIGURES</b> .....   | <b>xv</b>   |
| <b>NOTATION</b> .....  | <b>xvii</b> |
| <b>ABBREVIATIONS</b> .....   | <b>xix</b>  |
| <br>   |             |
| <b>PART I: FUNDAMENTALS</b> .....  | <b>xx</b>   |
| <b>1 INTRODUCTION</b> .....  | <b>1</b>    |
| 1.1 Motivation and Scope.....  | 1           |
| 1.2 Contribution of the Thesis.....  | 8           |
| 1.3 Structure of the Thesis.....   | 9           |
| <b>2 DYNAMIC OPTIMIZATION: METHODS AND ALGORITHMS</b> .....  | <b>11</b>   |
| 2.1 Direct Methods .....   | 12          |
| 2.1.1 <i>NLP Solution Algorithms</i> .....   | 13          |
| 2.1.2 <i>Computational Aspects</i> .....   | 17          |
| 2.2 Indirect Methods .....   | 18          |
| <br>   |             |
| <b>PART II: EFFICIENT DYNAMIC OPTIMIZATION USING INDIRECT METHODS</b><br>.....                               | <b>24</b>   |
| <b>3 DYNAMIC OPTIMIZATION OF CONSTRAINED BATCH AND SEMI-BATCH<br/>PROCESSES USING INDIRECT METHODS</b> ..... | <b>25</b>   |
| 3.1 Solution Methodology .....   | 26          |
| 3.2 Case Studies .....   | 32          |
| 3.2.1 <i>Non-isothermal semi-batch reactor with a heat-removal constraint</i> .....                          | 32          |
| 3.2.2 <i>Batch binary distillation column with terminal purity constraints</i> .....                         | 39          |
| 3.2.3 <i>Fed-batch hydroformylation reactor with path constraints</i> .....                                  | 45          |
| 3.3 Summary .....  | 51          |

|  |            |
|--|------------|
| <b>4 DYNAMIC OPTIMIZATION COMBINING PONTRYAGIN'S MINIMUM PRINCIPLE AND PARSIMONIOUS INPUT PARAMETERIZATION .....</b> | <b>53</b>  |
| 4.1 Solution Methodology .....   | 54         |
| 4.2 Computation of Gradients .....   | 57         |
| 4.3 Case Studies .....   | 58         |
| 4.3.1 <i>Batch binary distillation with terminal purity constraints</i> .....  | 59         |
| 4.3.2 <i>Fed-batch hydroformylation reactor with path constraints</i> .....  | 61         |
| 4.4 Summary .....  | 63         |
| <br>   |            |
| <b>PART III: EFFICIENT NONLINEAR MODEL PREDICTIVE CONTROL .....</b>  | <b>64</b>  |
| <b>5 NONLINEAR MODEL PREDICTIVE CONTROL USING PONTRYAGIN'S MINIMUM PRINCIPLE.....</b>                                | <b>65</b>  |
| 5.1 Shrinking-horizon NMPC Problem.....  | 66         |
| 5.2 Case Study: Two-phase Semi-batch Hydroformylation Reactor under Uncertainty .....                                | 69         |
| 5.2.1 <i>Problem formulation</i> .....   | 69         |
| 5.2.2 <i>Estimation of time-varying back-offs</i> .....  | 71         |
| 5.2.3 <i>NMPC for product maximization</i> .....   | 73         |
| 5.2.4 <i>NMPC for batch time minimization</i> .....  | 78         |
| 5.2.5 <i>NMPC with constraint tracking</i> .....   | 80         |
| 5.3 Summary .....  | 83         |
| <b>6 NONLINEAR MODEL PREDICTIVE CONTROL USING PARSIMONIOUS INPUT PARAMETERIZATION.....</b>                           | <b>85</b>  |
| 6.1 The Parsimonious Shrinking-Horizon NMPC .....  | 86         |
| 6.2 Case Studies .....   | 89         |
| 6.2.1 <i>Batch binary distillation with terminal purity constraints under uncertainty</i> .....                      | 89         |
| 6.2.2 <i>Semi-batch reactor for the hydroformylation of 1-dodecene under uncertainty</i> .....                       | 95         |
| 6.3 Summary .....  | 102        |
| <b>7 SUMMARY AND OUTLOOK .....</b>   | <b>103</b> |
| 7.1 Summary .....  | 103        |
| 7.2 Outlook.....   | 105        |
| <b>8 BIBLIOGRAPHY .....</b>  | <b>107</b> |
| <b>9 APPENDICES .....</b>  | <b>114</b> |
| <b>APPENDIX 1: SIMPLE CASE STUDY SOLVED WITH THE FULLY PARAMETERIZED PMP-BASED METHOD.....</b>                       | <b>115</b> |
| <b>APPENDIX 2: MODEL PARAMETERS FOR PROBLEM 3.2.3 .....</b>  | <b>116</b> |

# LIST OF TABLES

|   |    |
|---|----|
| <b>Table 2.1:</b> Selected recent publications.....   | 23 |
| <b>Table 3.1:</b> Model parameters and initial conditions for Problem 3.2.1.....  | 33 |
| <b>Table 3.2:</b> Comparison of the indirect PMP-based, direct simultaneous and analytical solution strategies for Problem 3.2.1..... | 38 |
| <b>Table 3.3:</b> Model parameters and initial conditions for Problem 3.2.2.....  | 41 |
| <b>Table 3.4:</b> Comparison of the indirect PMP-based and direct simultaneous strategies for Problem 3.2.2.....                      | 44 |
| <b>Table 3.5:</b> Comparison of the indirect PMP-based and direct simultaneous strategies for Problem 3.2.3.....                      | 48 |
| <b>Table 5.1:</b> Parametric variations for the sh-NMPC problem.....  | 72 |
| <b>Table 5.2:</b> Tuning parameters for the PMP-based and DSM-based algorithms.....   | 73 |
| <b>Table 5.3:</b> Performance of PMP-based NMPC for free final time.....  | 78 |
| <b>Table 5.4:</b> Performance of PMP-based NMPC without and with constraint tracking for fixed final time.....                        | 82 |
| <b>Table 6.1:</b> Nominal model parameters and initial conditions for the batch distillation.....                                     | 90 |
| <b>Table 6.2:</b> Nominal parameter values and corresponding variations for the hydroformylation process.....                         | 95 |

# LIST OF FIGURES

|   |    |
|---|----|
| <b>Figure 1.1:</b> The illustration of sh-NMPC. ....  | 5  |
| <b>Figure 1.2:</b> The corresponding CPU times as the input grid gets finer. ....   | 6  |
| <b>Figure 1.3:</b> The convergence profiles for the toy example. ....   | 7  |
| <b>Figure 2.1:</b> Dynamic optimization methods in the literature. ....   | 22 |
| <b>Figure 3.1:</b> Illustration of the proposed indirect adjoining and activation steps. ....   | 30 |
| <b>Figure 3.2.a:</b> Optimal input and state profiles computed via the PMP-based method and the analytical solution 2 for Problem 3.2.1. ....           | 35 |
| <b>Figure 3.2.b:</b> Optimal input and state profiles computed via the direct simultaneous method and the analytical solution 3 for Problem 3.2.1. .... | 36 |
| <b>Figure 3.3.a:</b> Switching function $s_u$ for the PMP-based solution and the analytical solution 2. ....  | 37 |
| <b>Figure 3.3.b:</b> Switching function $s_T$ for the PMP-based solution and the analytical solution 2. ....  | 37 |
| <b>Figure 3.4.a:</b> Switching function $s_u$ for the direct simultaneous solution and the analytical solution 3. ....                                  | 37 |
| <b>Figure 3.4.b:</b> Switching function $s_T$ for the direct simultaneous solution and the analytical solution 3. ....                                  | 37 |
| <b>Figure 3.5:</b> Computational times for different discretizations N of Problem 3.2.1. ....   | 39 |
| <b>Figure 3.6:</b> Schematic of the batch distillation column. ....   | 42 |
| <b>Figure 3.7:</b> Optimal input and state profiles computed via the PMP-based method and the direct simultaneous strategy for Problem 3.2.2. ....      | 43 |
| <b>Figure 3.8:</b> Switching function $s_r$ for Problem 3.2.2. ....   | 44 |
| <b>Figure 3.9:</b> Computational times for different discretizations of Problem 3.2.2. ....   | 45 |
| <b>Figure 3.10.</b> Hydroformylation reaction network. ....   | 46 |
| <b>Figure 3.11.</b> Optimal input and state profiles computed via the PMP-based method and the direct simultaneous strategy for Problem 3.2.3. ....     | 49 |
| <b>Figure 3.12.</b> Switching functions obtained for Problem 3.2.3. ....  | 50 |
| <b>Figure 3.13.</b> Computational times for different discretizations of Problem 3.2.3. ....  | 51 |
| <b>Figure 4.1:</b> Fully parameterized DSM and PMP as well as parsimonious parameterized PMP results for Problem 4.2.1. ....                            | 60 |
| <b>Figure 4.2:</b> Computational times for Problem 4.3.1. ....  | 60 |
| <b>Figure 4.3:</b> Optimal solutions for Problem 4.3.2. ....  | 62 |
| <b>Figure 4.4:</b> Computational times of different methods for Problem 4.3.2. ....   | 63 |
| <b>Figure 5.1.a:</b> Open loop Monte Carlo simulations for total pressure. ....   | 74 |
| <b>Figure 5.1.b:</b> Open loop Monte Carlo simulations for product concentration. ....  | 74 |
| <b>Figure 5.2:</b> Performance of the observer for a single batch. ....   | 74 |
| <b>Figure 5.3:</b> NMPC profiles with DSM-based and PMP-based algorithms for a particular batch and fixed final time. ....                              | 75 |
| <b>Figure 5.4:</b> Computational times with DSM-based and PMP-based algorithms. ....  | 76 |
| <b>Figure 5.5:</b> NMPC profiles with PMP for 40 uncertainty realizations and fixed final time. ....  | 77 |

|  |     |
|--|-----|
| <b>Figure 5.6:</b> NMPC profiles with PMP for 40 uncertainty realizations and batch time minimization.....                         | 79  |
| <b>Figure 5.7:</b> NMPC with constraint tracking for 40 different uncertainty realizations and fixed final time. ....              | 81  |
| <b>Figure 5.8:</b> Computational times using PMP-based NMPC without and with constraint tracking.....                              | 82  |
| <b>Figure 6.1:</b> Optimal <i>open-loop</i> profiles for Problem 6.2.1 with full direct simultaneous and parsimonious methods..... | 91  |
| <b>Figure 6.2:</b> Optimal <i>closed-loop</i> profiles for Problem 6.2.1 with standard and parsimonious sh-NMPC.....               | 93  |
| <b>Figure 6.3:</b> Computational times with standard and parsimonious sh-NMPC for the batch distillation.....                      | 93  |
| <b>Figure 6.4:</b> Optimal <i>closed-loop</i> profiles for Problem 6.2.1 with parsimonious sh-NMPC for 40 different batches. ....  | 94  |
| <b>Figure 6.5:</b> Optimal <i>open-loop</i> profiles for Problem 6.2.2 with full direct simultaneous and parsimonious methods..... | 97  |
| <b>Figure 6.6:</b> Hierarchical sh-NMPC structure for the semi-batch hydroformylation reactor...                                   | 98  |
| <b>Figure 6.7:</b> Optimal <i>closed-loop</i> profiles for Problem 6.2.2 with standard and parsimonious sh-nmpc. ....              | 99  |
| <b>Figure 6.8:</b> Computational times with standard and parsimonious sh-NMPC for the hydroformylation semi-batch reactor. ....    | 100 |
| <b>Figure 6.9:</b> Optimal <i>closed-loop</i> profiles for Problem 6.2.2 with parsimonious sh-NMPC for 40 different batches. ....  | 101 |

# NOTATION

|                      |  |  |
|----------------------|--|--|
| $t_f$                | Final time   | h, min, s                                  |
| $\phi$               | Objective function   | mol, kmol, mol/L                           |
| $x$                  | State equations  | mol, kmol, mol/L                           |
| $t$                  | Time   | h, s, min                                  |
| $\dot{x}$            | Dynamics of the system equations                                   | mol/h, kmol/h, mol/s, mol/min              |
| $x_0$                | Initial conditions of the states                                   | mol, kmol, mol/L                           |
| $u$                  | Inputs of the optimization problem                                 | K, mol/h, kmol/h, mol/s, mol/min           |
| $S$                  | Inequality Path constraints  | J/h, kmol/h, mol/s, mol/h, kmol/min, L     |
| $T$                  | Inequality Terminal constraints                                    | kmol/kmol                                  |
| $f$                  | Objective function of an NLP                                       | -  |
| $g$                  | Inequality constraints   | -  |
| $h$                  | Equality constraints   | -  |
| $L$                  | Lagrangian   | -  |
| $F$                  | Right hand side of the dynamics of the state equations             | J/h, kmol/h, mol/s, mol/min, kmol/h, L/min |
| $\tilde{F}, \hat{F}$ | Normalized right hand sides of the dynamics of the state equations | J, kmol, mol, mol, mol, kmol, L            |
| $\lambda$            | Lagrange multipliers for equality constraints (co-states)          | -  |
| $G$                  | Right hand side of the dynamics of the co-state equations          | -  |
| $\mu$                | Lagrange multipliers for inequality constraints                    | -  |
| $v$                  | Lagrange multipliers for terminal constraints                      | -  |
| $z$                  | Slack variables  | -  |
| $\nabla_x$           | Gradient   | -  |
| $\nabla_{xx}$        | Hessian  | -  |
| $d_x, d_z$           | Change of variables in an iteration                                | -  |
| $e$                  | Identity matrix  | -  |
| $s$                  | Switching function   | -  |
| $U$                  | Augmented input matrix   | K, mol/h, kmol/h, mol/s, mol/min           |
| $n$                  | Relative degree of a path constraint                               | -  |
| $H$                  | Hamiltonian  | -  |
| $N$                  | Discretization level for the inputs                                | -  |
| $h$                  | Number of iterations   | -  |
| $M$                  | Augmented constraint matrix  | J/h, kmol/h, mol/s, mol/h, kmol/h, L       |
| $\alpha_0$           | Initial step size  | -  |
| $K$                  | Penalty parameter  | -  |
| $\varepsilon$        | Optimality threshold value   | -  |
| $B$                  | Approximated Hessian   | -  |
| $\beta$              | BFGS parameter   | -  |
| $V$                  | Reactor volume   | L  |
| $q_{rx}$             | Heat release from the reactor                                      | J/h  |
| $T(t)$               | Temperature  | K  |

|                  |  |                                |
|------------------|--|--------------------------------|
| $c$              | Concentration                          | mol/L                          |
| $r$              | Internal reflux ratio                  | -                              |
| $L$              | Liquid flow rate in the distillation   | kmol/h                         |
| $V$              | Vapor flow rate in the distillation    | kmol/h                         |
| $MH$             | Molar hold-up per plate                | kmol                           |
| $V_{liq}$        | Liquid volume inside the reactor       | mL                             |
| $V_{gas}$        | Gas volume inside the reactor          | mL                             |
| $\pi$            | Parsimonious input vector              | s, h, mol/min                  |
| $t_k$            | Time at sampling instant $k$           | h, s, min                      |
| $\hat{x}_k$      | Estimated states                       | mol, kmol, mol/L               |
| $\tilde{\theta}$ | Uncertain model parameters             | -                              |
| $\bar{\theta}$   | Estimated model parameters             | -                              |
| $\delta$         | Sample time of the controller          | h, s, min                      |
| $b$              | Back-off term for the path constraints | kmol/h, mol/s, mol/h, kmol/min |

# ABBREVIATIONS

|               |   |
|---------------|---|
| NMPC          | Nonlinear Model Predictive Control      |
| NLP           | Nonlinear Programming                   |
| MPC           | Linear Model Predictive Control         |
| MHE           | Moving Horizon Estimation               |
| PMP           | Pontryagin's Minimum Principle          |
| DSM, DS       | Direct Simultaneous Method              |
| SQP           | Sequential Quadratic Programming        |
| QP            | Quadratic Problem                       |
| NCO           | Necessary Conditions of Optimality      |
| BFGS          | Broyden-Fletcher-Goldfarb-Shanno        |
| $H_2$         | Hydrogen                                |
| <i>CO</i>     | Carbon monoxide                         |
| <i>nC12en</i> | 1-dodecene                              |
| <i>iC12en</i> | internal dodecene                       |
| <i>nc13al</i> | n-tridecanal                            |
| <i>ic13al</i> | iso-aldehyde                            |
| <i>nC12an</i> | n-dodecane                              |
| FTIR          | Fourier Transform Infrared Spectroscopy |
| PID           | Proportional-Integral-Derivative        |
| MV            | Manipulated Variable                    |
| CV            | Controlled Variable                     |
| PIM           | Parsimonious Input Model                |

# **PART I**

## **Fundamentals**

*Science cannot solve the ultimate mystery of nature. And that is because, in the last analysis, we ourselves are a part of the mystery that we are trying to solve.*

---

Max Planck (1858 – 1947)

# 1 INTRODUCTION

## 1.1 Motivation and Scope

Batch and semi-batch processes have wide application areas in the specialty industries for the production of low-volume, high-added-value products. Typical examples are the pharmaceutical, food, fine chemical and microelectronic industries. Batch processing has often been used to scale-up processes from the laboratory to large-scale industrial facilities. However, in recent years, the trends in the process industry toward high-tech, low-volume and high-added-value products boosted the interest in semi-batch processing. In addition, these processes often represent flexible production environments. Accordingly, the optimal operation of semi-batch processes has moved from scheduling (better flexibility) to optimization (better profitability) (Bonvin, 1998; Srinivasan et al., 2003b; Bonvin, 2006; Marchetti et al., 2006; Nagy et al., 2007).

The optimal operation of batch and semi-batch processes requires overcoming many challenges. Unlike continuous operation, batch and semi-batch processes exhibit inherently transient behaviour as well as strong nonlinearity since the process does not operate around a steady operating point. In other words, batch and semi-batch processes have start-up behaviour. Moreover, the presence of both path and terminal constraints and tight product quality limits result in challenging and non-convex optimization problems. In addition, the lack of accurate models due to the limited amount of experimental data results in considerable uncertainty and hence hinders the usage of offline-computed optimal profiles. Furthermore, batch and semi-batch processes usually have constraints on end-product quality, and the

ability to influence the process (or flexibility) often decreases with time. If there is a deviation in product quality, the charge has to be almost always discarded (Terwiesch et al., 1994; Bonvin, 1998; Srinivasan et al., 2003b; Jung et al., 2015). Hence, the open-loop implementation of off-line computed optimal control profiles may result in sub-optimal, or worse, infeasible operation and the loss of the batch. Moreover, the operating conditions might change from batch to batch and cause unacceptable variations of product quality. Consequently, the application of online, measurement-based, optimizing feedback schemes is of great importance for the optimal operation of batch and semi-batch processes (Eaton and Rawlings, 1990; Ruppen et al., 1995; Ruppen et al., 1998; Bonvin et al., 2001; Bonvin, 2006; Kadam et al., 2007; Welz et al., 2008; Mesbah et al., 2011; Bonvin and François, 2017).

In summary, the important challenges related to the optimal control (dynamic optimization) of batch and semi-batch processes can be stated as follows:

- Transient start-up behaviour
- Strong nonlinearity
- No-steady state, classical PID methods are not applicable
- Tight product-quality limits
- Reduction of flexibility with time
- Irreversible behaviour
- Considerable plant-model mismatch

The nonlinear dynamic optimization of batch and semi-batch processes is becoming more and more popular due to industrial competitiveness and strict environmental regulations. If a reliable dynamic process model is available, dynamic optimization (or optimal control) is considered as a promising method for reducing production costs, improving product quality and meeting safety as well as environmental regulations. Moreover, nonlinear dynamic optimization is at the core of nonlinear model predictive control (NMPC) and plays an important role in terms of feedback optimizing control. The available methods in the literature to solve dynamic optimization problems can be classified as direct and indirect methods.

Direct methods are usually the selected procedure to solve the constrained nonlinear dynamic optimization problems. Basically, they convert the dynamic optimization problems into nonlinear programming problems (NLPs). However, for large time horizons (which is the

case for the batch and semi-batch processes), the standard NLP algorithms might turn out to be computationally demanding due to the required matrix factorizations related with the solution steps (Cannon et al., 2008; Biegler, 2010). This issue will be discussed in greater detail in the next sections.

Model predictive controller (MPC) has been used extensively in industry (García et al., 1989; Qin and Badgwell, 2003). On the basis of a (most often linear) process model, these controllers predict the future behaviour of the states and outputs. At each iteration, the algorithm updates the initial conditions using measurements and solves a dynamic optimization problem for some cost function such as the minimization of a tracking stage cost or the maximization of a final cost. Only the first part of the computed optimal inputs is implemented, then the horizon is shifted by one sampling time and the procedure is repeated iteratively. Since MPC is capable of addressing multivariable constrained nonlinear systems and can use different types of models and performance criteria, it possesses a suitable and flexible structure for real-time optimizing control (Diehl et al., 2002; Adetola and Guay, 2010; De Souza et al., 2010; Huang et al., 2010; Lucia et al., 2014). A detailed discussion and survey on MPC can be found in (Morari and Lee, 1999).

Due to the strong nonlinear behaviour of batch and semi-batch processes, linear MPC is often not the method of choice. Moreover, batch and semi-batch processes usually require strictly constrained operation since the ability to influence the performance and feasibility of the process decreases with time (Bonvin, 1998). This motivates the use of shrinking-horizon nonlinear model predictive controllers (NMPC), for which the optimization is performed with respect to the full time horizon and includes both path and terminal constraints (Nagy and Braatz, 2003; Nagy et al., 2007).

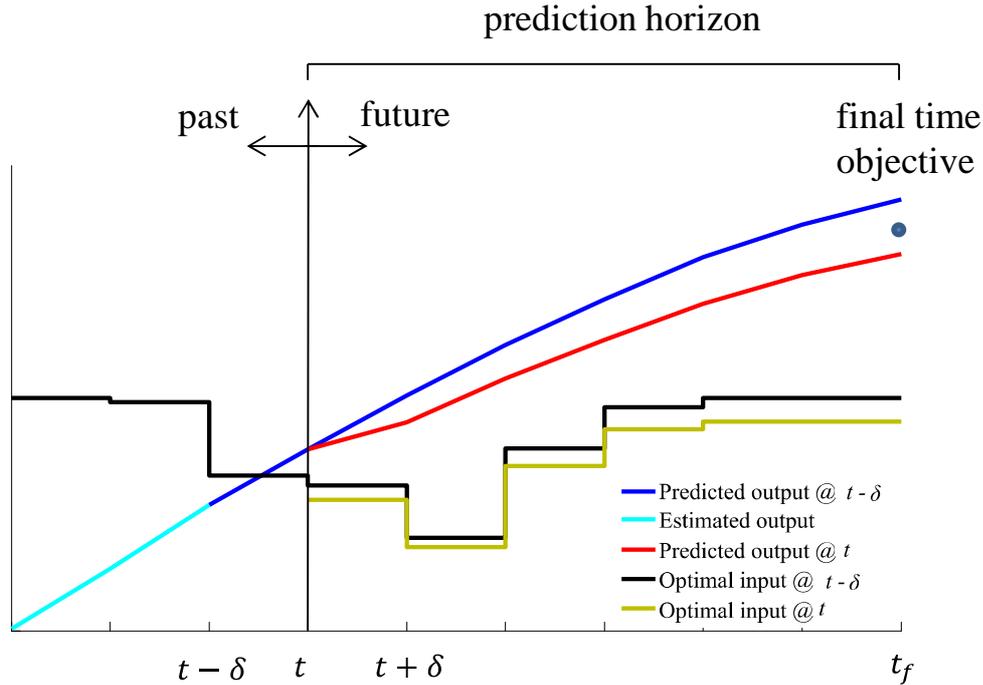
Several studies on the applicability of NMPC to batch and semi-batch processes have been reported in the literature. (Lakshmanan and Arkun, 1999) used linear parameter-varying models for the estimation and control of nonlinear batch processes. (Seki et al., 2001) proposed an NMPC structure for the industrial application on polymerization reactors. (Nagy and Braatz, 2003) studied a robust NMPC scheme for batch crystallization, whereby parametric uncertainties are taken into account explicitly. (Valappil and Georgakis, 2002) suggested a min-max NMPC scheme with successive linearization for the control of the end-point properties in batch reactors. (Nagy et al., 2007) studied the real-time implementation of sh-NMPC to industrial batch reactors. (Mesbah et al., 2011) compared different optimization algorithms for sh-NMPC of a semi-batch crystallizer. (Lucia et al., 2013) suggested a multi-

stage NMPC scheme to deal with uncertainties, and a scenario-tree approach was used to optimize a semi-batch polymerization reactor. Recently, (Jang et al., 2016) proposed a multi-stage NMPC scheme for semi-batch reactors using backoffs on path constraints. (Binette and Srinivasan, 2016) compared the performance of different tracking objectives for the NMPC of batch processes without parameter adaptation. (Zubov et al., 2017) discussed the online implementation of NMPC to a semi-batch pilot-plant copolymerization reactor.

Shrinking-horizon nonlinear model predictive control (sh-NMPC) has been proposed as a successful platform for the optimal operation of semi-batch processes, with the prediction horizon always being until the final batch time (Bosley and Edgar, 1992; Bonvin, 1998, 2006; Aydin et al., 2017a). The idea of sh-NMPC is illustrated in Fig. 1.1. The nonlinear dynamic optimization problem is solved always until the final time, but only the first parts of the inputs are implemented. Then, the states are measured or estimated, the horizon is shrunk by the sampling time and the same procedure is repeated until the end of the batch.

Unlike its linear counterpart MPC, NMPC takes into account a nonlinear model (usually first-principles models) to perform dynamic optimization, which results in challenging, non-convex and constrained nonlinear optimization problems (NLP). Moreover, as mentioned before, to solve these problems using direct methods, sh-NMPC requires expensive matrix factorizations due to large prediction and control horizons (Cannon, 2004; Cannon et al., 2008).

Advanced fast solution algorithms are essential in terms of the application of NMPC or moving horizon estimation (MHE) in real time. Fast real time update usually increases the performance of the closed-loop optimizing control either by tackling the effect of feedback delay or by enabling faster sampling to increase optimization frequency (Zavala et al., 2008a, 2008b; Huang et al., 2009; Zavala and Biegler, 2009; Wolf et al., 2011; Wolf and Marquardt, 2016; Cao et al., 2017). Note that it might sometimes be possible to find a compromise between computational time and performance between linear MPC and NMPC for steady-state processes (Gros et al., 2016). Yet, for batch and semi-batch processes, linear MPC is usually not preferred. Unfortunately, there is always a certain computational time associated with the solution of the corresponding non-convex optimal control problems in real-time, which in turn may lead to non-negligible feedback delay in closed-loop operation.



**Figure 1.1.** The illustration of sh-NMPC.

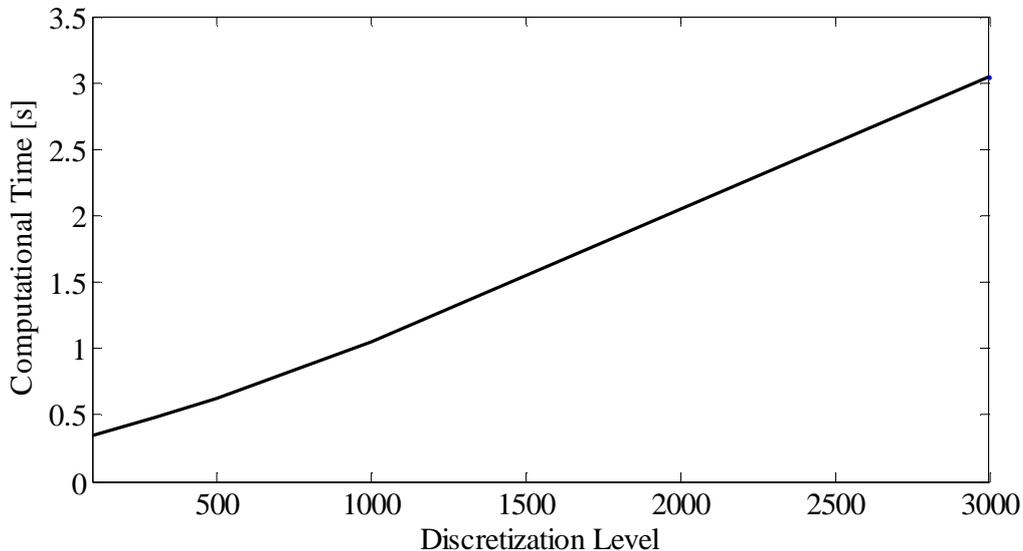
This delay may result in suboptimal, or worse, infeasible operation (Findeisen and Allgöwer, 2004; Gros et al., 2016). Hence, it is of great importance, and still an open research field both in academia and industry, to reduce the CPU time needed for the efficient real-time implementation of NMPCs (Wolf and Marquardt, 2016). For a deep review of the broad class of fast computational methods for NMPC (suboptimal, explicit, hierarchical, sensitivity-based), the reader is referred to (Wolf and Marquardt, 2016).

Indirect methods have been used to solve MPC problems in the literature. (Cannon et al., 2008) designed a MPC strategy for input-constrained linear systems, whereby the inputs are represented in terms of co-states and the problem is solved using active-set methods. The matrix factorizations performed by general direct solvers can be efficiently replaced by the cooperation of states, co-states and Lagrange multipliers for the path constraints using PMP. This way, the complexity per iteration increases only linearly with the length of the time horizon, which can be a computational advantage for batch and semi-batch processes that typically have large prediction and control horizons due to the shrinking-horizon approach. However, until very recently, there did not exist a fast convergent method to solve path-constrained optimal control problems using PMP (Hartl et al., 1995; Chachuat, 2007). In this thesis, firstly, an indirect, convergent and gradient-based dynamic optimization algorithm for

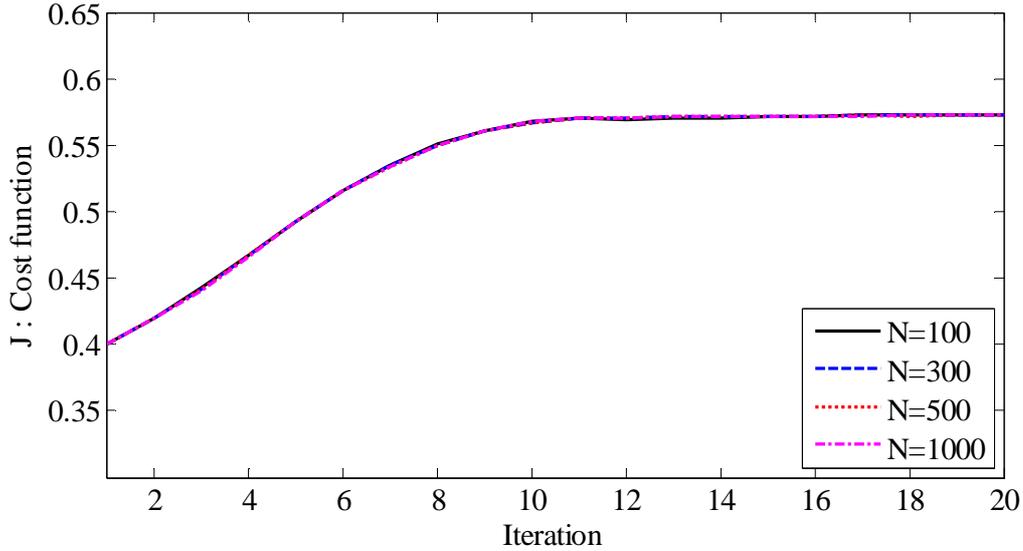
the non-control affine and constrained batch and semi-batch processes is proposed. The algorithm uses indirect adjoining to deal with path constraints, which allows the explicit and fast computation of the inputs to meet the path constraints at each iteration step.

Here, in order to provide more insight about the relation between the discretization level and the computational time, a simple problem which includes only input bounds is solved using the proposed novel indirect algorithm. The corresponding computational times and the iteration profiles are given in Fig. 1.2 and Fig. 1.3, respectively. For more detail about the problem, the reader is referred to Appendix 1. Note that a fixed step size is used in this problem. Although the speed of the algorithm can be increased via adaptive line search algorithms, it is clear that the computational time increases almost linearly when the grid gets finer. In addition, all solutions exhibit almost the same optimal cost.

A direct algorithm might be faster than the PMP-based method for input discretizations less than 50 elements. However, if we need larger time horizons or finer input discretization levels to get more accurate solution (usually the case for path constrained batch and semi-batch processes), PMP can turn out to be much more effective. Direct methods exhibit cubical increase for constrained problems in computational time as the time horizon or discretization level becomes larger. The computational performances of the direct and the proposed indirect methods will be compared in greater detail for more complex problems in the next sections.



**Figure 1.2.** The corresponding CPU times as the input grid gets finer.



**Figure 1.3.** The convergence profiles for the toy example.

The computational advantage of the proposed PMP formulation in the context of shrinking horizon nonlinear model predictive control represents another main motivation of this study. It will be proposed to apply the novel and convergent PMP-based solution algorithm to the shrinking-horizon NMPC of *nonlinear* batch and semi-batch processes under uncertainty and in the presence of nonlinear pure-state and mixed-state path constraints. The effect of uncertainties is handled by the introduction of time-varying backoffs (Visser et al., 2000; Srinivasan et al., 2003a; Shi et al., 2016). Since the prediction horizon is always until the final time, PMP can be expected to perform better than the classical direct methods, especially at the beginning of the batch.

The optimal inputs of batch and semi-batch processes can be characterized using different arcs. An optimal arc can be either on an input bound ( $u_{min}, u_{max}$ ), on a path constraint ( $u_{path}$ ), or inside the feasible region as a sensitivity-seeking arc ( $u_{sens}$ ). It is usually difficult to accurately compute the fine shapes of sensitivity-seeking arcs due to their lack of sensitivity. As a result, simplified solution models can be introduced, in which the inputs and most importantly the sensitivity-seeking arcs ( $u_{sens}$ ) are parameterized parsimoniously using switching times and low-order polynomials. This way, the number of decision variables and the complexity of the optimization problem can be reduced significantly. Thus, the required CPU time is expected to decrease significantly, with negligible reduction in the optimal cost (Welz et al., 2005; Schlegel et al., 2005; Welz et al., 2006; Aydin et al., 2017b). Another main contribution of this work is to detail the combination of these parsimonious solution models with indirect algorithms in the context of dynamic optimization and shrinking horizon NMPC.

In the sh-NMPC case, the optimization is performed at each sampling instant for the full time horizon but only the first part of the optimal inputs is applied to the process. Parsimonious sh-NMPC approximates the fine shapes of the optimal inputs at each sampling instant. Nevertheless, the optimal closed-loop behaviour could be captured accurately. In addition, since the full time horizon is taken into account, the loss in ability to influence the batch outcome (the loss of flexibility), can be prevented while still having a significant reduction in CPU time. Such parsimonious input parameterization scheme will be documented to design a fast computation method for the optimal operation of batch and semi-batch processes using sh-NMPC and to reduce the deteriorating effect of the computational delay in closed-loop operation.

## 1.2 Contribution of the Thesis

The contributions of this thesis can be summarized as follows:

- The algorithmic differences between the direct and indirect (PMP-based) dynamic optimization methods are compared in detail.
- A convergent and effective PMP-based algorithm is proposed for the dynamic optimization of constrained batch and semi-batch processes. The algorithmic steps and details are discussed in detail.
- The proposed indirect dynamic optimization method is extended to be used for the shrinking horizon NMPC for batch and semi-batch processes under parametric uncertainty.
- Constraint tracking is combined with sh-NMPC structure in order to further reduce the corresponding real-time computational effort.
- The proposed indirect dynamic optimization method is suggested to be united with an alternative and *parsimonious* input parameterization scheme in order to further decrease the computational complexity of the dynamic optimization problems.
- The performance of the suggested parsimonious indirect method is tested for the sh-NMPC of batch and semi-batch processes in the presence of parametric uncertainty.

### 1.3 Structure of the Thesis

- Chapter 2 compares the direct and indirect methods as well as the available solution algorithms and provides background information about the proposed novel indirect algorithms in this thesis.
- Chapter 3 suggests a convergent and effective indirect algorithm that is suited for the dynamic optimization of constrained batch and semi-batch processes. Detailed information and insight about the proposed algorithm is given and three different case studies are investigated to compare the computational performance of the proposed indirect method as opposed to a classical direct simultaneous method.
- Chapter 4 discusses the combination of the proposed indirect algorithm with parsimonious input parameterization in order to reduce the computational complexity of the dynamic optimization problems. Detailed information about the proposed parsimonious indirect algorithm is documented. The two case studies given in Chapter 3 are re-solved using the proposed parsimonious method.
- Chapter 5 extends the PMP-based method given in Chapter 3 for the sh-NMPC for batch and semi-batch processes under uncertainty. One of the examples studied in Chapter 3 (Hydroformylation of 1-dodecene) is selected and the performance of the PMP-based NMPC is investigated through closed-loop simulations in terms of both computational speed and robustness to plant-model mismatch.
- Chapter 6 incorporates the application of the proposed parsimonious indirect algorithm to the sh-NMPC for batch and semi-batch processes in order to decrease the computational complexity significantly. Two case studies are investigated under parametric uncertainty and the performance of the parsimonious and fully-parameterized methods are compared through closed-loop simulations in terms of both computational speed and robustness to plant-model mismatch.
- Chapter 7 summarizes and concludes this study and suggests further research areas as well as different types of applications.



