

© 2015

Jinjun Zhuge

ALL RIGHTS RESERVED

INTEGRATION OF PROCESS SCHEDULING AND CONTROL

by

JINJUN ZHUGE

A dissertation submitted to the

Graduate School-New Brunswick

Rutgers, The State University of New Jersey

In partial fulfillment of the requirements

For the degree of

Doctor of Philosophy

Graduate Program in Chemical & Biochemical Engineering

Written under the direction of

Marianthi G. Ierapetritou, Ph.D.

And approved by

---

---

---

---

---

New Brunswick, New Jersey

May, 2015

ABSTRACT OF THE DISSERTATION  
INTEGRATION OF PROCESS SCHEDULING AND CONTROL

By JINJUN ZHUGE

Dissertation Director:

Marianthi G. Ierapetritou, PhD

The objective of this dissertation is to develop integrated models and optimization methods to solve for chemical process scheduling and control problems. A traditional approach to handle process operations at scheduling and control levels is to consider them as separate optimization problems. However, scheduling and dynamic optimization at control level are naturally connected. An integrated decision making helps to achieve an overall optimality and thus improves the profitability of process operations. Integration of scheduling and control results in Mixed Integer Dynamic Optimization (MIDO) which is computationally expensive. To reduce the complexity brought by integration, research efforts of this dissertation target two goals focusing on first reducing the model complexity, and second reducing the solution computational time especially in the case of online implementations (i.e. closed loop implementations). In this dissertation, we first proposed an approach of implementing closed loop scheduling and control when the processes are subject to disturbance. Then we proposed a decomposition approach for the large size Mixed Integer Nonlinear Programming (MINLP) resulted from the integration of scheduling and control through sensitivity analysis. To facilitate online applications, we adopt multi-parametric Model Predictive Control (mp-MPC) at the control level and built a new integrated model using the explicit control solution generated by mp-MPC. We also developed an integrated model using a Piecewise Affine (PWA) model and used fast MPC at the control level to overcome the exponential dimension increasing in mp-MPC. Finally we discuss the uncertainty in process operations and present solution procedures of robust MPC for nonlinear problem at the

control level. Throughout this dissertation, detailed integrated models and the solution algorithms are developed and case studies are used to demonstrate the effectiveness of the proposed approaches.

## **ACKNOWLEDGEMENTS**

I would like to express my deepest gratitude to my advisor Prof. Marianthi Ierapetritou, for her excellent guidance and consistent encouragement throughout my course study and academic research. I learned from her the ways of investigating a new area, exploring the unknowns, and presenting the findings. They are invaluable fortunes that I gained during my PhD study. I believe that they will further benefit my career in the future.

I would also like to thank the committee members of the proposal and defense of this dissertation, Prof. Susan Albin, Prof. Ioannis Androulakis, Prof. Meenakshi Dutt, and Prof. Rohit Ramachandran, for their insightful and helpful comments.

I acknowledge the funding sources from National Science Foundation under grant CBET-1159244 that supported my research work.

I am grateful for the time spent in the Laboratory for Optimization and Systems Analysis. It's a pleasure working with all the friends and colleges in this group.

Finally, I deeply appreciate my wonderful parents and brother. Without their love and support, I wouldn't be able to go through the PhD stage. I would like to dedicate this dissertation to them.

## Table of Contents

|   |      |
|---|------|
| ABSTRACT OF THE DISSERTATION .....  | ii   |
| ACKNOWLEDGEMENTS .....  | iv   |
| List of tables.....   | viii |
| List of illustrations .....   | xi   |
| Chapter 1 Introduction .....  | 1    |
| 1.1 Scheduling and control for chemical processes .....   | 1    |
| 1.2 The significance of integration of scheduling and control.....  | 2    |
| 1.3 Modeling, computation, and challenges of integration of scheduling and control .....                                      | 3    |
| 1.4 Motivation and outline of the dissertation .....  | 4    |
| Chapter 2 A closed loop implementation of integrated scheduling and control .....   | 7    |
| 2.1 Integrated model for cyclic scheduling and control for continuous processes .....   | 7    |
| 2.1.1 General integrated model for scheduling and control .....   | 7    |
| 2.1.2 Scheduling constraints .....  | 10   |
| 2.1.3 Dynamic optimization problem .....  | 13   |
| 2.2 The closed loop implementation.....   | 15   |
| 2.3 Stability and robustness of closed loop implementation.....   | 17   |
| 2.3.1 Proof of the asymptotic stability of the closed loop system.....  | 17   |
| 2.3.2 Robustness analysis of the integrated scheduling and control .....  | 19   |
| 2.4 Case studies.....   | 20   |
| 2.4.1 Cyclic production in a multiproduct CSTR .....  | 20   |
| 2.4.2 Cyclic production of Multiproduct using Isothermal Tubular Reactor .....  | 28   |
| 2.5 Summary .....   | 35   |
| Chapter 3 A decomposition approach for the solution of scheduling including process dynamics<br>of continuous processes ..... | 38   |

|  |    |
|--|----|
| 3.1 General integrated model .....   | 38 |
| 3.2 A decomposition approach with analytical proof .....   | 40 |
| 3.2.1 Lemma 1: The perturbation on production time does not affect the transitions .....   | 43 |
| 3.2.2 Lemma 2: The optimal transition time is obtained as the inferior of its feasible values<br>.....                               | 44 |
| 3.2.3 Lemma 3: The perturbation on production time does not affect the production sequence<br>.....                                  | 45 |
| 3.2.4 Solution procedures for the decomposition approach.....  | 46 |
| 3.3 Case studies.....  | 47 |
| 3.3.1 Single state CSTR cyclic production .....  | 49 |
| 3.3.2 MMA production with multiple states .....  | 51 |
| 3.4 Summary .....  | 55 |
| Chapter 4 Integration of scheduling and control for batch processes using multi-parametric model<br>predictive control (mp-MPC)..... | 58 |
| 4.1 Introduction.....  | 58 |
| 4.2 Modeling the integration of scheduling and control for batch processes.....  | 59 |
| 4.3 Simultaneous scheduling and control incorporating mp-MPC .....   | 61 |
| 4.4 Detailed integrated model of scheduling and control incorporating mp-MPC .....   | 64 |
| 4.4.1 Constraints at scheduling level .....  | 65 |
| 4.4.2 Constraints at control level .....   | 68 |
| 4.4.3 The linking variables and constraints.....   | 69 |
| 4.4.4 The objective of the integrated problem .....  | 71 |
| 4.5 Case studies.....  | 73 |
| 4.5.1 A simple batch process .....   | 73 |
| 4.5.2 A more complex batch process .....   | 78 |
| 4.6 Summary .....  | 92 |

|  |     |
|--|-----|
| Chapter 5 An integrated framework for scheduling and control using fast Model Predictive Control ..... | 94  |
| 5.1 Introduction.....  | 94  |
| 5.2 Integration of scheduling and control based on (Piece-Wise Affine) PWA model .....                 | 96  |
| 5.2.1 A brief review of the PWA identification techniques .....  | 97  |
| 5.2.2 A PWA identification technique using optimization methods.....                                   | 100 |
| 5.3 Integrated problem incorporating PWA systems .....   | 103 |
| 5.4 Fast MPC for PWA systems .....   | 111 |
| 5.4.1 The role of fast MPC.....  | 111 |
| 5.4.2 Fast MPC for PWA systems .....   | 112 |
| 5.4.3 Stability analysis .....   | 114 |
| 5.5 Case studies.....  | 115 |
| 5.5.1 Single-Input Single-Output (SISO) CSTR.....  | 115 |
| 5.5.2 Multi-Input Multi-Output (MIMO) CSTR.....  | 122 |
| 5.6 Summary .....  | 130 |
| Chapter 6 Dealing with uncertainties in the integration of scheduling and control.....                 | 133 |
| 6.1 Uncertainty in process operations, origin and models .....   | 133 |
| 6.2 Approaches of dealing with uncertainty .....   | 134 |
| 6.3 Robust MPC for control problems .....  | 135 |
| 6.3.1 Robust MPC formulation for PWA systems.....  | 135 |
| 6.3.2 A general formulation of robust MPC for nonlinear systems .....                                  | 139 |
| Chapter 7 Conclusions and future perspectives .....  | 140 |
| Appendix A (Supporting Material for the decomposition proof in section 3.2) .....                      | 142 |
| Bibliography .....   | 151 |



## List of tables

|   |    |
|---|----|
| Table 2.1 Five products with steady state information and market information with CSTR .....  | 20 |
| Table 2.2 Simultaneous scheduling and control results for the original CSTR integrated problem.<br>no state deviation .....   | 22 |
| Table 2.3 Simultaneous scheduling and control results for closed loop implementation on CSTR,<br>state deviation occurs in transition from C to B.....                                | 24 |
| Table 2.4 Simultaneous scheduling and control results for open loop implementation on CSTR,<br>state deviation occurs in transition from C to B.....                                  | 24 |
| Table 2.5 Quantitatively comparison of open loop strategy and closed loop strategy on CSTR,<br>state deviation occurs in transition from C to B.....                                  | 25 |
| Table 2.6 Simultaneous scheduling and control results for closed loop implementation on CSTR,<br>state deviation occurs in production period of B (Case 1).....                       | 26 |
| Table 2.7 Simultaneous scheduling and control results for closed loop implementation on CSTR,<br>state deviation occurs in production period of B (Case 2).....                       | 27 |
| Table 2.8 Simultaneous scheduling and control results for open loop implementation on CSTR,<br>state deviation occurs in production period of B .....                                 | 27 |
| Table 2.9 Quantitatively comparison of open loop strategy and closed loop strategy, disturbance<br>occurs in production period of B .....   | 27 |
| Table 2.10 Five products with steady state information and market information with PFR .....  | 30 |
| Table 2.11 Simultaneous scheduling and control results for the original PFR problem, no state<br>deviation, corresponding to Figure 2.11 .....  | 33 |
| Table 2.12 Simultaneous scheduling and control results for closed loop implementation on PFR,<br>state deviation occurs in transition from D to B, corresponding to Figure 2.12 ..... | 34 |
| Table 2.13 Simultaneous scheduling and control results for open loop implementation on PFR,<br>state deviation occurs in transition from D to B, corresponding to Figure 2.14 .....   | 34 |

|   |     |
|---|-----|
| Table 2.14 Quantitatively comparison of open loop strategy and closed loop strategy, state deviation occurs in transition from D to B.....  | 34  |
| Table 3.1 Demands assignment in a single state CSTR cyclic production .....   | 50  |
| Table 3.2 Estimation of transition time (hour) and transition cost (\$) between products (single state CSTR), for instance, transition time from product C to B is 4 hours and the corresponding transition cost is \$10000 ..... | 50  |
| Table 3.3 Results of single state CSTR cyclic production, comparison between three scenarios .  | 51  |
| Table 3.4 Parameters in the kinetic model of MMA polymerization in a CSTR.....  | 52  |
| Table 3.5 Variables in the kinetic model of MMA polymerization in a CSTR .....  | 53  |
| Table 3.6 Steady state information for each product of MMA production.....  | 54  |
| Table 3.7 Estimation of transition time (hour) and transition cost (\$) between products (MMA case study), for instance, transition time from product C to B is 0.16h and the corresponding transition cost is \$85.68 .....      | 54  |
| Table 3.8 Results of MMA case study, comparison between three scenarios .....   | 54  |
| Table 4.1 Data for unit specification and market information of example 1: a simple batch process .....   | 74  |
| Table 4.2 Data for unit specification and market information of example 2: a complex batch process .....  | 79  |
| Table 4.3 Comparison of the quantitative results for four scenarios of the complex batch process case study.....  | 87  |
| Table 4.4 Results of the MINLP and the derived MILP (case 1) .....  | 89  |
| Table 4.5 Results of the MINLP and the derived MILP (case 2) .....  | 91  |
| Table 5.1 Comparison of enumeration and binary search based on their CPU time used in searching for a LTI in a 3-dimensional PWA.....   | 113 |
| Table 5.2 Steady state information of a CSTR with cyclic production .....   | 116 |
| Table 5.3 Results of the integrated problem of SISO CSTR case .....   | 117 |

|   |     |
|---|-----|
| Table 5.4 Products at different steady states for MIMO CSTR case study .....                            | 123 |
| Table 5.5 Process specifications of MIMO CSTR.....  | 123 |
| Table 5.6 Scheduling solutions for the integrated problem of MIMO CSTR case.....                        | 124 |
| Table 5.7 Analysis of the computation time for fast MPC and mp-MPC, online steps only.....              | 128 |
| Table 5.8 Notations associated with Table 5.7 .....   | 128 |
| Table 5.9 Comparison of CPU time in implementing fast MPC and mp-MPC using a<br>1.86GHz/4G RAM PC ..... | 130 |
| Table 6.1 Possible combinations of future uncertainty, and the associated possible states.....          | 136 |

## List of illustrations

|  |    |
|--|----|
| Figure 2.1 Cyclic production of continuous processes .....   | 7  |
| Figure 2.2 Inventory is represented as product accumulation with respect to time .....   | 10 |
| Figure 2.3 Flow chart of closed loop implementation .....  | 16 |
| Figure 2.4 Solution of the CSTR original problem: profile of manipulated variable and state<br>variable, pre-calculated off line .....   | 22 |
| Figure 2.5 Closed loop implementation on CSTR, state deviation occurs in transition from C to B,<br>new solution is generated as transition from C to E .....  | 23 |
| Figure 2.6 Open loop implementation on CSTR, state deviation occurs in transition from C to B,<br>original solution is implemented and product B is not produced.....  | 24 |
| Figure 2.7 Closed loop implementation, state deviation occurs in production period of B (Case 1),<br>new solution is generated as transition from partial B to A.....  | 25 |
| Figure 2.8 Closed loop implementation, state deviation occurs in production period of B (Case 2),<br>B is retrieved right after the deviation .....  | 26 |
| Figure 2.9 Open loop implementation, state deviation occurs in production period of B, original<br>solution is implemented and product B is partially produced.....  | 26 |
| Figure 2.10 Transition profile between products in PFR cyclic production .....   | 31 |
| Figure 2.11 Solution for the PFR original integrated problem, no state deviation .....   | 32 |
| Figure 2.12 Closed Loop Implementation, state deviation occurs at 29 hours in the transition from<br>D to B, scheduling solution and control profile are updated as targeting C right after the deviation<br>.....   | 32 |
| Figure 2.13 enlarged illustration of the state deviation in Figure 2.12 .....  | 32 |
| Figure 2.14 Open Loop Implementation, state deviation occurs at 29 hours in the transition from<br>D to B, scheduling solution and control profile are the same as the original solutions, the<br>production period of B is shorter than the original one in Figure 2.11 ..... | 33 |

|  |    |
|--|----|
| Figure 2.15 An enlarged illustration of the deviation in Figure 2.14, production period of B is shortened due to the negligence of state deviation ..... | 33 |
| Figure 3.1 Decomposition for the solutions of integrated problem of continuous processes .....   | 41 |
| Figure 3.2 Decomposition with variables for the optimization problem of continuous processes   | 41 |
| Figure 3.3 Perturbation on production does not affect transition within one slot .....   | 44 |
| Figure 3.4 Perturbation on production does not affect all the transitions .....  | 46 |
| Figure 3.5 Solutions for integrated problem at different cycles assigned with different demands  | 50 |
| Figure 3.6 MMA polymerization in a CSTR .....  | 53 |
| Figure 4.1 The working mode of conventional MPC .....  | 63 |
| Figure 4.2 The working mode of mp-MPC.....   | 63 |
| Figure 4.3 A scheme of integrated scheduling and control using MPT toolbox.....  | 64 |
| Figure 4.4 Demonstration of the integration using event point based scheduling formulation .....   | 65 |
| Figure 4.5 Discretization with fixed step size.....  | 72 |
| Figure 4.6 Discretization with varying step size but fixed number of steps .....   | 73 |
| Figure 4.7 Flow sheet of a simple batch process .....  | 74 |
| Figure 4.8 State task network of a simple batch process .....  | 74 |
| Figure 4.9 Discretization of step size $h$ .....   | 76 |
| Figure 4.10 Explicit solution for mp-MPC .....   | 77 |
| Figure 4.11 Dynamic profile of the reactor. $x$ represents the concentration of raw material and $u$ represents the scaled temperature .....             | 78 |
| Figure 4.12 Scheduling solution for example 1 a simple batch process .....   | 78 |
| Figure 4.13 Flow sheet for a complex batch process .....   | 79 |
| Figure 4.14 Temperature profile obtained by applying the explicit control solution to the reaction .....   | 82 |
| Figure 4.15 Conversion profile and the approximation (empirical relation) .....  | 82 |
| Figure 4.16 Utility amount profile and the approximation (empirical relation) .....  | 82 |