*I believe in intuition and inspiration. Imagination is more important than knowledge. Knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution.*

---

Albert Einstein (1879 – 1955)

## 2 DYNAMIC OPTIMIZATION: METHODS AND ALGORITHMS

This chapter provides insight about the available dynamic optimization algorithms and methods in the literature.

The dynamic optimization problem for batch and semi-batch processes is often stated as follows (Srinivasan et al., 2003b):

$$\begin{aligned}
 \min_{t_f, u(t)} \quad & J := \phi(x(t_f)) \\
 \text{s.t.} \quad & \dot{x} = F(x, u, \theta), \quad x(0) = x_0 \\
 & S(x, u, \theta) \leq 0, \quad T(x(t_f)) \leq 0
 \end{aligned} \tag{2.1}$$

where  $J$  is a scalar performance index which can always be formulated with respect to the states at the final time  $t_f$ ,  $\phi$  is the objective function,  $x$  is the  $n_x$ -dimensional state vector with the corresponding initial conditions  $x_0$ ,  $u$  is the  $n_u$ -dimensional input vector,  $\theta$  is the vector of model parameters,  $S$  is the  $n_S$ -dimensional vector of inequality path constraints that include input bounds, and  $T$  is the  $n_T$ -dimensional vector of inequality terminal constraints. The nonlinear differential equations describing the system dynamics are included in the formulation as equality constraints. The solution methods that are available in the literature can be divided into two major categories, namely, the direct and indirect (or PMP-based) approaches (Srinivasan et al., 2003b).

## 2.1 Direct Methods

In direct optimization approaches, the solution methodology is applied directly to the original optimization problem given by Eq. 2.1, by using either sequential or simultaneous numerical techniques.

In the class of direct sequential methods (named also as direct single shooting), the input vector is parameterized using polynomial functions, the state equations are integrated from the given initial conditions up to the final time, where the states are needed for evaluating the objective function. This way, the dynamic optimization problem is converted into a nonlinear programming problem (NLP). Then, the optimal input parameters are computed by a NLP solver (Vassiliadis et al., 1994; Schlegel and Marquardt, 2006b). The use of time integration is the reason for calling the sequential techniques as “feasible-path methods”. However, depending on the type of the problem and the NLP solver available, a sequential method can be slow and thus computationally expensive, in particular while dealing with state path constraints (Srinivasan et al., 2003b). Furthermore, in direct sequential methods, the input profiles are often represented using a coarse discretization grid to ensure computational efficiency (Schlegel and Marquardt, 2004). Nevertheless, please note that a fine input discretization might be needed to accurately detect switching times and satisfy path constraints.

Another direct solution algorithm is the direct multiple shooting approach, which represents a mid-way between sequential and simultaneous algorithms. In this approach, the time interval is divided into stages, and the initial conditions of the stages are taken as decision variables for the optimization problem. This procedure is an ‘infeasible-path’ method but the integration is as accurate as in sequential methods (Srinivasan et al., 2003b). Direct multiple shooting has been used extensively in NMPC problems (Keil, 1999; Bock et al., 2000; Diehl et al., 2002; Diehl et al., 2006; Schäfer et al., 2007; Findeisen et al., 2007). In addition, (Diehl et al, 2005) suggested an efficient real-time iteration scheme which uses the idea of direct multiple shooting. Nevertheless, direct multiple shooting is not in the scope of this thesis.

In contrast, in the class of direct simultaneous methods (DSM), the entire optimization problem (system equations, input profiles, objective function and constraints) is discretized with respect to time, using for example collocation techniques, thus resulting in a large system of algebraic equations. Then, an NLP solver simultaneously interpret the governing dynamic

system equations and optimizes the cost (Cervantes and Biegler, 1998; Biegler et al., 2002) (Biegler, 2007). Since the dynamic system equations are not integrated, but approximated at discrete time instants, this approach is referred as “infeasible-path method”. Although simultaneous techniques allow the efficient solution of large-scale optimization problems, the trade-off between approximation and optimization must be considered carefully (Srinivasan et al., 2003b). In addition, the method might require good initial guess so as to guarantee robust convergence due to the discretization of the states and constraints. The NLP solver ‘Ipopt’ is one of the state of the art for both sequential and simultaneous methods nowadays, and it implements an interior point NLP algorithm (Wächter and Biegler, 2009).

Direct simultaneous methods were reported to be effective for the optimization and NMPC of large-scale problems (Cervantes and Biegler, 1998; Biegler et al., 2002; Wächter and Biegler, 2006; Kameswaran and Biegler, 2006; Biegler, 2007; Huang et al., 2009; Jang et al., 2016). Moreover, (Zavala and Biegler, 2009) introduced an ‘advanced-step’ DSM to deal with the computational delay associated with the time required to compute the solution in real-time implementations. Later, (Huang et al., 2010) extended this method for the combination of NMPC and moving horizon estimation.

### 2.1.1 NLP Solution Algorithms

Upon applying the direct solution algorithms, the dynamic optimization problem given in Eq. 2.1 is reformulated into the following general NLP (static) as follows:

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & g(x) \leq 0 \\ & h(x) = 0 \end{aligned} \tag{2.2}$$

where  $f$  is the objective function,  $g$  the inequality constraints and  $h$  the equality constraints. The Lagrangian  $L$  of the given NLP is defined by:

$$L(x, \lambda, \mu) = f(x) + \lambda(x, \lambda, \mu)^T h(x) + \mu^T g(x) \tag{2.3}$$

where  $\lambda$  and  $\mu$  are the Lagrange multipliers of the equality and inequality constraints, respectively. Assuming that  $x^*$  is a *local minimizer*, the first order necessary conditions of optimality (also referred as the Karush-Kuhn-Tucker, KKT, conditions) can be written as follows:

$$\begin{aligned}
\frac{\partial L(x^*, \lambda^*, \mu^*)}{\partial x} &= 0 && \text{stationary conditions} \\
g(x) &\leq 0 && \text{primal feasibility conditions} \\
h(x) &= 0 && \text{primal feasibility conditions} \\
\mu^* &\geq 0 && \text{dual feasibility conditions} \\
\mu^{*T} g(x) &= 0 && \text{complementarity slackness conditions}
\end{aligned} \tag{2.4}$$

and  $(x^*, \lambda^*, \mu^*)$  is referred as a KKT point assuming that conditions given in Eq. 2.4 hold (Sager, 2005; Fletcher, 2013). The two common solution algorithms for such NLP problems, sequential quadratic programming (SQP, active-set) and interior point (barrier-type), will be discussed in the next section.

### 2.1.1.1 Sequential Quadratic Programming (SQP)

The general idea of the SQP method is to solve approximate quadratic problems of the original NLP successively in order to converge to the optimal solution. This method can be categorized as an active-set type (Goldsmith, 1999; Boggs and Tolle, 2000; Sager, 2005). The NLP problem given in Eq. 2.2 can be reconstructed by adding the slack variables ( $z$ ) for the inequality constraints as follows:

$$\begin{aligned}
\min_{x,z} \quad & f(x) \\
\text{s.t.} \quad & g(x) + z = 0, \quad z \geq 0 \\
& h(x) = 0
\end{aligned} \tag{2.5a}$$

Upon defining the extended Lagrangian ( $\mathcal{L}$ ), it can be shown that solving Eq. 2.5a is equivalent to solve the following system of equations:

$$\begin{aligned}
\min_{x,z} \quad & \mathcal{L}(x, z, \lambda, v) = f(x) + \lambda^T h(x) + v^T (g(x) + z) \\
\text{s.t.} \quad & g(x) + z = 0, \quad z \geq 0 \\
& h(x) = 0
\end{aligned} \tag{2.5b}$$

where  $v$  is also the Lagrange multiplier vector for the reformulated equality constraints. The standard approximation (linearization) method for the SQP is the Taylor series expansion to the original NLP. Given an arbitrary value of the variables  $(x_k, z_k)$  and the Lagrange

multiplier vectors  $(\lambda_k, v_k)$ , the approximated quadratic problem for a change  $(d_x, d_z)$  can be written as follows:

$$\begin{aligned}
 \min_{d_x, d_z} \quad & \frac{1}{2} d_x^T \nabla_{xx} \mathcal{L}(x_k, \lambda_k, v_k) d_x + \nabla f(x_k) d_x \\
 \text{s.t.} \quad & \nabla g(x_k)^T d_x + d_z = -(g(x_k) + z_k) \\
 & \nabla h(x_k)^T d_x = -h(x_k) \\
 & d_z \geq -z_k
 \end{aligned} \tag{2.6}$$

Please note that in most cases the Hessian matrix  $\nabla_{xx} \mathcal{L}(x_k, \lambda_k, v_k)$  is approximated using finite differences or BFGS-like algorithms to approximate the original Hessian as positive definite (Boggs and Tolle, 2000; Biegler, 2010). The problem given in Eq. 2.6 should be solved by applying Newton's method, which requires solving the following linear system:

$$\begin{bmatrix} \nabla_{xx} \mathcal{L}(x_k, \lambda_k, v_k) & 0 & \nabla h(x_k) & \nabla g(x_k) \\ \nabla h(x_k)^T & 0 & 0 & 0 \\ \nabla g(x_k)^T & I & 0 & 0 \\ 0 & V_k & 0 & Z_k \end{bmatrix} \begin{bmatrix} d_x \\ d_z \\ d_\lambda \\ d_v \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) \\ -h(x_k) \\ -g(x_k) - z_k \\ -Z_k v_k \end{bmatrix} \tag{2.7}$$

where  $V_k$  and  $Z_k$  are diagonal matrices including  $x_k$  and  $z_k$ . Then, the new iteration can be obtained as follows:

$$\begin{aligned}
 x_{k+1} &= x_k + d_x, \\
 z_{k+1} &= z_k + d_z, \\
 \lambda_{k+1} &= \lambda_k + d_\lambda, \\
 v_{k+1} &= v_k + d_v
 \end{aligned} \tag{2.8}$$

The steps given in Eq. 2.7 and Eq. 2.8 are followed recursively until the optimal solution is obtained. The reader is referred to (Biegler, 2010) for specially tailored SQP algorithms applicable to large-scale problems. Globalization of the algorithm can be carried out using line search and filter methods along with the differences computed from Eq. 2.7.

### 2.1.1.2 Interior-Point Method

Interior-point methods represent an alternative to active-set strategies by relaxing the complementarity conditions and solving the relaxed problems. The basic idea is to include

log-based penalty terms to the objective function in order to force the convergence inside the feasible region (Wright, 1997). First of all, to illustrate the basic idea of the algorithm, the problem given by Eq. 2.2 is re-written by modifying the inequality constraints as follows (Biegler, 2010):

$$\begin{aligned} \min_x \quad & f(x) \\ \text{s.t.} \quad & h(x) = 0, x \geq 0 \end{aligned} \quad (2.9)$$

Then, the penalty term is included into the problem such as:

$$\begin{aligned} \min_x \quad & f^{new}(x) = f(x) - \mu \sum_{i=1}^{n_x} \ln(x_i) \\ \text{s.t.} \quad & h(x) \leq 0, x > 0 \end{aligned} \quad (2.10)$$

where  $n_x$  is the number of inequality constraints in the original problem and  $\mu$  is the penalty term. Note that the log-barrier term becomes unbounded at  $x = 0$ . Thus, the path generated at each optimization iteration must lie in the strictly positive region for the reformulated inequality constraints. As the barrier-parameter value decreases, the solution approaches to the solution to the problem given by Eq. 2.9. This solution satisfies the first-order conditions given as follows:

$$\begin{aligned} \nabla f(x) + \nabla h(x)\lambda - \mu X^{-1}e &= 0 \\ h(x) &= 0 \end{aligned} \quad (2.11)$$

where  $X = \text{diag}\{x\}$ ,  $e = [1, 1, \dots, 1]^T$ . Eq. 2.11 is known as the primal optimality conditions. However, it is usually difficult to solve the direct barrier problem due to the high nonlinearity. This motivates the introduction and solution of a primal-dual system given as follows:

$$\begin{aligned} \nabla f(x) + \nabla h(x)\lambda - u &= 0 \\ Xu &= \mu e \\ h(x) &= 0 \end{aligned} \quad (2.12)$$

where  $u$  are the strictly positive dual variables for the barrier problem replaced with the barrier term. The substitutions and linearization enable the straightforward solution of the problem (Biegler, 2010). Given an iterate  $(x_k, \lambda_k, u_k)$ , the search directions can be computed as follows:

$$\begin{bmatrix} \nabla_{xx}\mathcal{L}(x_k, \lambda_k, v_k) & \nabla h(x_k) & -I \\ \nabla h(x_k)^T & 0 & 0 \\ U_k^T & 0 & X_k \end{bmatrix} \begin{bmatrix} d_x \\ d_\lambda \\ d_u \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) + \nabla h(x_k)\lambda_k - u_k \\ h(x_k) \\ X_k u_k - \mu e \end{bmatrix} \quad (2.13)$$

where  $X = \text{diag}\{x\}$  and  $U = \text{diag}\{u\}$ . Afterwards, iterations are recursively performed, similar to Eq. 2.8, until the optimal solution is found. Line search and filter algorithms are very important for the efficient globalization and implementation of the aforementioned method. For more detailed information, the reader is referred to (Biegler et al., 2002; Wächter and Biegler, 2006; Biegler, 2010).

### 2.1.2 Computational Aspects

The iteration steps related to the direct methods require the solution to the equation systems given in Eq. 2.7 and in Eq. 2.13, which might turn out to be relatively expensive to factorize for certain types of problems and constraints and for large discretization levels, although the system is linear. This holds for using both active-set and barrier type methods. The complexity of the iteration usually increases cubically as the time horizon or the input discretization level increases due to the expansion of the matrix systems (Cannon, 2004; Cannon et al., 2008). Several sensitivity-based methods have been proposed in the literature to reduce the computational requirements of NLPs in the context of NMPC. These methods usually rely on previously computed solutions and NLP sensitivities. (Diehl et al., 2002) suggested a real-time iteration scheme, in which, at each sampling time, instead of a full NLP, only a quadratic problem around the solution of a previous QP is solved. Another sensitivity-based method of choice is the advanced-step NMPC, which was proposed by (Zavala and Biegler, 2009). In this method, the NLP is solved in advance with respect to a predicted initial state. Then, as soon as the new state measurements (or estimates) are obtained, the NLP solution is updated using a fast sensitivity-update step and the IPOPT solver (Wächter and Biegler, 2006; Jäschke et al., 2014; Suwartadi et al., 2017). Successful implementations have been documented in the literature, in particular for large-scale processes (Zavala et al., 2008b; Huang et al., 2009). However, the performance of these sensitivity-based methods is still an open question for batch and semi-batch processes that are characterized by high nonlinear effects and large perturbations. For a detailed review on the recent advances in the sensitivity-based NMPC, the reader is referred to (Biegler et al., 2015).

In contrast, indirect-based methods can be adapted to avoid these expensive factorizations by interplaying the inputs  $u$  and Lagrange multipliers  $\lambda$  and  $\mu$ , instead of recursively updating the Lagrange multipliers through Newton-type solution steps. Consequently, the complexity of the optimization can be tailored to increase almost linearly with increasing horizon lengths (Cannon et al., 2008; Aydin et al., 2017a). Indirect methods will be detailed in the next section.

## 2.2 Indirect Methods

In indirect optimization methods, the optimization problem is reformulated as the minimization of a Hamiltonian function (Bryson, 1975). The reformulated problem is then solved to satisfy the necessary conditions of optimality (NCO) using Pontryagin's Minimum Principle (PMP). This reformulation results in multi-point boundary value problems (MT-BVP) for constrained problems. For simple problems, the optimal solution can usually be computed analytically. More complex and, in particular, constrained problems require a numerical solution which is often computed using the shooting method (Miele, 1978). The necessary conditions of optimality can be stated as follows:

*Assuming Problem 2.1 has a feasible optimal solution  $u^*(\cdot)$  with state profiles  $x^*(\cdot)$ , it is stated that there exist Lagrange multipliers  $\lambda^*(\cdot)$ ,  $\mu^*(\cdot)$  and  $v^*$  such that following equations hold for  $t \in (t_0, t_f)$ :*

$$\begin{aligned}
H^*(x^*(t), u^*(t), \lambda^*(t), \mu^*(t)) &= \lambda^{*T} F(x^*(t), u^*(t)) + \mu^{*T}(t) S(x^*(t), u^*(t)), \\
\dot{x}^*(t) &= H_\lambda(x^*(t), u^*(t), \lambda^*(t), \mu^*(t)), && \text{states} \\
\dot{\lambda}^{*T}(t) &= -H_x(x^*(t), u^*(t), \lambda^*(t), \mu^*(t)), && \text{co-states} \\
x^*(t_0) &= x_0, && \text{initial conditions} \\
\lambda^{*T}(t_f) &= \left. \frac{\partial \phi}{\partial x} \right|_{t_f} + v^{*T} \left. \frac{\partial T}{\partial x} \right|_{t_f}, && \text{terminal sensitivities} \\
0 &= H_u(x^*(t), u^*(t), \lambda^*(t), \mu^*(t)), && \text{path sensitivities (stationarity)} \\
0 &\geq S(x^*(t), u^*(t)), && \text{path feasibility} \\
0 &= \mu^{*T}(t) S(x^*(t), u^*(t)), && \text{path feasibility (compl. slackness)} \\
0 &\leq \mu^*(t), && \text{dual path feasibility} \\
0 &\geq T(x^*(t_f)), && \text{terminal feasibility} \\
0 &= v^{*T} T(x^*(t_f)), && \text{terminal feasibility (compl. slackness)}
\end{aligned} \tag{2.14}$$

$$0 \leq v^*, \quad \text{dual terminal feasibility}$$

$$H(t_f) = 0, \quad \text{transversality condition}$$

where  $H$  is the Hamiltonian function,  $\lambda$  is the  $n_x$ -dimensional vector of Lagrange multipliers (also called co-states or adjoints) for the system equations,  $\theta$  is the vector of model parameters,  $\mu$  is the  $n_s$ -dimensional vector of Lagrange multipliers for the path constraints, and  $v$  is the  $n_T$ -dimensional vector of Lagrange multipliers for the terminal constraints. For the detailed derivation of NCO, the reader is referred to (Biegler, 2010).

The PMP approach has been applied to various engineering optimization problems since the 70's. (Jaspan and Coull, 1971) suggested a boundary condition iteration (BCI) solution scheme for unconstrained chemical reactor optimization problems. For input-affine systems, (Visser et al., 2000) proposed an online optimizing structure that uses a switching function along with the PMP-based optimality conditions; then, a cascade optimization scheme that tracks the necessary conditions of optimality was designed and tested on a fed-batch penicillin fermentation process. (Cannon et al., 2008) designed a model predictive control strategy for input-constrained linear systems using PMP. In this approach, the inputs can be represented in terms of co-states, and the problem can then be solved using active-set methods. This work represents a nice example for the interplay between the states and co-states in order to reduce the complexity of the optimization problem. (Kim and Rousseau, 2012) used PMP for the optimal control of hybrid electric vehicles. (Palanki and Vemuri, 2005) proposed an end-point dynamic optimization scheme using PMP for semi-batch processes with a single reaction. (Roubos et al., 1997) studied the use of PMP with an unconstrained gradient-based solution technique for the optimization of fed-batch biological problems. In order to account for path constraints, they penalized the value of the objective function in case of a constraint violation. (Ali and Wardi, 2015) implemented a shooting method, where the inputs are expressed analytically in terms of the states and co-states. (Hannemann-Tamás and Marquardt, 2012) used PMP to verify the inputs computed by a direct sequential method. For a given optimal control problem, they computed “the true solution” using a PMP-based multiple shooting algorithm for the purpose of verifying the results of the direct sequential optimization algorithm. Recently, (Zhang et al., 2017) applied PMP in the context of MPC for a plug-in vehicle. In this method, the values of the co-states are determined by trial and error, which may turn out to be non-convergent for some problems.

Tracking the necessary conditions of optimality (NCO tracking) has also been proposed as a real-time optimization algorithm (Srinivasan & Bonvin, 2007). The optimal inputs are first computed via off-line optimization of the nominal model. The main assumption is that the solution structure (sequence and types of arcs) does not change with uncertainty. Hence, instead of performing online explicit optimization, the necessary conditions of optimality, given by PMP (computed off-line), are tracked with the help of feedback controllers. In other words, the optimization problem is converted into a model-free control problem (Srinivasan and Bonvin, 2007; Srinivasan et al., 2008; Chachuat et al., 2009; Ebrahim et al., 2016).

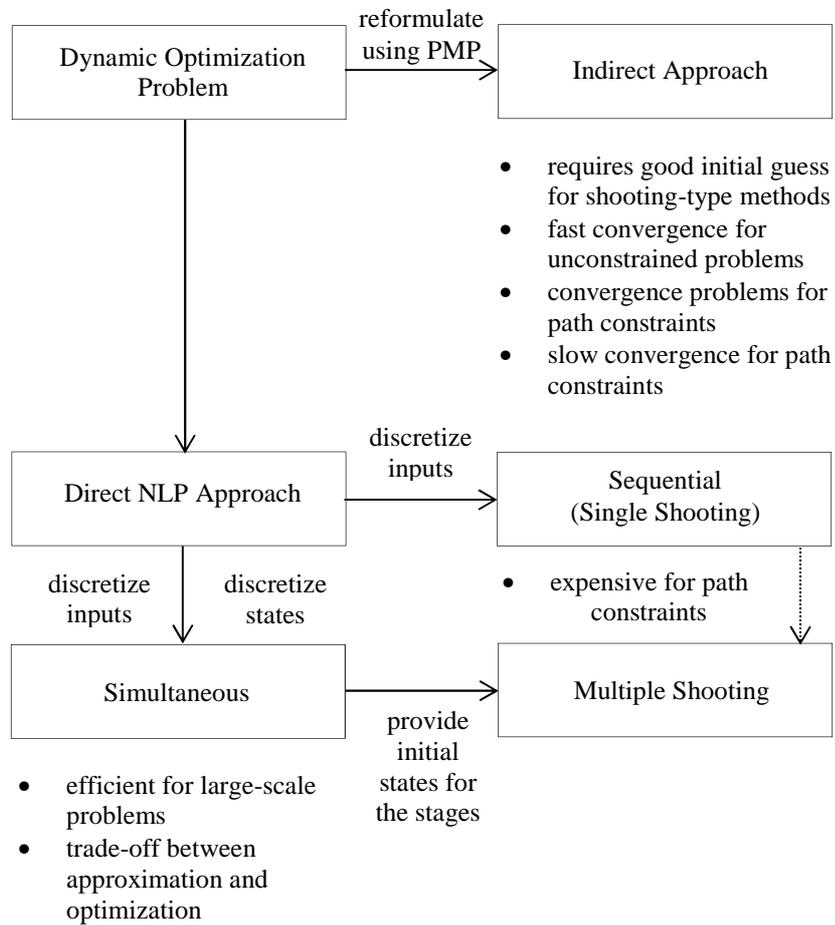
PMP has been applied to various type of engineering optimization problems (Palanki; Jaspán and Coull, 1971; Visser et al., 2000; Thomas et al., 2004; Palanki and Vemuri, 2005; Kim and Rousseau, 2012; Ali and Wardi, 2015; Zhang et al., 2017; Aydin et al., 2018b). It can be used to generate input and state trajectories as functions of initial and terminal states (Bryson, 1975). It is worth noting again that for simple problems, the optimal solution to the indirect problem can usually be computed analytically or by solving a standard BVP problem numerically. On the other hand, indirect formulation results in multi-point boundary value problems for constrained systems, which may be difficult to solve.

Solving dynamic optimization problems that include nonlinear path constraints is a challenging task for PMP-based (indirect) approaches. The convergence of existing shooting-type and gradient methods depends on many conditions (Chachuat, 2007; Biegler, 2010). In fact, the main problem of the shooting methods is that the integration of the co-state equations forward in time may introduce instabilities that result in convergence problems, especially in the absence of a good initial guess. However, for gradient-based type algorithms, initial guess is not required for convergence, but might be useful to speed up convergence. Hence, instead of integrating the states and co-states simultaneously forward in time, the inputs can be parameterized (or discretized) and then, sequentially, the state equations are integrated forward in time and the co-states backward in time. Eventually, optimization can be performed using a gradient-based algorithm, for which a good initial guess is not required for convergence (Bryson, 1975; Hartl et al., 1995; Srinivasan et al., 2003b; Chachuat, 2007). Graichen and Käpernick (2012) proposed a similar so-called ‘gradient projection’ approach for input-constrained problems, which possess similar computational advantages as opposed to classical direct optimization methods.

The computational advantage of indirect methods is the main motivation for this work. A convergent quasi-Newton and PMP-based method for the optimization of path and terminal

constrained batch and semi-batch processes will be introduced in 3<sup>rd</sup> Chapter. In particular, indirect adjoining will be used to reformulate the Hamiltonian function with mixed state-input constraints. This allows computing certain inputs explicitly so as to satisfy the infeasible path constraints. This way, the Lagrange multipliers for the path constraints can be eliminated from the optimization steps. As a result, the complexity of the solution algorithm can be reduced to the size of the inputs and, hence, only the inputs should be updated using Newton iteration steps. This would be an important advantage for the real-time application of model-based optimizing control in the presence of large time horizons or in the necessity of fine discretization levels.

The numerical algorithms present in the literature are summarized in Fig. 2.1 (Biegler, 2010):



**Figure 2.1.** Dynamic optimization methods in the literature (Biegler, 2010).

For a comprehensive overview of the dynamic optimization literature until 2003, the reader is referred to (Srinivasan et al., 2003b). Some more recent publications are given in Table 2.1.

**Table 2.1.** Selected recent publications.

| <b>Publication</b>              | <b>Method of Choice</b>  | <b>Subject</b>  |
|---------------------------------|--------------------------|---|
| (Schlegel et al., 2005)         | Direct Sequential        | Reducing the problem size using adaptive control vector                 |
| (Schlegel and Marquardt, 2006a) | Direct Sequential        | Reducing the problem size using adaptive switching times and structures |
| (Schlegel and Marquardt, 2004)  |                          |   |
| (Kadam et al., 2007)            | NCO Tracking             | Robust optimization using measurements and solution models              |
| (Srinivasan et al., 2008)       | NCO Tracking             | NCO Tracking using barrier-functions for active constraints             |
| (Biegler, 2007)                 | Direct Simultaneous      | Overview of recent direct simultaneous strategies                       |
| (Kameswaran and Biegler, 2006)  |                          |   |
| (Logist et al., 2011)           | Direct Multiple Shooting | Robust, multi-objective dynamic optimization                            |
| (Assassa and Marquardt, 2014)   | Direct Multiple Shooting | Adaptive multiple shooting  |

# **PART II**

## **Efficient Dynamic**

## **Optimization**

## **Using Indirect Methods**

*Everything has been thought of before, but  
the problem is to think of it again.*

---

Johann Wolfgang von Goethe (1749 – 1832)

# **3 DYNAMIC OPTIMIZATION OF CONSTRAINED BATCH AND SEMI-BATCH PROCESSES USING INDIRECT METHODS**

This chapter investigates the numerical dynamic optimization of constrained batch and semi-batch processes based on indirect methods. Direct methods are often the methods of choice to solve the constrained dynamic optimization problems, but they may exhibit certain limitations related to the compromise between feasibility and computational load. Indirect methods, such as Pontryagin's Minimum Principle (PMP), reformulate the dynamic optimization problem. This reformulation may turn out to be computationally advantageous. The main solution technique related with the indirect methods is the shooting method, which however often results in convergence problems and instabilities caused by the integration of the co-state equations forward in time (Srinivasan et al., 2003b; Chachuat, 2007).

This chapter introduces an alternative, effective and convergent indirect solution technique for the dynamic optimization of constrained batch and semi-batch processes. Specifically, instead of integrating the states and co-states forward in time, the proposed algorithm parameterizes or discretizes the inputs, integrates the state equations forward in time and the co-state equations backward in time, which in turn, leads to a gradient-based optimization approach. Constraints are handled using indirect adjoining to the Hamiltonian function, which allows meeting the active constraints explicitly at every iteration step.

### 3.1 Solution Methodology

The necessary conditions of optimality given in Eq. 2.14 (in the context of PMP) can be summarized as follows (Srinivasan et al., 2003b):

$$\begin{aligned}
 \min_{t_f, u(t)} \quad & H(t) := \lambda^T F(x, u, \theta) + \mu^T S(x, u, \theta) \\
 \text{s.t.} \quad & \dot{x} = F(x, u, \theta); \quad x(0) = x_0; \\
 & \dot{\lambda}^T = -\frac{\partial H}{\partial x}, \quad \lambda^T(t_f) = \frac{\partial \phi}{\partial x} \Big|_{t_f} + v^T \frac{\partial T}{\partial x} \Big|_{t_f}; \\
 & \mu^T S = 0; \quad v^T T = 0
 \end{aligned} \tag{3.1}$$

where  $H$  is the Hamiltonian function,  $\lambda$  is the  $n_x$ -dimensional vector of Lagrange multipliers for the system equations,  $\theta$  is the vector of model parameters,  $\mu$  is the  $n_S$ -dimensional vector of Lagrange multipliers for the path constraints, and  $v$  is the  $n_T$ -dimensional vector of Lagrange multipliers for the terminal constraints.  $\mu^T S = 0$  and  $v^T T = 0$  are the complementary slackness conditions that will be satisfied at the optimum. Moreover, the following necessary conditions must hold at any (local) optimum:

$$\frac{\partial H(t)}{\partial u} = \lambda^T \frac{\partial F}{\partial u} + \mu^T \frac{\partial S}{\partial u} = 0 \tag{3.2}$$

$$H(t_f) = (\lambda^T F + \mu^T S) \Big|_{t_f} = 0 \tag{3.3}$$

Eq. 3.2 indicates that the partial derivatives of the Hamiltonian function with respect to the inputs must all be equal to zero to be at an optimal solution. If the final time of the dynamic optimization problem is fixed, then Eq. 3.3, which is the transversality condition, is not necessary (Biegler, 2010). Note that, for each input  $u_i$ , the first term on the right-hand side of Eq. 3.2 is the switching function,  $s_{u_i} := \lambda^T \frac{\partial F}{\partial u_i}$ , which is useful to determine whether a given optimal input arc  $u_i(t)$  is constraint-seeking ( $s_{u_i} \neq 0$ ) or sensitivity-seeking ( $s_{u_i} = 0$ ).

Assuming that the state and co-state (adjoint) equations are differentiable, the proposed solution steps can be listed as follows:

- 1) Build the problem as the solution of differential equations for both the states and the co-states. This can be done by the differentiation of the Hamiltonian function with respect to the states as given in Eq. 35. The Matlab Symbolic Toolbox can be used for this purpose. This step is only necessary for the initialization of the problem.
- 2) Indirect adjoining should be used to deal with pure-state path constraints of the form  $S(x) \leq 0$ . In this method, the state constraints are differentiated with respect to time until at least one of the inputs appears explicitly (Hartl et al., 1995). The resulting expression is  $S^{(n)}(x, u) \leq 0$ , where  $n$  represents the relative degree of a constraint with respect to an input, that is, the number of differentiations required for an input to appear explicitly (Srinivasan and Bonvin, 2007). Then, instead of the original state constraints  $S(x) \leq 0$ , the differentiated version  $S^{(n)}(x, u) \leq 0$  is used to construct the Hamiltonian. This way, it is more effective to deal with path constraints when they become active. Usually, the corresponding path constraint can be activated via a single explicit calculation when the iteration is infeasible. If the constraint cannot be activated via a single step computation, then a root-finding algorithm should be considered. Consequently, the Hamiltonian function reads  $H(t) = \lambda^T F(x, u) + \mu^T S^{(n)}(x, u)$ . Because of the complementary slackness,  $\mu^T S^{(n)}(x, u) = 0$ , the penalty term  $\mu^T S^{(n)}(x, u)$  vanishes when all the constraints are satisfied. Nevertheless, if some of the constraints are not satisfied during the optimization iteration, the penalty term  $\mu^T S^{(n)}(x, u)$  will be positive, which in turn forces the convergence inside the feasible region. A restoration phase might be necessary if a feasibility problem between the inputs and constraints occurs (Miele, 1978).

**Remark 3.1.** Input saturation can be implemented such that:

$$u(t) = \begin{cases} u_{min}, & \text{if a lower constraint is violated} \\ u_{max}, & \text{if an upper constraint is violated} \end{cases}$$

- 3) Discretize the input profiles as  $u(t) = U(U)$ , where  $U$  is a  $(n_u \times N)$  matrix that contains  $N$  discrete input values for the  $n_u$  inputs. For example, the input profiles can be approximated by piecewise-constant functions. The choice of  $N$  depends on the nature of the problem.

- 4) Initialize  $U$  corresponding to the initial input profiles, and first integrate the state equations  $\dot{x} = F(x, u)$  forward in time. Then, integrate the co-state equations  $\dot{\lambda} = G(x, u, M, v)$  backward in time. Please note that the final condition of the co-states equations can be obtained through the PMP formulation as given in Eq. 3.1.
- 5) Discretize the Lagrange multiplier vectors as  $\mu(t) = M(M)$ , where  $M$  is a  $(n_S \times N)$  matrix. If the condition  $S_j^{\{n\}}(x, U(., k)) \leq 0$  is not satisfied at the discrete time instant  $k$ , set  $M(j, k) = K > 0$ , and compute the value of  $U(., k)$  that makes  $S_j^{\{n\}}(x, U(., k)) = 0$ . Otherwise, set  $M(j, k) = 0$ .

**Remark 3.2.** Here, the choice of the value of  $K$  is arbitrary. Yet, it is suggested to choose  $K$  as large as possible to guarantee the feasibility of the path constraints.

- 6) Update the input values  $U$  via a Quasi-Newton step (or steepest-descent) until a pre-defined optimality criterion is satisfied, such as the threshold value  $\varepsilon$  for  $norm\left(\frac{\partial H}{\partial U}\right)$ . Adjoining the inequality path constraints into the Hamiltonian enables the original constrained optimization problem to be solved as an unconstrained problem. Furthermore, the penalty terms  $\mu^T S^{\{n\}}(x, u)$  ensure that the update direction goes through the feasible region. Since the use of the analytic Hessian might be problematic and results in singularities, a robust BFGS update algorithm that ensures the positive-definiteness of the Hessian matrix is used (Biegler, 2010). In addition, the Hessian matrix is updated if the iteration is inside the feasible region. Otherwise, the Hessian matrix remains the same, and the optimization direction is set as  $\alpha\left(\frac{\partial H}{\partial U}\right)$ , that is, a steepest-descent algorithm is used. Applying these sequential steps, the states and Lagrange multipliers are interplayed so that the optimization is performed with respect to the inputs only. As a result, the cubic computational complexity increase for finer input discretization levels related with the solution of direct method steps is avoided.

**Remark 3.3.** Steps 2, 5 and 6 represent together the interplay of states, co-states and Lagrange multipliers, which reduces the complexity of the solution steps. While the co-states are handled via integration (as given by PMP), the Lagrange multipliers for the path constraints are eliminated from the optimization through Step 2.

**Remark 3.4.** Since the choice of  $K$  is arbitrary, the computed gradients might be ill conditioned. To avoid this, the gradients used in the optimization are normalized. An adaptive line-search algorithm is employed as opposed to a fixed step size. In this line-search, Wolfe conditions are followed as long as the iteration is inside the feasible region (Wolfe, 1971). When the iteration is infeasible, the initial step size ( $\alpha_0$ ) is used.

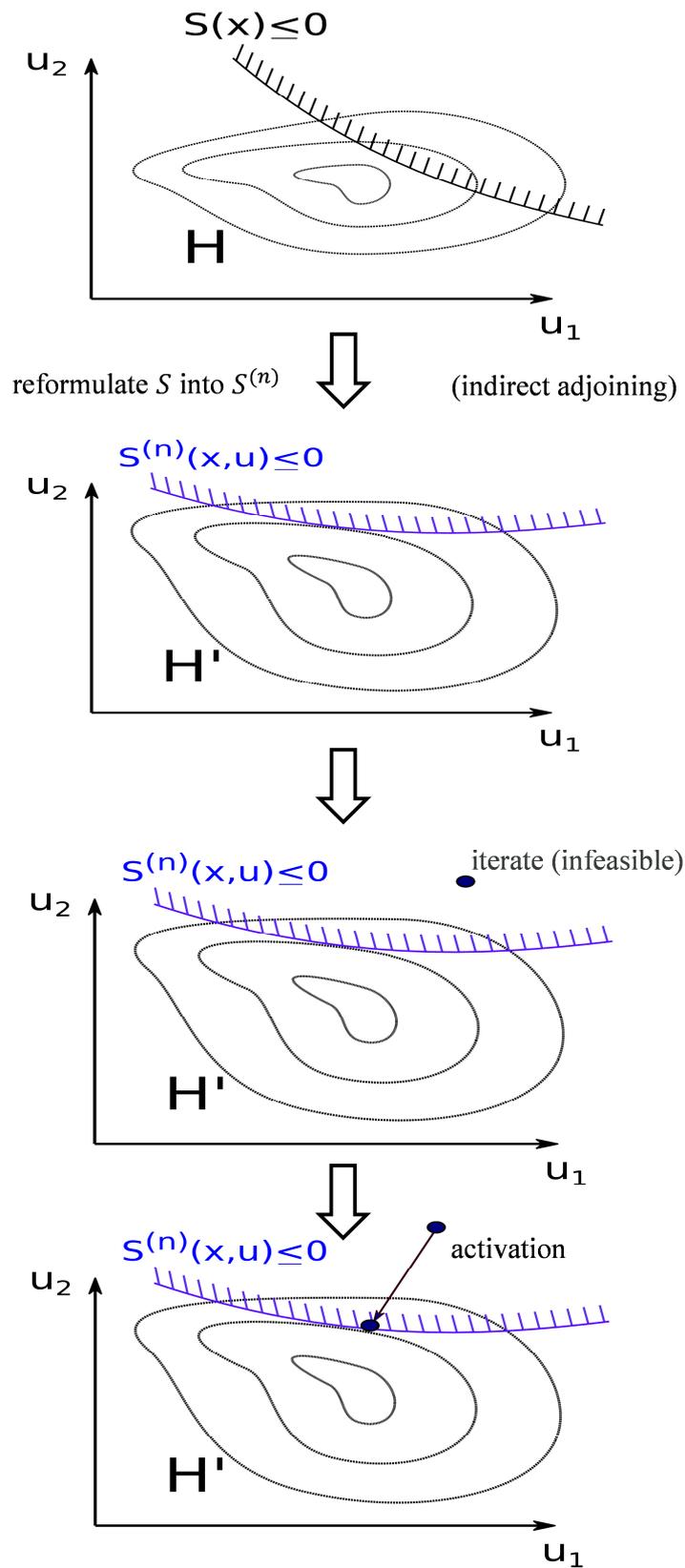
**Remark 3.5.** Constraint relaxation is highly recommended, because errors stemming from numerical integration or round off might contribute to larger convergence times.