|   |     |
|---|-----|
| Figure 4.17 Scenario 1, scheduling solution for MINLP. Red number represents the index of task-unit-eventpoint (ijn) and black number represents the (amount of material/conversion/utility consumption) .....  | 85  |
| Figure 4.18 Scenario 2, apply control to pre-obtained scheduling solution. Red number represents the index of task-unit-event point (ijn) and black number represents the (amount of material/conversion/utility consumption) .....   | 86  |
| Figure 4.19 Scenario 3, implicit integration based on the recipe obtained in control problem. Red number represents the index of task-unit-event point (ijn) and black number represents the (amount of material/conversion/utility consumption).....                           | 86  |
| Figure 4.20 Scenario 4, original integrated problem MIDO discretized into a MINLP using implicit RK method, solved using GAMS/DICOPT. Red number represents the index of task-unit-event point (ijn) and black number represents the (amount of material/conversion/utilit..... | 86  |
| Figure 4.21 Scenario 4, original integrated problem MIDO discretized into a MINLP using implicit RK method, solved using GAMS/BARON. Red number represents the index of task-unit-event point (ijn) and black number represents the (amount of material/conversion/utility .... | 86  |
| Figure 5.1 Cascade strategy for the online integration of scheduling and control.....   | 96  |
| Figure 5.2 Example of one-dimension PWA approximation.....  | 99  |
| Figure 5.3 Rectangular partition of the domain of two-dimensional functions .....   | 101 |
| Figure 5.4 The end of the transition locates at the point inside the bounds with the prior one locating outside the bounds.....   | 108 |
| Figure 5.5 CSTR cyclic production, feeding flow rate is the manipulated variable and raw material concentration is the state variable .....   | 116 |
| Figure 5.6 CSTR cyclic production, each slot is composed of a transition period and a production period .....   | 116 |
| Figure 5.7 Profiles of the nonlinear dynamic and the PWA .....  | 117 |

|  |     |
|--|-----|
| Figure 5.8 Transition profile from product 2 to product 5 in SISO CSTR, obtained by fast MPC .....   | 119 |
| Figure 5.9 Transition profile from product 5 to product 4 in SISO CSTR, obtained by fast MPC .....   | 119 |
| Figure 5.10 Disturbance with magnitude -0.04 (less than threshold 0.08) at time 0.7 h (in transition), handled by fast MPC, scheduling solution does not change .....  | 121 |
| Figure 5.11 Disturbance with magnitude -0.09 (great than threshold 0.08) at time 1.2 h (in transition), steady state value (scheduling solution) updated by integrated problem, process goes to product 4 (SS=0.393) instead of product 5 (SS=0.5) ..... | 121 |
| Figure 5.12 Disturbance with magnitude +0.05 (less than threshold 0.08) at time 2.6 h (in production), handled by fast MPC, scheduling solution does not change .....  | 122 |
| Figure 5.13 A MIMO CSTR process .....  | 123 |
| Figure 5.14 Transition profile from steady state 3 to steady state 1 in MIMO CSTR, obtained by fast MPC .....  | 125 |
| Figure 5.15 Transition profile from steady state 1 to steady state 2 in MIMO CSTR, obtained by fast MPC .....  | 126 |
| Figure 5.16 Locate the critical regions and evaluate the explicit solution obtained by mp-MPC  | 127 |
| Figure 5.17 Explicit solution for the control problem in transition from product 3 to 1 of MIMO CSTR case .....  | 129 |
| Figure 6.1 Origin of uncertainty in process operations .....   | 133 |
| Figure 6.2 Performance comparison between robust MPC and conventional MPC .....  | 138 |

## **Chapter 1 Introduction**

### **1.1 Scheduling and control for chemical processes**

Scheduling of process operations is an important activity especially when multiple multi-purpose units and a variety of products are involved in order to achieve the optimal production capacity and thus maximize plant profitability. Generally the solutions of scheduling problems involve the allocation of resources (material and equipments) at appropriate time over a production horizon following a pre-defined recipe for the production of specific products. For a short-term scheduling problem, the solutions are composed of equipment assignment, amount of material processed in units, task sequence and task starting/ending times. During the past years significant efforts have been made in the development of modeling and optimization approaches for batch scheduling problems. Detailed review regarding the problem formulations and optimization methods in batch scheduling can be found in (Floudas and Lin, 2004), (Mendez et al., 2006), (Mouret et al., 2011) and (Maravelias, 2012).

Control problems of continuous processes focus on transition periods between different products, especially for polymerization processes that are usually carried out in continuous reactors such as CSTR and PFR. Mahadevan et al. (Mahadevan et al., 2002) investigated classic control strategies (i.e. based on transfer function, such as PID) for the transition periods in polymer processes. They solved the control problem under different scheduling policies and compared the corresponding solutions. Feather et al. (Feather et al., 2004) built a mixed integer Model Predictive Control for grade transition problems. They introduced integer variables to approximate the nonlinear process, and used heuristics to make selections. Padhiyar et al. (Padhiyar et al., 2006) proposed a differential evolution method to solve the optimization problem of grade transitions. We review control strategies mainly in polymerization processes because most of the work regarding the integration of scheduling and control are in this area and since our case studies are also based on the published case studies.



Unlike the steady state operations in continuous processes, batch processes are operated at transient state. In batch operations the state variables are varying under constraints stemmed from mass and energy conservation law (Sundaram et al., 2005). The task of control problem is to obtain the optimal state profiles by minimizing the operation cost under these constraints (Bonvin, 2006). Thus the objective of batch control problem is different from the one addressing continuous processes. For continuous processes, the units operate at steady state and the optimal control strategies are designed to reduce off-spec products during transitions between different production stages. In contrast, the objective for batch control problem is to improve the productivity during transient periods by generating optimal operating conditions.

## **1.2 The significance of integration of scheduling and control**

A traditional approach to handle production scheduling and dynamic optimization problems in chemical processes is to consider them separately. However, there are variables that link the scheduling and dynamic optimization problems, for example, the states at the beginning and end of the transient periods, duration of the transient periods and steady state periods, and amount of material in production stages. Scheduling and dynamic optimization are naturally connected by the linking variables and thus cannot be considered completely separately. Using an integrated approach, information can be shared between scheduling and dynamic optimization level, leading to the integrated decision making which improves the profitability of process operations (Engell and Harjunkski, 2012) (Harjunkski et al., 2009) (Mitra et al., 2009).

Previous published papers in this area have demonstrated the advantages of integration. Bhatia and Biegler (Bhatia and Biegler, 1996) proposed a framework for the integration of design, scheduling and control of multiproduct batch plants. The authors addressed flowshop scheduling and replaced batch recipes with detailed dynamic models and show the improvement of profitability due to their integration using numerical examples. Mishra et al. (Mishra et al., 2005) compared the performance of standard recipe approach (SRA) with the overall optimization

approach (OOA) in optimizing general batch plants. In SRA the production scheduling problem follows the basis of standardized recipes and is formulated using event-point-based continuous-time formulation proposed in Ierapetritou and Floudas (Ierapetritou and Floudas, 1998), whereas in OOA the dynamic models of the reaction tasks are incorporated into the STN framework. The OOA approach leads to a MIDO problem which is discretized and solved as a MINLP. Their results shown that the OOA outperformed SRA in terms of profitability. Recently Nie et al. (Nie et al., 2012) employed state equipment network (SEN) to represent a process and formulated mixed-logic dynamic optimization (MLDO) for integrated scheduling and control. They demonstrated the advantages of integrated approach over conventional scheduling method through numerical results. Chu and You (Chu and You, 2013b) proposed to use Benders Decomposition to decrease the computation complexity, and two-stage stochastic program (Chu and You, 2013c) to handle the uncertainties in the integration of scheduling and dynamic optimization for batch processes. They also proposed a new online integrated method in which a reduced integrated problem is developed using a rolling horizon approach (Chu and You, 2014b).

### **1.3 Modeling, computation, and challenges of integration of scheduling and control**

Recently a number of publications appear in the literature to address the integration of scheduling and dynamic optimization problems for continuous processes. Two approaches can be identified in this area: simultaneous modeling and decomposition based approach. With simultaneous modeling, the process dynamics is treated as constraints and incorporated into the scheduling problems, forming a integrated problem which is a Mixed Integer Dynamic Optimization (MIDO) problem (Allgor and Barton, 1999). The MIDO can be discretized using collocation point methods and a Mixed Integer Nonlinear Programming (MINLP) is obtained (Flores-Tlacuahuac and Grossmann, 2006) (Terrazas-Moreno et al., 2007) (Terrazas-Moreno et al., 2008b). Using decomposition methods, the scheduling problem (master problem) is modeled as Mixed Integer

Linear Programming and the control problem (primal problem) as dynamic optimization. This approach is implemented by iterating between the master problem and the primal problem until convergence is achieved (Nystrom et al., 2005) (Nystrom et al., 2006).

Integration of scheduling and dynamic optimization results in a large size MINLP which is computational very expensive. In order to achieve better computational performance, (Terrazas-Moreno et al., 2008a) proposed Lagrangean decomposition for the integrated model and solved it with iteration between a primal control problem and a master scheduling problem. (Chu and You, 2013b) and (Chu and You, 2013a) explored the structure of the integrated problem and applied Benders Decomposition to decrease the computational complexity. Their approach effectively lowered the computation time and achieved optimality of the integrated problem.

Challenges of integration of scheduling and control for batch processes mainly lie in modeling and computation. Since scheduling and control levels have different dynamics and different time scales, the integration requires high-fidelity representations of both the scheduling problem and the dynamics of the plant (Engell and Harjunkski, 2012). Moreover, the resulting MIDO is typically nonlinear and nonconvex, and the discretized MINLP involves a large number of variables and constraints and requires excessive computation effort to get solved (Allgor and Barton, 1999) (Flores-Tlacuahuac and Biegler, 2005). Thus the problem complexity limits the scale of the problems that can be considered in online applications.

## **1.4 Motivation and outline of the dissertation**

As described above, challenges of integration of scheduling and control mainly lie in modeling and computation. In this dissertation we intend to develop new integrated models and optimization methods to reduce the complexity brought by integration, so as to facilitate the online implementations.

In Chapter 2 we build a closed loop implementation for simultaneous scheduling and control, which mitigates the effects of disturbances by generating new solutions once the state deviation due to disturbances is detected. In order to test the performance of the proposed methodology, we use two case studies, and compare its performance with that of open loop implementation. The results of the case studies verify the effectiveness of closed loop strategy in dealing with process disturbances.

In Chapter 3 we continue with the integrated model presented in Chapter 2 and propose a decomposition approach that result in the efficient solution of the integrated problem. Sensitivity Analysis is utilized to prove that the production sequence and transition times are independent of products' demands. The proof leads to the decomposition of the integrated problem into two sub problems that can be solved separately without the need for iterations. Results of case studies verify the feasibility and effectiveness of proposed approach in reducing the computational complexity of the integrated problem but also obtaining the optimal solution.

In Chapter 4 and 5 we focus on the online applications. In Chapter 4 we incorporate multiparametric Model Predictive Control (mp-MPC) in the integration of scheduling and control. The proposed methodology involves the development of an integrated model using continuous time event point formulation for the scheduling level and the derived constraints from explicit MPC for the control level. Results of case studies of batch processes prove that the proposed approach guarantees efficient computation and thus facilitates the online implementation. In Chapter 5 we propose a framework for the integration of scheduling and control. We identify a piece-wise affine (PWA) model from the first principle model and integrate it with the scheduling level leading to a new integration that has linear constraints. At the control level we employ fast Model Predictive Control (fast MPC) which overcomes the increasing dimensionality of multiparametric Model Predictive Control (mp-MPC) in our previous study. (Zhuge and Ierapetritou,

2014) Results of CSTR case studies prove that the proposed approach reduces the computing time by at least two orders of magnitude compared to the integrated solution using mp-MPC.

In Chapter 6 we discuss the uncertainty in process operations and present solution procedures of robust MPC for nonlinear problem at the control level.

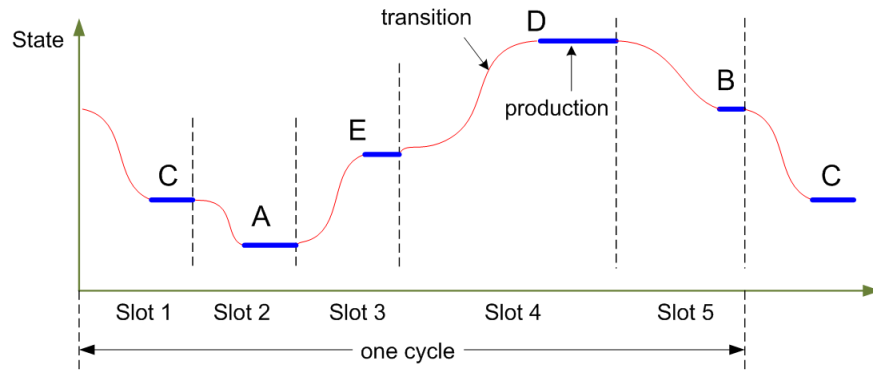
Finally in Chapter 7 we summarize major conclusions and discuss the future perspectives in the area of integration of scheduling and control for chemical processes.

## Chapter 2 A closed loop implementation of integrated scheduling and control

### 2.1 Integrated model for cyclic scheduling and control for continuous processes

#### 2.1.1 General integrated model for scheduling and control

In this section we present the general scheduling and dynamic optimization models for continuous cyclic productions, and explain how the scheduling and dynamic optimization problems are integrated.



**Figure 2.1 Cyclic production of continuous processes**

In continuous processes cyclic production, slot based formulation is adopted (Flores-Tlacuahuac and Grossmann, 2006) and in each slot transition period (dynamic profile) is followed by production period (steady state profile) (Figure 2.1). The optimization problem at scheduling level determines the production time and production sequence in order to satisfy demands. During transitions between steady state production stages optimal control (i.e. dynamic optimization) are needed which is obtained from the solution of problem (2.1) in which state variables govern the system dynamics at slot  $k$ .

$$\begin{aligned} & \max_{u_k} J_k(x_k, u_k, q_k) \\ & s.t. \begin{cases} \dot{x}_k = f_k(x_k, u_k) \\ q_k = h_k(x_k, u_k) \\ (x_k, u_k, q_k) \in \mathbb{W}_{xuq} \end{cases} \end{aligned} \quad (2.1)$$

In problem (2.1)  $J_k$  is the local profit, and  $u_k$  and  $q_k$  are the manipulated variables and outputs, respectively, whereas  $\Omega_{xuq}$  is the domain for  $x_k, u_k, q_k$ . With simultaneous modeling, the process dynamics (differential equations derived from mass and energy balance) are considered as constraints and incorporated into the scheduling problem, resulting in the integrated model in formula (2.2). More specifically,  $\dot{x}_k = f_k(x_k, u_k)$  are the differential equations describing the process dynamics which can be solved under the initial value of state and manipulated variables  $x_{k0}$  and  $u_{k0}$ , leading to  $x_k = ff_k(f_k, u_k, x_{k0}, u_{k0})$ . Here  $ff_k$  is the solution for  $x_k$ . It is determined by the form of  $f_k$  and the manipulated variable  $u_k$  and the initial conditions  $x_{k0}$  and  $u_{k0}$ .  $ff_k$  is not necessarily in an explicit form. Note that  $x_{k0}$  and  $u_{k0}$  are equal to the steady state values of the prior time slot, which are relevant to the scheduling solution  $\mathcal{Y}$ . In continuous processes because the transition between products is followed by the steady state period, the sequence of products determines the set point of the transition in a certain slot. Thus we obtain  $x_k = ff_k(f_k, u_k, x_{k0}(\mathcal{Y}), u_{k0}(\mathcal{Y}))$  which is integrated with the scheduling model as shown in problem (2.2).