**Remark 3.6.** Regarding the choice of the initial step size  $\alpha_0$ , values between 0.01 and 0.05 are usually quite effective for fully normalized problems.

**Remark 3.7.** The optimal solutions of batch and semi-batch processes often contain active constraints. However, if the proposed algorithm computes  $M_1 = 0$  at the first iteration, it is likely that the iterative scheme converges to a sub-optimal solution with all the constraints satisfied but inactive. To prevent this, a threshold on the number of iterations (e.g.,  $h > 10$ ) is used, which would enable the algorithm to search for active constraints. With the examples given in this thesis, it is yielded that 10 iterations suffice with a proper line search algorithm; however, for more difficult problems, larger values might be necessary.

The idea of indirect adjoining and constraint activation is illustrated in Fig. 3.1. Fig. 3.1.a shows the original problem with a pure-state path constraint, while Fig. 3.1.b illustrates the new optimization problem after indirect adjoining with the new constraint  $S^{(n)}(x,u,\theta) \leq 0$  and the new objective function  $H'(t)$ . Fig. 3.1.c and Fig. 3.1.d represent the location of the infeasible iterate and the activation of the constraint. The following remarks can be made at this point:

- 1) Although the original optimization problem is converted into a different problem, the optimal cost values are expected to be very close (Hartl et al., 1995; Aydin et al., 2017a).
- 2) In Fig. 3.1.d, the arrow illustrates the activation of the path constraint. Here, it is assumed that both inputs ( $u_1$  and  $u_2$ ) become explicit simultaneously through indirect adjoining. Yet, the method can also be applied if only one input is explicit after indirect adjoining. In that case, the arrow would be perpendicular to the axes of the implicit inputs.



**Figure 3.1.** Illustration of the proposed indirect adjoining and activation steps.

Assuming that the system and adjoint equations are differentiable, the algorithmic steps proposed for solving the indirect problem can be summarized as follows:

---

### PMP-based Solution Algorithm

---

Select values for the penalty term  $K > 0$ , the initial step size  $\alpha_0$ , the coefficient  $\beta$ , the threshold  $\varepsilon$ , the number of discrete input values  $N$  and the maximal number of iterations  $iter\_max$ . Initialize the input matrix  $U_0$ ,  $M_0$ ,  $v_0$  and the Hessian matrix corresponding to the  $k$ -th time instant,  $B_0^k := I_{n_u}$ . Discretize the Lagrange multipliers for the path constraints as  $\mu(t) = M(M)$ , where  $M$  is a  $(n_s \times N)$  matrix. Finally, let us write the co-states as  $\dot{\lambda} = G(x, u, M, v)$ .

**do**  $h = 0 \rightarrow iter\_max$

- 1) Solve the state equations  $\dot{x} = F(x, u)$  by forward integration and the co-state equations  $\dot{\lambda} = G(x, u, M, v)$  by backward integration and compute the constraint matrix  $M$  as follows: if the  $j$ -th constraint is satisfied at the  $k$ -th discrete time instant, set  $M_h(j, k) := 0$ , otherwise set  $M_h(j, k) := K$ .
- 2) Evaluate the matrix of first-order gradients  $(\nabla_U H)_h$  by using pre-computed analytical expressions.
- 3) **if**  $\|(\nabla_U H)_h\| < \varepsilon$ , set  $U_{opt} := U_h$ , **STOP**
- 4) Compute the next inputs as follows:

**do**  $k = 1 \rightarrow N$ ,

$$u_h^k := U_h(\cdot, k)$$

$$\nabla_h^k H := (\nabla_{U(\cdot, k)} H)_h$$

**if** the  $i_{th}$  terminal constraint is such that  $T^i(x(t_f)) < 0$ ,

set  $v_h(i) = 0$ . Otherwise, set  $v_h(i) = K$ , for  $i=1, \dots, n_T$ .

**end if**

**if**  $S(x, u_h^k, \theta) \leq 0$

4.1. Apply line search for  $\alpha_0$  and estimate  $\alpha$

4.2. Compute  $u_{h+1}^k = u_h^k - \alpha (B_h^k)^{-1} \nabla_h^k H$

4.3 Update the Hessian matrix  $B_{h+1}^k$  as follows:

$$s := u_{h+1}^k - u_h^k; \quad y := \nabla_{h+1}^k H - \nabla_h^k H$$

**if**  $s^T y \geq \beta \|s\|^2$ , set  $B_{h+1}^k := B_h^k + \frac{yy^T}{s^T y} - \frac{B_h^k s s^T B_h^k}{s^T B_h^k s}$

**else** set  $B_{h+1}^k := B_h^k$

**end if**

**else** compute  $u_{h+1}^k$  that gives  $S^{\{n\}}(x, u_{h+1}^k, \theta) = 0$  and set  $B_{h+1}^k := B_h^k$

**end if**

$$U_{h+1}(\cdot, k) := u_{h+1}^k$$

**end do**

**end do**

---

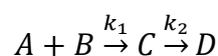
## 3.2 Case Studies

For illustrating the application of the proposed methodology to the dynamic optimization of constrained batch and semi-batch processes, three case studies are presented in this section. The first problem is a dynamic reactor optimization taken directly from (Srinivasan et al., 2003b). The second problem is the dynamic optimization of a batch binary distillation column with terminal purity constraints. Finally, the third problem involves the dynamic optimization of a complex fed-batch chemical process taken from (Hentschel et al., 2015).

All problems were solved using both a direct simultaneous method and the indirect PMP-based quasi-Newton proposed in this work. The CasADi toolbox (Andersson and Diehl, 2012) was applied for the implementation of the direct simultaneous method, along with the nonlinear programming solver IPOPT (Wächter and Biegler, 2006). In this thesis, for the direct simultaneous methods, collocation on finite elements method with a uniform grid is used for discretization. The degree of the interpolating polynomial is 4. The input discretization levels are problem specific and their values are given in the next sections. All computational results were obtained with an Intel i-3-2100 machine (CPU 3.10 GHz 4 GB RAM). Please note that the initializations of the problems are also considered in the computational results.

### 3.2.1 Non-isothermal semi-batch reactor with a heat-removal constraint

Consider a fed-batch reactor in which the following series reactions take place:



The objective is to maximize the molar content of the desired product  $C$  at a specified final time (Srinivasan et al., 2003b). The two inputs are the feedrate of  $B$ ,  $u(t)$ , and the reactor temperature,  $T(t)$ . The path constraints include input bounds as well as upper limits on the heat generated by the chemical reactions,  $q_{rx}$ , and the reactor volume,  $V$ . Note that an energy balance is not considered explicitly, but the temperature effect is included in  $q_{rx}$  as proposed by (Srinivasan et al., 2003b). The final time  $t_f$  is fixed at 0.5 h. Accordingly, the optimization problem can be formulated as follows:

$$\begin{aligned}
& \max_{u(t), T(t)} \quad J = c_c(t_f)V(t_f) \\
& \text{s.t.} \quad \dot{c}_A = -k_1 c_A c_B - \frac{u}{V} c_A; \quad c_A(0) = c_{A0}; \\
& \quad \dot{c}_B = -k_1 c_A c_B + \frac{u}{V} (c_{B,in} - c_B); \quad c_B(0) = c_{B0}; \\
& \quad \dot{c}_C = k_1 c_A c_B - k_2 c_C - \frac{u}{V} c_C; \quad c_C(0) = c_{C0}; \\
& \quad \dot{V} = u; \quad V(0) = V_0; \\
& \quad k_1 = k_{10} e^{\frac{-E_1}{RT}}; \quad k_2 = k_{20} e^{\frac{-E_2}{RT}}; \\
& \quad T_{min} \leq T(t) \leq T_{max}; \\
& \quad u_{min} \leq u(t) \leq u_{max}; \\
& \quad (-\Delta H_1)k_1 c_A(t)c_B(t)V(t) + (-\Delta H_2)k_2 c_C(t)V(t) \leq q_{rx,max}; \\
& \quad V(t) \leq V_{max} \tag{3.4}
\end{aligned}$$

The model parameters, initial conditions and constraints are given in Table 3.1

**Table 3.1.** Model parameters, initial conditions and constraints for Problem 3.2.1.

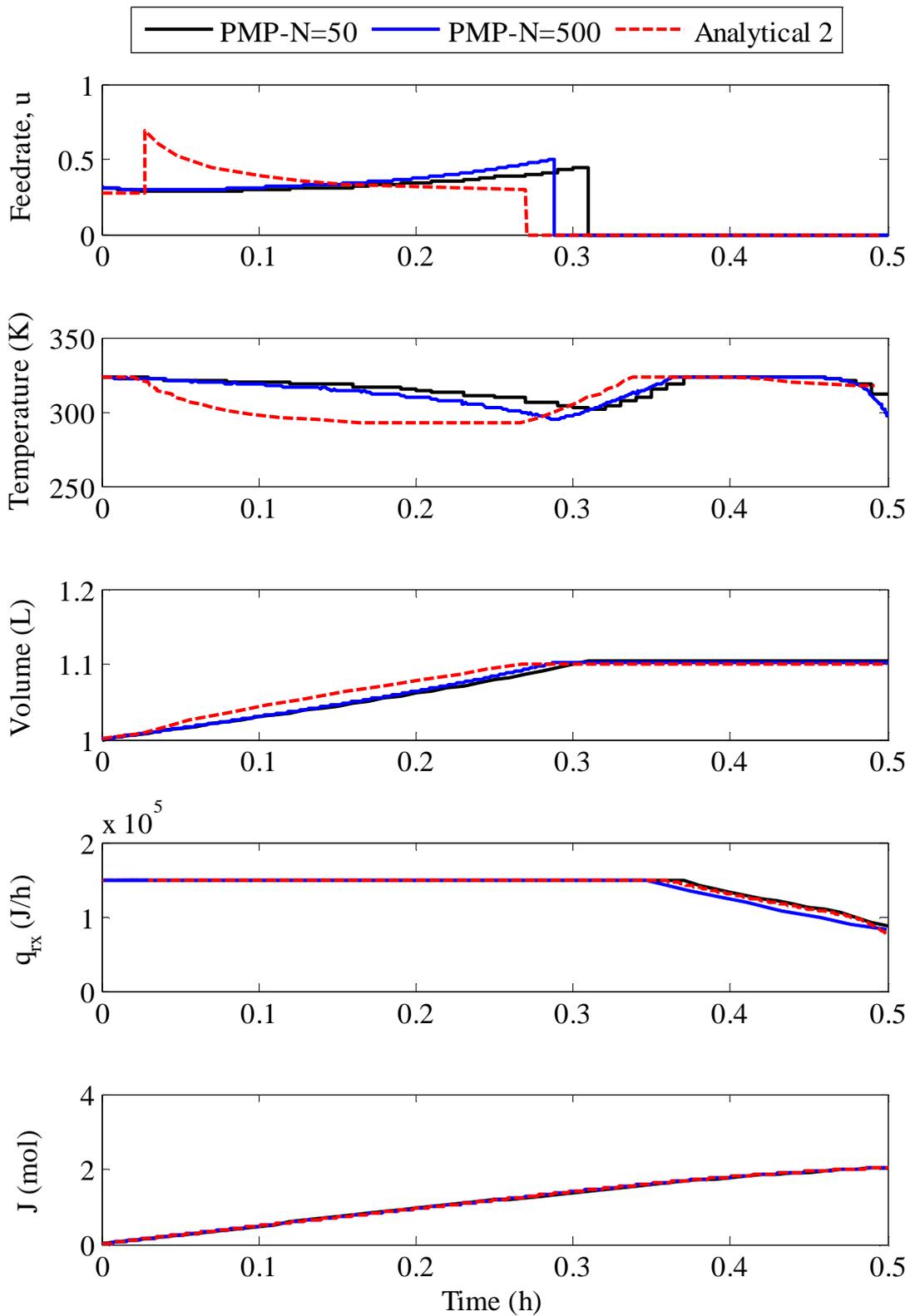
| Parameter    | Value                  | Parameter    | Value                 |
|--------------|------------------------|--------------|-----------------------|
| $k_{10}$     | 4 l/(mol h)            | $T_{min}$    | 293.15 K              |
| $k_{20}$     | 800 l/h                | $T_{max}$    | 323.15 K              |
| $E_1$        | $6 \times 10^3$ J/mol  | $V_{max}$    | 1.1 L                 |
| $E_2$        | $20 \times 10^3$ J/mol | $q_{rx,max}$ | $1.5 \times 10^5$ J/h |
| $R$          | 8.314 J/(mol K)        | $c_{A0}$     | 10 mol/L              |
| $\Delta H_1$ | $-3 \times 10^4$ J/mol | $c_{B0}$     | 1.1685 mol/L          |
| $\Delta H_2$ | $-10^4$ J/mol          | $c_{C0}$     | 0 mol/L               |
| $u_{min}$    | 0 L/h                  | $V_0$        | 1 L                   |
| $u_{max}$    | 1 L/h                  | $c_{B,in}$   | 20 mol/L              |

### 3.2.1.1 Computed optimal solutions

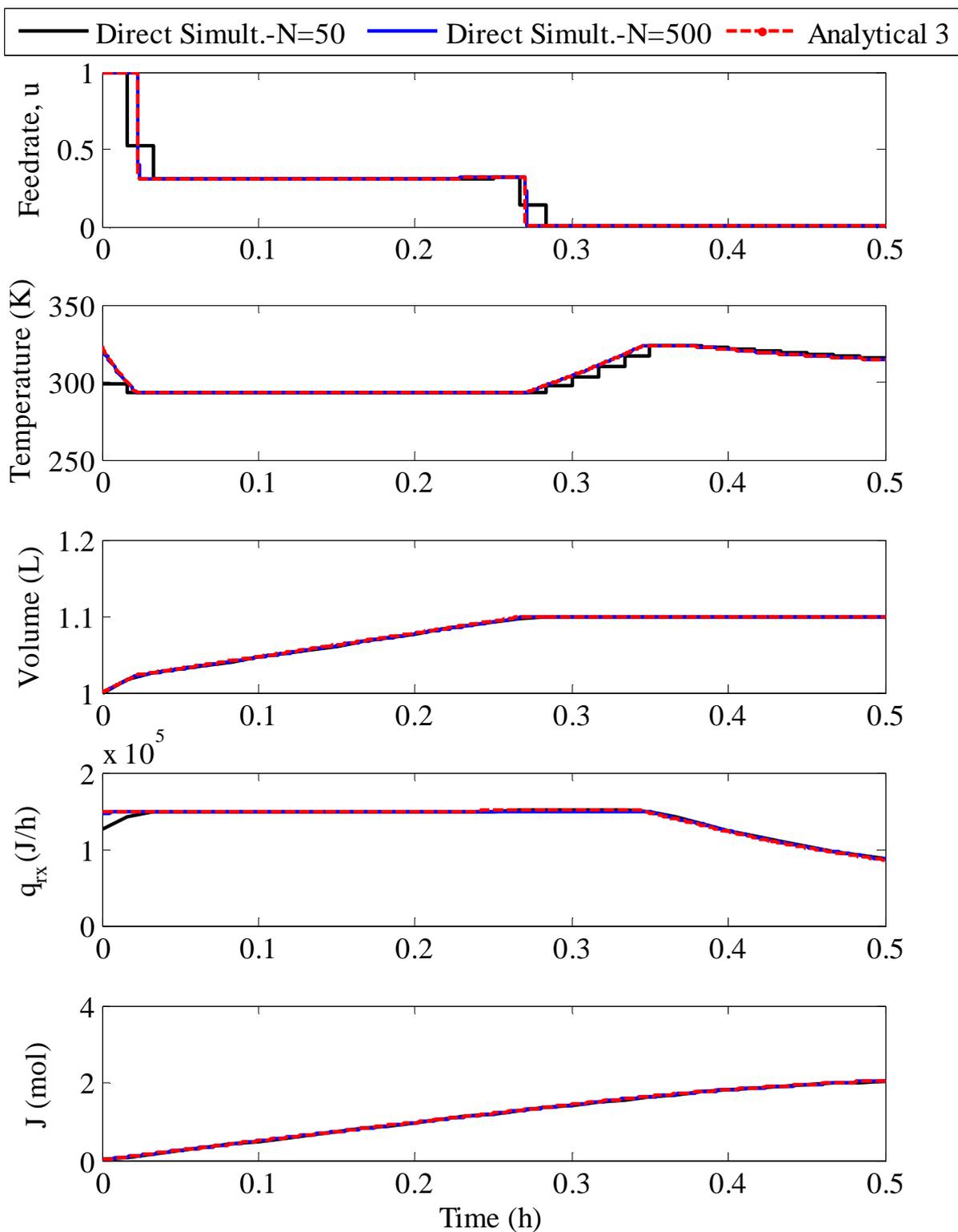
In this problem, there are several local solutions, three of which are given analytically by (Binette et al., 2016). In fact, any feasible combination of the arcs  $(u_{\min}, u_{\text{path}}, u_{\max})$  and  $(T_{\min}, T_{\text{path}}, T_{\text{sens}}, T_{\max})$  described in that paper can be a local solution to the problem.

The optimal input and state profiles computed with different numerical techniques are given in Figs 3.2.a and 3.2.b. Fig. 3.2.a shows the PMP-based solutions for the discretization levels  $N=50$  and  $N=500$ , along with the analytical solution 2 (Binette et al., 2016). The parameter values  $\alpha = 0.01$ ,  $K = 50$  and  $\varepsilon = 0.05$  are used in the PMP-based approach. Similarly, Fig. 3.2.b shows the direct simultaneous solution for  $N=50$  and  $N=500$  along with the analytical solution 3 (Binette et al., 2016). Accordingly, some remarks are given as follows:

1. In the PMP-based algorithm, the heat removal constraint is adjoined indirectly into the Hamiltonian function, but it is activated through a root finding algorithm; because, it is not straightforward to activate this constraint via only one explicit computation due to the logarithmic rate terms.
2. Although all strategies converge to a solution with nearly the same cost (between 2.050 and 2.053 moles of reactant B, as given in Table 3.2), there are significant differences in the computed optimal profiles, which is an indication that the two numerical strategies converge to *different* local solutions.
3. The heat-removal constraint is active during the first part of the run. The volume constraint is active in the second part of the run (batch mode without feeding).
4. With the PMP-based solution strategy, the input profiles are not very similar with the analytical solution 2 in the second arc characterized by  $T_{\text{sens}}$ . This is due to the lack of sensitivity of the objective function with respect to the sensitivity-seeking inputs.
5. In the direct simultaneous solution for  $N=50$ , the heat-removal constraint is not active initially because the time discretization is relatively coarse. However, the constraint becomes active when finer input discretization is used. This shows the necessity of the finer input discretization to get more reliable solutions. It should also be noted that coarser discretization results in infeasible solution of the original problem.

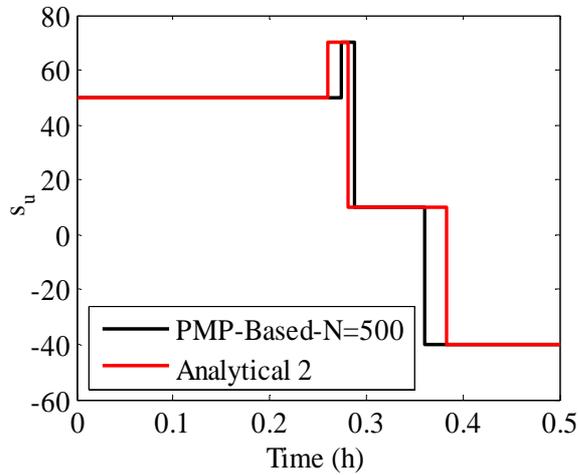


**Figure 3.2.a.** Optimal input and state profiles computed via the PMP-based method and the analytical solution 2.

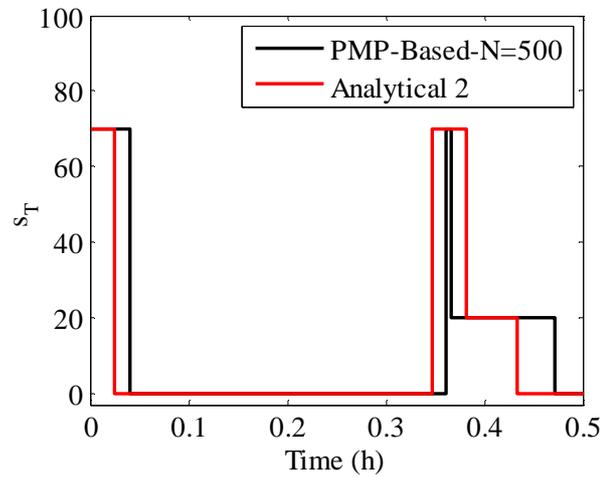


**Figure 3.2.b.** Optimal input and state profiles computed via the direct simultaneous method and the analytical solution 3.

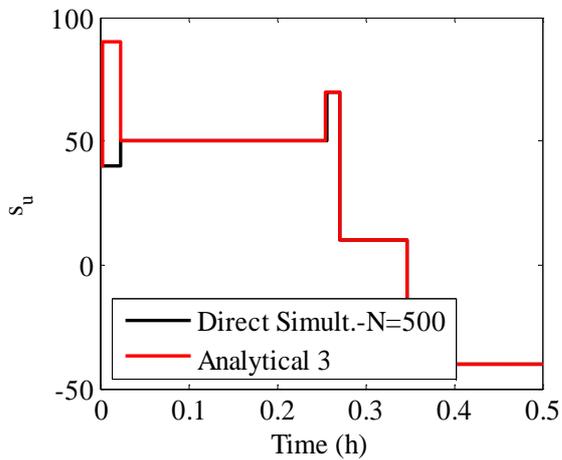
The switching functions  $s_u$  and  $s_T$  computed at the optimal solution are given in Figs 3.3 and 3.4 for the PMP-based solution and the direct simultaneous solution, respectively. It is seen that  $s_u$  is never zero, which means that the feed rate  $u$  is never sensitivity seeking. In contrast,  $s_T = 0$  in certain intervals, which are thus sensitivity seeking.



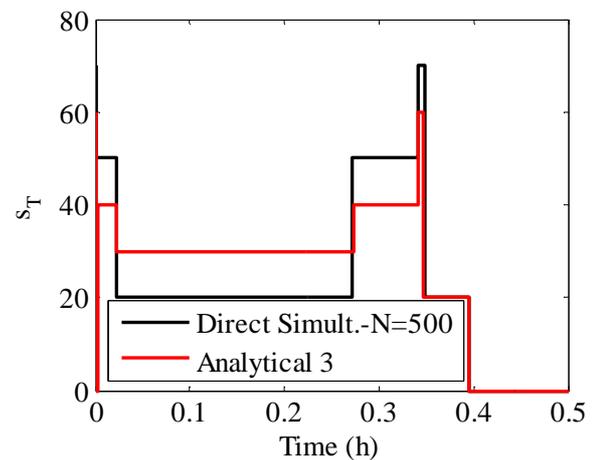
**Figure 3.3.a.** Switching function  $s_u$  for the PMP-based solution and the analytical solution 2.



**Figure 3.3.b.** Switching function  $s_T$  for the PMP-based solution and the analytical solution 2.



**Figure 3.4.a.** Switching function  $s_u$  for the direct simultaneous solution and the analytical solution 3.



**Figure 3.4.b.** Switching function  $s_T$  for the direct simultaneous solution and the analytical solution 3.

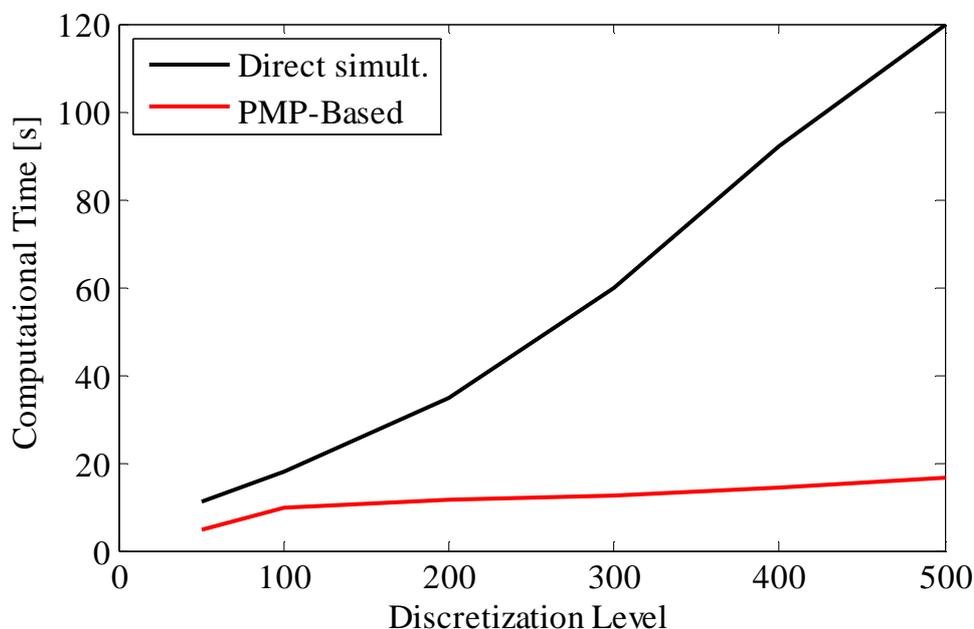
**Table 3.2.** Comparison of the indirect PMP-based, direct simultaneous and analytical solution strategies for Problem 3.2.1.

| Optimization Strategy         | Solution Structure |                   |                   |                  |                   |
|-------------------------------|--------------------|-------------------|-------------------|------------------|-------------------|
|                               | Arc 1              | Arc 2             | Arc 3             | Arc 4            | Arc 5             |
| <u>Indirect PMP-based:</u>    |                    |                   |                   |                  |                   |
| N=50, J=2.050                 | $u_{\text{path}}$  | $u_{\text{path}}$ | $u_{\text{min}}$  | $u_{\text{min}}$ | $u_{\text{min}}$  |
| N=500, J=2.052                | $T_{\text{max}}$   | $T_{\text{sens}}$ | $T_{\text{path}}$ | $T_{\text{max}}$ | $T_{\text{sens}}$ |
| <u>Analytical Solution 2:</u> |                    |                   |                   |                  |                   |
| J=2.050                       |                    |                   |                   |                  |                   |
| <u>Direct simultaneous:</u>   |                    |                   |                   |                  |                   |
| N=50, J=2.051                 | $u_{\text{max}}$   | $u_{\text{path}}$ | $u_{\text{min}}$  | $u_{\text{min}}$ | $u_{\text{min}}$  |
| N=500, J=2.053                | $T_{\text{path}}$  | $T_{\text{min}}$  | $T_{\text{path}}$ | $T_{\text{max}}$ | $T_{\text{sens}}$ |
| <u>Analytical Solution 3:</u> |                    |                   |                   |                  |                   |
| J=2.053                       |                    |                   |                   |                  |                   |

The comparison of the cost values and solution structures obtained with the various strategies are summarized in Table 3.2. The two numerical schemes converge to different solutions, which are in fact the analytical solution 2 and 3 given in Binette et al. (2016). The PMP-based solution suggests that the reactor temperature profile starts at its upper limit, with the feed rate  $u(t)$  adjusted to satisfy the heat-removal constraint. Then, the temperature follows  $T_{\text{sens}}(t)$  to find a compromise between producing much of the desired C and minimize the undesired by-product D. Once the reactor is completely filled, the feed rate is set to zero and the temperature is adjusted to still keep the path constraint active. When  $T_{\text{max}}$  is reached, the temperature is kept there until there is some advantage in reducing it and following  $T_{\text{sens}}(t)$  again. Although the input profiles of the PMP-based solution and the analytical solution 2 (Binette et al., 2016) are a bit different, the arc types and sequence are exactly the same.

On the other hand, the direct simultaneous solution with the Ipopt solver comes fairly close to the analytical solution 3. Optimal operation starts with maximal feeding of reactant B, with the temperature being used to meet the heat-removal constraint. When the minimal

temperature is reached, it is kept there, and the feed rate is adjusted to keep the path constraint active. Once the reactor is filled, the feed rate is set to zero and the temperature is increased to still keep the path constraint active. From that point on, the sequence of arcs is the same as for the PMP-based solution.



**Figure 3.5.** Computational times for different discretization levels  $N$  of Problem 3.2.1.

Fig. 3.5 shows the computational times required to obtain the solution with the PMP-based and direct simultaneous methods. It is clearly seen that the PMP-based method requires significantly less computational time when the grid gets finer, because the direct method becomes more expensive due to the factorization of the expanding matrices.

### 3.2.2 Batch binary distillation column with terminal purity constraints

The optimization of batch distillation columns using PMP has been documented in the literature. For example, (Coward, 1967) solved a time-optimal problem for a batch binary distillation column using PMP. The solution is based on an adaptive shooting strategy that requires good initial guesses for the adjoints. (Mayur and Jackson, 1971) studied PMP for binary and multicomponent batch distillation problems with adaptive solution techniques. (Welz et al., 2005) used the necessary conditions of optimality to design an implicit optimization scheme for a batch binary distillation column. In this section of the thesis, it is

proposed to compare the PMP-based quasi-Newton method with a direct simultaneous approach to optimize the operation of a batch binary distillation column.

Consider a batch distillation column with only three equilibrium plates, in which the components A and B (more volatile) are separated from each other. The objective is to maximize the molar amount of B in the distillate for a given batch time, while satisfying the terminal purity constraints of at least 80 mol % of B in the distillate and at most 20 mol% of B in the bottom product. The final time  $t_f$  is fixed at 3.0 h. The only path constraints are the input bounds on the reflux ratio.

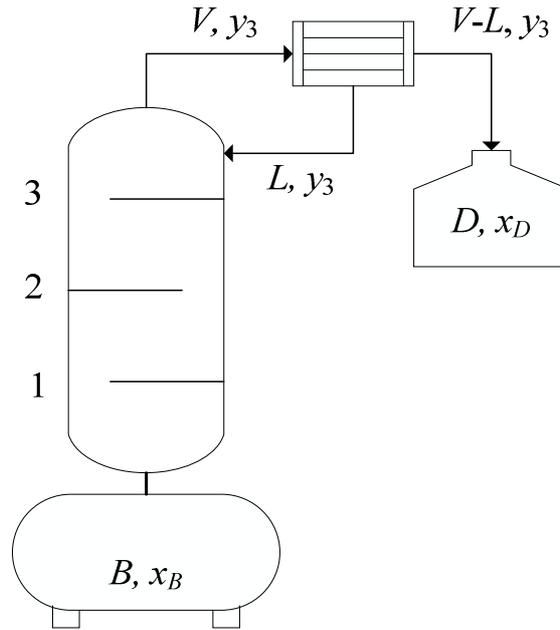
A schematic of the column is given in Fig. 3.6, with the molar amounts  $B$  and  $D$  in the bottoms and in the distillate tank, respectively. The vapor flow rate is represented by  $V$  and the liquid flow rate is given by  $L$ . The internal reflux ratio  $r = \frac{L}{V}$  is the only input variable. Assuming perfect mixing on all stages, negligible vapor hold-up, constant vapor flow through the column, total condensation in the condenser of negligible hold-up, constant liquid hold-up on all trays and constant relative volatility, the optimization problem can be stated as follows (Note that the material balances are written in terms of more volatile component B):

$$\begin{aligned}
& \max_{r(t)} && J = D(t_f) \\
& \text{s.t.} && \dot{D} = V(1 - r); \quad D(0) = D_0 \\
& && \dot{B} = V(r - 1); \quad B(0) = B_0 \\
& && \dot{n}_B = \dot{x}_B B + \dot{B} x_B = V(-y_B + r x_1); \quad n_B(0) = n_{B_0} \\
& && \dot{n}_1 = \dot{x}_1 M H = V(y_B - y_1 + r(x_2 - x_1)); \quad n_1(0) = n_{1_0} \\
& && \dot{n}_2 = \dot{x}_2 M H = V(y_1 - y_2 + r(x_3 - x_2)); \quad n_2(0) = n_{2_0} \\
& && \dot{n}_3 = \dot{x}_3 M H = V(y_2 - y_3 + r(y_3 - x_3)); \quad n_3(0) = n_{3_0} \\
& && \dot{n}_D = V(1 - r)y_3; \quad n_D(0) = n_{D_0} \\
& && y_m = \frac{\alpha x_m}{1 + (\alpha - 1)x_m}; \quad m = B, 1, \dots, 3 \\
& && x_D(t_f) = n_D(t_f)/D(t_f) \geq 0.8 \\
& && x_B(t_f) = n_B(t_f)/B(t_f) \leq 0.2 \\
& && 0 \leq r(t) \leq 1
\end{aligned} \tag{3.5}$$

where  $B_0$  is the initial charge,  $n_{B_0}$  the moles of B in the charge,  $n_m$  the moles of B in the liquid phase on the  $m$ -th tray,  $x_{B_0}$  the mole fraction of B in the initial charge,  $x_m$  the mole fraction of B in the liquid phase of the  $m$ -th tray,  $y_m$  the mole fraction of B in the vapor phase leaving the  $m$ -th tray,  $x_D$  the mole fraction of B in the distillate tank,  $x_B$  the mole fraction of B in the bottoms,  $y_B$  the mole fraction of B in the vapor leaving the bottoms,  $\alpha$  the relative volatility, and  $MH$  the liquid hold-up on each tray. The trays are numbered from the bottom to the top of the column. Because of total condensation, the composition of the refluxed liquid is equal to the vapor composition leaving the upper plate. It is also assumed that all plates are initially charged with the same liquid mixture as the reboiler and thus the initial concentration of B on each tray is  $x_{B_0}$ . The dynamic column model has 7 state variables and a single input. The model parameters and initial conditions are given in Table 3.3.

**Table 3.3.** Model parameters and initial conditions for Problem 3.2.2.

| <b>Parameter</b>                            | <b>Value</b> |
|---|--------------|
| Vapor flow rate, $V$                        | 50 kmol/h    |
| Relative volatility, $\alpha$               | 2.35         |
| Initial charge, $B_0$                       | 115 kmol     |
| Concentration of B in the charge, $x_{B_0}$ | 0.4          |
| Molar hold-up per plate, $MH$               | 5 kmol       |



**Figure 3.6.** Schematic of the batch distillation column.

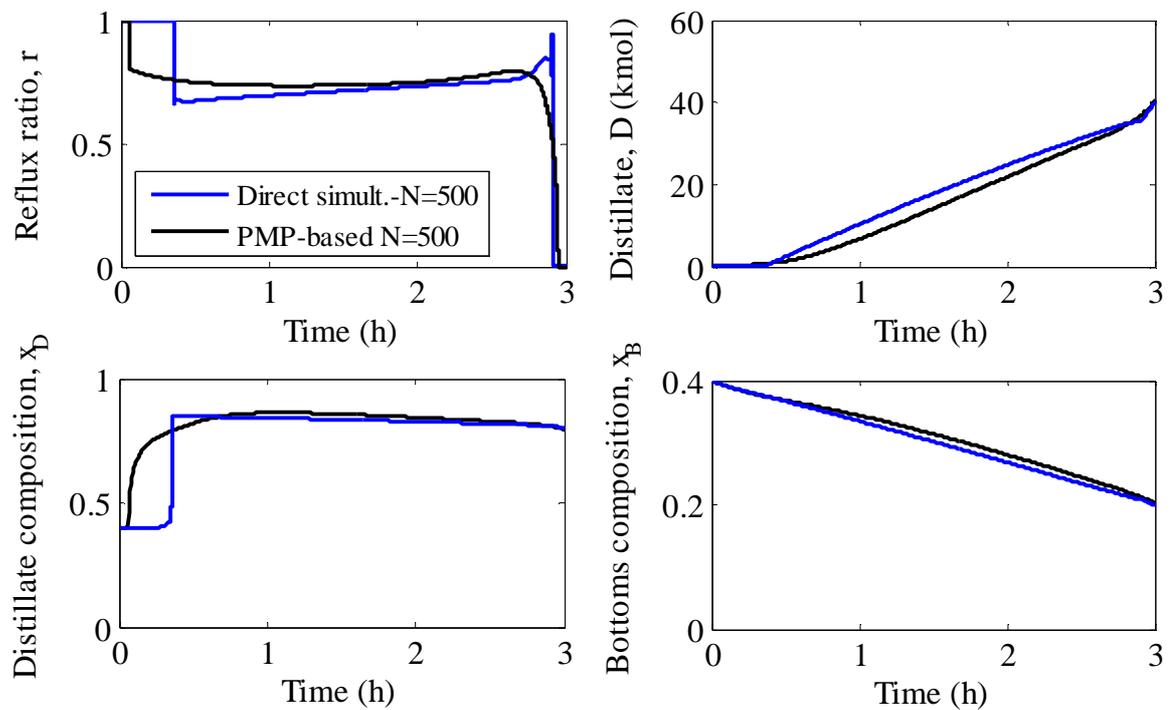
### 3.2.2.1 Computed optimal solutions

The input vector is parameterized using  $N$  equidistant piecewise-constant elements. The terminal constraints are enforced in the PMP-based solution by setting the final values of the adjoints as stated in Eq. 3.1. The parameter values  $\alpha_0 = 0.1$ ,  $K = 100$  and  $\varepsilon = 0.05$  are used in the PMP-based approach.