$$\begin{aligned} & \max_{w, y, u} J(w, y, x, u, q) \\ & s.t. \begin{cases} g_i(w, y) \geq demand(i) \quad \forall i \in I \\ x_k = ff_k(f_k, u_k, x_{k0}(y), u_{k0}(y)) \quad \forall k \in K \\ q_k = h_k(x_k, u_k) \quad \forall k \in K \\ (x_k, q_k, u_k, w) \in \Omega_{xquw} \quad \forall k \in K \end{cases} \end{aligned} \quad (2.2)$$

where  $I$  is the product set and  $K$  is the slot set.  $\mathcal{Y}$  is a vector of binary variables representing assignment of products in slots;  $w$  is the vector of production times which are decision variables

in scheduling problem; and  $\Omega_{xquw}$  is the domain for  $w$ , the state variables  $x$ , the manipulated variables  $u$ , and the output variables  $q$  defined in problem (2.1). In this study infinite amounts of raw material and resources are considered. In the scheduling problem demands should be satisfied by the production amount  $g_i$ . The objective  $J(w, y, x, u, q)$  represents the profit in unit time or in general the total profit in a pre-determined time period.

To achieve economically optimal operations of chemical processes, we utilize the objective of maximizing profit per unit time, which can be calculated as follows:

Profit per unit time = (Revenue – Inventory cost – Raw material cost)/Cycle time

$$\Phi = \Phi_1 - \Phi_2 - \Phi_3 \quad (2.3)$$

$$\Phi_1 = \sum_{i=1}^{N_p} \frac{C_i^p W_i}{T_c} \quad (2.4)$$

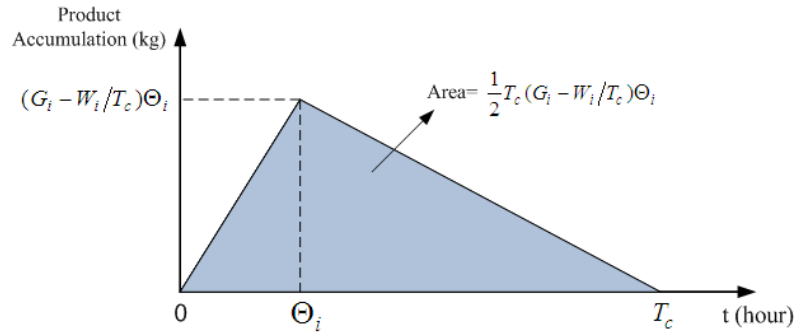
$$\Phi_2 = \sum_{i=1}^{N_p} \frac{1}{2} C_i^s \left( G_i - \frac{W_i}{T_c} \right) \Theta_i \quad (2.5)$$

$$\Phi_3 = \left( \sum_{k=1}^{N_s} \sum_{e=1}^{N_e} C^r \left( u_{ke}^1 + \dots + u_{ke}^m \right) h_k \theta_k^t + \sum_{k=1}^{N_s} C^r \left( u_k^{-1} + \dots + u_k^{-m} \right) p_k \right) / T_c \quad (2.6)$$

We consider a comprehensive economic objective incorporating market economics which we believe are very important in making scheduling decisions.  $\Phi_1, \Phi_2, \Phi_3$  represent the revenue rate, inventory cost rate, and raw material cost rate, respectively. The total revenue is given as the amount produced ( $W_i$ ) times the product price ( $C_i^p$ ). The accumulation of product  $i$  with respect to time is shown in Figure 2.2. When  $t \leq \Theta_i$  the accumulation rate is given by the production rate  $G_i$  minus the depletion rate  $W_i/T_c$ , and at  $t = \Theta_i$  (i.e. the end of production period) the accumulation reaches its peak. At  $t \geq \Theta_i$  the accumulation decreases due to pure product



depletion. The total inventory cost of product  $i$  is thus calculated as the inventory cost  $C_i^s$  times the area of the triangle i.e.  $\frac{1}{2}T_c(G_i - w_i/T_c)\Theta_i$ . Unlike the objective in Flores-Tlacuahuac and Grossmann (Flores-Tlacuahuac and Grossmann, 2006), we do not incorporate state variation in  $\Phi_3$ . The main reason for this is that it is difficult to quantify state variation economically. For continuous processes, raw material is consumed both in transition periods and production periods. Thus  $\Phi_3$  is composed of two terms which quantify the raw material consumption in these two periods. It should be noticed that the objective in the model of (Terrazas-Moreno et al., 2007) quantifies raw material consumption during product transitions as well but with a different manner compared to our model.



**Figure 2.2 Inventory is represented as product accumulation with respect to time**

### 2.1.2 Scheduling constraints

#### Product assignment

$$\sum_{k=1}^{N_s} y_{ik} = 1, \quad \forall i \in I \quad (2.7)$$

$$\sum_{i=1}^{N_p} y_{ik} = 1, \quad \forall k \in K \quad (2.8)$$

$$y'_{ik} = y_{i,k-1}, \quad \forall i \in I, k \in K, k \neq 1 \quad (2.9)$$

$$y'_{i1} = y_{i,N_s}, \quad \forall i \in I \quad (2.10)$$

$$z_{ipk} \geq y'_{ik} + y_{pk} - 1, \quad \forall i \in I, p \in I, k \in K \quad (2.11)$$

$$z_{ipk} \leq y'_{ik}, \quad \forall i \in I, p \in I, k \in K \quad (2.12)$$

$$z_{ipk} \leq y_{pk}, \quad \forall i \in I, p \in I, k \in K \quad (2.13)$$

where binary variables  $y_{ik}$  indicate the assignment of products in slots,  $y_{ik} = 1$  if product  $i$  is produced in slot  $k$ , otherwise  $y_{ik} = 0$ ;  $y'_{ik}$  are auxiliary variables. Constraints (2.7) enforces that each product can only be produced in one slot and (2.8) implies that only one product is produced in one slot. In constraints (2.11)-(2.13) if binary  $z_{ipk} = 1$ , there is a transition from product  $i$  to product  $p$  within slot  $k$ ; otherwise  $z_{ipk} = 0$ . In the case studies of this work it is assumed that  $N_s = N_p$ .

It should be noticed that the production sequences that have the same relative positions in the production wheel are equivalent (i.e. cyclic feature). For example, sequences CADEB, ADEBC, DEBCA, EBCAD, BCADE are equivalent solutions in terms of producing the same transition profiles and the same objective values.

#### Demand constraints

$$W_i \geq D_i T_c, \quad \forall i \in I \quad (2.14)$$

$$W_i = G_i \Theta_i, \quad \forall i \in I \quad (2.15)$$

$$G_i = F^0 (1 - X_i), \quad \forall i \in I \quad (2.16)$$

Inequality (2.14) implies that the total production amount of each product  $w_i$ , which is calculated by Equation (2.15), should be greater than the demand  $D_i$  in current production cycle. Equation (2.16) computers the production rate as a product of feed flow rate  $F^0$  and conversion  $X_i$ .

### Processing times

$$\theta_{ik} \leq \theta^{\max} y_{ik}, \forall i \in I, k \in K \quad (2.17)$$

$$\Theta_i = \sum_{k=1}^{N_s} \theta_{ik}, \forall i \in I \quad (2.18)$$

$$p_k = \sum_{i=1}^{N_p} \theta_{ik}, \forall k \in K \quad (2.19)$$

Inequality (2.17) sets up an upper bound for the time allowed for producing product  $i$  in slot  $k$ . In Equations (2.18) and (2.19),  $\Theta_i$ ,  $p_k$  define the production time of product  $i$  and slot  $k$ , respectively.