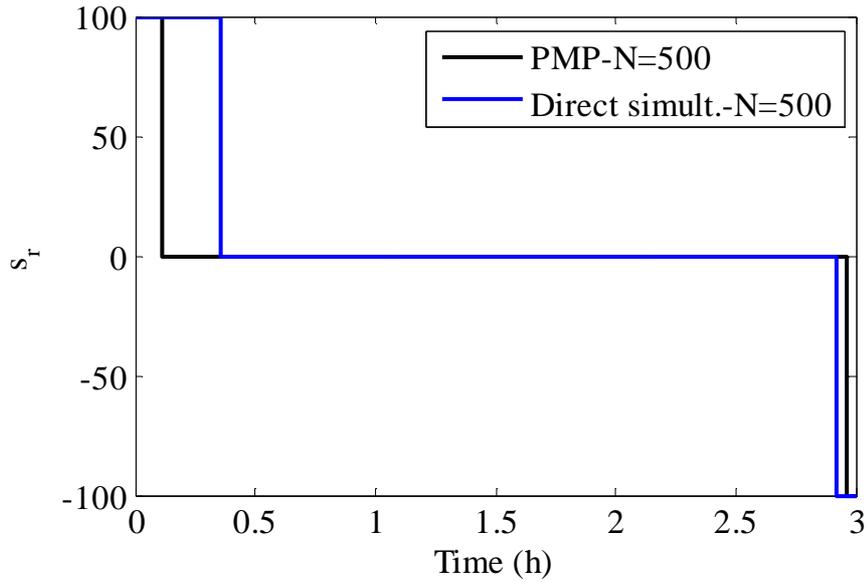
Fig. 3.7 shows the optimal input and state profiles computed with the two strategies for the discretization level  $N=500$ . Both PMP-based and the direct simultaneous solutions converge to a 3-arc solution. They suggest total reflux in the beginning of the batch to increase the composition at the top of the column. Then, they both follow a sensitivity-seeking arc to produce as much distillate as possible with the required purity. Finally, a short third arc with zero reflux is used to recover the high-purity material that is still in the column.

The corresponding switching functions are shown in Fig. 3.8. The following remarks can be made:

1. Since there are no path constraints besides the input bounds, the only possible arcs are  $r_{max}$ ,  $r_{min}$  and  $r_{sens}$ .
2. Although the two numerical schemes lead to the same sequences and types of arcs, namely  $r_{max}$ , followed by  $r_{sens}$  and  $r_{min}$ , and nearly the same optimal cost  $J$  (cf. Table 3.4), the computed input profiles are noticeably different. This is due to the lack of sensitivity of the objective function with respect to the input  $r_{sens}(t)$ . As discussed earlier, this is a common feature of sensitivity-seeking arcs, which significantly complicates the numerical computation of optimal solutions.



**Figure 3.7.** Optimal input and state profiles computed via the PMP-based method and the direct simultaneous strategy.

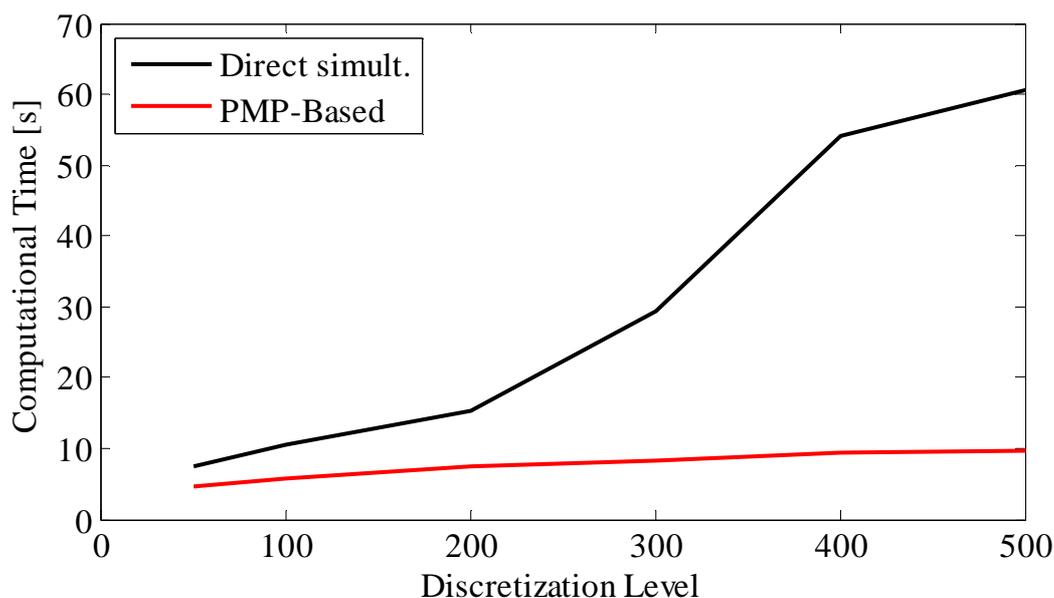


**Figure 3.8.** Switching function  $s_r$ .

**Table 3.4.** Comparison of the indirect PMP-based and direct simultaneous strategies for Problem 3.2.2.

| Optimization Strategy | Solution Structure |                   |            |
|-----------------------|--------------------|-------------------|------------|
|                       | Arc 1              | Arc 2             | Arc 3      |
| Indirect PMP-based    |                    |                   |            |
| N=50, J=40.01 kmol    | $r_{\max}$         | $r_{\text{sens}}$ | $r_{\min}$ |
| N=500, J=40.02 kmol   |                    |                   |            |
| Direct simultaneous   |                    |                   |            |
| N=50, J=40.02 kmol    | $r_{\max}$         | $r_{\text{sens}}$ | $r_{\min}$ |
| N=500 J=40.03 kmol    |                    |                   |            |

Finally, Fig. 3.9 compares the computational times required for the two numerical schemes as functions of the discretization level. One sees that the indirect PMP-based method has a clear advantage when finer grids are applied.

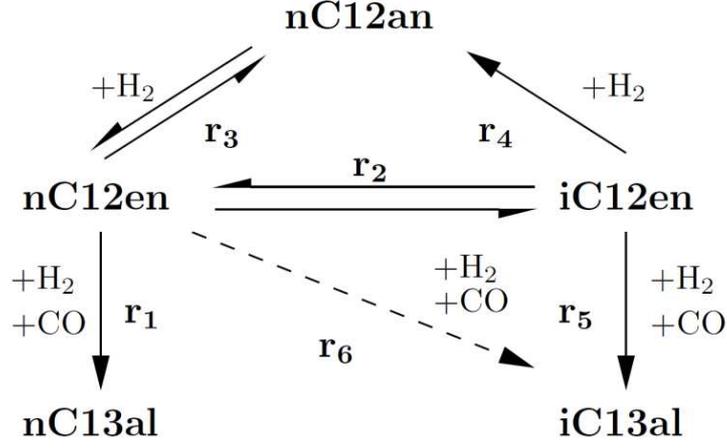


**Figure 3.9:** Computational times for different discretization levels of Problem 3.2.2.

### 3.2.3 Fed-batch hydroformylation reactor with path constraints

Due to their chemical nature, long-chain olefins are potential renewable feedstock to be integrated into existing petrochemical production networks. Hydroformylation is a suitable way of converting these feedstocks into valuable intermediates like aldehydes. A carbon double bond can be converted into an aldehyde group with the addition of  $H_2$  and  $CO$  in hydroformylation using a homogeneous catalyst (Hentschel et al., 2015; Kaiser et al., 2016).

Consider the optimization of a fed-batch reactor to maximize the production of n-tridecanal (nC13al) from 1-dodecene (nC12en) that reacts with syngas ( $H_2 + CO$ ). The reaction network is illustrated in Fig. 3.10 (Hentschel et al., 2015). A stirred tank reactor with gas feeding is used in semi-batch mode of operation. The input variables are the reactor temperature  $T(t)$  and the feedrate  $u(t)$  of syngas ( $H_2 + CO$ ). The gas and liquid phases are modeled as ideally mixed phases. The model parameters have been estimated and validated using experimental data (Hentschel et al., 2015). The aim is to maximize the concentration of n-tridecanal (nC13al) at a specified final time. In addition to the bounds on the input variables, the total pressure of the gas phase must be kept within the specified limits.



**Figure 3.10.** Hydroformylation reaction network.

The dynamic optimization problem is formulated as follows:

$$\max_{u(t), T(t)} J = c_{nc13al}(t_f)$$

$$\text{s.t. } \dot{c}_{liq,i} = j_i^{GL} + c_{cat} M_{cat} \sum_{j \in \text{ER}} v_{j,i} r_j ; \quad c_{liq,i}(0) = c_{liq,i0}; \quad i=1, 2, \dots, 7;$$

$$\dot{p}_i = \frac{RT}{V_{gas}} (u x_i - V_{liq} j_i^{GL}) \quad (i \in \text{gas}); \quad p_i(0) = p_{i0} ;$$

$$x_i = 0.5 \left( \frac{\text{mol}}{\text{mol}} \right); \quad i = 1, 2;$$

$$j_i^{GL} = \begin{cases} (k_L a)_i (c_i^* - c_{liq,i}), & (\text{if } i \in \text{gas}); \quad i = 1, 2; \\ 0, & (\text{else}); \quad i = 3, 4, \dots, 7 \end{cases} ;$$

$$r_1 = \frac{k_{1,0}(T) c_{nC12en} c_{H_2} c_{CO}}{1 + K_{1,1} c_{nC12en} + K_{1,2} c_{nC13al} + K_{1,3} c_{H_2}} ;$$

$$r_2 = \frac{k_{2,0}(T) (c_{nC12en} - \frac{c_{iC12en}}{K_{p,2}})}{1 + K_{2,1} c_{nC12en} + K_{2,2} c_{iC12en}} ;$$

$$r_3 = \frac{k_{3,0}(T) (c_{nC12en} c_{H_2} - \frac{c_{nC12an}}{K_{p,3}})}{1 + K_{3,1} c_{nC12en} + K_{3,2} c_{nC13an} + K_{3,3} c_{H_2}} ;$$

$$r_4 = k_{4,0}(T) c_{iC12en} c_{H_2} ;$$

$$r_5 = k_{5,0}(T) c_{iC12en} c_{H_2} c_{CO} ;$$

$$r_6 = k_{6,0}(T) c_{nC12en} c_{H_2} c_{CO} ;$$

$$\begin{aligned}
k_j(T) &= k_{0,j} \exp\left(-\frac{E_{A,j}}{R}\left(\frac{1}{T} - \frac{1}{T_{ref}}\right)\right); \\
K_{p,j} &= \exp\left(\frac{-\Delta G_j}{RT}\right); \\
-\Delta G_j &= a_{0,j} + a_{1,j}T + a_{2,j}T^2; \\
c_{cat} &= \frac{c_{cat,tot}}{1 + K_{cat,1}c_{CO}^{K_{cat,3}} + K_{cat,2}\frac{c_{CO}^{K_{cat,3}}}{c_{H_2}}}; \\
c_i^* &= \frac{p_i}{H_i}; \\
H_i &= H_i^0 \exp\left(\frac{-E_{A,H,i}}{RT}\right); \\
p_{total}(t) &= p_{H_2}(t) + p_{CO}(t); \\
1 \text{ bar} &\leq p_{total}(t) \leq 20 \text{ bar}; \\
0 &\leq u(t); \\
368.15 \text{ K} &\leq T(t) \leq 388.15 \text{ K}
\end{aligned} \tag{3.6}$$

where  $i$  represents the component index ( $i=1,2,\dots,7$  for the liquid phase and  $i=1,2$  for the gas phase),  $j$  stands for the reaction index and  $R$  is the reaction set. The final time  $t_f$  is fixed at 80 min. All related model parameters are given in Appendix 2. Equal molar content of  $CO$  and  $H_2$  in the syngas is assumed. The liquid volume  $V_{liq}$  and the gas volume  $V_{gas}$  inside the reactor are assumed constant, namely 900 mL each. The initial molar amount of the main reactant 1-dodecene is 0.85 mol, while all other initial conditions for the chemical species in the liquid phase are set to zero. The initial partial pressures of the  $CO$  and  $H_2$  in the gas phase are 10 bar.

### 3.2.3.1 Computed optimal solutions

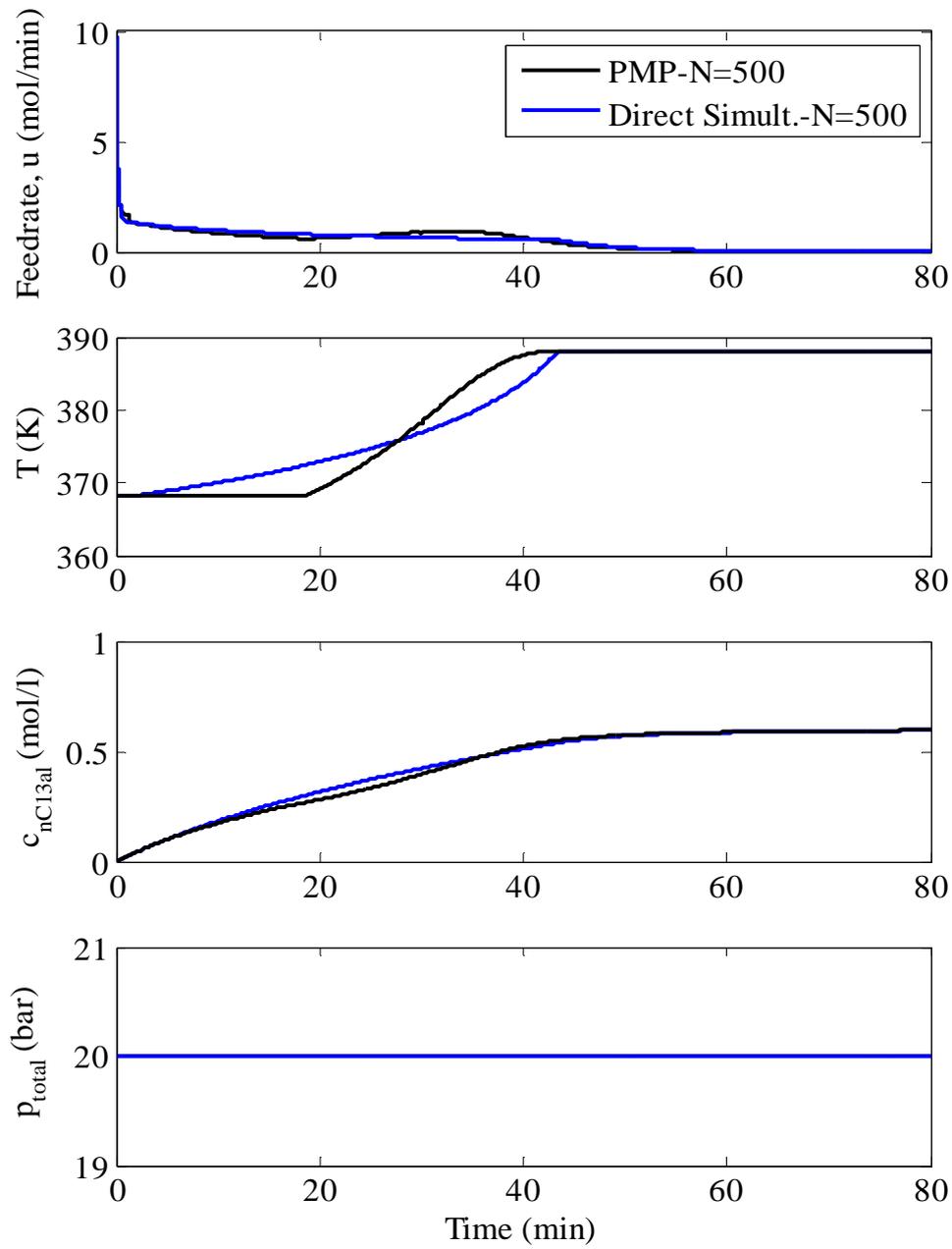
The parameter values  $\alpha_0 = 0.02$ ,  $K = 100$  and  $\varepsilon = 0.05$  are used in the PMP-based approach. Due to the lack of sensitivity of the cost function with respect to the fine shape of the input profiles in some of the arcs, a relatively fine input discretization ( $N \geq 100$ ) is necessary for accurate results. The optimal input trajectories and the corresponding concentration of the desired product and total pressure are given in Fig. 3.11 for  $N=500$ . The temperature is initially at the lower bound to favor the desired reaction.

With the effect of gas feeding, the concentration of the desired product increases and approaches its maximal value after about 50 min. Then, the temperature is set to its upper limit to suppress the undesired side reactions. This results in a relatively small increase in the concentration of n-tridecanal in the last part of the batch run.

The switching functions  $s_u$  and  $s_T$  are illustrated in Fig. 3.12. The solution structure and the performance of both numerical schemes for two discretization levels are given in Table 3.5.

**Table 3.5.** Comparison of the indirect PMP-based and direct simultaneous strategies for Problem 3.2.3

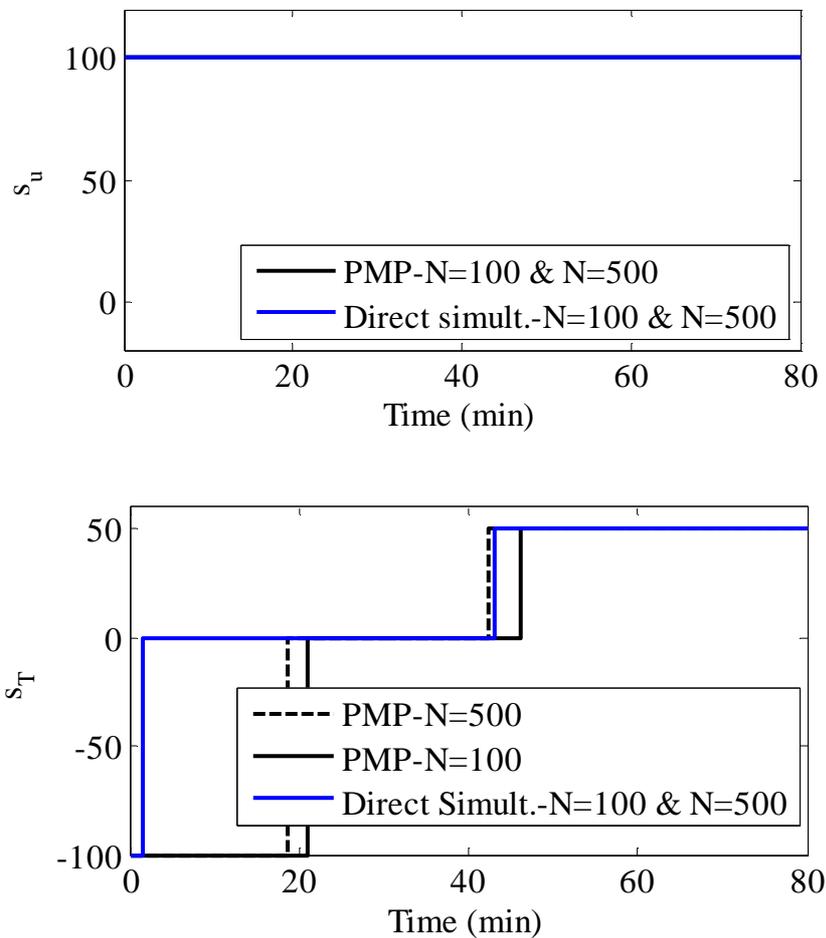
| Algorithm            | Solution Structure |                   |                   |
|----------------------|--------------------|-------------------|-------------------|
|                      | Arc 1              | Arc 2             | Arc 3             |
| Indirect PMP-based   | $u_{\text{path}}$  | $u_{\text{path}}$ | $u_{\text{path}}$ |
| N=100, J=0.593 mol/L | $T_{\text{min}}$   | $T_{\text{sens}}$ | $T_{\text{max}}$  |
| N=500, J=0.595 mol/L |                    |                   |                   |
| Direct simultaneous  | $u_{\text{path}}$  | $u_{\text{path}}$ | $u_{\text{path}}$ |
| N=100, J=0.595 mol/L | $T_{\text{min}}$   | $T_{\text{sens}}$ | $T_{\text{max}}$  |
| N=500, J=0.596 mol/L |                    |                   |                   |



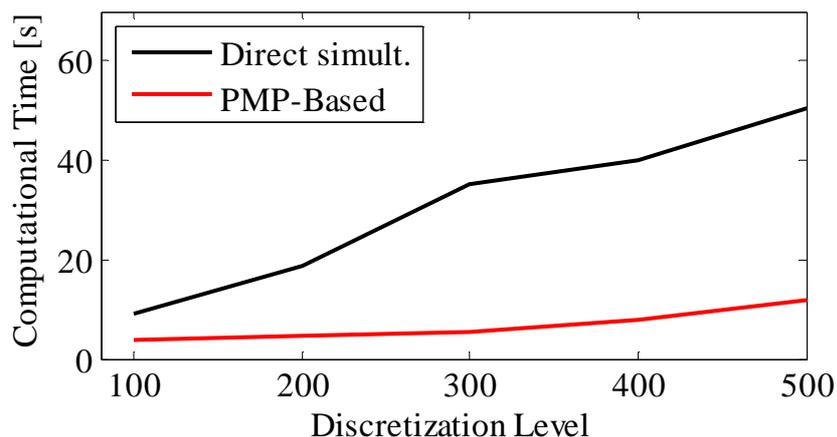
**Figure 3.11.** Optimal input and state profiles computed via the PMP-based method and the direct simultaneous strategy.

A few remarks are made at this point:

1. Both numerical schemes exhibit a 3-arc solution for different discretization levels as shown in Table 3.5.
2. The pressure upper bound is active throughout the batch run. This is enforced by adjusting the gas feed rate  $u(t)$ , which is therefore constraint seeking throughout. Toward the end of the batch, the feed rate is very close, but not exactly equal, to zero.
3. There is a significant difference in the two sensitivity-seeking temperature profiles. Again, this is due to the lack of sensitivity of the objective function with respect to the temperature  $T_{sens}(t)$ .



**Figure 3.12.** Switching functions  $s_u$  and  $s_T$  .



**Figure 3.13.** Computational times for different discretization levels.

Fig. 3.13 shows that the computational time of the PMP-based strategy is significantly shorter than that of the direct simultaneous method.

### 3.3 Summary

A PMP-based quasi-Newton algorithm has been proposed for solving constrained dynamic optimization of batch and semi-batch chemical processes. This algorithm constructs the Hamiltonian function by indirectly adjoining the inequality path constraints via their time derivatives so that the inputs can be easily enforced to satisfy the active path constraints at each iteration step. Symbolic differentiation of the Hamiltonian function with respect to the states is only necessary at the initialization step. The results show that the proposed PMP-based quasi-Newton algorithm can solve the investigated constrained optimization problems significantly faster than direct simultaneous methods as the discretization grid gets finer.

This chapter also shows that, although there are only negligible differences between the optimal costs determined with various strategies, the actual input profiles can differ significantly and even correspond to different local solutions. The main reason for this observation is the lack of sensitivity of the objective function with respect to the sensitivity-seeking parts of certain inputs. Hence, it may be useful to parameterize these input profiles in a so-called parsimonious way, for example by using switching times and low-order polynomial approximations rather than piecewise-constant or piecewise-linear approximations. In the next chapter, the proposed PMP-based solution method is combined with such parsimonious parameterization schemes to speed-up the numerical solution of the dynamic optimization problems of constrained semi-batch processes.



*The only thing that will redeem mankind is cooperation.*

---

Bertrand Russell (1872 – 1970)

# 4 DYNAMIC OPTIMIZATION COMBINING PONTRYAGIN'S MINIMUM PRINCIPLE AND PARSIMONIOUS INPUT PARAMETERIZATION

In dynamic optimization problems, the optimal input profiles consist of various arcs (Srinivasan et al., 2003b). These arcs can be categorized depending on their characteristics as follows: An optimal input arc might be on a lower or an upper bound ( $u_{\min}$  or  $u_{\max}$ ), on a path constraint ( $u_{\text{path}}$ ) (which activates the corresponding path constraint) or inside the feasible region and behaving as a sensitivity-seeking arc ( $u_{\text{sens}}$ ). Accurate computation of a sensitivity-seeking arc can be burdensome since the fine configuration of a sensitivity-seeking arc advances the optimal cost negligibly. Accordingly, most solution schemes (direct as well as indirect) demand significantly finer input discretization levels to obtain accurate solutions. On the other hand, as an alternative to full discretization/parameterization, the sensitivity-seeking arcs can be parameterized alternatively, in a *parsimonious* manner. This way, the number of decision variables in the dynamic optimization problem can be reduced significantly (Welz et al., 2005; Schlegel et al., 2005; Welz et al., 2006). This chapter details the combination of the proposed efficient PMP-based solution algorithm in Chapter 3 with

such parsimonious input parameterizations so as to further reduce the corresponding computational load of constrained nonlinear dynamic optimization problems.

## 4.1 Solution Methodology

The proposed indirect solution algorithm can be used to solve the constrained problem given in Eq. 2.1. It parameterizes the inputs using  $N$  piecewise-constant elements, integrates the state equations forward in time and the co-state equations backwards in time, leading to a gradient-based control vector iteration approach. Pure state path constraints are handled by indirect adjoining into the Hamiltonian function, that enables the explicit computation of the values of the inputs to activate the infeasible path constraints in the iteration. If a path or terminal constraint is violated, the corresponding Lagrange multiplier is penalized so as to keep the optimization iterates within the feasible region. This way, the complexity of the optimization problem can be reduced.

Although indirect-based methods have been shown to be efficient for the dynamic optimization of batch and semi-batch processes, there is still the requirement of fine input discretization to obtain reliable solutions. This requirement might result in significant computational effort for small as well as larger-scale problems. Nevertheless, a parameterization of the sensitivity-seeking arcs with respect to switching times can decrease the computational complexity of the problem further (Visser et al., 2000; Welz et al., 2006; Kadam et al., 2007; Aydin et al., 2017b).

Given the optimal solution structure (the types and sequence of arcs), an alternative and parsimonious parameterization of the form  $u(t) = U(\pi)$  can be postulated to reformulate the problem. Then, the effective indirect solution algorithm can be applied to the reformulated parsimonious problem.

For example, a sensitivity-seeking input arc can be expressed as a linear arc between the two values  $u_1$  and  $u_2$  at the two adjustable switching times  $t_1$  and  $t_2$ , which represent the beginning and the end of the arc, thus resulting in the new input vector  $\pi = (t_1, t_2, u_1, u_2)^T$ . This reformulation allows applying the NCO as follows:

$$\begin{aligned}
\min_{\pi} \quad & \tilde{H}(t) := \lambda^T \tilde{F}(x, \pi) + \mu^T \tilde{S}(x, \pi) \\
\text{s.t.} \quad & \dot{x} = \tilde{F}(x, \pi); \quad x(0) = x_0; \\
& \lambda^T = -\frac{\partial \tilde{H}}{\partial x}, \quad \lambda^T(t_f) = \frac{\partial \phi}{\partial x} \Big|_{t_f} + v^T \frac{\partial \tilde{T}}{\partial x} \Big|_{t_f}; \\
& \mu^T \tilde{S}(x, \pi) = 0; \quad v^T T(x(t_f)) = 0; \\
& \frac{\partial \tilde{H}(t)}{\partial \pi} = \lambda^T \frac{\partial \tilde{F}(x, \pi)}{\partial \pi} + \mu^T \frac{\partial \tilde{S}(x, \pi)}{\partial \pi} = 0
\end{aligned} \tag{4.1}$$

If the solution structure consists of the 3 arcs  $u_{\max}$ ,  $u_{sens}$  and  $u_{\min}$ , the sensitivity-seeking arc  $u_{sens}$  can be approximated using linear interpolation between the two switching times  $t_1$  and  $t_2$ , thus giving:

$$\tilde{u}(t) = \begin{cases} u_{\max} & \text{if } 0 \leq t < t_1; \\ u_{sens}(t) = u_{\max} + \frac{u_{\min} - u_{\max}}{t_2 - t_1} (t - t_1) & \text{if } t_1 \leq t < t_2; \\ u_{\min} & \text{if } t_2 \leq t < t_f \end{cases} \tag{4.2}$$

Then, the problem given by Eq. 4.4 is solved using the proposed PMP-based algorithm given in Chapter 3.

**Remark 4.1.** The proposed algorithm requires an initial solution to the problem given by Eq.2.1, using either a direct or an indirect method, in order to decide on the parameterization scheme. However, this solution need not be obtained using fine discretization to ensure feasibility. It is only a pre-analysis step for suitable parameterization candidates.

**Remark 4.2.** The parsimonious parameterization is usually problem specific. It can be extended to more complex problems by using higher-order polynomials or multiple switching times instead of linear relations.

**Remark 4.3.** Parsimonious parameterization was found particularly effective for problems with a small number of inputs.

**Remark 4.4.** The inputs that activate the path constraints  $u_{\text{path}}$  can usually be computed using the model equations, without any optimization (Srinivasan et al., 2003b; Aydin et al., 2018b). This will be detailed in the second case study.

The overall indirect parsimonious algorithm can be formulated as follows:

---

## Parsimonious PMP-based Solution Algorithm

---

Consider the optimization problem given by Eq. 4.1, with  $N$  discrete time instants between 0 and  $t_f$ . Let us discretize the Lagrange multipliers for the path constraints as  $\mu(t) = M(M)$ , where  $M$  is a  $(n_s \times N)$  matrix. Finally, let us write the co-states as  $\dot{\lambda} = \tilde{G}(x, \pi, M, v)$ .

Select values for the penalty term  $K > 0$ , maximum iteration  $iter\_max$ , the step size  $\alpha$ , the threshold  $\varepsilon$ , the number of discrete input values  $N$ . Initialize the iteration counter  $h = 0$  and the input vector  $\pi_0, M_0, v_0$

**do**  $h = 1 \rightarrow iter\_max$

- I. Solve the state equations by forward integration of  $\dot{x} = \tilde{F}(x, \pi)$ , and the co-state equations by backward integration of  $\dot{\lambda} = \tilde{G}(x, \pi, M, v)$ .
- II. **if** the  $j$ th path constraint is such that  $\tilde{S}^j(x, \pi) \leq 0$  at the time instant  $k$ , set  $M_h(j, k) = 0$ . Otherwise, set  $M_h(j, k) = K$ ,  $u_h(j, k) = u_{path}$ , for  $j=1, \dots, n_s$ ,  $k=1, \dots, N$ .  
**end if**
- III. **if** the  $i$ th terminal constraint is such that  $T^i(x(t_f)) < 0$ , set  $v_h(i) = 0$ . Otherwise, set  $v_h(i) = K$ , for  $i=1, \dots, n_T$ .  
**end if**
- IV. Evaluate the value of the gradient  $(\frac{\partial \tilde{H}}{\partial \pi})_h$  using analytical expressions given by Eq. 4.1.
- V. **if**  $\left\| \alpha \left( \frac{\partial \tilde{H}}{\partial \pi} \right)_h \right\| < \varepsilon$ , **stop**, set  $\pi_{opt} := \pi_h$   
**else** set  $\pi_{h+1} := \pi_h - \alpha \left( \frac{\partial \tilde{H}}{\partial \pi} \right)_h$ , where  $\alpha=0.05$ . Go to I  
**end if**

**end do**

---

**Remark 4.5** Constant step-size ( $\alpha$ ) is being used for the parsimonious indirect method. Note that an adaptive line search, e.g. Wolfe criterion (Wolfe, 1971), could speed up the method.

## 4.2 Computation of Gradients

The parsimonious parameterization results in discrete decision variables along the time horizon that include the switching times as well as other input parameters. Since the general PMP algorithm defines the sensitivity of the Hamiltonian with respect to the input variables, one needs to evaluate  $\frac{\partial \tilde{H}(t)}{\partial \pi}$  in order to perform the optimization shown in the algorithm. This can be done by considering each arc individually and performing a time transformation of the system equations as detailed next.

Consider the system equation in Problem 4.1:

$$\dot{x} = \tilde{F}(x, \pi), \quad \text{for } t \in [t_0, t_f] \quad (4.3)$$

For simplicity of presentation, let us assume that the optimal solution consists of the three following arcs:

$$\tilde{u}(t) = \begin{cases} 1 & \text{if } t_0 < t \leq t_1; \\ u_0 & \text{if } t_1 < t \leq t_2; \\ 0 & \text{if } t_2 < t \leq t_f \end{cases} \quad (4.4)$$

where  $u_0$  is a scalar decision variable, along with  $t_1$  and  $t_2$ . Hence,  $\pi = (t_1, t_2, u_0)^T$ , and we need to compute  $\frac{\partial \tilde{H}}{\partial t_1}, \frac{\partial \tilde{H}}{\partial t_2}, \frac{\partial \tilde{H}}{\partial u_0}$ .

Let us consider the first arc with the single decision variable  $t_1$  and introduce the dimensionless time  $\tau$  valid between  $t_0$  and  $t_1$ . Time  $t$  relates to the dimensionless time  $\tau$  as follows:

$$t(\tau) := t_0 + \tau(t_1 - t_0), \quad \tau \in [0,1] \quad (4.5)$$

which allows writing the states, their derivatives and the system equations in terms of  $\tau$ :

$$\begin{aligned} \hat{x}(\tau) &:= x(t(\tau)) \\ \frac{d}{d\tau} \hat{x}(\tau) &= \frac{d}{dt} x(t) \frac{dt}{d\tau} = \frac{d}{dt} x(t) t_1 \\ \hat{F}_1(\hat{x}(\tau)) &:= \tilde{F}(x(t), u = 1) \end{aligned} \quad (4.6)$$

Repeating this procedure for the second and third arcs gives:

$$\frac{d}{d\tau} \hat{x}(\tau) = \begin{cases} t_1 \hat{F}_1(\hat{x}(\tau)) & \text{if } 0 < t \leq t_1; \\ (t_2 - t_1) \hat{F}_2(\hat{x}(\tau), u_0) & \text{if } t_1 < t \leq t_2; \\ (t_f - t_2) \hat{F}_3(\hat{x}(\tau)) & \text{if } t_2 < t \leq t_f \end{cases} \quad (4.7)$$

where  $\hat{F}_1(\hat{x}(\tau)) := \tilde{F}(x(\tau), u = 1)$ ,  $\hat{F}_2(\hat{x}(\tau), u_0) := \tilde{F}(x(\tau), u = u_0)$  and

$$\hat{F}_3(\hat{x}(\tau)) := \tilde{F}(x(\tau), u = 0).$$

Finally, differentiating Eq. 4.7 with respect to  $\pi = (t_1, t_2, u_0)^T$  gives:

$$\begin{aligned} \frac{\partial \hat{F}}{\partial t_1} &= \hat{F}_1(\hat{x}(\tau)) - \hat{F}_2(\hat{x}(\tau), u_0); \\ \frac{\partial \hat{F}}{\partial t_2} &= \hat{F}_2(\hat{x}(\tau), u_0) - \hat{F}_3(\hat{x}(\tau)); \\ \frac{\partial \hat{F}}{\partial u_0} &= \frac{\partial \hat{F}_2(\hat{x}(\tau), u_0)}{\partial u_0}; \\ \frac{\partial \tilde{H}(t)}{\partial \pi} &= \lambda^T \frac{\partial \tilde{F}(x, \pi)}{\partial \pi} + \mu^T \frac{\partial \tilde{S}(x, \pi)}{\partial \pi} = 0 \end{aligned} \quad (4.8)$$

Note that the gradient  $\frac{\partial \mathcal{S}}{\partial \pi}$  can be obtained in a similar way.

### 4.3 Case Studies

To investigate the application of the proposed methodology to the dynamic optimization of constrained semi-batch processes, two case studies are presented. The first problem is the dynamic optimization of a batch binary distillation column with terminal purity constraints given in Chapter 3 and in (Aydin et al., 2017a). The second problem, which deals with the dynamic optimization of a complex fed-batch chemical process in the presence of path constraints, which is discussed also in Chapter 3 and is taken from Hentschel et al. (2015).

The two problems are solved using both a direct simultaneous method and the indirect (PMP-based) parsimonious method proposed in this work. The CasADi toolbox (Andersson and Diehl, 2012) is used for the implementation of the direct simultaneous method, along with the NLP solver IPOPT (Wächter and Biegler, 2006).