### Timing constraints

$$\theta_k^t = \sum_{i=1}^{N_p} \sum_{p=1}^{N_p} \theta_{pi}^t z_{ipk}, \forall k \in K \quad (2.20)$$

$$t_1^s = 0 \quad (2.21)$$

$$t_k^e = t_k^s + p_k + \sum_{i=1}^{N_p} \sum_{p=1}^{N_p} \theta_{pi}^t z_{ipk}, \forall k \in K \quad (2.22)$$

$$t_k^s = t_{k-1}^e, \forall k \in K, k \neq 1 \quad (2.23)$$

$$t_k^e \leq T_c, \forall k \in K \quad (2.24)$$

Equation (2.20) computes the time of transition  $\theta_k^t$  from product  $i$  to  $p$  in slot  $k$ . The starting  $t_k^s$  and the ending time point  $t_k^e$  for each slot are presented by Equations (2.22) and (2.23). The starting point is equal to the ending point of the prior slot. The ending point is calculated as the

sum of the starting point, the transition time, and the production time. As shown in inequality (2.24) all slots should finish in the cyclic time.

### 2.1.3 Dynamic optimization problem

In discretizing the dynamic model, we adopt implicit Runge-Kutta methods due to their high stability (Frank et al., 1985). A general form of implicit RK methods has Butcher tableau as following.

$$\begin{pmatrix} c_1 & a_{11} & a_{12} & \cdots & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \cdots & a_{ss} \\ & b_1 & b_2 & \cdots & b_s \end{pmatrix} = \begin{pmatrix} \mathbf{c} & \mathbf{A} \\ & \mathbf{b}^T \end{pmatrix} \quad (2.25)$$

Here we use Hammer-Hollingsworth with Butcher tableau as follows:

$$\begin{pmatrix} \frac{3-\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{3+\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\ & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \quad (2.26)$$

Thus we obtain the following discretization of the dynamic model:

$$\dot{x}_{ke}^n = f^n(t_{ke}, x_{ke}^1, \dots, x_{ke}^n, u_{ke}^1, \dots, u_{ke}^m), \forall n \in S_x, k \in K, e \in E \quad (2.27)$$

$$K1_{ke}^n = f^n(t_{ke} + 0.2113h_k, x_{ke}^n + h_k(0.25K1_{ke}^n - 0.0387K2_{ke}^n), u_{ke}^1, \dots, u_{ke}^m), \quad (2.28)$$

$$\forall n \in S_x, k \in K, e \in E$$

$$K2_{ke}^n = f^n(t_{ke} + 0.7887h_k, x_{ke}^n + h_k(0.5387K1_{ke}^n + 0.25K2_{ke}^n), u_{ke}^1, \dots, u_{ke}^m), \quad (2.29)$$

$$\forall n \in S_x, k \in K, e \in E$$

$$h_k = \frac{\theta_k^t}{N_e}, \forall k \in K \quad (2.30)$$

$$x_{k,e+1}^n = x_{ke}^n + h_k(0.5K1_{ke}^n + 0.5K2_{ke}^n), \forall n \in S_x, k \in K, e \in E \quad (2.31)$$

The derivatives of the state variables at each sample step are calculated using Equation (2.27) where  $f$  represents the detailed dynamic model; and sub-index  $e$  represent elements in each slot. Through the computing of intermediate variables  $K1, K2$  in Equations (2.28) and (2.29), the state of the next sample step is obtained by Equation (2.31). Equation (2.30) defines the sample size(length of each element) in slot  $k$ .

#### Lower and upper bounds of states and manipulated variables

$$x_{\min}^n \leq x_{ke}^n \leq x_{\max}^n, \quad \forall n \in S_x, k \in K, e \in E \quad (2.32)$$

$$u_{\min}^m \leq u_{ke}^m \leq u_{\max}^m, \quad \forall m \in S_u, k \in K, e \in E \quad (2.33)$$

Inequality (2.32) and (2.33) provide lower and upper bounds for the state and manipulated variables at each sample point.

#### Constraints that link scheduling and control level

$$x_{in,k}^n = \sum_{i=1}^{N_p} x_{ss,i}^n y_{i,k}, \quad \forall n \in S_x, k \in K \quad (2.34)$$

$$\bar{x}_k^n = \sum_{i=1}^{N_p} x_{ss,i}^n y_{i,k+1}, \quad \forall n \in S_x, k \in K, k \neq N_s \quad (2.35)$$

$$\bar{x}_{N_s}^n = \sum_{i=1}^{N_p} x_{ss,i}^n y_{i,1}, \quad \forall n \in S_x \quad (2.36)$$

$$u_{in,k}^m = \sum_{i=1}^{N_p} u_{ss,i}^m y_{i,k}, \quad \forall m \in S_u, k \in K \quad (2.37)$$

$$\bar{u}_k^m = \sum_{i=1}^{N_p} u_{ss,i}^m y_{i,k+1}, \quad \forall m \in S_u, k \in K, k \neq N_s - 1 \quad (2.38)$$

$$\bar{u}_{N_s}^m = \sum_{i=1}^{N_p} u_{ss,i}^m y_{i,1}, \quad \forall m \in S_u \quad (2.39)$$

$$x_{k,1}^n = x_{in,k}^n, \quad \forall k \in K \quad (2.40)$$

$$u_{k,1}^m = u_{in,k}^m, \quad \forall k \in K \quad (2.41)$$

$$x_{k,N_e}^n = \bar{x}_k^n, \quad \forall k \in K \quad (2.42)$$

The steady state values for each slot  $x_{ss,i}^n, u_{ss,i}^m$  are computed beforehand by simulating the process at steady state. Equations (2.35) (2.36) and (2.38) (2.39) calculate the desired state and manipulated variables at each slot. Equations (2.34) and (2.37) provide the initial values of state and manipulated variables in slot  $k$ . (2.40) and (2.41) link the initial values with the state and control value of the first element in the transitions. (2.42) enforces that the state values at the end of transition are equal to the desired values (steady state values) in the current slot. Constraints (2.34)-(2.41) demonstrate how scheduling and control are linked. For example, sequence  $y$  in (2.34) and dynamic in (2.31) are linked through (2.40).

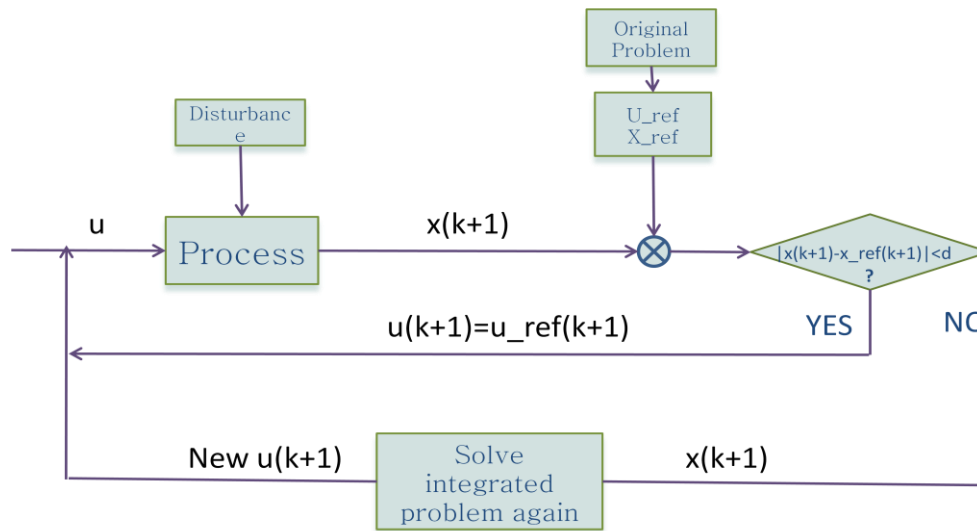
## 2.2 The closed loop implementation

In real applications, processes are subject to disturbances. There are a few disturbances that CSTRs or PFRs may be subject to, such as disturbances on feed flow rate, feed flow composition, and feed flow temperature. Even though the disturbances on these variables can be handled by local controllers, their fluctuations cause a joint effect on state variables (concentration), which may degrade products quality. In most cases this joint effect is difficult to track analytically. Therefore it is easier to measure the state variables (concentration) and update the control profile (feed flow rate) if concentration exceeds the quality bounds.