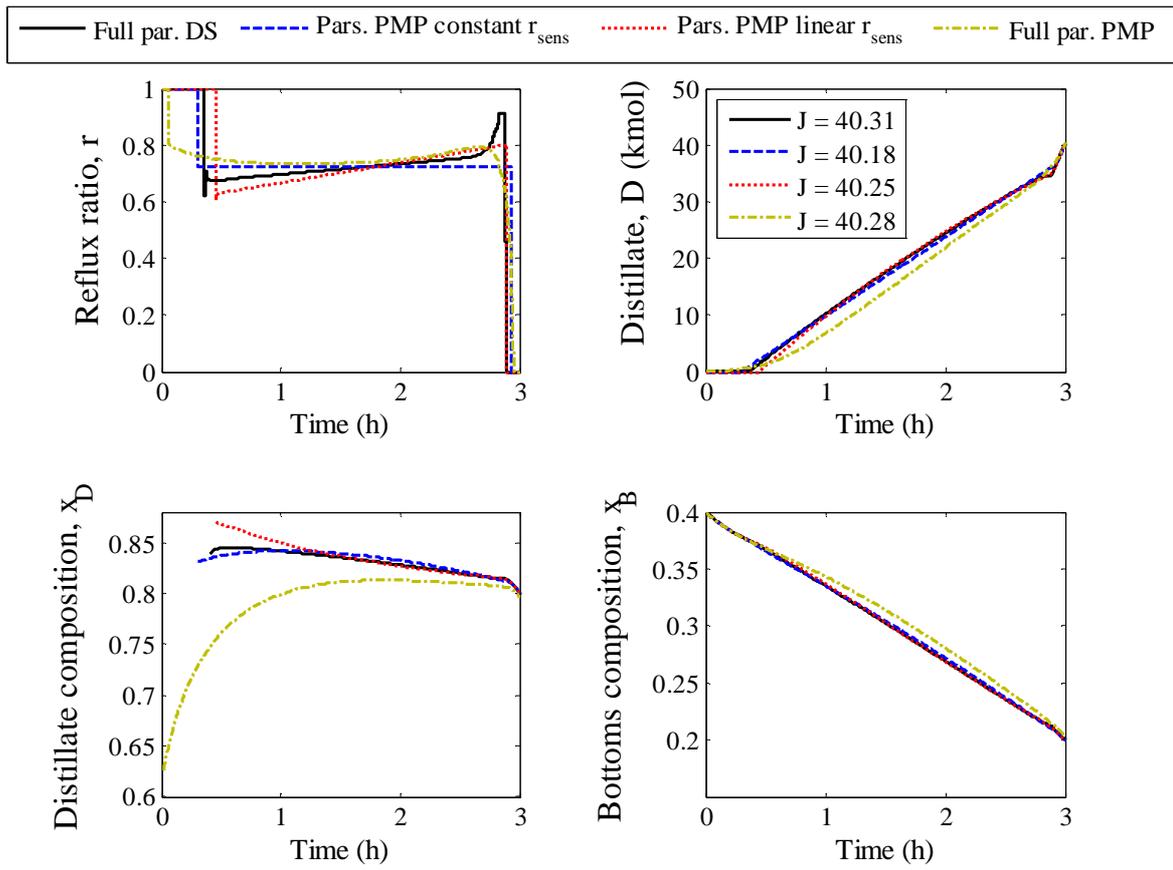
### 4.3.1 Batch binary distillation with terminal purity constraints

Consider Problem 3.2.3, which is a batch binary distillation column with three equilibrium plates, in which the components A and B (more volatile) are separated from each other. The objective is to maximize the molar amount of  $D$  in the distillate for the given fixed batch time  $t_f = 3$  h, while satisfying the terminal purity constraints of at least 80 mol % of B in the distillate ( $x_D$ ) and at most 20 mol % of B in the bottom product ( $x_B$ ). The internal reflux ratio  $r = L/V$  is the input variable. The reader is referred to Chapter 3 for more detail related to the case study and model parameters. The optimization problem can be summarized as follows:

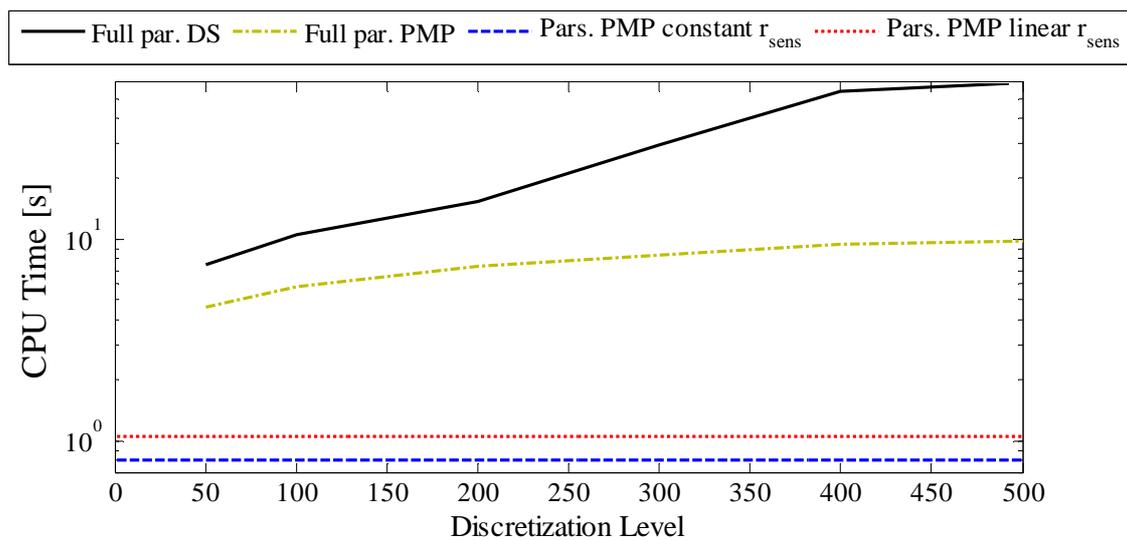
$$\begin{aligned} \max_{r(t)} \quad & J = D(t_f) \\ \text{s.t.} \quad & \text{dynamic model eqns. (from Eq. 3.5);} \\ & x_0; x_D(t_f) \geq 0.8; x_B(t_f) \leq 0.2; 0 \leq r(t) \leq 1 \end{aligned} \tag{4.9}$$

The optimal solutions computed by the direct simultaneous (DS) method (500 piecewise-continuous input parameterization and collocation on finite elements) and the fully parameterized PMP (500 piecewise-continuous input parameterization) are given in Fig. 4.1. As discussed in Chapter 3, both solutions suggest total reflux at the beginning of the batch to enhance the composition at the top of the column. Then, the sensitivity-seeking arc  $r_{sens}$  produces maximum distillate with the required purity. Finally, a quick third arc with zero reflux is followed to recover the high-purity material that is still present at the top of the column.

In this problem, since  $r_{sens}$  does not start at the maximum value 1 and end up with the minimum value 0, we firstly postulate a constant reflux ratio ( $r_b$ ) for this sensitivity-seeking arc. As a result, the decision variables for the parsimonious method become  $\pi = (t_1, t_2, r_b)^T$ . Alternatively, the reflux ratio could also be defined as varying linearly between the two switching times. Accordingly, the decision variables would read  $\pi = (t_1, t_2, r_{b1}, r_{b2})^T$ . The solutions given by the two PMP-based parsimonious parameterization methods are also given in Fig. 4.1. These solutions result in fairly close optimal value of the fully parameterized methods. Furthermore, the addition of one parameter ( $r_{b1}$  and  $r_{b2}$  instead of  $r_b$ ) increases the optimal cost further. Fig. 4.2 shows that parsimonious parameterization combined with PMP-based method significantly reduces the computational time, in particular with fine grids.



**Figure. 4.1.** Fully parameterized DS and PMP as well as parsimonious parameterized PMP results.



**Figure. 4.2.** Computational times with different solution methods.

### 4.3.2 Fed-batch hydroformylation reactor with path constraints

Consider Problem 3.2.3 given in Chapter 3, which is the production of n-tridecanal (nC13al) from 1-dodecene (nC12en) that reacts with syngas ( $H_2 + CO$ ). The model parameters are given in Table 3.2. The operational aim is to maximize the concentration of tridecanal (nC13al) at the final time  $t_f = 80$  min. In addition to the bounds on the input variables, the total pressure of the gas phase must be kept within specified limits for safety reasons.

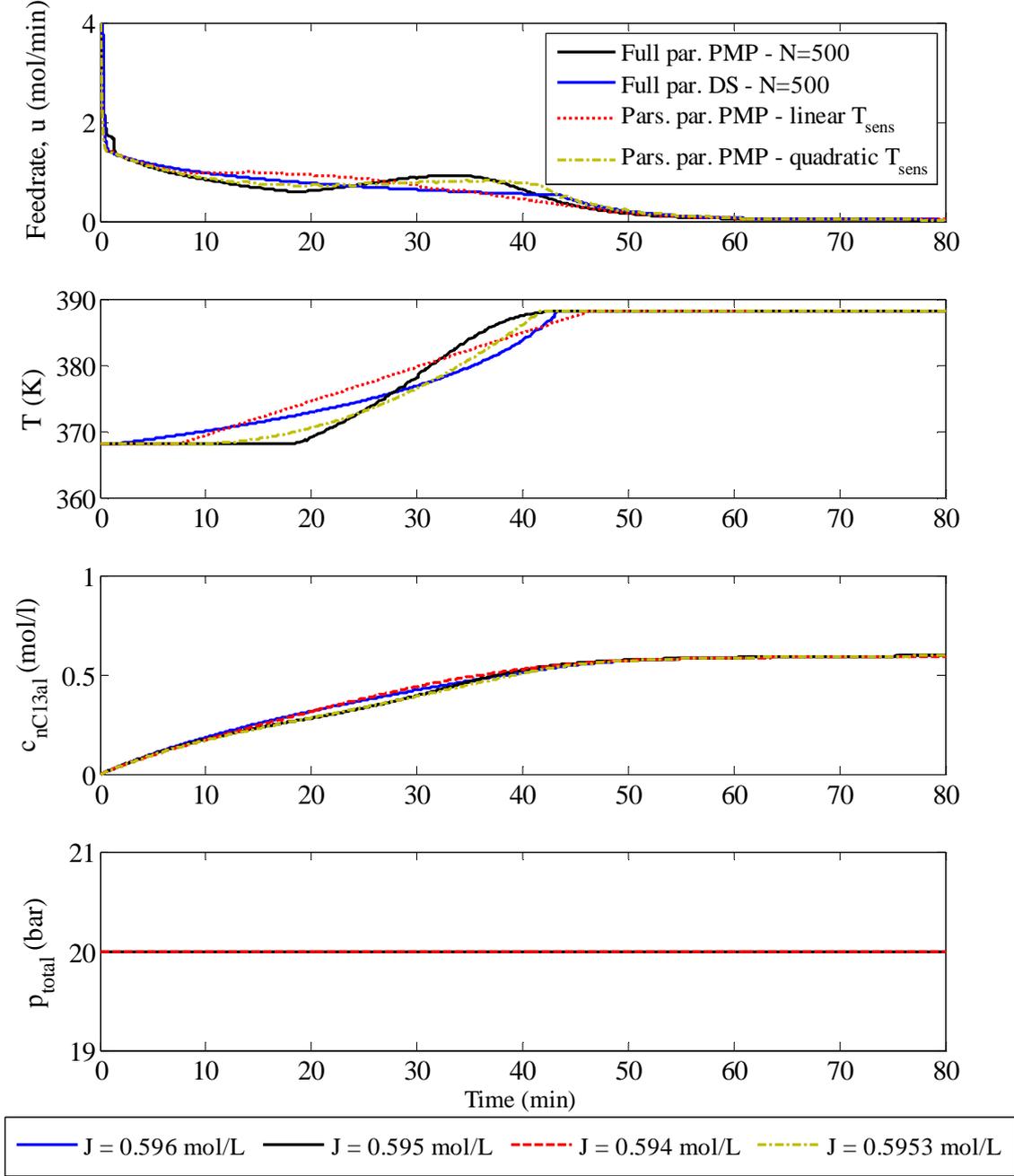
$$\begin{aligned}
 & \max_{u(t), T(t)} J = c_{nC13al}(t_f) \\
 & \text{s.t.} \quad \text{dynamic model eqns. ; } x_0; \\
 & \quad \text{gas – liquid mass transfer eqns. given by (Eq. 3.6);} \\
 & \quad \text{physical constraints; rate expressions;} \\
 & \quad 0 \leq u(t); 368.15 \text{ K} \leq T(t) \leq 388.15 \text{ K} \\
 & \quad 1 \text{ bar} \leq p_{total}(t) \leq 20 \text{ bar;}
 \end{aligned} \tag{4.10}$$

The solutions obtained using DS and fully parameterized PMP-based algorithms are given in Fig. 4.3. As discussed earlier, both solutions suggest that the optimal temperature starts at the minimum level  $T_{min}$  to favor the desired reaction. Then, it follows the sensitivity-seeking arc  $T_{sens}$  to speed up the production of nC13al by boosting the forward reaction  $r_1$ , while restraining the equilibrium reaction  $r_2$  to the product side. Finally, the optimal temperature follows the upper level  $T_{max}$  to suppress the undesired reactions. It is seen that the optimal temperature profile consists of 3 arcs. Accordingly, it can be postulated to parameterize  $T_{sens}$  linearly between the two adjustable switching times. Alternatively,  $T_{sens}$  can also be parameterized using quadratic interpolation between the two switching times.

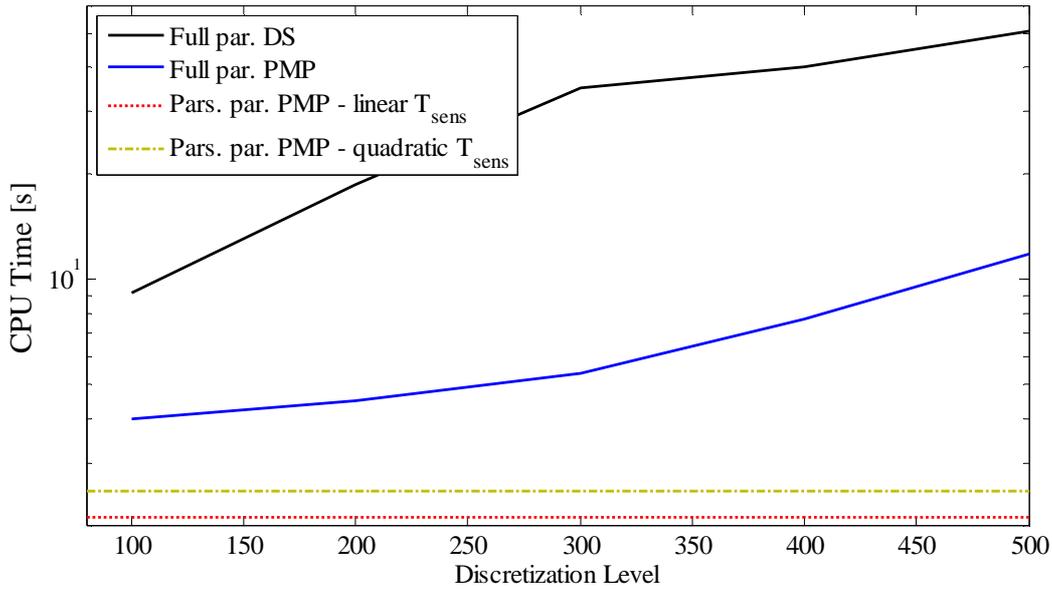
Since the pressure constraint is active throughout the batch, the optimal feedrate of syngas can be determined by tracking the pressure path constraint. This is done analytically from Eq. 3.2, as follows:  $\dot{p}_{total}(t) = \dot{p}_1(t) + \dot{p}_2(t) = 0$ , which gives  $u_{path}(t) = V_{liq} (j_1^{GL}(t) + j_2^{GL}(t))$ .

It is observed that the cost values obtained by the different methods are very similar to each other, as given in Fig. 4.3. Note that increasing the order of the polynomial interpolation results in a slightly better cost value.

Fig. 4.4 compares the computational times needed to solve the problem using the various approaches. It is clear that both parsimonious PMP-based methods are much faster than the fully parameterized ones.



**Figure. 4.3.** Optimal solutions.



**Figure 4.4.** Computational times of different solution methods.

## 4.4 Summary

This chapter details an alternative indirect solution method that parameterizes the sensitivity-seeking inputs parsimoniously and uses a PMP-based indirect method to solve the dynamic optimization for constrained semi-batch processes. The performance of the algorithm is compared with DS and fully parameterized PMP-based algorithms. It is observed that the proposed method can solve the corresponding problems much faster for very similar optimal cost values.

Indirect-based parsimonious methods may possess certain advantages for more complex applications such as stochastic and multi-level optimization, real-time application of nonlinear model predictive control and multi-stage algorithms, where fast implementation are required. In addition, application of indirect-based parsimonious algorithms to the large-scale optimization problems can turn out to be promising, but is still an open field.

# **PART III**

**Efficient**

**Nonlinear**

**Model Predictive Control**

*People are sometimes afraid to ask questions out of fear of seeming "stupid". Yet the smartest people on the planet are often the ones who ask the most questions.*

---

Albert Einstein (1879 – 1955)

# 5 NONLINEAR MODEL PREDICTIVE CONTROL USING PONTRYAGIN'S MINIMUM PRINCIPLE

Nonlinear model predictive control (NMPC) is an important tool for the real-time optimal operation of batch and semi-batch processes. Direct methods are often the methods of choice to solve the corresponding optimal control problems, in particular for large-scale problems. However, as discussed in Chapter 2, the matrix factorizations associated with large prediction horizons can be computationally demanding. In contrast, as discussed in Chapter 3 and 4, indirect methods can be competitive for small and mid-scale problems. Furthermore, the interplay between states, co-states and Lagrange multipliers for path constraints in the context of Pontryagin's Minimum Principle (PMP) might turn out to be computationally quite efficient (Cannon et al., 2008).

This chapter proposes to use the indirect solution technique discussed in Chapter 3 for the shrinking-horizon NMPC (sh-NMPC) of semi-batch processes. In particular, the technique deals with path constraints via indirect adjoining, which allows dealing with the path constraints explicitly at each iteration. Uncertainties are handled by introducing time-varying backoff terms for the path constraints. The resulting sh-NMPC algorithm is tested on a two-phase semi-batch reactor for the hydroformylation of 1-dodecene in the presence of

uncertainty, and its performance is compared to that of NMPC that uses a direct simultaneous optimization method.

## 5.1 Shrinking-horizon NMPC Problem

In shrinking-horizon NMPC (sh-NMPC) of batch and semi-batch processes, the optimal control problem to be solved on-line at each iteration can be written as follows:

$$\begin{aligned}
\min_{u(t)} \quad & J = \phi(x(t_f, \tilde{\theta})) \\
\text{s.t.} \quad & \dot{x} = F(x, u, \tilde{\theta}), \quad x(t_k) = x_k \\
& S(x, u, \tilde{\theta}) \leq 0, \quad T(x(t_f), \tilde{\theta}) \leq 0, \quad t \in [t_k, t_f]
\end{aligned} \tag{5.1}$$

where  $t_k$  is the  $k$ -th sampling time,  $J$  is a scalar performance index that depends on the values of the states at the final time  $t_f$ ,  $\phi$  is the objective function,  $x$  is the  $n_x$ -dimensional state vector with the corresponding initial conditions  $x_k$ ,  $u$  is the  $n_u$ -dimensional input vector,  $S$  is the  $n_S$ -dimensional vector of inequality path constraints that include input bounds,  $T$  is the  $n_T$ -dimensional vector of inequality terminal constraints, and  $\tilde{\theta}$  is the vector of parameters which are uncertain and associated with plant-model mismatch. After solving Problem (5.1), the first part  $u[t_k, t_k + \delta]$  of the optimal inputs is implemented in the plant, the horizon is shrunk by the sampling interval  $\delta$ , and a new optimal control problem is solved. This procedure is repeated iteratively until the final batch time is reached. Sh-NMPC is illustrated in Fig.1.1

Several methods are available in the literature to cope with uncertainties in the context of stochastic programming or two-level approaches (Sahinidis, 2004; Li et al., 2008; Mesbah et al., 2014; Puschke et al., 2016; Mesbah, 2016; Puschke and Mitsos, 2016). However, the computational time associated with these methods can still be a limitation for real-time optimization. To deal with the effect of uncertainties, time-varying backoffs will be introduced (Visser et al., 2000; Srinivasan et al., 2003a; Shi et al., 2016). Furthermore, it is assumed that  $x_k$  can be measured or estimated using on-line sensors and state estimation (Allgöwer et al., 1999; Rao et al., 2001; Rao and Rawlings, 2002; Schneider and Georgakis, 2013). Using small sampling times and frequent on-line measurements, the conservatism associated with the robust backoff approach can be reduced.

The optimal control problem given in Eq. 5.1 can be reformulated using PMP and the constraint backoffs as follows:

$$\begin{aligned}
\min_{u(t)} \quad & H(t) = \lambda^T F(x, u, \bar{\theta}) + \mu^T [S(x, u, \bar{\theta}) + b_s] \\
\text{s.t.} \quad & \dot{x} = F(x, u, \bar{\theta}); \quad x(t_k) = x_k; \quad t \in [t_k, t_f] \\
& \dot{\lambda}^T = -\frac{\partial H}{\partial x}, \quad \lambda^T(t_f) = \frac{\partial \phi}{\partial x} \Big|_{t_f} + v^T \frac{\partial T}{\partial x} \Big|_{t_f}; \\
& v^T [T(x(t_f), \bar{\theta}) + b_T] = 0 \\
& \frac{\partial H(t)}{\partial u} = \lambda^T \frac{\partial F}{\partial u} + \mu^T \frac{\partial S}{\partial u} = 0
\end{aligned} \tag{5.2}$$

where  $H$  is the Hamiltonian function,  $\lambda$  the  $n_x$ -dimensional vector of Lagrange multipliers (also called co-states or adjoints) for the system equations,  $\bar{\theta}$  the estimated parameters,  $\mu$  the  $n_s$ -dimensional vector of Lagrange multipliers for the path constraints, and  $v$  the  $n_T$ -dimensional vector of Lagrange multipliers for the terminal constraints,  $b_s$  and  $b_T$  are the backoffs associated with the path and terminal constraints, respectively. The terms  $\mu^T [S(x, u, \bar{\theta}) + b_s] = 0$  and  $v^T [T(x(t_f), \bar{\theta}) + b_T]$  are the complementary slackness conditions that will be satisfied at the optimum. Additionally, the partial derivatives of the Hamiltonian function with respect to the inputs must all be equal to zero at the optimum.

---

## Robust PMP-based Solution Algorithm for sh-NMPC

---

Select values for the penalty term  $K > 0$ , the step size  $\alpha$ , the coefficient  $\beta$ , the threshold  $\varepsilon$ , the number of discrete input values  $N$ , the maximal number of iterations  $iter\_max$ , and the backoffs  $b_s$ . Initialize the input matrix  $U_0$ ,  $M_0$ ,  $v_0$  and the Hessian matrix corresponding to the  $k$ -th time instant,  $B_0^k := I_{n_u}$ .

**do**  $h = 0 \rightarrow iter\_max$

- 1) Solve the state equations by forward integration and the co-state equations  $\dot{\lambda} = G(x, u, M, v)$  by backward integration and compute the constraint matrix  $M$  as follows: if the  $j$ -th constraint is satisfied at the  $k$ -th discrete time instant, set  $M_h(j, k) := 0$ , otherwise set  $M_h(j, k) := K$ .
- 2) Evaluate the matrix of first-order gradients  $(\nabla_U H)_h$  by using pre-computed analytical expressions.
- 3) **if**  $\|(\nabla_U H)_h\| < \varepsilon$ , set  $U_{opt} := U_h$ , **STOP**
- 4) Compute the next inputs as follows:

**do**  $k = 1 \rightarrow N$ ,

$$u_h^k := U_h(\cdot, k)$$

$$\nabla_h^k H := (\nabla_{U(\cdot, k)} H)_h$$

**if** the  $i_{th}$  terminal constraint is such that  $T^i(x(t_f)) < 0$ ,

set  $v_h(i) = 0$ . Otherwise, set  $v_h(i) = K$ , for  $i=1, \dots, n_T$ .

**end if**

**if**  $S(x, u_h^k, \bar{\theta}) + b_s \leq 0$

4.1. Apply line search for  $\alpha_0$  and estimate  $\alpha$

4.2. Compute  $u_{h+1}^k = u_h^k - \alpha (B_h^k)^{-1} \nabla_h^k H$

4.3 Update the Hessian matrix  $B_{h+1}^k$  as follows:

$$s := u_{h+1}^k - u_h^k; \quad y := \nabla_{h+1}^k H - \nabla_h^k H$$

**if**  $s^T y \geq \beta \|s\|^2$ , set  $B_{h+1}^k := B_h^k + \frac{yy^T}{s^T y} - \frac{B_h^k s s^T B_h^k}{s^T B_h^k s}$

**else** set  $B_{h+1}^k := B_h^k$

**end if**

**else** compute  $u_{h+1}^k$  that gives  $S^{\{n\}}(x, u_{h+1}^k, \bar{\theta}) = 0$  and set  $B_{h+1}^k := B_h^k$

**end if**

$$U_{h+1}(\cdot, k) := u_{h+1}^k$$

**end do**

**end do**

---

## 5.2 Case Study: Two-phase Semi-batch Hydroformylation Reactor under Uncertainty

### 5.2.1 Problem formulation

Similar to Problem 3.2.1, consider the semi-batch operation of hydroformylation in a two-phase stirred-tank reactor. The objective is to maximize the final concentration of n-tridecanal (nC13al) from 1-dodecene (nC12en) that reacts with syngas ( $H_2 + CO$ ). The final time is fixed at 70 min. For more detail, the reader is referred to Chapter 3.

The manipulated variables are the reactor temperature  $T(t)$  and the syngas feedrate  $u(t)$ . Equimolar content of  $CO$  and  $H_2$  in the syngas is assumed. The gas and liquid phases are modeled as ideally mixed phases. Nominal model parameters are given in Appendix 2 (Hentschel et al., 2015). In addition to bounds on the input variables, the total pressure of the gas phase must be kept within specified limits. It is seen from Eq. 3.6 that the first time derivative of the total pressure ( $\dot{p}_{total}$ ) contains the input  $u(t)$  explicitly, thus implying relative degree 1 for this constraint. As a result, this constraint can be indirectly adjoined into the Hamiltonian and will be activated at each infeasible iteration as illustrated in Fig. 3.1. Concretely, if the constraint  $S(x, \theta) = p_{total}(x, \theta) \leq (20 \text{ bar} - b_s)$  is violated, the indirectly adjoined constraint  $S^1(x, u, \theta) = \dot{p}_{total}(x, u, \theta)$  will be activated by computing the value of  $u$  that makes  $\dot{p}_{total}(x, u, \theta) = 0$ .

Including the backoff term  $b_s$ , the optimal control problem to be solved on-line at each NMPC iteration is as follows:

$$\begin{aligned}
 & \max_{u(t), T(t)} \quad J = c_{nc13al}(t_f) \quad t \in [t_k, t_f] \\
 & \text{s.t} \quad \text{dynamic model eqns.;} \\
 & \quad \text{gas - liquid mass transfer eqns. given by (Eq. 3.6)} \\
 & \quad c_{liq,i}(t_k) = \hat{c}_{liq,i,k}; \quad p_i(t_k) = p_{i,k}; \quad i=1, 2, \dots, 7; \\
 & \quad 1 \text{ bar} \leq p_{total}(t) \leq 20 \text{ bar} - b_s(t); \\
 & \quad 0 \text{ mol/min} \leq u(t) \leq 7 \text{ mol/min}; \\
 & \quad 368.15 \text{ K} \leq T(t) \leq 388.15 \text{ K} \\
 & \quad \frac{|T(t) - T(t_k)|}{t - t_k} \leq \Delta T_{max} \tag{5.3}
 \end{aligned}$$

where  $t_k$  is the time at the  $k$ -th sampling instant,  $i$  represents the component index ( $i=1,2,\dots,7$  for the liquid phase and  $i=1,2$  for the gas phase),  $\hat{c}_{liq,i,k}$  is the estimated concentration of component  $i$  in the liquid phase,  $p_{i,k}$  is the partial pressure of gas  $i$  in the gas phase,  $b_s$  is the time-varying backoff associated with the pressure upper limit, whose value can be calculated by open-loop Monte Carlo simulations (Shi et al., 2016). Parametric variations are given in Table 5.1. The total partial pressure and the species concentrations are assumed to be measured at each NMPC iteration, e.g. using an on-line IR spectroscopy.

The optimal reactor temperature calculated by NMPC serves as set point for a thermostat that regulates the reactor temperature by adjusting the flowrate of heating/cooling fluid. Hence, a rate constraint on the temperature change is introduced ( $\Delta T_{max}$ ) so that the controller is capable of reaching the set point before the next NMPC iteration. This rate constraint depends on the heating/cooling capacity of the thermostat. On the other hand, the optimal feed flowrate can be implemented directly (Abel et al., 2000; Abel and Marquardt, 2003). It was observed that a relatively fine input discretization ( $N \geq 100$ ) is necessary to get accurate and feasible optimal results, especially with regard to the pressure constraint (Aydin et al., 2017a).

**Remark 5.1.** For steady-state problems, closed-loop stability can be enforced by adding an extra terminal constraint to the problem (García et al., 1989; Diehl et al., 2011; Angeli et al., 2012). The prescribed algorithm can also tackle the terminal constraints. This serves as an additional advantage of the proposed algorithm compared to other indirect methods in the context of NMPC for continuous processes (Cannon et al., 2008).

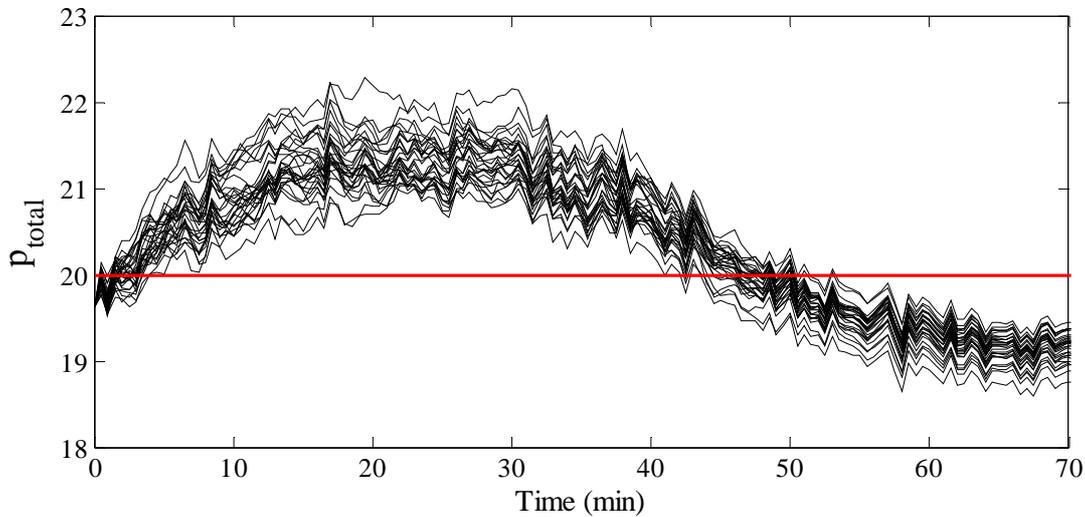
**Remark 5.2.** To speed-up the real-time algorithm, NMPC can be initialized at  $t = 0$  with the nominal optimal control profiles as initial guesses. Then, when the horizon shrinks at each iteration, the last computed input profiles are extrapolated linearly for the new horizon and serve as initial guesses for the next optimization.

**Remark 5.3.** Since the proposed algorithm searches for a feasible point at each iteration, it can be implemented in a sub-optimal fashion by setting a minimal number of iterations to further reduce the computational effort (Findeisen et al., 2007). This can be beneficial for fast processes.

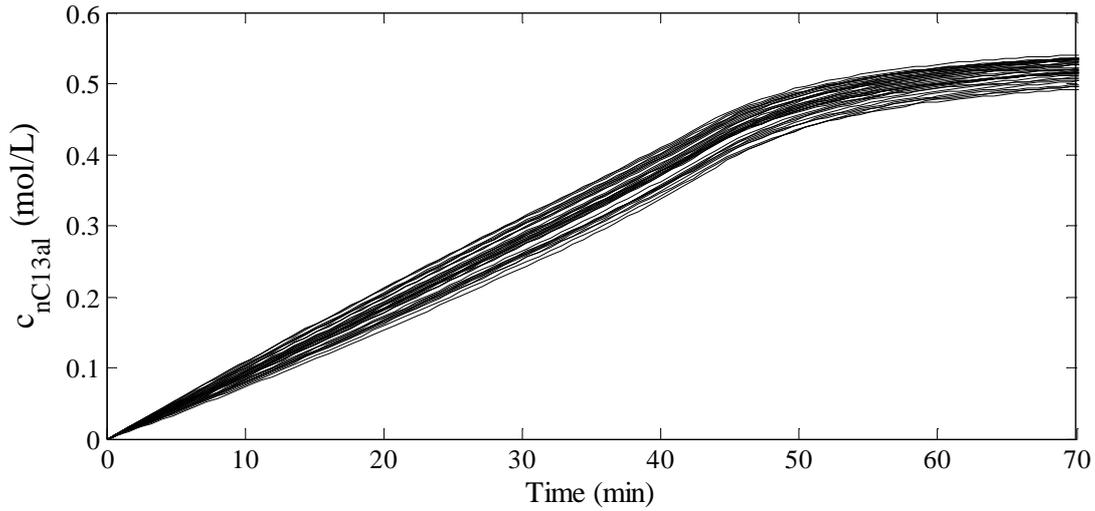
## 5.2.2 Estimation of time-varying back-offs

Back-offs can be useful to enforce feasibility under parametric uncertainty. The conservative nature of back-offs can be reduced through the use of small sampling times. In order to decide on the back-off term  $b_s$ , the multi-step approach of (Shi et al., 2016) is used in this part. First, the nominal optimal input profiles are computed. This was done in Chapter 3. Then, Monte-Carlo simulations are performed using the optimal inputs and sampling various uncertainties. Finally, a time-varying back-off is determined from the standard deviation of constraint violations. Note that the Monte-Carlo approach might require a significant computational effort in the presence of multiple uncertainties, but this work is done off-line.

In this study, the rate constants  $k_{i,0}$  and the catalyst activity  $\gamma$  are assumed to vary within a certain range from batch to batch according to a uniform distribution. On the other hand, the gas-liquid mass-transfer coefficients  $(k_L a)_i$  are assumed to vary within a batch. The uncertainty ranges for the parameters are given in Table 5.1. The results of open-loop Monte Carlo simulations for 40 realizations of multiple uncertainties are depicted in Fig. 5.1



**Figure 5.1.a.** Open-loop Monte Carlo simulations of total pressure for 40 uncertainty realizations.



**Figure 5.1.b.** Open-loop Monte Carlo simulations of product concentration for 40 uncertainty realizations.

**Table 5.1.** Parametric variations:  $(k_L a)_i$  vary within batch, while  $k_{i,0}$  and  $\gamma$  vary from batch to batch.

| <b>Parameter</b> | <b>Nominal Value</b><br>(Hentschel et al., 2015) | <b>Minimal Value</b> | <b>Maximal Value</b> |
|------------------|--|----------------------|----------------------|
| $(k_L a)_1$      | 9.57   | 8.0                  | 10.1                 |
| $(k_L a)_2$      | 7.08   | 5.5                  | 7.6                  |
| $k_{1,0}$        | 4.904  | 2.9                  | 5.2                  |
| $k_{2,0}$        | 4.878  | 3.8                  | 5.8                  |
| $k_{3,0}$        | 2.724  | 1.7                  | 3.7                  |
| $k_{4,0}$        | 2.958  | 1.9                  | 3.9                  |
| $k_{5,0}$        | 3.702  | 2.7                  | 4.7                  |
| $k_{6,0}$        | 3.951  | 2.9                  | 4.9                  |
| $\gamma$         | -  | 80 %                 | 100 %                |

Accordingly, the time-varying back-off is chosen as follows:

$$b_s(t) = \begin{cases} 1.3 & \text{if } t \leq 30 \\ 0.7 & \text{if } 30 < t \leq 55 \\ 0 & \text{if } t > 55 \end{cases}$$

**Remark 5.4** The back-off values can also be updated on a batch-to-batch manner so as to increase the performance of future batches (Bonvin and François, 2017).

### 5.2.3 NMPC for product maximization

The PMP-based algorithm for the NMPC was implemented in the Matlab environment. Simulink is used for closed-loop simulations. All computational results (excluding the initializations of the problems) were obtained using an Intel i-3-2100 machine (CPU 3.10 GHz 4 GB RAM). The tuning parameters for the DSM-based and the PMP-based algorithms are summarized in Table 5.2.

**Table 5.2.** Tuning parameters for the PMP-based and DSM-based algorithms.

| DSM-based Algorithm (with Ipopt)                 | PMP-based Algorithm  |
|--|--|
| control sampling time = 15 s                     | control sampling time = 15 s   |
| measurement sampling time = 30 s                 | measurement sampling time = 30 s                                     |
| measurement delay = 5 s                          | measurement delay = 5 s  |
| $N = 100$ ; ipopt.max_iter = 100;                | $\alpha = 0.05$ ; $\beta = 0.1$ ; $\varepsilon = 0.01$ ; $N = 100$ ; |
| ipopt.tol = 1e-4; ipopt.mu_init = 1e-6           | $K=20$ ; iter_max = 15   |
| $\Delta T_{max} = (0.35 \text{ K})/(15\text{s})$ | $\Delta T_{max} = (0.35 \text{ K})/(15\text{s})$                     |

As discussed by Remark 5.2, the input profiles computed off-line using the nominal model are used as initial guesses at the beginning of the batch for both algorithms. Later, the optimal inputs computed at a given iteration are used to generate by extrapolation the initial guesses for the next optimization.