We propose a closed loop strategy for implementing the integration of scheduling and control in chemical processes whose state variables are subject to disturbances. State variables are deviated because of disturbances. The deviation is a joint result of the disturbances on the variables mentioned previously. To make it more applicable to real processes, we assume disturbances can occur both in transition and production periods. In real operations, the state is monitored and

state's deviation from reference is measured at every step. A threshold (also defined as quality bounds) is set to determine whether the manipulated variables for the next step remain the same as the reference or should be updated using the solution of the integrated problem for the remaining time slots. The only limitation is that the time needed for solving the integrated problem should be less than the sample step.

More specifically, as shown in Figure 2.3, we first solve the integrated problem off-line and obtain the scheduling solution and control input as reference. Then the solution is implemented in the process. If the state deviation from the reference is less than the threshold (i.e. within the quality bounds), no update is needed. If it is greater, which means off-spec product is produced, the integrated problem is solved again for the remaining part of the production cycle based on the current state information. A new solution for the integrated problem is generated at the point of significant deviation. Thus both the scheduling solution and control input are updated, which ensures that the operations thereafter are optimal.



**Figure 2.3 Flow chart of closed loop implementation**

The main steps of the proposed approach are as follows:

Assuming that the wheel time is divided into  $N_{spl}$  sample points having the same interval length so that each time point  $n$  satisfies the following constraint:  $1 \leq n \leq N_{spl}$ .

Step 0: Initialization:  $n=1$ .

Step 1: Solve original problem, obtain solution of scheduling and control as reference

Step 2: At time point  $n$ . Implement the solution, detect state deviation from reference

Step 3: Compare the deviation to the threshold

Step 4: If deviation is smaller than threshold, go to step 6

Step 5: If deviation is greater than threshold, re-solve the integrated problem, generate new solution of scheduling and control, and go to step 6

Step 6:  $n=n+1$ , if  $1 \leq n \leq N_{spl}$ , go to step 2; otherwise stop.

## 2.3 Stability and robustness of closed loop implementation

The proposed strategy in section 2.2 is implemented in a closed loop manner. For every sample step state information is feedback and new solutions are generated if its deviation exceeds a predefined threshold. The new solutions however are obtained by optimizing the cost function on the remaining slots in the production cycle. In a sense, this implementation is equivalent to constrained Nonlinear Model Predictive Control with varying prediction horizon. Due to the presence of disturbance and model mismatching, stability and robustness are investigated in this study. The nonlinearity in the prediction model necessitates the use of Lyapunov stability theory (Mayne et al., 2000).

### 2.3.1 Proof of the asymptotic stability of the closed loop system

Suppose a local Lyapunov function  $F$  can be constructed to satisfy the following constraint:

Assumption 1:

$$F(f(x, u(x))) - F(x) + l(x, u(x)) \leq 0 \quad \forall x \in \Omega_{x_{ss}} \quad (2.43)$$



where  $f(\cdot, \cdot)$  governs the state dynamic i.e.  $x_{k+1} = f(x_k, u_k)$ ,  $l(\cdot, \cdot)$  is the cost of current step, and  $\Omega_{x_{ss}}$  is the set of steady state value of state  $x$ .

Under the principle of optimality, we have  $J_1(x) = l(x, u^*(x)) + J_0(f(x, u^*(x)))$ , where  $J_i(x)$  is the optimal objective value at state  $x$ , and  $i$  is the prediction horizon. Since  $u^*(x)$  is the optimal control solution,  $l(x, u^*(x)) + J_0(f(x, u^*(x))) \leq l(x, u(x)) + J_0(f(x, u(x)))$  in which  $J_0(f(x, u(x)))$  can be substituted with  $F(f(x, u(x)))$  based on the definition of  $J_0$  and  $F$ .

Referring to Assumption 1, we obtain  $l(x, u(x)) + F(f(x, u(x))) \leq F(x) = J_0(x)$ . Hence,

$$J_1(x) \leq J_0(x) \quad \forall x \in \Omega_{x_{ss}} \quad (2.44)$$

Thus if we prove that  $J_1(x) \leq J_0(x) \quad \forall x \in \Omega_{x_{ss}}$ , then  $J_{i+1}(x) \leq J_i(x) \quad \forall x \in \Omega_{x_{ss}} \quad \forall i \geq 0$ .

Assuming monotonicity of  $J_i(x)$  we have:  $J_i(x) \leq J_{i-1}(x) \quad \forall x \in \Omega_{x_{ss}}$ , then

$$\begin{aligned} J_{i+1}(x) - J_i(x) &= l(x, u(x)) + J_i(f(x, u(x))) \\ &\quad - l(x, u(x)) - J_{i-1}(f(x, u(x))) \\ &= J_i(f(x, u(x))) - J_{i-1}(f(x, u(x))) \\ &\leq 0 \end{aligned} \quad (2.45)$$

To establish the monotonicity of  $J_i(x)$  is necessary to construct a local Lyapunov function  $F$  such that assumption 1 is satisfied.

We define  $F$  as the terminal cost so we have the following objective function

$$J(x, u) = \sum_{i=0}^{N-1} l(x(i), u(i)) + F(x(N)) \quad (2.46)$$

In the simultaneous scheduling and control problem, from the MPC point of view, the final stage in the prediction horizon is the production stage, i.e. steady state is achieved. Thus we have:

$$f(x, u(x)) = x \quad (2.47)$$

So  $F(f(x, u(x))) - F(x) = 0$  always hold and we only need to prove that

$l(x, u(x)) \leq 0 \quad \forall x \in \Omega_{x_{ss}}$  which means that the cost of each prediction step in the final

production stage is negative. This is satisfied in principle, because the process is profitable at the whole production stage, which implies that the cost is negative. Thus assumption 1 is always satisfied given any expression of  $F(\cdot)$ , which eliminates the necessity of constructing an explicit

$F(\cdot)$ .

Based on the above argument, monotonicity of  $J_i(x)$  is proved and, hence, asymptotic stability of the close loop system is established.

### 2.3.2 Robustness analysis of the integrated scheduling and control

Suppose the real system is  $x_{k+1} = f_r(x_k, u_k, w_k)$ , the model is  $f(\cdot, \cdot)$ , due to modeling errors,

$$f(\cdot, \cdot) \neq f_r(\cdot, \cdot, \cdot).$$

From the proof of stability, we have

$$J_N(x, u(x)) \leq J_{N-1}(x, u(x)) \quad (2.48)$$

Due to the optimality we have:

$$J_{N-1}(f(x, u(x))) = J_N(x) - l(x, u(x)) < J_N(x) \quad (2.49)$$

Hence we obtain monotonicity:

$$J_N(f(x, u(x))) < J_N(x) \quad (2.50)$$

If monotonicity holds, closed-loop stability is guaranteed (De Nicolao et al., 1999).

In order to guarantee closed-loop stability with the presence of model mismatch, we must have

$$J(f_r(x, u(x)), N) < J(x, N) \quad (2.51)$$

Combined with the above, we obtain

$$J(f_r(x, u(x)), N) - J(f(x, u(x)), N) \leq l(x, u(x)) \quad (2.52)$$

Since  $l(x, u(x)) > 0$ , the above inequity guarantees that closed-loop stability is preserved if model mismatch is small enough such that the difference between real cost and nominal cost is less than the cost of current step.

## 2.4 Case studies

In order to test the effectiveness of closed loop implementation and compare with the open loop implementation, we applied our approach in two case studies (multiproduct CSTR and multiproduct PFR) and made quantitative comparisons between open and closed loop strategy. Both cases address cyclic production in which five products are produced.