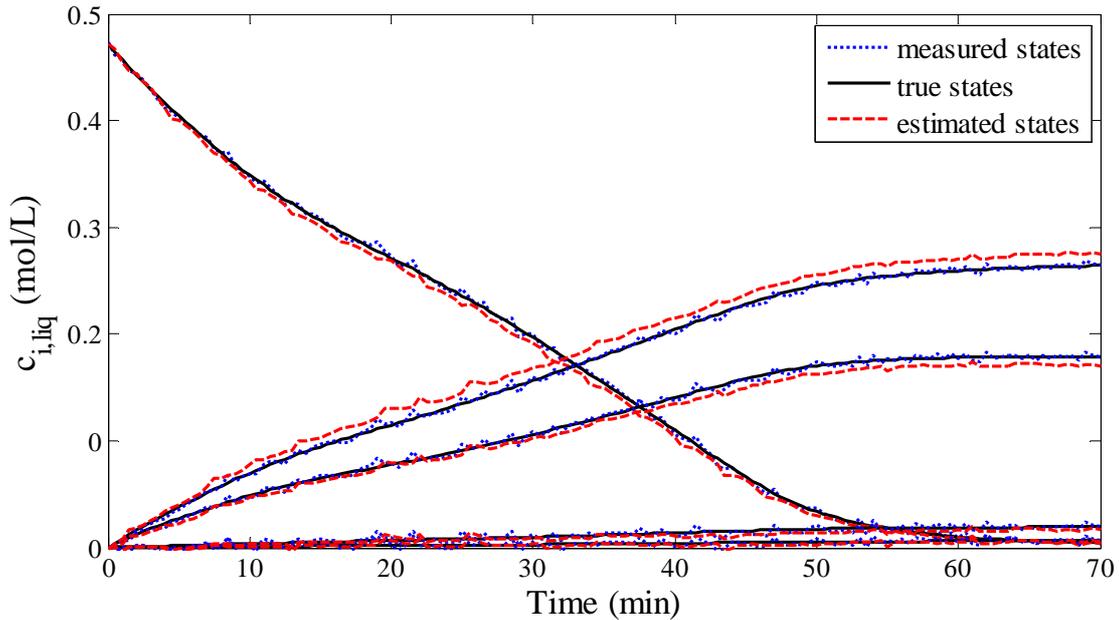
All measurements are corrupted with white noise. Because the sampling times for the measurements and the controller are not the same, and in addition there is some measurement

delay, an observer is designed to estimate the concentrations in the liquid phase using the model equations and a linear update term:

$$\hat{x}_k = \int_{t_{k-1}}^{t_k} f(\hat{x}, u) dt + d_{k-1} \quad (5.4)$$

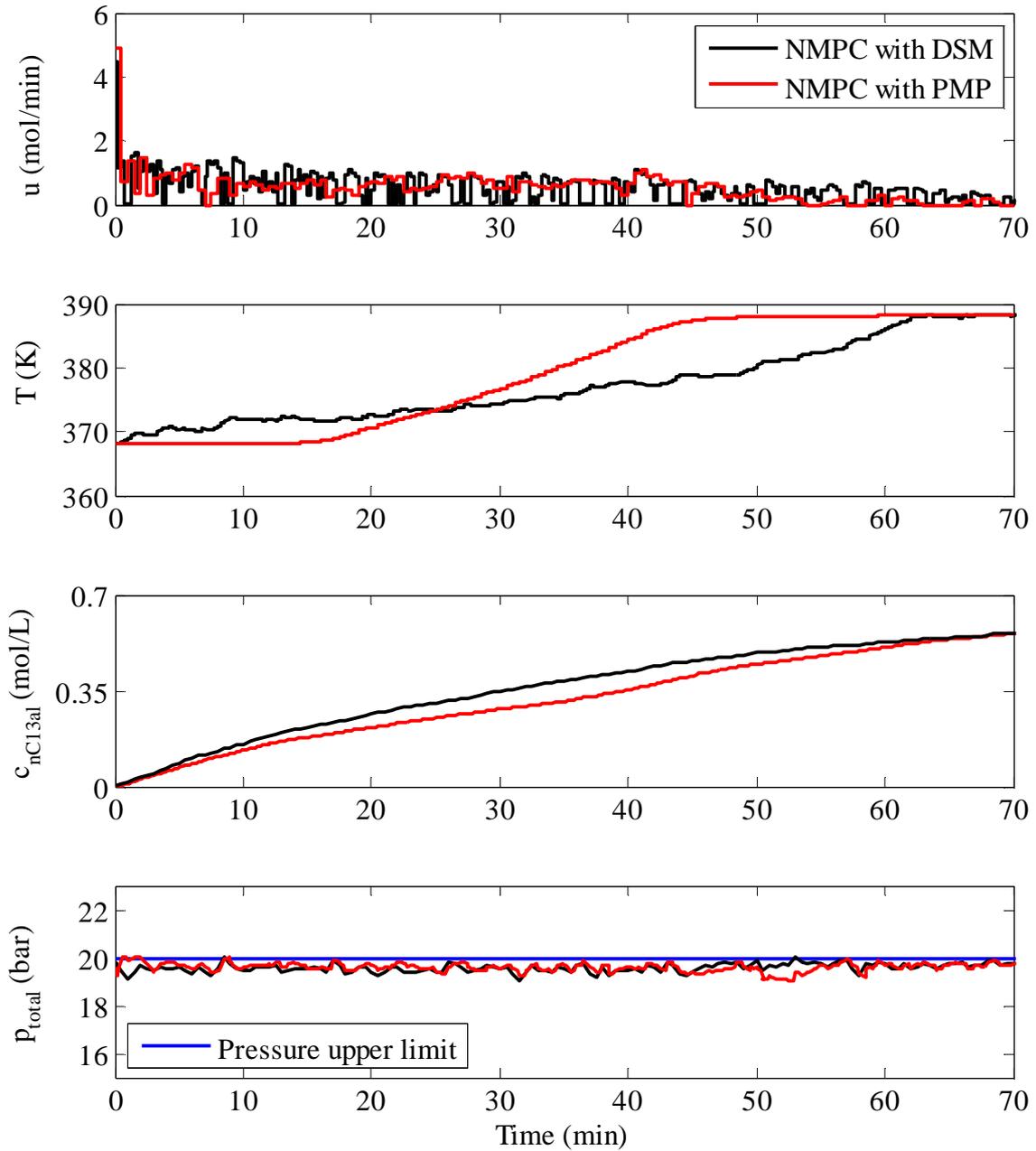
$$d_k = L(x_k - \hat{x}_k)$$

where  $d_k$  is the linear update term,  $\hat{x}_k = \hat{c}_{liq,k}$ ,  $x_k$  are the measured states,  $\hat{x}_k$  are the estimated states and  $L = diag[0.75 \ 0.75 \ 0.65 \ 0.75 \ 0.75 \ 0.75 \ 0.75]$  is the observer gain matrix. All concentrations in the liquid phase are assumed to be measured every 30 sec (+ 5 sec delay) using an on-line FTIR, and the pressure in the gas phase is assumed to be measured every second with no delay. The NMPC algorithm takes into account the estimated states at each sampling time as the initial conditions of the optimal control problem and the linear term  $d_k$  is updated as soon as the new measurements are obtained. The performance of the observer is given in Fig. 5.2.

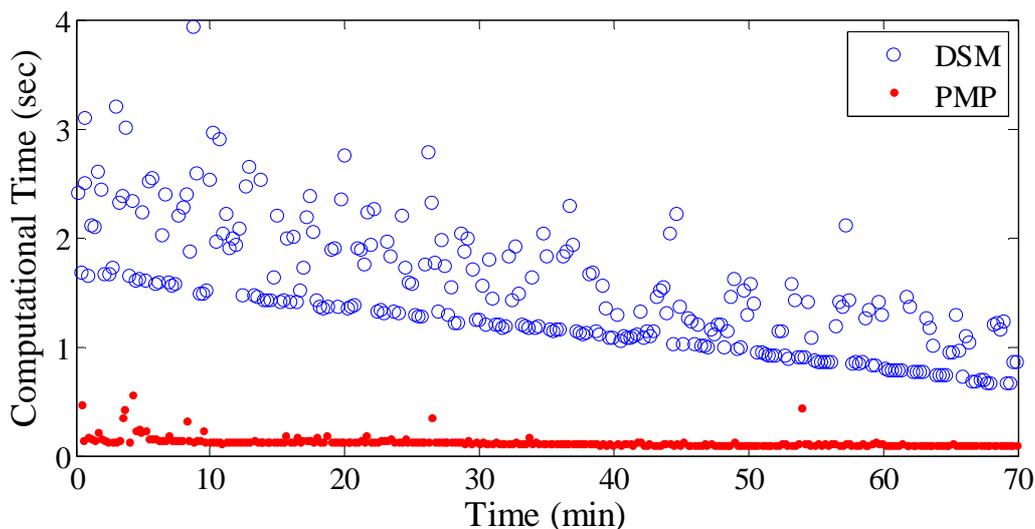


**Figure 5.2.** Performance of the observer for a single batch.

The performance of the DSM-based and PMP-based algorithms for the same disturbance realizations within a batch are compared in Fig. 5.3. The corresponding computational times for the individual iterations are given in Fig. 5.4. Slight variations between the true and estimated concentrations do not affect the feasibility of the closed-loop operation. Nevertheless, performance increases with better state estimation.



**Figure 5.3.** NMPC profiles with DSM-based and PMP-based algorithms for a particular batch and fixed final time.

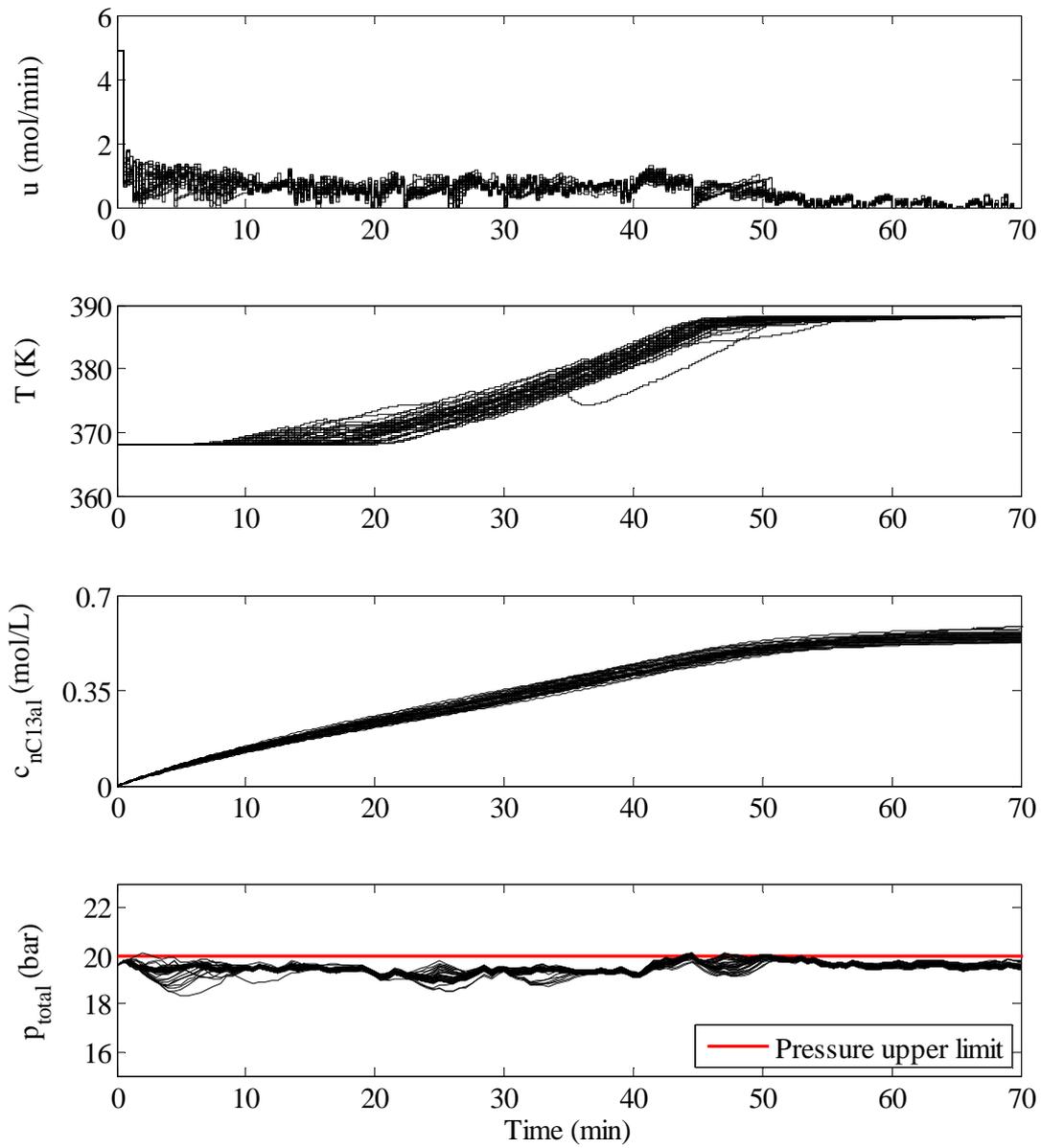


**Figure 5.4.** Computational times with DSM-based and PMP-based algorithms.

Fig. 5.3 shows that, with both methods, very similar closed-loop performance in terms of optimal cost can be achieved in the presence of parametric plant-model mismatch. In addition, the upper pressure limit is never violated thanks to the backoff term, and the rate constraint on temperature is satisfied at each NMPC iteration. Although the closed-loop input trajectories are slightly different, the optimal costs are very similar. Finally, as seen in Fig. 5.4, PMP-based NMPC is much faster than DSM-based NMPC, especially at the beginning of the batch.

Almost 70% of the CPU time required for the PMP-based method is used for integration of the states and co-states. Hence, CPU time does not decrease significantly with PMP as the horizon shrinks. Faster performance may be obtained using fast integration algorithms or discretization methods. However, the speed and the performance of the PMP-based solution is still an open issue for large-scale systems that require high computational time for integration.

In order to test the robustness of the PMP-based NMPC, simulations were performed for 40 uncertainty realizations (Fig. 5.5). PMP-based NMPC is able to sustain feasible operation in all these batches in the presence of uncertainty, with a mean final nC13al concentration of 0.554 mol/L. This indicates that, under closed-loop operation, nearly 9% increase in the final amount of the desired product can be obtained compared to the infeasible (because of pressure violation) operation in Fig. 5.1.



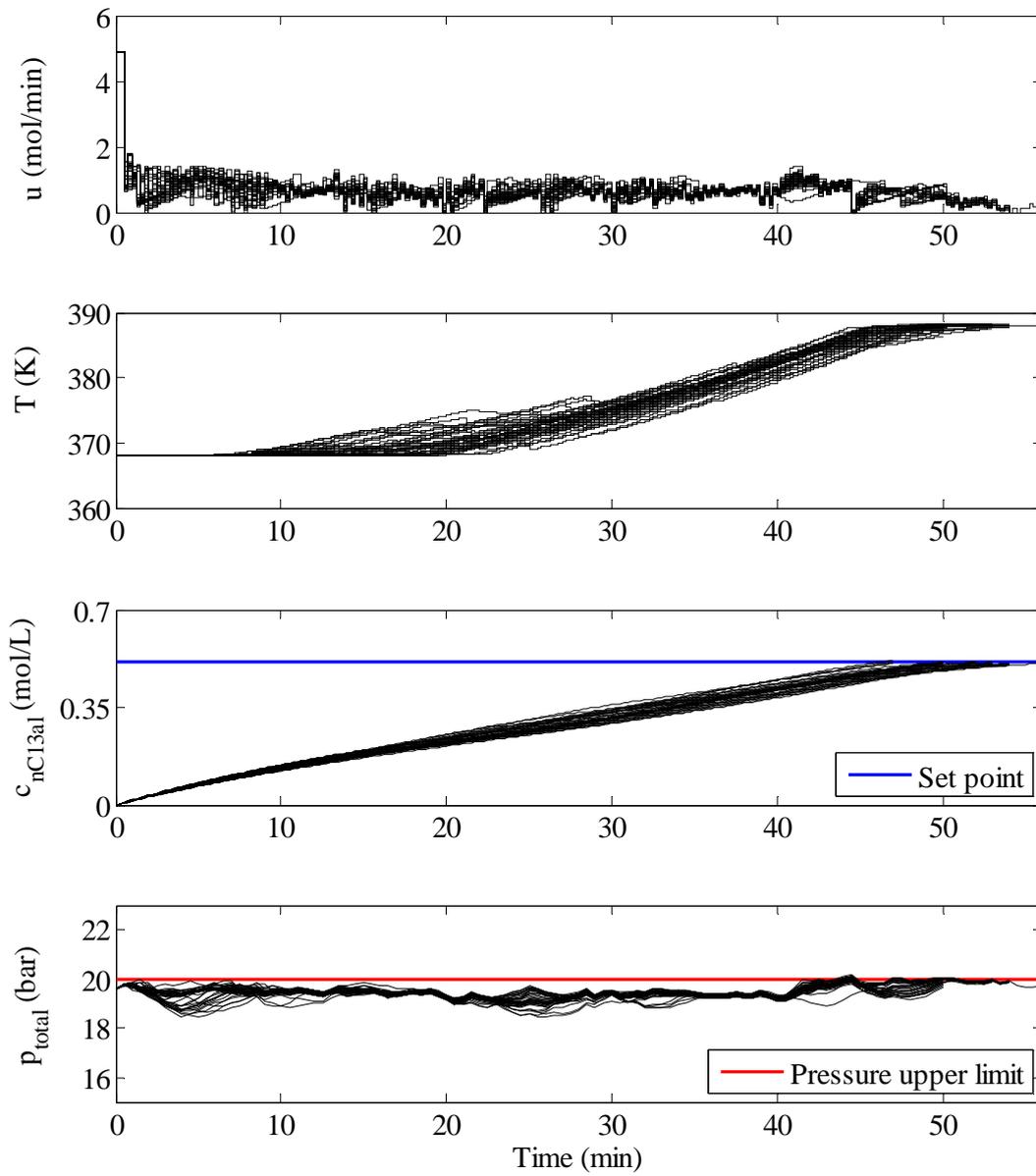
**Figure 5.5.** NMPC profiles with PMP for 40 uncertainty realizations and fixed final time.

## 5.2.4 NMPC for batch time minimization

Furthermore, to check the effect of closed-loop operation on batch time reduction, NMPC problem is reformulated such that the open-loop optimal concentration of tridecanal ( $c_{nc13al}(t_f) = 0.51$  mol/L, Fig. 5.1.b) is given as a set-point to the controller, while the final time  $t_f$  is let free. The corresponding closed-loop results for 40 different batches are given in Fig. 5.6, which shows that, with PMP-based NMPC, the overall batch time can be reduced from 70 to 51.36 min, corresponding to a 26.5% reduction (Table 5.3).

**Table 5.3.** Performance of PMP-based NMPC for free final time.

| Method   | mean $t_f$<br>(for $c_{nc13al}(t_f) = 0.51$ mol/L) | median $t_f$ | st. dev. $t_f$ |
|--|--|--------------|----------------|
| Open-loop nominal optimization<br>(infeasible) | 70 min   | 70 min       | —              |
| NMPC   | 51.36 min  | 51 min       | 2.28 min       |



**Figure 5.6.** NMPC profiles with PMP for 40 uncertainty realizations and batch time minimization.

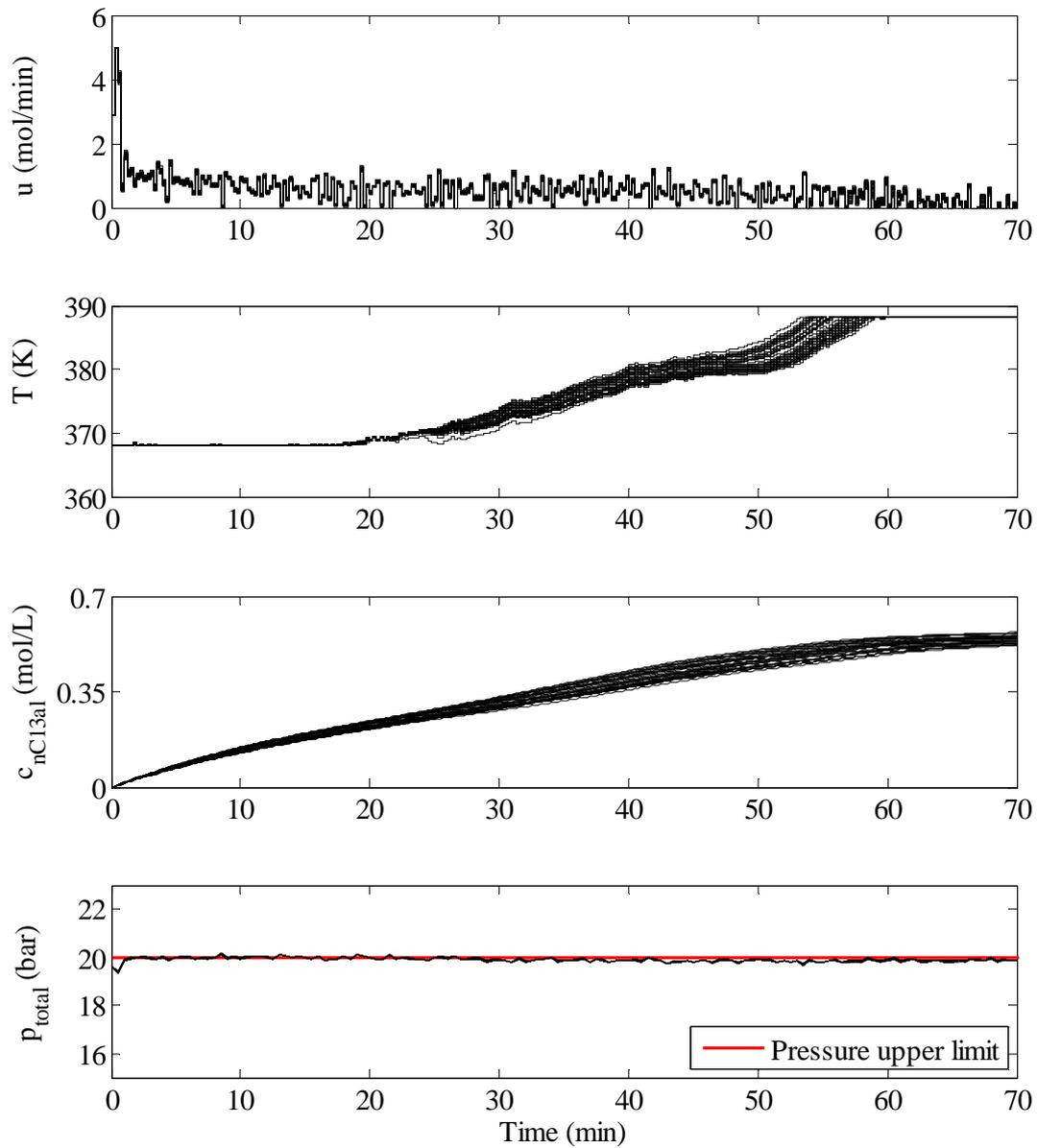
## 5.2.5 NMPC with constraint tracking

Figs. 5.5 and 5.6 illustrate that the pressure is very close to the upper limit of 20 bar in all 80 batches. The optimal solution computed off-line with the nominal model also suggests that this pressure constraint is active throughout the operation (Aydin et al., 2017a). Accordingly, a constraint-tracking framework can be suggested to further reduce the computational complexity.

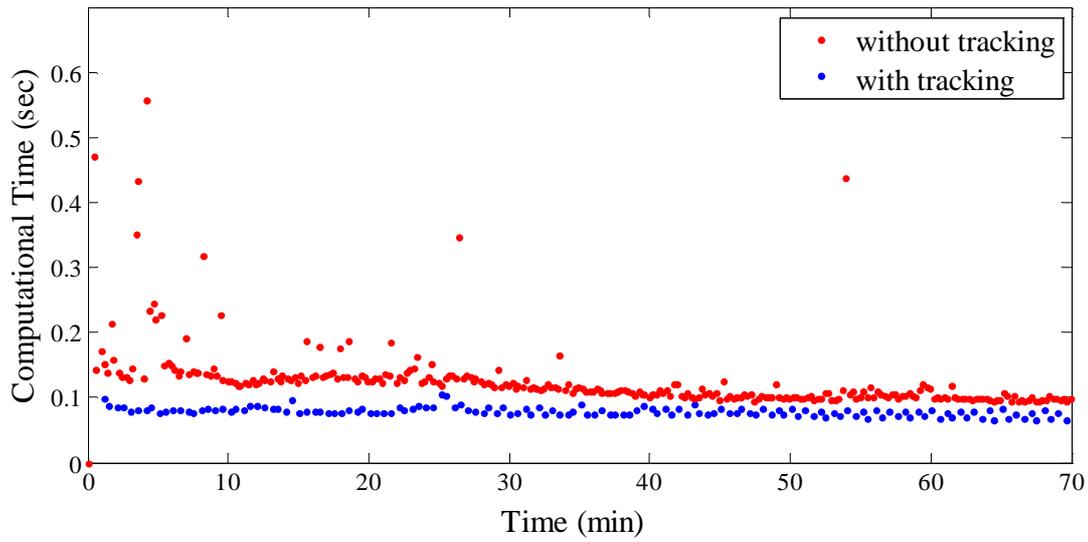
The syngas feedrate  $u(t)$  is adjusted to keep the pressure at 20 bar, while the temperature  $T(t)$  is used to maximize the final concentration of nC13al. The control can be done implicitly with the model, that is,  $u(t)$  is computed to keep  $P_{total}(t)$  constant at 20 bar or, equivalently,  $\dot{p}_{total}(t) = \dot{p}_1(t) + \dot{p}_2(t) = 0$ . From Eq. (3), this gives

$$u(t) = V_{liq} \left( j_1^{GL}(t) + j_2^{GL}(t) \right), \quad (5.5)$$

which keeps the total pressure constant. This way,  $u(t)$  can be removed from the set of decision variables in Eq. 5.3. However,  $p_{total}(t)$  has to be kept constant at 20-bar, which is done via PID control. As seen in Fig. 5.7, the controller is able to keep the pressure very close to the desired value. In addition, because the pressure limit is enforced by feedback control, a coarser input discretization can be used for the temperature ( $N=50$  instead of  $N=100$ ). The performance of NMPC with constraint tracking is shown in Fig. 5.7 for 40 different uncertainty realizations, with the corresponding computational times given in Fig. 5.8. This optimization scheme results in feasible operation, with a mean final nC13al concentration of almost 0.531 mol/L and reduced computational effort. Table 5.4 compares the performance of NMPC without and with constraint tracking. The introduction of constraint tracking reduces performance only by 2%.



**Figure 5.7.** NMPC with constraint tracking for 40 different uncertainty realizations and fixed final time.



**Figure 5.8.** Computational times using PMP-based NMPC without and with constraint tracking.

**Table 5.4.** Performance of PMP-based NMPC without and with constraint tracking for fixed final time.

| Method   | mean $c_{nc13al}(t_f)$ | median $c_{nc13al}(t_f)$ | st. dev. $c_{nc13al}(t_f)$ |
|--|------------------------|--------------------------|----------------------------|
| Open-loop nominal optimization ( <b>infeasible</b> ) | 0.51 mol/L             | 0.511 mol/L              | 0.0126                     |
| NMPC without tracking                                | 0.554 mol/L            | 0.55 mol/L               | 0.010                      |
| NMPC with tracking                                   | 0.543 mol/L            | 0.539 mol/L              | 0.014                      |

### 5.3 Summary

This chapter has proposed to extend the indirect PMP-based solution scheme for the shrinking-horizon NMPC (sh-NMPC) of semi-batch processes. The application of sh-NMPC to a two-phase semi-batch Hydroformylation reactor under uncertainty is illustrated. A time-varying backoff approach is used to deal with parametric uncertainties. The pressure path constraint is indirectly adjoined into the Hamiltonian function and activated at each infeasible iteration. Simulation results show that the computational burden stemming from the matrix factorization in large-horizon problems is successfully reduced by the interplay of states, co-states and Lagrange multipliers. Accordingly, PMP-based NMPC has a computational advantage over direct simultaneous method-based NMPC, especially at the beginning of the batch. In addition, finer input discretization with the PMP-based NMPC can be expected to increase the closed-loop performance but not the computational time significantly, whereas cubic increase in the computational time is anticipated with the DSM-based NMPC. Furthermore, the PMP-based solution algorithm can be extended to active constraint tracking. For example, for the semi-batch hydroformylation reactor, further reduction in computational time was obtained via tracking of the active pressure constraint. Note that the computational speed of the PMP-based algorithms can be further increased by discretizing the state and co-state equations instead of relying on integration. Yet, it is still an open question how PMP-based NMPC performs for large-scale problems, where integration requires more effort.



*Simple can be harder than complex: You have to work hard to get your thinking clean to make it simple. But it's worth it in the end because once you get there, you can move mountains.*

---

Steve Jobs (1955 – 2011)

# 6 NONLINEAR MODEL PREDICTIVE CONTROL USING PARSIMONIOUS INPUT PARAMETERIZATION

The optimal inputs of batch and semi-batch processes computed via dynamic optimization can be characterized using different arcs. An optimal arc can be either on an input bound ( $u_{min}, u_{max}$ ), on a path constraint ( $u_{path}$ ), or inside the feasible region as a sensitivity-seeking arc ( $u_{sens}$ ). It is usually difficult and burdensome to compute the fine shapes of sensitivity-seeking arcs accurately due to their lack of sensitivity. To deal with this issue, simplified solution models can be introduced, in which the inputs and most importantly the sensitivity-seeking arcs  $u_{sens}$  are parameterized parsimoniously using switching times and low-order polynomials. This way, the number of decision variables and the complexity of the optimization problem can be reduced significantly. In other words, instead of a full NLP, a parsimoniously parameterized NLP can be solved, which reduces the load of the corresponding non-convex dynamic optimization problem. Hence, the required CPU time is expected to decrease significantly, without affecting much the optimal cost (Welz et al., 2005, 2006; Welz et al., 2008). This is also the motivation of Chapter 4, in which such a parsimonious scheme is combined with an indirect method to be applied to dynamic optimization.

The main contribution of this chapter is to detail the application of the parsimonious parameterization models in the context of sh-NMPC. Parsimonious sh-NMPC approximates the fine shapes of the optimal inputs at each sampling instant. As stated earlier, the optimization is performed at each sampling instant for the full time horizon but only the first part of the optimal inputs is applied to the process. Nevertheless, the optimal closed-loop behavior might be captured accurately. In addition, since the full time horizon is taken into account, the loss in ability to influence the batch outcome (which is one of the most important challenges in batch processing) can be prevented, while still having a significant reduction in CPU time.

## 6.1 The Parsimonious Shrinking-Horizon NMPC

Direct and indirect methods exist in the literature to solve the problem given by Eq. 5.1 (Srinivasan et al., 2003b; Biegler, 2007, 2010; Aydin et al., 2017a). The input profiles are typically discretized as  $u(t) = \mathcal{U}(U)$ , where  $U$  is a  $(n_u \times N)$  input matrix that contains  $N$  discrete input values for the  $n_u$  inputs. In an earlier publication, it is observed that at least  $N=100$  is required to obtain reliable offline optimal profiles for the batch and semi-batch problems (Aydin et al., 2017a).

Accordingly, the standard sh-NMPC algorithm can be formulated as follows:

---

### Standard sh-NMPC Algorithm

---

Set  $k = 0$  and specify  $x_0$ .

while  $t_k \in [t_0, t_f]$  **do**

1. Measure/estimate  $x_k$  and assign  $x(t_k) := x_k$
2. Solve Problem (5.1) for the decision variables  $U(n_u \times N)$
3. Inject  $u[t_k, t_k + \delta] = \mathcal{U}(U(n_u \times 1))$  to the plant and wait for  $t_{k+1} = t_k + \delta$
4. Set  $k := k + 1$

**end do**

---

Numerical optimization schemes often require fine input discretization levels to be able to compute accurate solutions. In addition, as discussed earlier, the complexity of the

optimization problem increases cubically with the time horizon due to the matrix factorizations required in the solution steps. Furthermore, it is sometimes necessary to include additional terminal constraints to the NMPC problems for continuous processes to guarantee closed-loop stability (Mayne et al., 2000; Findeisen et al., 2007). This may increase the size of the prediction horizon significantly. Consequently, the CPU time required to solve the sh-NMPC problem may turn out to be large, especially at the beginning of the batch when the time horizon is the largest. On the other hand, the expensive matrix factorizations can be avoided by decreasing the number of decision variables using parsimonious parameterizations, while keeping the full length of the time horizon. Afterwards, the proposed indirect algorithms can be applied to the problem, which decreases the computational complexity significantly.

The strategy behind building the parsimonious solution models starts with computing the offline solution to Problem 2.1. It is typically assumed that the uncertainty does not change the types and sequence of optimal arcs in closed-loop operation, which is reasonable for batch processes (Kadam et al., 2007; Srinivasan and Bonvin, 2007). Given the optimal solution structure, the input bounds ( $u_{min}, u_{max}$ ) and the sensitivity-seeking arcs ( $u_{sens}$ ) can be approximated using polynomial profiles and adjustable switching times between arcs (Welz et al., 2005, 2006; Schlegel and Marquardt, 2006b; Welz et al., 2008). These solution models have also been used recently in the context of the NCO tracking to design multivariable controllers by pairing the inputs (MVs) with the active constraints and appropriate sensitivities (CVs) using physical insight, relative gain analysis and sensitivity analysis (Visser et al., 2000; Srinivasan et al., 2003a; Bonvin, 2006; Srinivasan et al., 2008; Welz et al., 2008; Ebrahim et al., 2016).

In this chapter, it is proposed to use the concept of parsimonious solution models for solving the sh-NMPC Problem (1). As discussed in Chapter 4, given the optimal solution structure (the types and sequence of arcs), it is possible to reformulate the optimization problem using a parsimonious input parameterization of the form  $u(t) = U(\pi)$ . For example, a sensitivity-seeking arc can be expressed as a linear arc between the two switching times  $t_1$  and  $t_2$  with the values  $a_1$  and  $a_2$  at  $t_1$  and  $t_2$ , respectively. This results in  $\pi = (t_1, t_2, a_1, a_2)^T$ , with the parsimonious input model:

$$\mathcal{U}(\pi) = \begin{cases} u_{\max} & \text{if } 0 \leq t < t_1; \\ a_1 + \frac{a_2 - a_1}{t_2 - t_1}(t - t_1) & \text{if } t_1 \leq t < t_2; \\ u_{\min} & \text{if } t_2 \leq t < t_f \end{cases} \quad (6.1)$$

The reformulated optimal control problem to be solved online at each sampling instant reads:

$$\begin{aligned} \min_{\pi} \quad & \tilde{J} = \phi(x(t_f), \bar{\theta}) \\ \text{s.t.} \quad & \dot{x}(t) = \tilde{F}(x(t), \mathcal{U}(\pi), \bar{\theta}), \quad x(t_k) = x_k \\ & \tilde{S}(x(t), \mathcal{U}(\pi), \bar{\theta}) \leq 0, \\ & T(x(t_f), \bar{\theta}) \leq 0, \quad t \in [t_k, t_f] \end{aligned} \quad (6.2)$$

where  $\pi$  is the new vector of decision variables,  $\mathcal{U}(\pi)$  the input vector given by Eq. 6.1,  $\tilde{F}$  represents the system of equations expressed in terms of  $\mathcal{U}(\pi)$  instead of  $u(t)$ ,  $\tilde{J}$  is the scalar performance index for Problem (3),  $\bar{\theta}$  is the vector of estimated parameters,  $\tilde{S}$  the vector of inequality path constraints expressed in terms of  $\mathcal{U}(\pi)$  instead of  $u(t)$ . Finally, the dynamic optimization problem given by Eq. 6.2 can be solved using the method given in Chapter 4.

Accordingly, the parsimonious sh-NMPC algorithm can be formulated as follows:

---

### Parsimonious sh-NMPC Algorithm

---

- I. Solve Problem given in Eq. 2.1 numerically offline for the nominal values of the parameters  $\bar{\theta} = \theta_0$ .
- II. Build a parsimonious solution model by parameterizing the inputs with respect to the switching times and low-order polynomials to obtain  $\mathcal{U}(\pi)$ .
- III. Set  $k = 0$  and specify  $x_0$ .

while  $t_k \in [t_0, t_f]$  **do**

1. Measure/estimate  $x_k$  and assign  $x(t_k) := x_k$
2. Solve Problem 6.2 using indirect methods for the decision variables  $\pi$
3. Inject  $u[t_k, t_k + \delta] = \mathcal{U}[t_k, t_k + \delta](\pi)$  to the plant and wait for  $t_{k+1} = t_k + \delta$
4. set  $k := k+1$

**end do**

---

**Remark 6.1.** The polynomials used in the parsimonious models are often very problem specific, with piecewise-constant or piecewise-linear functions often resulting in accurate approximations to the sensitivity-seeking arcs. Problem-specific information regarding the parsimonious parameterization will be detailed in the case studies of next section.

**Remark 6.2.** As discussed in earlier chapters, depending on their relative degree, the inputs that activate the path constraints  $u_{path}$  can sometimes be computed online using the model equations, that is, without any optimization (Srinivasan et al., 2003b; Aydin et al., 2017b). Another alternative is to track the corresponding path constraints with the help of feedback controllers using  $u_{path}$  as manipulated variables (Srinivasan and Bonvin, 2007).

## 6.2 Case Studies

To illustrate the application of parsimonious sh-NMPC to batch and semi-batch processes, two case studies are selected. The first example is the batch binary distillation column with terminal purity constraints and the second one is the semi-batch hydroformylation reactor with path constraints. In order to test the performance and robustness of the controllers, closed-loop simulations are performed under parametric uncertainties. The standard sh-NMPC cases are solved using a direct simultaneous method. The CasADI toolbox and Matlab Simulink are used for both sh-NMPC methods (Wächter and Biegler, 2006; Andersson and Diehl, 2012).

### 6.2.1 Batch binary distillation with terminal purity constraints under uncertainty

Recall the batch distillation column with three equilibrium plates, in which components A and B (more volatile) are separated from each other. The operational goal is the maximization of the amount B in the distillate, while satisfying two terminal constraints. The only path constraint is on the input variable, namely, the reflux ratio. Accordingly, the optimal control problem to be solved online in the context of sh-NMPC can be written as follows:

$$\begin{aligned}
& \max_{r(t), t_f} \quad J = D(t_f) \\
& \text{s.t.} \quad \text{dynamic model eqns. (from Eq. 3.5);} \\
& \quad D(t_k) = D_k; B(t_k) = B_k; \quad n_B(t_k) = n_{B_k}; n_1(t_k) = n_{1k}; \\
& \quad n_2(t_k) = n_{2k}; n_3(t_k) = n_{3k}; n_D(t_k) = n_{D_k} \\
& \quad y_m = \frac{\alpha x_m}{1 + (\alpha - 1)x_m}; \quad m = B, 1, \dots, 3 \\
& \quad x_D(t_f) = n_D(t_f)/D(t_f) \geq 0.8 \\
& \quad x_B(t_f) = n_B(t_f)/B(t_f) \leq 0.2 \\
& \quad 3 h \leq t_f \leq 3.25 h \\
& \quad 0 \leq r(t) \leq 1 \quad , \quad t \in [t_k, t_f] \tag{6.3}
\end{aligned}$$

where  $t_k$  is the time at the  $k$ -th iteration,  $B_k$  the charge,  $n_{B_k}$  the moles of B in the charge,  $n_m$  the moles of B in the liquid phase on the  $m$ -th tray,  $y_m$  the mole fraction of B in the vapor phase leaving the  $m$ -th tray,  $n_D$  the moles of B in the distillate tank,  $x_D$  the mole fraction of B in the distillate tank,  $n_B$  the moles of B in the bottoms,  $x_B$  the mole fraction of B in the bottoms,  $\alpha$  the relative volatility,  $MH$  the liquid hold-up on each tray, and  $t_f$  the free final time. Because of the assumption of total condensation, the composition of the refluxed liquid is equal to the vapor composition leaving the upper plate. It is also assumed that all plates are initially charged with the same liquid mixture as the reboiler. The nominal model parameters and the initial conditions are given in Table 6.1.

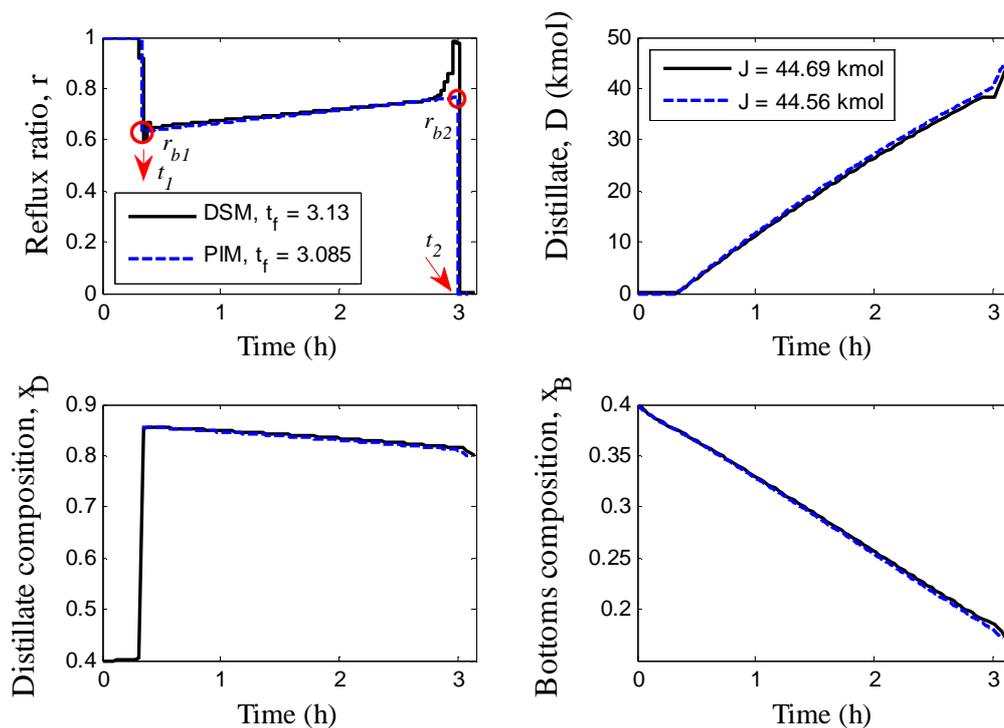
**Table 6.1.** Nominal model parameters and initial conditions for the batch distillation.

| Parameter                                   | Value     |
|---|-----------|
| Vapor flowrate, $V$                         | 50 kmol/h |
| Relative volatility, $\alpha$               | 2.35      |
| Initial charge, $B_0$                       | 115 kmol  |
| Concentration of B in the charge, $x_{B_0}$ | 0.4       |
| Molar hold-up per plate, $M$                | 5 kmol    |

In this batch distillation study, state feedback information is assumed in order to focus only on the computational aspects of the proposed scheme. This is a reasonable assumption considering the advances in state estimation and online spectroscopy (Rao and Rawlings, 2002; Zavala et al., 2008b; Schneider and Georgakis, 2013).

### 6.2.1.1 Nominal Open-loop Optimal Policy

The optimal control problem given by Eq. 2.1 is first solved offline, using the estimated parameters given in Table 6.1. The optimal profiles, computed using a direct simultaneous method and 100 piecewise-constant elements, are given in Fig. 6.1. The optimal cost is 44.69 [kmol].



**Figure 6.1.** Optimal *open-loop* profiles for Problem 6.2.1 with a direct simultaneous method (DSM) and parsimonious input model (PIM).

Fig. 6.1 shows that the solution structure starts with total reflux to increase the purity at the top of the column. Then, a sensitivity-seeking arc represents the best compromise between

producing more distillate and satisfying the required purity. Finally, a no-reflux third arc recovers the high-purity material that is still at the top of the column.

Similar to Chapter 4, analysing the optimal input profile, a parsimonious solution model can be proposed, in which the sensitivity-seeking arc varies linearly between the two switching times  $t_1$  and  $t_2$ . Furthermore, the values of  $r_{sens}$  at the switching times are also considered as the decision variables  $r_{b1}$  and  $r_{b2}$ , because  $r_{sens}$  does not start at 1 and end at 0. Accordingly, the new vector of decision variables for this parsimonious solution model is  $\pi = (t_1, t_2, r_{b1}, r_{b2}, t_f)^T$ , and the parameterized reflux ratio reads:

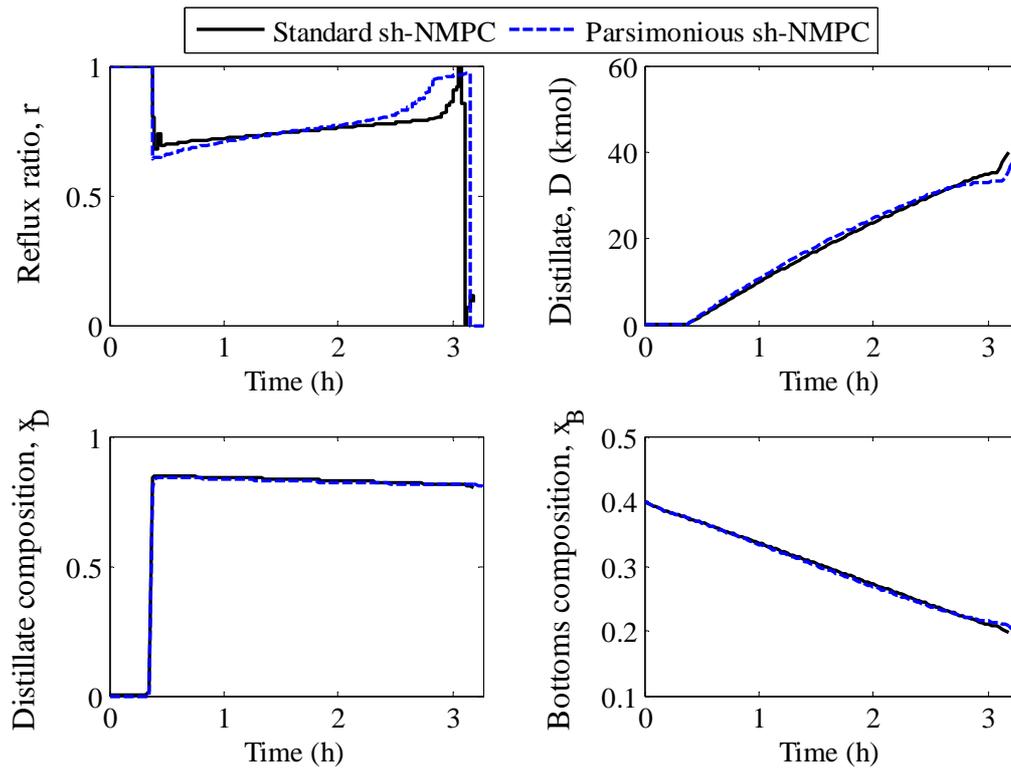
$$r(\pi) = \begin{cases} 1 & \text{if } 0 \leq t < t_1; \\ r_{b1} + \frac{r_{b2} - r_{b1}}{t_2 - t_1}(t - t_1) & \text{if } t_1 \leq t < t_2; \\ 0 & \text{if } t_2 \leq t < t_f \end{cases} \quad (6.4)$$

The optimal profiles obtained with this simple solution model are also shown in Fig. 6.1, with an optimal cost of 44.56 [kmol]. It is observed that the nominal and approximated open-loop optimal profiles are very similar.

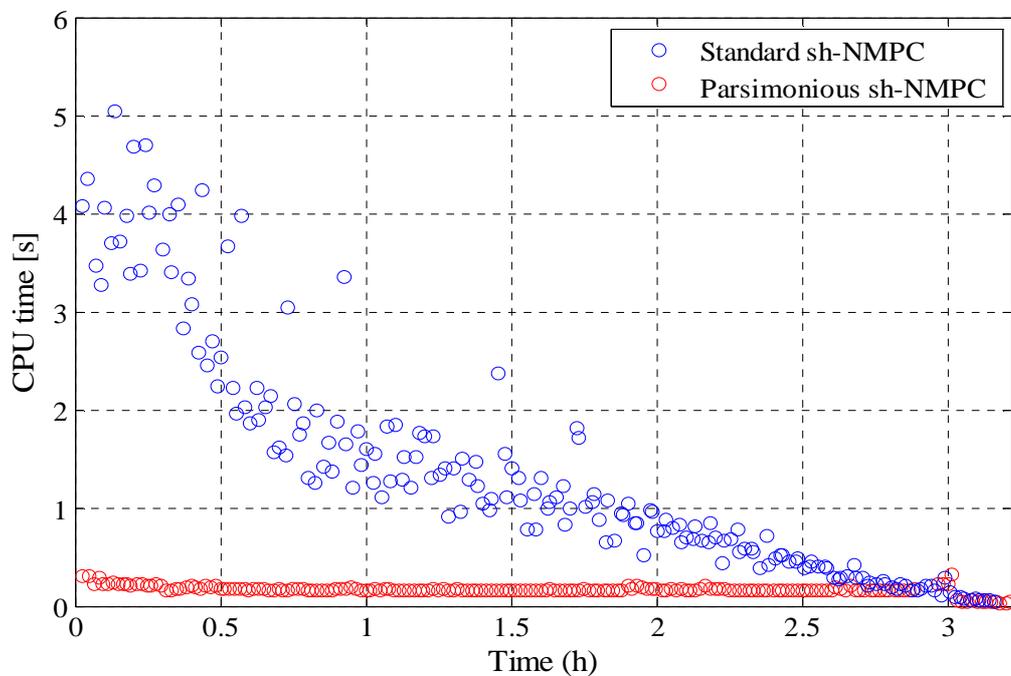
### 6.2.1.2 Closed-loop Simulation with sh-NMPC

In order to check the performances of both sh-NMPC schemes, uncertainty is added to the relative volatility parameter ( $\alpha$ ), which varies randomly from batch to batch between -15% and 0%. In addition, the vapor flowrate ( $V$ ) is randomly perturbed by  $\pm 3$  kmol/h within the batch. The controller sampling time is 1 min.

Firstly, the performances of standard and parsimonious sh-NMPC are compared for one particular batch. The optimal closed-loop behaviours are given in Fig. 6.2, which shows that very similar performance can be achieved with both methods. In other words, the closed-loop optimal behaviour can be approximated accurately by using the parsimonious solution model. Only a batch time difference of 3.5 min is required to have the same distillate amount at the end of the batch.



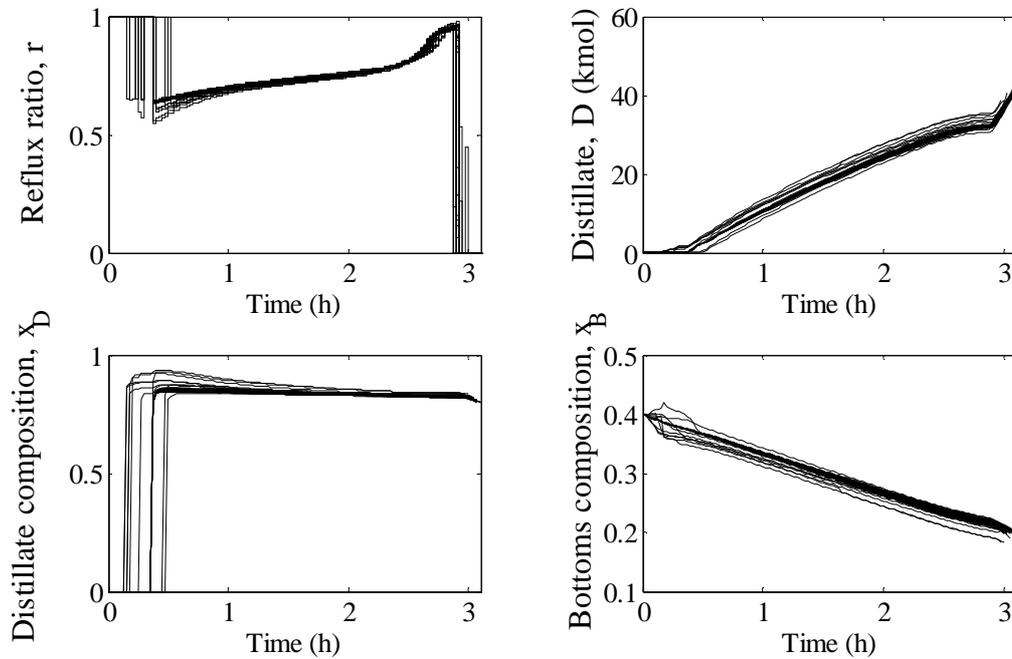
**Figure 6.2.** Optimal *closed-loop* profiles for Problem 6.2.1 with standard and parsimonious sh-NMPC.



**Figure 6.3.** Computational times with standard and parsimonious sh-NMPC for the batch distillation.

On the other hand, it is observed that parsimonious sh-NMPC requires significantly less CPU time and is much faster than standard sh-NMPC, especially at the beginning of the batch (Fig. 6.3).

Finally, to validate the robustness and check the performance of parsimonious sh-NMPC for different uncertainty realizations, 40 closed-loop simulations are performed. In all cases, the simulation is stopped as soon as the desired purity levels are achieved. The results are given in Fig. 6.4. It is observed that parsimonious sh-NMPC results in feasible operation. Furthermore, with CPU-time reduction at each iteration, faster sampling and control can be used, which in turn helps to deal with plant-model mismatch. This way, parsimonious sh-NMPC can outperform standard sh-NMPC in real-life implementation.



**Figure 6.4.** Optimal *closed-loop* profiles for Problem 6.3 with parsimonious sh-NMPC for 40 different batches.

## 6.2.2 Semi-batch reactor for the hydroformylation of 1-dodecene under uncertainty

This section investigates the application of sh-NMPC to a two-phase fed-batch reactor for the hydroformylation of 1-dodecene (nC12en) in the presence of uncertainty. Note that the same problem has been detailed in Eq. 3.6 and 5.3.

A stirred-tank reactor with the dosing of syngas ( $H_2 + CO$ ) is used for the operation. The manipulated variables are the reactor temperature  $T(t)$  and the feedrate of syngas  $u(t)$ . The operational objective is to maximize the amount of n-tridecanal (nC13al) at the end of the batch. Fixed batch time is 70 min. Input bounds and limits on the total pressure in the gas phase represent the path constraints that should be satisfied throughout the operation. Earlier studies showed that plant-model mismatch affects the process conditions and feasibility significantly, and therefore should be taken into account (Kaiser et al., 2016; Aydin et al., 2018b). For detailed information, the reader is referred to Chapter 3.

**Table 6.2.** Nominal parameter values and corresponding variations for the hydroformylation process:  $(k_L a)_i$  varies within batch, while  $k_{i,0}$  and  $\gamma$  vary from batch to batch.

| Parameter   | Nominal Value<br>(Hentschel et al., 2015) | Minimal Value | Maximal Value |
|-------------|---|---------------|---------------|
| $(k_L a)_1$ | 9.57                                      | 8.57          | 10.57         |
| $(k_L a)_2$ | 7.08                                      | 6.08          | 8.08          |
| $k_{1,0}$   | 4.904                                     | 3.8           | 6.0           |
| $k_{2,0}$   | 4.878                                     | 3.78          | 5.98          |
| $k_{3,0}$   | 2.724                                     | 1.72          | 3.72          |
| $k_{4,0}$   | 2.958                                     | 1.8           | 4.0           |
| $k_{5,0}$   | 3.702                                     | 2.6           | 4.8           |
| $k_{6,0}$   | 3.951                                     | 2.8           | 5.0           |
| $\gamma$    | 100 %                                     | 80 %          | 100 %         |

### 6.2.2.1 Nominal Open-loop Optimal Policy

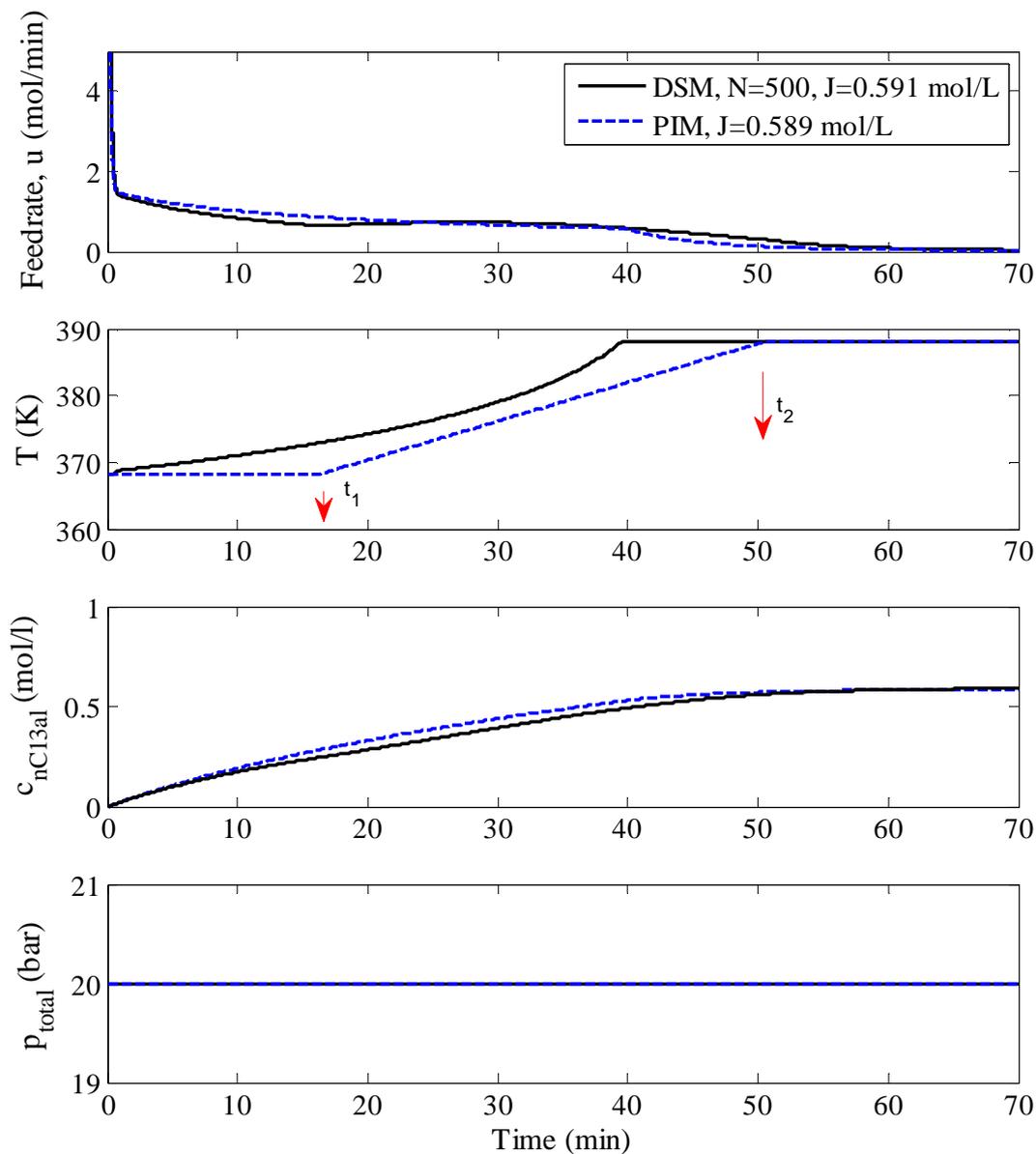
The optimal control problem given in Eq. 3.6 is solved offline for the nominal model parameters using a direct simultaneous method. The input parameterization uses 500 piecewise-constant elements. The optimal profiles are given in Fig. 6.5. It is seen that (i) the optimal solution exhibits a single arc for the feedrate  $u$ , and (ii) the upper pressure limit is always active,  $p_{total}(t) = 20$  bar. Furthermore, the optimal temperature profile starts at the lower limit ( $T_{min}$ ) to favor the desired reactions, then follows a sensitivity-seeking arc ( $T_{sens}$ ) and ends up at the upper limit ( $T_{max}$ ) to suppress the undesired reactions. The optimal cost with the fully parameterized NLP is 0.591 [mol/L].

A parsimonious solution model can be introduced to reduce the computational load of the online control problem. In this model, the temperature  $T(t)$  is parameterized using the switching times  $t_1$  and  $t_2$  and a linear profile between the lower and upper bounds between these switching times. On the other hand, the feedrate input  $u$  is set to keep the pressure at its upper limit.

As discussed in the previous chapters, Eq. 3.6 shows that the pressure constraint has relative degree of 1. In other words, the input becomes explicit after the first time derivative of this constraint. Hence, the value of  $u(t)$  that keeps the total pressure active can be computed from  $\dot{p}_{total}(t) = \dot{p}_1(t) + \dot{p}_2(t) = 0$ , which gives  $u(t) = V_{liq} (j_1^{GL}(t) + j_2^{GL}(t))$ . As a result, the vector of decision variables for the reformulated problem is  $\pi = (t_1, t_2)^T$ . Accordingly, the parsimonious input model reads:

$$u(\pi) = \begin{cases} T(\pi) = \begin{cases} 368.15 & \text{if } 0 \leq t < t_1; \\ 368.15 + \frac{20}{t_2 - t_1}(t - t_1) & \text{if } t_1 \leq t < t_2; \\ 388.15 & \text{if } t_2 \leq t < t_f \end{cases} \\ u(t) = V_{liq} (j_1^{GL}(t) + j_2^{GL}(t)) & \forall t \in [0, t_f] \end{cases} \quad (6.5)$$

The optimal profiles obtained via the simplified solution model are also shown in Fig. 6.5. The optimal cost obtained using the parsimonious parameterization is 0.589 [mol/L].



**Figure 6.5.** Optimal *open-loop* profiles for Problem 6.2.2 with a direct simultaneous method (DSM) and parsimonious input model (PIM).

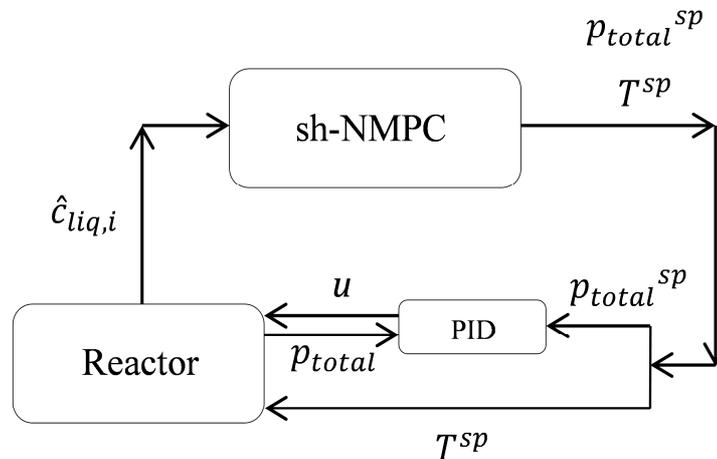
It is observed that, although the input profiles computed with both methods differ to some extent, very similar cost values can be achieved (Binette et al., 2016; Aydin et al., 2017a). On the other hand, note that the number of decision variables is reduced from 500 to 2 through the use of the parsimonious solution model, which proves the significant reduction in online computational effort.

### 6.2.2.2 Closed-loop Simulations with sh-NMPC

The performance and robustness of both sh-NMPC schemes for the hydroformylation reactor are compared under the parametric variations given in Table 6.2. The controller sampling time is 30 s. It is assumed that the concentrations of each component can be measured via online spectroscopy every 30 s, with a 5 s measurement delay. The total pressure in the gas phase is assumed to be measured every second with no delay. Moreover, all measurements are corrupted with Gaussian white noise. The linear observer, which is given in Eq. 5.4, is used to estimate the concentrations of all species in the liquid phase at each sampling instant.

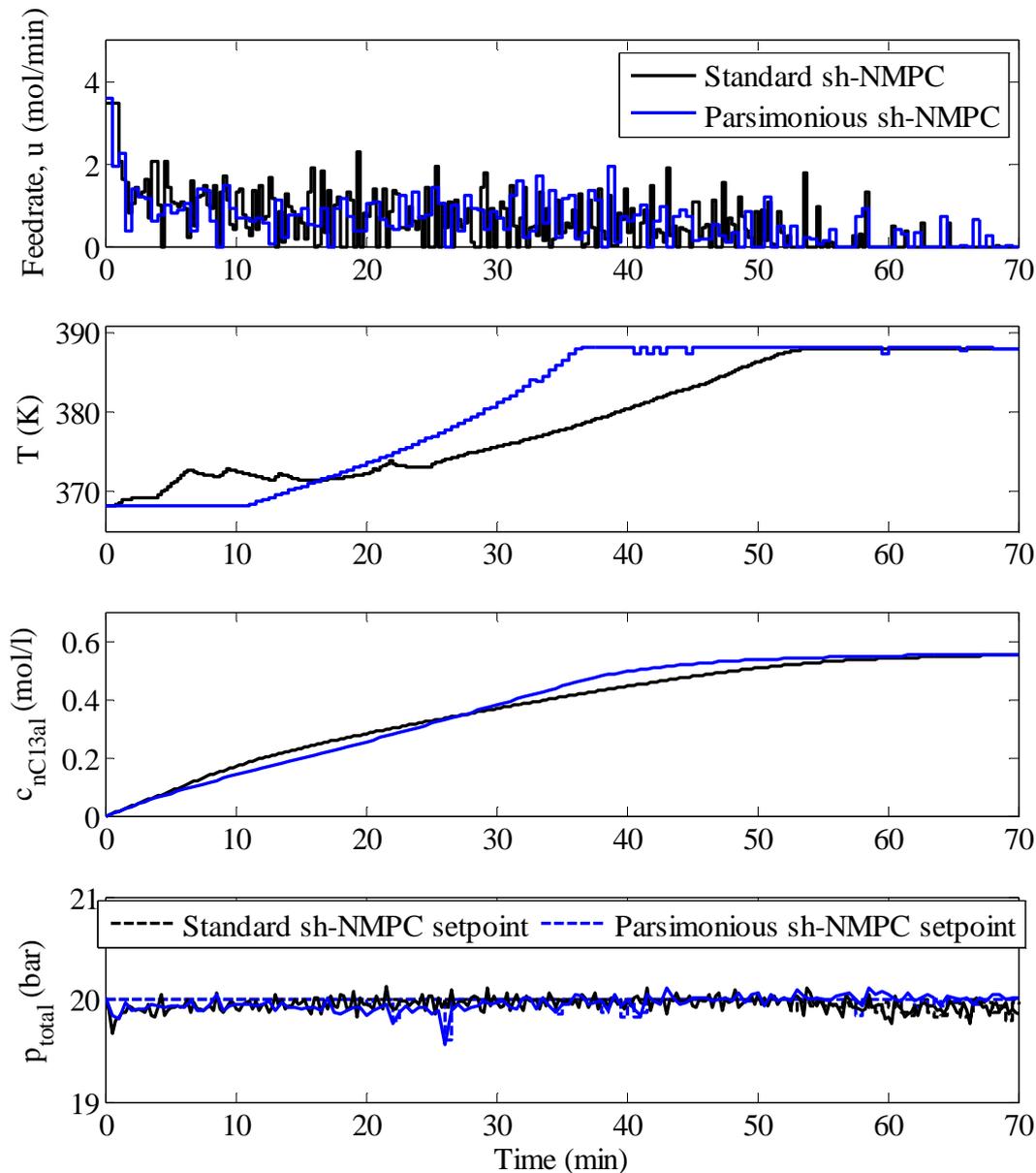
**Remark 6.3.** The  $\Delta T_{max}$  constraint can be enforced directly with standard sh-NMPC. However, with parsimonious sh-NMPC, this constraint can be included in the optimization via a constraint on the two switching times, namely,  $t_2 - t_1 > 15$  min.

Unlike Chapter 5, in order to reject the effects of parametric uncertainty on the pressure path constraint, a hierarchical control structure is recommended instead of a back-off approach. In this scheme, the total pressure in the gas phase ( $p_{total}$ ) computed via optimization is sent as setpoint to a low-level PID controller that tracks the pressure by adjusting the feedrate of syngas  $u(t)$ . This way, the fast perturbations on the path constraint are rejected via the low-level controller, while the slow perturbations on the cost are reduced through the upper level sh-NMPC. Note that, similar to the nominal parsimonious optimization case, implicit control of this path constraint via the system equations is also possible as discussed in Chapter 5. The suggested hierarchical structure is illustrated in Fig. 6.6, with the thermostat and the state estimator hidden inside the reactor.



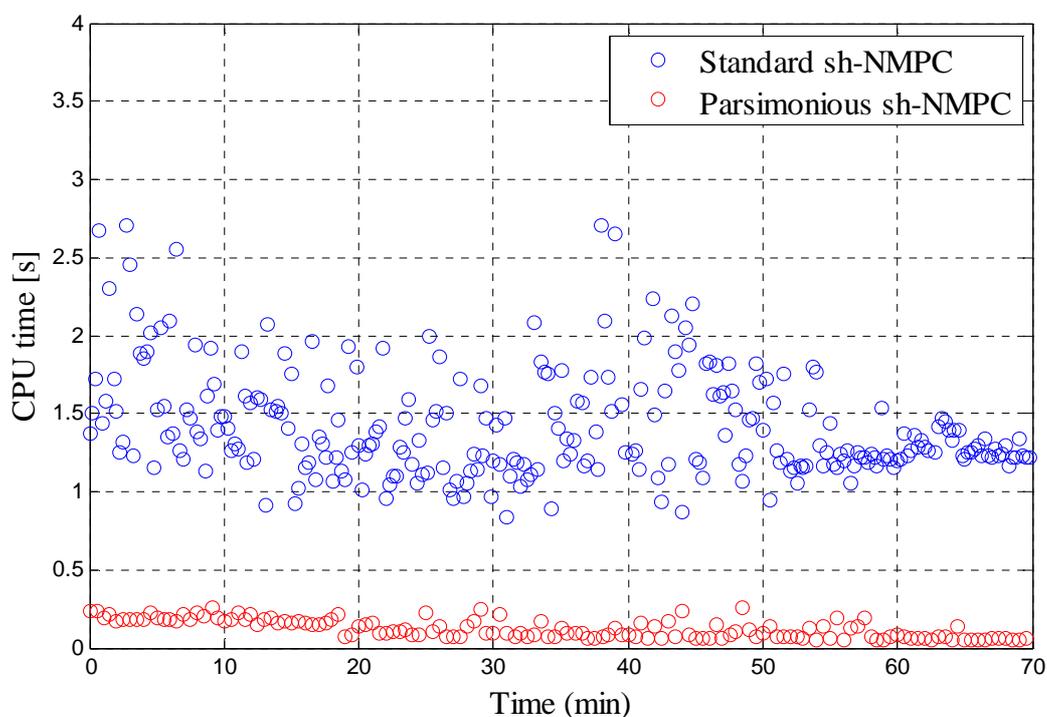
**Figure 6.6.** Hierarchical sh-NMPC structure for the semi-batch hydroformylation reactor.

To be able to compare the performance of both sh-NMPC methods, closed-loop simulations are performed for a particular batch, in which the parametric variations are the same throughout the operation. Also, to have a fair comparison in terms of CPU times, the inputs of standard sh-NMPC are parameterized with 50 piecewise-constant elements. Here, note that feasibility is achieved via hierarchical control. However, finer input discretization is usually necessary to have reliable solutions for single stage problems, as discussed in Chapter 3. The results are reported in Fig. 6.7.

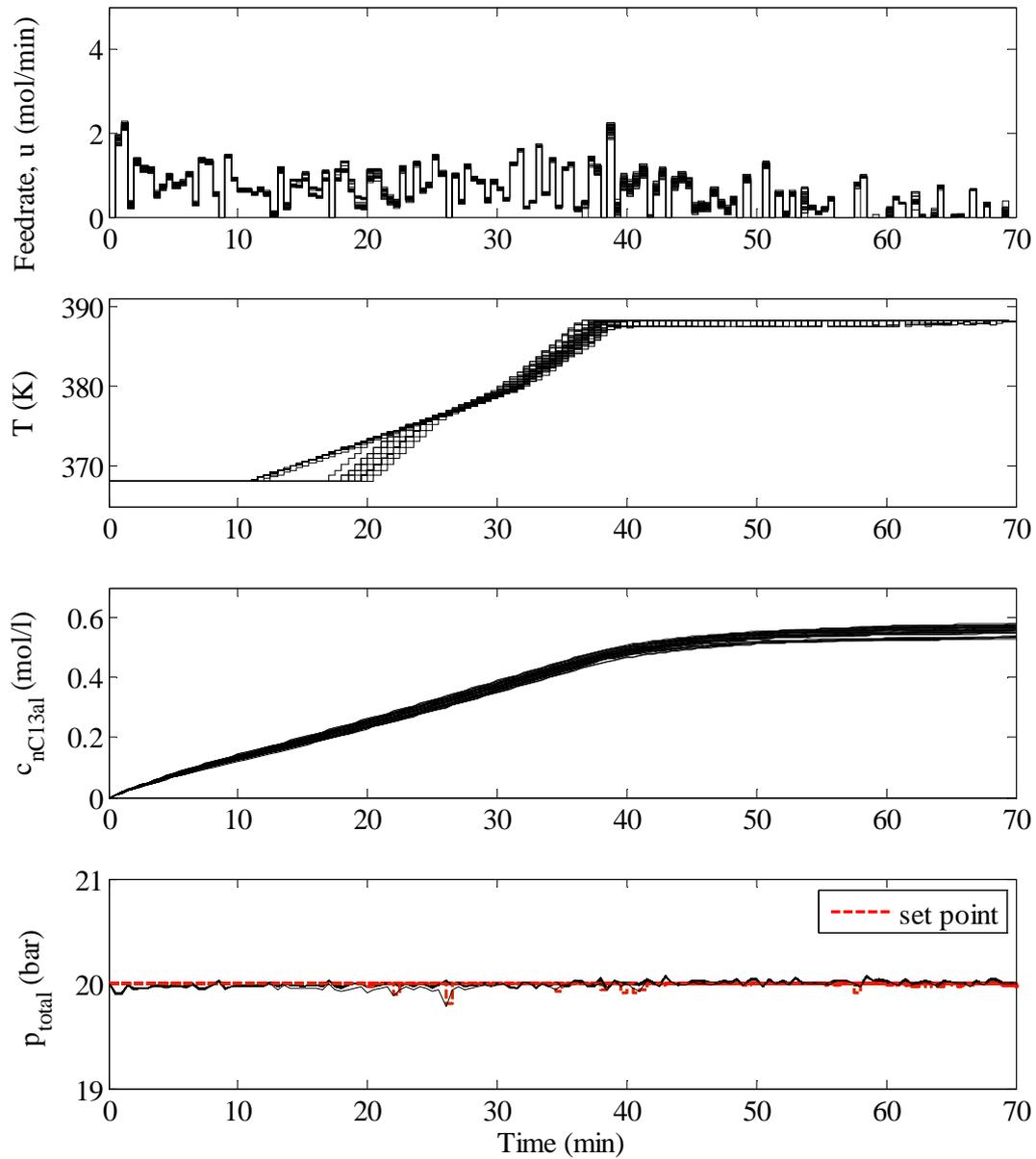


**Figure 6.7.** Optimal *closed-loop* profiles with standard and parsimonious sh-NMPC.

As shown in Fig. 6.7, both control schemes contribute to very similar final concentrations of tridecanal (nC13al), even though the input profiles are different. Furthermore, it is observed that the temperature profiles exhibit the same 3-arc shape as the nominal solution. The resulting optimal costs are 0.5527 [mol/L] for standard sh-NMPC and 0.5525 [mol/L] for parsimonious sh-NMPC. On the other hand, parsimonious sh-NMPC is computationally far superior to standard sh-NMPC. For both methods, the CPU times at each sampling time are reported in Fig. 6.8. Also, one may expect that, due to the fact that the computational (or feedback) delay is reduced with parsimonious sh-NMPC, better closed-loop performance can be achieved in real-life operation. Additionally, faster sampling is enabled by fast computation, which may increase the performance of the closed loop by increasing optimization frequency.



**Figure 6.8.** Computational times with standard and parsimonious sh-NMPC for the hydroformylation semi-batch reactor.



**Figure 6.9.** Optimal *closed-loop* profiles with parsimonious sh-NMPC for 40 different batches.

Finally, parsimonious sh-NMPC is tested for 40 different uncertainty realizations. The optimal profiles are given in Fig. 6.9. The parsimonious sh-NMPC scheme results in feasible operation, with a mean final concentration of tridecanal of 0.5562 [mol/L].

## 6.3 Summary

This part of the thesis has combined the use of simplified solution models with shrinking-horizon NMPC for semi-batch processes. A parsimonious parameterization of the optimal inputs computed offline has been postulated, which reduces the complexity of the optimization problem and therefore also the CPU times at each sampling instant. This decrease in online computational effort is important in practice. Faster computation enables higher optimization frequencies, which in turn may lead to better closed-loop performance. The resulting scheme, labeled ‘parsimonious sh-NMPC’, has been applied to two case studies simulated in the presence of uncertainty.

Parsimonious sh-NMPC is compared to standard sh-NMPC in terms of both performance and robustness. It turns out that the performance of parsimonious sh-NMPC is very close to that of standard sh-NMPC. Significant reduction in real-time computational effort has been observed in both case studies. Although sh-NMPC only approximates the optimal input profiles, the closed-loop behavior is accurately captured, mostly because only the first part of the inputs is implemented at each iteration. The input profiles computed with parsimonious and standard sh-NMPC differ to some extent, but both schemes exhibit the same solution structures and arc types. It turns out that the performance of parsimonious sh-NMPC is very close to that of standard sh-NMPC in terms of optimal cost.

Furthermore, it can be stated that the computational superiority of the parsimonious sh-NMPC scheme makes it very suited for real-time applications of optimizing control to batch and semi-batch processes. Finally, due to its fast-NMPC features, other application areas such as stochastic or multi-stage NMPC should be envisioned.

*If one day, my words are against science, choose science.*

---

M. Kemal Atatürk (1881 – 1938)

# 7 SUMMARY AND OUTLOOK

## 7.1 Summary

In this thesis, a convergent PMP-based quasi-Newton algorithm is proposed for solving constrained batch semi-batch optimization problems. This algorithm reformulates the Hamiltonian function by indirectly adjoining the inequality path constraints so that the inputs can activate the path constraints at each infeasible iteration step via one single explicit computation. This way, the dynamic optimization algorithm can be tailored to have nearly linear increase in complexity with respect to increasing input discretization grids and time horizons.

The results in Chapter 3 indicate that the proposed PMP-based quasi-Newton algorithm can solve the corresponding constrained optimization problems significantly faster than direct simultaneous methods as the discretization grid gets finer. Chapter 3 also demonstrates that, although the differences between the optimal costs computed with various strategies are negligible, the actual input profiles can differ significantly and correspond to *different* local solutions. The main reason is the lack of sensitivity of the objective function with respect to the sensitivity-seeking parts of the optimal inputs. Therefore, it may be useful to parameterize these input profiles in an alternative way, for example by using switching times and low-order polynomial approximations rather than piecewise-constant or piecewise-linear approximations.

Following this idea, Chapter 4 details an alternative indirect solution method that parameterizes the sensitivity-seeking inputs parsimoniously and uses the PMP-based method

to solve the dynamic optimization of constrained batch and semi-batch processes. It is observed that the proposed parsimonious indirect method can solve the corresponding problems much faster for very similar optimal cost values.

Chapter 5 suggests using the indirect solution scheme proposed in Chapter 3 for the shrinking-horizon NMPC (sh-NMPC) for batch and semi-batch processes. The application of sh-NMPC to a two-phase semi-batch hydroformylation reactor under uncertainty is investigated. A time-varying backoff approach is used to deal with the effect of parametric uncertainties. The computational burden due to the matrix factorization in large-horizon problems is reduced through PMP via the interplay of states, co-states and Lagrange multipliers for path constraints. Accordingly, PMP-based NMPC has a clear computational advantage over NMPC based on direct simultaneous methods, especially at the beginning of the batch. Furthermore, the PMP-based solution algorithm can be extended to track the active constraints. This way, further reduction in computational time is possible.

Chapter 6 extends the use of parsimonious input parameterization together with the indirect solution algorithms for the shrinking-horizon NMPC for batch and semi-batch processes. As a result, the complexity of the optimization problem and, therefore, also the CPU times at each sampling instant can be reduced significantly. This decrease in online computational effort is important for real-life implementations of advanced control methods to tackle the effect of computational delay in feedback. In addition, the proposed method can be applied to real-time embedded systems, where time constants are much faster but there are limitations with respect to computational power due to short battery life. The resulting scheme is applied to two case studies simulated in the presence of uncertainty. The effect of parametric uncertainty is dealt with via hierarchical control. Parsimonious sh-NMPC is compared to standard sh-NMPC in terms of both performance and robustness. The performance of parsimonious sh-NMPC is very similar to that of standard sh-NMPC in terms of optimal cost. But, significant reduction in real-time computational effort can be observed through the use of parsimonious sh-NMPC in both case studies.

In conclusion, the computational advantage of the indirect methods for the dynamic optimization and nonlinear model predictive control for constrained batch and semi-batch processes represents the main outcome of this thesis. Especially for high input discretization levels and large time horizons (the case in batch and semi-batch processes), indirect methods exhibit certain advantages regarding the reduction of the computational complexity. In the dynamic optimization community, it has usually been reported that there were no convergent

indirect algorithm available for constrained problems. Nevertheless, this thesis shows that indirect methods can be tailored to be convergent and quite effective in terms of computational time for both dynamic optimization and sh-NMPC.

## 7.2 Outlook

First of all, the results given in this thesis are obtained *in silico*, in other words via simulations. In order to demonstrate the applicability and advantages of the proposed indirect methods, real-time applications should be performed. In addition, the effect of computational delay for nonlinear model predictive control strategies is still quite under-investigated, both in terms of applicability and closed-loop control performance for real-life problems.

It can be stated that the computational superiority of the parsimonious sh-NMPC scheme makes it very suited for industrial applications of optimizing control. On the other hand, we should also note that advanced estimation/measurement techniques are always as important as the optimal control algorithms and they have a vital impact on the closed-loop performance. Equal attention should also be paid to the development of more sophisticated and powerful estimation/measurement techniques, as well as new measuring devices.

Tailored indirect methods can exhibit certain advantages for more complex applications such as stochastic optimization, mixed-integer, multi-level optimization, moving horizon estimation and multi-stage algorithms where fast implementations are needed. Moreover, application of indirect-based algorithms to the large-scale optimization problems and optimal experimental design can also turn out to be promising. The computational speed of these algorithms can be further increased by discretizing the state and co-state equations or using collocation on finite elements instead of relying on integration. Yet, it is still an open question how PMP-based NMPC performs for large-scale problems, where the state and co-states equations are integrated.

Finally, we are still in need of advanced optimization algorithms requiring less computational effort, power and CPU time even though computers and parallelization algorithms are being improved every day. This requirement applies not only to dynamic optimization and NMPC for batch and semi-batch processes, but also for the currently popular artificial intelligence and machine learning strategies together with ‘*smart manufacturing*’ and ‘*internet of things*’. Wireless systems and connected networks may require reduced

computational effort for operational reasons such as limited battery life. Tailored indirect methods may possess computational benefits for these areas as well.

# 8 BIBLIOGRAPHY

- Abel, O., Helbig, A., Marquardt, W., Zwick, H., & Daszkowski, T. (2000). Productivity optimization of an industrial semi-batch polymerization reactor under safety constraints. *Journal of Process Control*, 10, 351-362.
- Abel, O., & Marquardt, W. (2003). Scenario-integrated on-line optimisation of batch reactors. *Journal of Process Control*, 13, 703-715.
- Adetola, V., & Guay, M. (2010). Integration of real-time optimization and model predictive control. *Journal of Process Control*, 20, 125-133.
- Ali, U., & Wardi, Y. (2015). Multiple shooting technique for optimal control problems with application to power aware networks. *IFAC-PapersOnLine*, 48, 286-290.
- Allgöwer, F., Badgwell, T. A., Qin, J. S., Rawlings, J. B., & Wright, S. J. (1999). Nonlinear predictive control and moving horizon estimation—An introductory overview. In *Advances in control* (pp. 391-449): Springer.
- Andersson, J., & Diehl, M. (2012). Dynamic optimization with CasADi. In *51st IEEE Conference on Decision and Control* (pp. 681-686).
- Angeli, D., Amrit, R., & Rawlings, J. B. (2012). On average performance and stability of economic model predictive control. *IEEE Transactions on Automatic Control*, 57, 1615-1626.
- Assassa, F., & Marquardt, W. (2014). Dynamic optimization using adaptive direct multiple shooting. *Computers & Chemical Engineering*, 60, 242-259.
- Aydin, E., Bonvin, D., & Sundmacher, K. (2017a). Dynamic optimization of constrained semi-batch processes using Pontryagin's minimum principle—An effective quasi-Newton approach. *Computers & Chemical Engineering*, 99, 135-144.
- Aydin, E., Bonvin, D., & Sundmacher, K. (2017b). Dynamic Optimization of Constrained Semi-Batch Processes using Pontryagin's Minimum Principle and Parsimonious Parameterization. *Computer Aided Chemical Engineering* (Vol. 40, pp. 2041-2046): Elsevier.
- Aydin, E., Bonvin, D., & Sundmacher, K. (2018a). Computationally Efficient NMPC for Batch and Semi-Batch Processes Using Parsimonious Input Parameterization. *Journal of Process Control*, In Press.
- Aydin, E., Bonvin, D., & Sundmacher, K. (2018b). NMPC using Pontryagin's Minimum Principle-Application to a two-phase semi-batch hydroformylation reactor under uncertainty. *Computers & Chemical Engineering*, 108, 47-56.

- Aydin, E., Bonvin, D., & Sundmacher, K. (2018c). Toward Fast Dynamic Optimization: An Indirect Algorithm that Uses Parsimonious Input Parameterization. *Industrial & Engineering Chemistry Research*, Under Review.
- Biegler, L., Yang, X., & Fischer, G. (2015). Advances in sensitivity-based nonlinear model predictive control and dynamic real-time optimization. *Journal of Process Control*, 30, 104-116.
- Biegler, L. T. (2007). An overview of simultaneous strategies for dynamic optimization. *Chemical Engineering and Processing: Process Intensification*, 46, 1043-1053.
- Biegler, L. T. (2010). *Nonlinear programming: concepts, algorithms, and applications to chemical processes* (Vol. 10): SIAM.
- Biegler, L. T., Cervantes, A. M., & Wächter, A. (2002). Advances in simultaneous strategies for dynamic process optimization. *Chemical Engineering Science*, 57, 575-593.
- Binette, J.-C., & Srinivasan, B. (2016). On the use of nonlinear model predictive control without parameter adaptation for batch processes. *Processes*, 4, 27.
- Binette, J., Srinivasan, B., & Bonvin, D. (2016). On the various local solutions to a two-input dynamic optimization problem. *Computers & Chemical Engineering*, 95, 71-74.
- Bock, H. G., Diehl, M. M., Leineweber, D. B., & Schlöder, J. P. (2000). A direct multiple shooting method for real-time optimization of nonlinear DAE processes. In F. Allgöwer & A. Zheng (Eds.), *Nonlinear Model Predictive Control* (pp. 245-267). Basel: Birkhäuser Basel.
- Boggs, P. T., & Tolle, J. W. (2000). Sequential quadratic programming for large-scale nonlinear optimization. *Journal of Computational and Applied Mathematics*, 124, 123-137.
- Bonvin, D. (1998). Optimal operation of batch reactors—a personal view. *Journal of Process Control*, 8, 355-368.
- Bonvin, D. (2006). Control and optimization of batch processes. *IEEE Control Systems*, 26, 34-45.
- Bonvin, D., & François, G. (2017). Chapter 11 - Control and Optimization of Batch Chemical Processes A2 - Rohani, Sohrab. In *Coulson and Richardson's Chemical Engineering (Fourth Edition)* (pp. 441-503): Butterworth-Heinemann.
- Bonvin, D., Srinivasan, B., & Ruppen, D. (2001). Dynamic optimization in the batch chemical industry. In *Preprints CPC6, Tucson, Arizona*, 283–307.
- Bosley, J., & Edgar, T. (1992). Application of nonlinear model predictive control to optimal batch distillation. *IFAC Proceedings Volumes*, 25, 303-308.
- Bryson, A. E. (1975). *Applied Optimal Control: Optimization, Estimation and Control*: CRC Press.
- Cannon, M. (2004). Efficient nonlinear model predictive control algorithms. *Annual Reviews in Control*, 28, 229-237.
- Cannon, M., Liao, W., & Kouvaritakis, B. (2008). Efficient MPC optimization using Pontryagin's minimum principle. *International Journal of Robust and Nonlinear Control*, 18, 831-844.
- Cao, Y., Acevedo, D., Nagy, Z. K., & Laird, C. D. (2017). Real-time feasible multi-objective optimization based nonlinear model predictive control of particle size and shape in a batch crystallization process. *Control Engineering Practice*, 69, 1-8.
- Cervantes, A., & Biegler, L. T. (1998). Large-scale DAE optimization using a simultaneous NLP formulation. *AIChE Journal*, 44, 1038-1050.
- Chachuat, B. (2007). *Nonlinear and dynamic optimization: From theory to practice*. Lecture Notes.
- Chachuat, B., Srinivasan, B., & Bonvin, D. (2009). Adaptation strategies for real-time optimization. *Computers & Chemical Engineering*, 33, 1557-1567.

- Coward, I. (1967). The time-optimal problem in binary batch distillation. *Chemical Engineering Science*, 22, 503-516.
- De Souza, G., Odloak, D., & Zanin, A. C. (2010). Real time optimization with model predictive control (MPC). *Computers & Chemical Engineering*, 34, 1999-2006.
- Diehl, M., H.G. Bock, J.P. Schlöder, A Real-Time Iteration Scheme for Nonlinear Optimization in Optimal Feedback Control, *SIAM Journal on Control and Optimization*, 43 (2005) 1714-1736.
- Diehl, M., Amrit, R., & Rawlings, J. B. (2011). A Lyapunov function for economic optimizing model predictive control. *IEEE Transactions on Automatic Control*, 56, 703-707.
- Diehl, M., Bock, H. G., Diedam, H., & Wieber, P.-B. (2006). Fast direct multiple shooting algorithms for optimal robot control. In *Fast Motions in Biomechanics and Robotics* (pp. 65-93): Springer.
- Diehl, M., Bock, H. G., Schlöder, J. P., Findeisen, R., Nagy, Z., & Allgöwer, F. (2002). Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12, 577-585.
- Eaton, J. W., & Rawlings, J. B. (1990). Feedback control of chemical processes using on-line optimization techniques. *Computers & Chemical Engineering*, 14, 469-479.
- Ebrahim, T., Hernandez, R., Subramanian, S., Kalliski, M., Krämer, S., & Engell, S. (2016). NCO-tracking with changing set of active constraints using multiple solution models. *IFAC-PapersOnLine*, 49, 79-84.
- Findeisen, R., & Allgöwer, F. (2004). Computational delay in nonlinear model predictive control. *IFAC Proceedings Volumes*, 37, 427-432.
- Findeisen, R., Allgöwer, F., & Biegler, L. T. (2007). *Assessment and Future Directions of Nonlinear Model Predictive Control* (Vol. 358): Springer.
- Fletcher, R. (2013). *Practical Methods of Optimization*: John Wiley & Sons.
- García, C. E., Prett, D. M., & Morari, M. (1989). Model predictive control: Theory and practice—A survey. *Automatica*, 25, 335-348.
- Goldsmith, M. J. (1999). *Sequential Quadratic Programming Methods based on Indefinite Hessian Approximations*: Stanford University.
- Graichen, K., & Käpernick, B. (2012). A real-time gradient method for nonlinear model predictive control. In *Frontiers of Model Predictive Control*.
- Gros, S., Zanon, M., Quirynen, R., Bemporad, A., & Diehl, M. (2016). From linear to nonlinear MPC: bridging the gap via the real-time iteration. *International Journal of control*, 1-19.
- Hannemann-Tamás, R., & Marquardt, W. (2012). How to verify optimal controls computed by direct shooting methods? – A tutorial. *Journal of Process Control*, 22, 494-507.
- Hartl, R. F., Sethi, S. P., & Vickson, R. G. (1995). A survey of the maximum principles for optimal control problems with state constraints. *SIAM review*, 37, 181-218.
- Harvey Jr, P. S., Gavin, H. P., & Scruggs, J. T. (2012). Optimal performance of constrained control systems. *Smart Materials and Structures*, 21, 085001.
- Hentschel, B., Kiedorf, G., Gerlach, M., Hamel, C., Seidel-Morgenstern, A., Freund, H., & Sundmacher, K. (2015). Model-based identification and experimental validation of the optimal reaction route for the hydroformylation of 1-dodecene. *Industrial & Engineering Chemistry Research*, 54, 1755-1765.
- Huang, R., Biegler, L. T., & Patwardhan, S. C. (2010). Fast offset-free nonlinear model predictive control based on moving horizon estimation. *Industrial & Engineering Chemistry Research*, 49, 7882-7890.
- Huang, R., Zavala, V. M., & Biegler, L. T. (2009). Advanced step nonlinear model predictive control for air separation units. *Journal of Process Control*, 19, 678-685.

- Jang, H., Lee, J. H., & Biegler, L. T. (2016). A robust NMPC scheme for semi-batch polymerization reactors. *IFAC-PapersOnLine*, 49, 37-42.
- Jäschke, J., Yang, X., & Biegler, L. T. (2014). Fast economic model predictive control based on NLP-sensitivities. *Journal of Process Control*, 24, 1260-1272.
- Jaspan, R., & Coull, J. (1971). Trajectory optimization techniques in chemical reaction engineering: I. Boundary condition iteration methods. *AIChE Journal*, 17, 111-115.
- Jung, T. Y., Nie, Y., Lee, J., & Biegler, L. (2015). Model-based on-Line optimization framework for semi-batch polymerization reactors. *IFAC-PapersOnLine*, 48, 164-169.
- Kadam, J., Schlegel, M., Srinivasan, B., Bonvin, D., & Marquardt, W. (2007). Dynamic optimization in the presence of uncertainty: From off-line nominal solution to measurement-based implementation. *Journal of Process Control*, 17, 389-398.
- Kaiser, N. M., Flassig, R. J., & Sundmacher, K. (2016). Probabilistic reactor design in the framework of elementary process functions. *Computers & Chemical Engineering*, 94, 45-59.
- Kameswaran, S., & Biegler, L. T. (2006). Simultaneous dynamic optimization strategies: Recent advances and challenges. *Computers & Chemical Engineering*, 30, 1560-1575.
- Keil, F. (1999). *Scientific Computing in Chemical Engineering II: Simulation, Image Processing, Optimization, and Control*: Springer Science & Business Media.
- Kim, N., & Rousseau, A. (2012). Sufficient conditions of optimal control based on Pontryagin's minimum principle for use in hybrid electric vehicles. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, 226, 1160-1170.
- Lakshmanan, N., & Arkun, Y. (1999). Estimation and model predictive control of non-linear batch processes using linear parameter varying models. *International Journal of Control*, 72, 659-675.
- Li, P., Arellano-Garcia, H., & Wozny, G. (2008). Chance constrained programming approach to process optimization under uncertainty. *Computers & Chemical Engineering*, 32, 25-45.
- Logist, F., Houska, B., Diehl, M., & Van Impe, J. F. (2011). Robust multi-objective optimal control of uncertain bio-chemical processes. *Chemical Engineering Science*, 66, 4670-4682.
- Lucia, S., Andersson, J. A., Brandt, H., Diehl, M., & Engell, S. (2014). Handling uncertainty in economic nonlinear model predictive control: A comparative case study. *Journal of Process Control*, 24, 1247-1259.
- Lucia, S., Finkler, T., & Engell, S. (2013). Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty. *Journal of Process Control*, 23, 1306-1319.
- Marchetti, A., Amrhein, M., Chachuat, B., & Bonvin, D. (2006). Scale-up of batch processes via decentralized control. *IFAC Proceedings Volumes*, 39, 221-226.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Sokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36, 789-814.
- Mayur, D., & Jackson, R. (1971). Time-optimal problems in batch distillation for multicomponent mixtures and for columns with holdup. *The Chemical Engineering Journal*, 2, 150-163.
- Mesbah, A. (2016). Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems*, 36, 30-44.
- Mesbah, A., Huesman, A. E., Kramer, H. J., Nagy, Z. K., & Van den Hof, P. M. (2011). Real-time control of a semi-industrial fed-batch evaporative crystallizer using different direct optimization strategies. *AIChE Journal*, 57, 1557-1569.

- Mesbah, A., Streif, S., Findeisen, R., & Braatz, R. D. (2014). Stochastic nonlinear model predictive control with probabilistic constraints. In *American Control Conference, 2014* (pp. 2413-2419).
- Miele, A. (1978). *Gradient Algorithms for the Optimization of Dynamic Systems*. Rice University Press.
- Morari, M., & Lee, J. H. (1999). Model predictive control: past, present and future. *Computers & Chemical Engineering*, 23, 667-682.
- Nagy, Z. K., & Braatz, R. D. (2003). Robust nonlinear model predictive control of batch processes. *AIChE Journal*, 49, 1776-1786.
- Nagy, Z. K., & Braatz, R. D. (2004). Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis. *Journal of Process Control*, 14, 411-422.
- Nagy, Z. K., Mahn, B., Franke, R., & Allgöwer, F. (2007). Real-time implementation of nonlinear model predictive control of batch processes in an industrial framework. In *Assessment and Future Directions of Nonlinear Model Predictive Control* (pp. 465-472): Springer.
- Onori, S., Serrao, L., & Rizzoni, G. (2016). Pontryagin's Minimum Principle. In *Hybrid Electric Vehicles* (pp. 51-63): Springer.
- Palanki, S. Chapter 16: The Application of Pontryagin's Minimum Principle for Endpoint Optimization of Batch Processes. In *Control and Optimization with Differential-Algebraic Constraints* (pp. 327-337).
- Palanki, S., & Vemuri, J. (2005). Optimal operation of semi-batch processes with a single reaction. *International Journal of Chemical Reactor Engineering*, 3.
- Puschke, J., & Mitsos, A. (2016). Robust dynamic optimization of a semi-batch emulsion polymerization process with parametric uncertainties—A heuristic approach. *IFAC-PapersOnLine*, 49, 907-912.
- Puschke, J., Zubov, A., Kosek, J., & Mitsos, A. (2016). Multi-model approach based on parametric sensitivities—A heuristic approximation for dynamic optimization of semi-batch processes with parametric uncertainties. *Computers & Chemical Engineering*.
- Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control Engineering Practice*, 11, 733-764.
- Rao, C. V., & Rawlings, J. B. (2002). Constrained process monitoring: Moving-horizon approach. *AIChE Journal*, 48, 97-109.
- Rao, C. V., Rawlings, J. B., & Lee, J. H. (2001). Constrained linear state estimation—a moving horizon approach. *Automatica*, 37, 1619-1628.
- Roubos, J., De Gooijer, C., Van Straten, G., & Van Boxtel, A. (1997). Comparison of optimization methods for fed-batch cultures of hybridoma cells. *Bioprocess Engineering*, 17, 99-102.
- Ruppen, D., Benthack, C., & Bonvin, D. (1995). Optimization of batch reactor operation under parametric uncertainty—Computational aspects. *Journal of Process Control*, 5, 235-240.
- Ruppen, D., Bonvin, D., & Rippin, D. (1998). Implementation of adaptive optimal operation for a semi-batch reaction system. *Computers & Chemical Engineering*, 22, 185-199.
- Sager, S. (2005). *Numerical methods for mixed-integer optimal control problems: Der andere Verlag Tönning, Lübeck, Marburg*.
- Sahinidis, N. V. (2004). Optimization under uncertainty: State-of-the-art and opportunities. *Computers & Chemical Engineering*, 28, 971-983.
- Schäfer, A., Kühl, P., Diehl, M., Schlöder, J., & Bock, H. G. (2007). Fast reduced multiple shooting methods for nonlinear model predictive control. *Chemical Engineering and Processing: Process Intensification*, 46, 1200-1214.

- Schlegel, M., & Marquardt, W. (2004). Direct sequential dynamic optimization with automatic switching structure detection. In Proc. of DYCOPS (Vol. 7).
- Schlegel, M., & Marquardt, W. (2006a). Adaptive switching structure detection for the solution of dynamic optimization problems. *Industrial & Engineering Chemistry Research*, 45, 8083-8094.
- Schlegel, M., & Marquardt, W. (2006b). Detection and exploitation of the control switching structure in the solution of dynamic optimization problems. *Journal of Process Control*, 16, 275-290.
- Schlegel, M., Stockmann, K., Binder, T., & Marquardt, W. (2005). Dynamic optimization using adaptive control vector parameterization. *Computers & Chemical Engineering*, 29, 1731-1751.
- Schneider, R., & Georgakis, C. (2013). How to not make the extended Kalman filter fail. *Industrial & Engineering Chemistry Research*, 52, 3354-3362.
- Seki, H., Ogawa, M., Ooyama, S., Akamatsu, K., Ohshima, M., & Yang, W. (2001). Industrial application of a nonlinear model predictive control to polymerization reactors. *Control Engineering Practice*, 9, 819-828.
- Shi, J., Biegler, L. T., Hamdan, I., & Wassick, J. (2016). Optimization of grade transitions in polyethylene solution polymerization process under uncertainty. *Computers & Chemical Engineering*, 95, 260-279.
- Srinivasan, Bonvin, D., Visser, E., & Palanki, S. (2003a). Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty. *Computers & Chemical Engineering*, 27, 27-44.
- Srinivasan, B., Biegler, L. T., & Bonvin, D. (2008). Tracking the necessary conditions of optimality with changing set of active constraints using a barrier-penalty function. *Computers & Chemical Engineering*, 32, 572-579.
- Srinivasan, B., & Bonvin, D. (2007). Real-Time Optimization of Batch Processes by Tracking the Necessary Conditions of Optimality. *Industrial & Engineering Chemistry Research*, 46, 492-504.
- Srinivasan, B., Palanki, S., & Bonvin, D. (2003b). Dynamic optimization of batch processes: I. Characterization of the nominal solution. *Computers & Chemical Engineering*, 27, 1-26.
- Suwartadi, E., Kungurtsev, V., & Jäschke, J. (2017). Sensitivity-based economic NMPC with a path-following approach. *Processes*, 5, 8.
- Terwiesch, P., Agarwal, M., & Rippin, D. W. (1994). Batch unit optimization with imperfect modelling: a survey. *Journal of Process Control*, 4, 238-258.
- Thomas, S., Pushpavanam, S., & Seidel-Morgenstern, A. (2004). Performance improvements of parallel– series reactions in tubular reactors using reactant dosing concepts. *Industrial & Engineering Chemistry Research*, 43, 969-979.
- Valappil, J., & Georgakis, C. (2002). Nonlinear model predictive control of end-use properties in batch reactors. *AIChE Journal*, 48, 2006-2021.
- Vassiliadis, V. S., Sargent, R. W. H., & Pantelides, C. C. (1994). Solution of a class of multistage dynamic optimization problems. 2. Problems with path constraints. *Industrial & Engineering Chemistry Research*, 33, 2123-2133.
- Visser, E., Srinivasan, B., Palanki, S., & Bonvin, D. (2000). A feedback-based implementation scheme for batch process optimization. *Journal of Process Control*, 10, 399-410.
- Wächter, A., & Biegler, L. (2009). IPOPT-an interior point optimizer. In <https://projects.coinor.org/Ipopt>.
- Wächter, A., & Biegler, L. T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106, 25-57.

- Welz, C., Srinivasan, B., & Bonvin, D. (2008). Measurement-based optimization of batch processes: Meeting terminal constraints on-line via trajectory following. *Journal of Process Control*, 18, 375-382.
- Welz, C., Srinivasan, B., Marchetti, A., Bonvin, D., & Ricker, N. (2005). Validation of a solution model for the optimization of a binary batch distillation column. In *Proc. of American Control Conference*, (pp. 3127-3132).
- Welz, C., Srinivasan, B., Marchetti, A., Bonvin, D., & Ricker, N. (2006). Evaluation of input parameterization for batch process optimization. *AIChE Journal*, 52, 3155-3163.
- Wolf, I. J., & Marquardt, W. (2016). Fast NMPC schemes for regulatory and economic NMPC—A review. *Journal of Process Control*, 44, 162-183.
- Wolf, I. J., Würth, L., & Marquardt, W. (2011). Rigorous solution vs. fast update: Acceptable computational delay in NMPC. In *Decision and Control and European Control Conference*, (pp. 5230-5235).
- Wolfe, P. (1971). Convergence Conditions for Ascent Methods. II: Some Corrections. *SIAM review*, 13, 185-188.
- Wright, S. J. (1997). *Primal-dual Interior-point Methods*: SIAM.
- Zavala, V. M., & Biegler, L. T. (2009). The advanced-step NMPC controller: Optimality, stability and robustness. *Automatica*, 45, 86-93.
- Zavala, V. M., Laird, C. D., & Biegler, L. T. (2008a). Fast implementations and rigorous models: Can both be accommodated in NMPC? *International Journal of Robust and Nonlinear Control*, 18, 800-815.
- Zavala, V. M., Laird, C. D., & Biegler, L. T. (2008b). A fast moving horizon estimation algorithm based on nonlinear programming sensitivity. *Journal of Process Control*, 18, 876-884.
- Zhang, S., Xiong, R., & Sun, F. (2017). Model predictive control for power management in a plug-in hybrid electric vehicle with a hybrid energy storage system. *Applied Energy*, 185, Part 2, 1654-1662.
- Zubov, A., Naeem, O., Hauger, S. O., Bouaswaig, A., Gjertsen, F., Singstad, P., Hungenberg, K. D., & Kosek, J. (2017). Bringing the on-Line control and optimization of semibatch emulsion copolymerization to the pilot plant. *Macromolecular Reaction Engineering*.

# 9 APPENDICES

## APPENDIX 1: SIMPLE CASE STUDY SOLVED WITH THE FULLY PARAMETERIZED PMP-BASED METHOD

Consider the optimization of a batch reactor which includes only input bounds, and is directly taken from (Biegler, 2010). The dynamic optimization problem is formulated as follows ( $t_f = 1$ ):

$$\begin{aligned} & \max_{u(t)} x_2(t_f) \\ \text{s.t.} \quad & \dot{x}_1 = -\left(u + \frac{u^2}{2}\right)x_1; \quad x_1(0) = 1; \\ & \dot{x}_2 = ux_1; \quad x_2(0) = 0; \\ & 0 \leq u(t) \leq 5 \end{aligned}$$

## APPENDIX 2: MODEL PARAMETERS FOR PROBLEM 3.2.3

| Reaction Kinetics     |                    |                        |  |                     |                   |                   |
|-----------------------|--------------------|------------------------|--|---------------------|-------------------|-------------------|
| component             | $E_A$<br>(kJ/mol)  | $k_0$                  | Unit   | $K_1$<br>(mL/mol)   | $K_2$<br>(mL/mol) | $K_3$<br>(mL/mol) |
| $r_1$                 | 113.08             | $4.904 \times 10^{16}$ | $\text{mL}^3/(\text{g} \cdot \text{min} \cdot \text{mol}^2)$ | 574876              | 3020413           | 11732838          |
| $r_2$                 | 136.89             | $4.878 \times 10^{16}$ | $\text{mL}/(\text{g} \cdot \text{min})$                      | 38632               | 226214            |                   |
| $r_3$                 | 76.11              | $5.411 \times 10^8$    | $\text{mL}^2/(\text{g} \cdot \text{min} \cdot \text{mol})$   | 2661.2              | 7100              | 1280              |
| $r_4$                 | 102.26             | $2.958 \times 10^4$    | $\text{mL}^2/(\text{g} \cdot \text{min} \cdot \text{mol})$   |                     |                   |                   |
| $r_5$                 | 120.84             | $7.619 \times 10^{10}$ | $\text{mL}^3/(\text{g} \cdot \text{min} \cdot \text{mol}^2)$ |                     |                   |                   |
| $r_6$                 | 113.08             | $3.951 \times 10^{10}$ | $\text{mL}^3/(\text{g} \cdot \text{min} \cdot \text{mol}^2)$ |                     |                   |                   |
| $c_{cat}$             |                    |                        |  | $3.041 \times 10^4$ | 0                 | 0.644             |
| Equilibrium Constants |                    |                        |  |                     |                   |                   |
| component             | $a_0$ (kJ/mol)     | $a_1$ (kJ/mol/K)       | $a_2$ (kJ/mol/K <sup>2</sup> )                               |                     |                   |                   |
| $\Delta G_2$          | -11.0034           | 0                      | 0  |                     |                   |                   |
| $\Delta G_3$          | -126.275           | 0.1266                 | $6.803 \times 10^{-6}$                                       |                     |                   |                   |
| Solubility            |                    |                        |  |                     |                   |                   |
| component             | $H_0$ (bar.mL/mol) | $E_{A,H}$ (kJ/mol)     | $k_L a$ (min <sup>-1</sup> )                                 |                     |                   |                   |
| H <sub>2</sub>        | 66400              | -3.06                  | 9.57   |                     |                   |                   |
| CO                    | 73900              | -0.84                  | 7.08   |                     |                   |                   |

## CURRICULUM VITAE

### CONTACT

**Address:** Mittelstraße 50/29 39114,  
Magdeburg, Germany  
**Phone:** +49 391 6110 391  
**E-mail:** aydin@mpi-magdeburg.mpg.de

### PERSONAL INFORMATION

**Citizenship:** Republic of Turkey  
**Birthplace:** Istanbul-Turkey  
**Birth Date:** 06/01/1990

### BACKGROUND

**Ph.D:** 2015 – 2018

- Max Planck Institute DCTS – PSE Department
  - Advisor: Prof. Dr. Kai Sundmacher
- International Max Planck Research School for Advanced Methods in Process and Systems Engineering (IMPRS)
- Otto von Guericke University Magdeburg  
Thesis: Tailored Indirect Algorithms for Efficient On-line Optimization of Batch and Semi-Batch Processes

**M.Sc:** 2013-2015

- Koc University – Chemical and Biological Eng. Department
  - Advisor: Prof. Dr. Yaman Arkun *C.Gpa: 3.66/4.00*
  - Thesis: Plant-Wide Modeling, Optimization and Control of an Industrial Diesel Hydroprocessing Plant

**B.Sc:** 2008-2013

- Middle East Technical University (METU) – Chemical Eng. Department
  - Advisor: Prof. Dr. Deniz Uner *C.Gpa: 3.47/4.00*
  - Thesis: Modeling the Gas-solid Reaction of Pyrite oxidation and its effects on coal gasification

### PATENT

Aydin, E., Celebi, A. D., Sildir, H., Arkun, Y., Canan, U., Is, G., Dikbas, M., Erdogan, M., Mutlu, M. (2015) An automation system for the modelling, optimization and control of an industrial diesel hydroprocessing plant, *European Patent Office* (Under Review)

## TEACHING EXPERIENCE

**2015 – 2018:** Teaching Assistant at Otto v.Guericke University

- Advanced Process Systems Engineering

**2013 – 2015:** Teaching Assistant at Koc University

- Introduction to Chemical Engineering
- Process Control
- Numerical Methods for Chemical Engineering
- Applied Optimal Control

**2012 – 2013:** Student Teaching Assistant at Middle East Technical University

- FORTRAN Language for Programming

## THESIS SUPERVISED

1) Giulia Slaviero – Adaptive Model Predictive Control of Semi-Batch Processes (M.Sc Thesis)

## WORK EXPERIENCE

1) Internship in NUH Cement Factory, 2012

2) Internship in TUPRAS Izmit Petroleum Refinery, 2011