### 2.4.1 Cyclic production in a multiproduct CSTR

The recipe and data set of this case study are imported from Flores-Tlacuahuac and Grossmann (Flores-Tlacuahuac and Grossmann, 2006). Reaction  $3R \xrightarrow{k} P$  takes place in an isothermal CSTR with reaction rate  $-r_R = kC_R^3$ . Five products, A, B, C, D, and E, are manufactured in a cyclic mode, which is shown in Figure 2.1. The following dynamic model is obtained applying mass balance in the reactor.

$$\frac{dC_R}{dt} = \frac{Q}{V}(C_0 - C_R) + r_R \quad (2.53)$$

where  $C_0$  is feed stream concentration,  $Q$  is the feed flow rate (i.e. manipulated variable), and  $C_R$  is the concentration of the outflow (i.e. state variable). The following values of design and kinetic parameters are assumed:  $C_0 = 1 \text{ mol/L}$ ,  $V = 5000L$ ,  $k = 2L^2/(mol^2h)$ . A constraint on manipulated variable (i.e. feed flow rate) is enabled as  $10L/h \leq Q \leq 3000L/h$ . Market information is provided in Table 2.1. The values of the steady state conditions (i.e. the specific qualities of products) of each product in Table 2.1 are calculated beforehand.

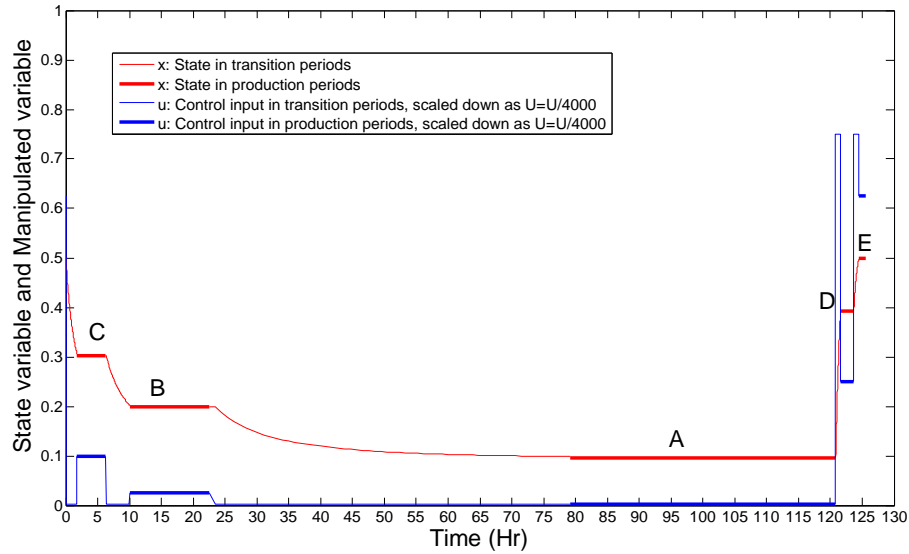
**Table 2.1 Five products with steady state information and market information with CSTR**

| Product | $Q(L/hour)$ | $C_R(mol/L)$ | Demand rate<br>( $kg/hour$ ) | Product price<br>( $\$/kg$ ) | Inventory cost ( $\$/kg$ ) |
|---------|-------------|--------------|------------------------------|------------------------------|----------------------------|
| A       | 10          | 0.0967       | 3                            | 200                          | 1                          |

|   |      |        |    |     |     |
|---|------|--------|----|-----|-----|
| B | 100  | 0.2    | 8  | 150 | 1.5 |
| C | 400  | 0.3032 | 10 | 130 | 1.8 |
| D | 1000 | 0.393  | 10 | 125 | 2   |
| E | 2500 | 0.5    | 10 | 120 | 1.7 |

The objective in this case is to maximize profit per hour which is expressed in Equation (2.3). Decision variables consist of sequence of production, production time, amount manufactured of each product, transition time and feed flow rate (i.e. manipulated variables) in transition periods. They are determined simultaneously by solving the integrated optimization problem. The dynamic model in Equation (2.53) is a nonlinear differential equation. It produces nonlinear equality constraints when discretized using implicit Runge-Kutta method. We formulated a MINLP on the basis of the discretized model in section 2.1.3. To find an appropriate number of elements in each slot, we divided the transition period using different number of elements and found that 60 elements produced an acceptable tradeoff between computational complexity and accuracy. The integrated problem has 2991 variables and 4055 constraints. It takes approximately 10 seconds to solve using GAMS/SBB on an 8 CPUs server. The transition periods are discretized with 60 points and the minimum time step is 18 seconds, which ensures that the computation is completed during each step. Because the model is non-convex, global optimum is not guaranteed. We also investigated the use of GAMS/BARON but the solution time was more than ten minutes which is unacceptable since it exceeds the time step in the real process. However, it should be noticed that the solution found by BARON is the same as that of SBB.

The solution for the original problem is demonstrated in Figure 4 and Table 2. Each slot has different length, indicating different processing time for each product. This figure also shows the state and control variable profiles during the transition and production stages. The state profile is obtained as reference.



**Figure 2.4 Solution of the CSTR original problem: profile of manipulated variable and state variable, pre-calculated off line**

**Table 2.2 Simultaneous scheduling and control results for the original CSTR integrated problem. no state deviation**

| Slot | Products | Transition Time (h) | Production Time (h) | Amount Produced (kg) |
|------|----------|---------------------|---------------------|----------------------|
| 1    | C        | 1.77                | 4.50                | 1254.89              |
| 2    | B        | 3.74                | 12.55               | 1003.91              |
| 3    | A        | 56.58               | 41.68               | 376.47               |
| 4    | D        | 0.77                | 2.07                | 1256.89              |
| 5    | E        | 0.83                | 1.00                | 1251.47              |

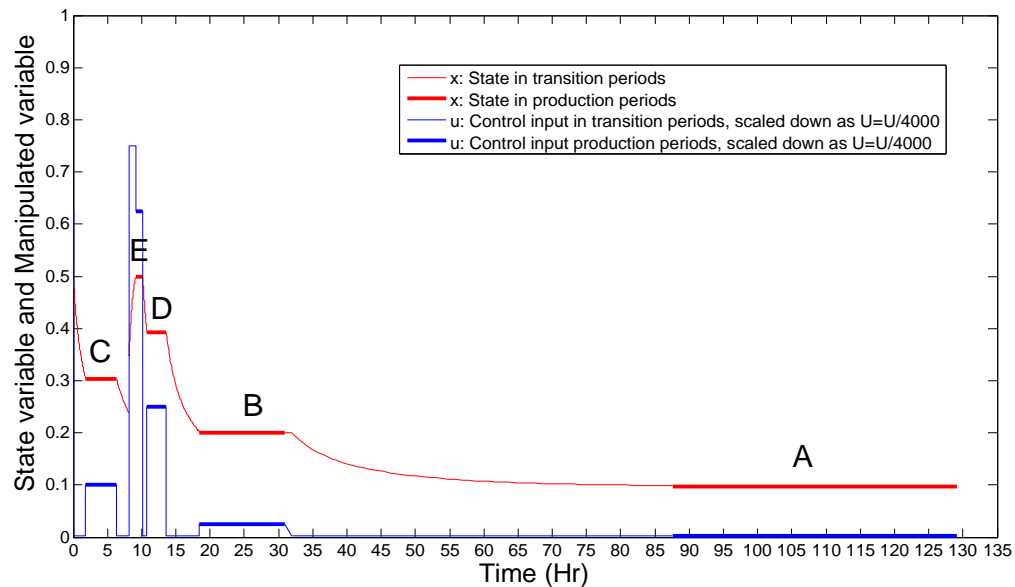
#### Quantitatively comparison of open loop strategy and closed loop strategy

In order to compare the performance of closed loop implementation and open loop implementation, we introduce three different state deviations both in transition and production periods in the original solution shown in Figure 2.4. The first state deviation from 0.25 mol/L to 0.35 mol/L occurs at 8 hours in the transition from product C to B, and the second from 0.2 mol/L to 0.15 mol/L occurs at 15 hours in the production period of B, and the third from 0.2 mol/L to 0.22 mol/L occurs at 15 hours in the production period of B. The threshold (i.e. quality bounds)

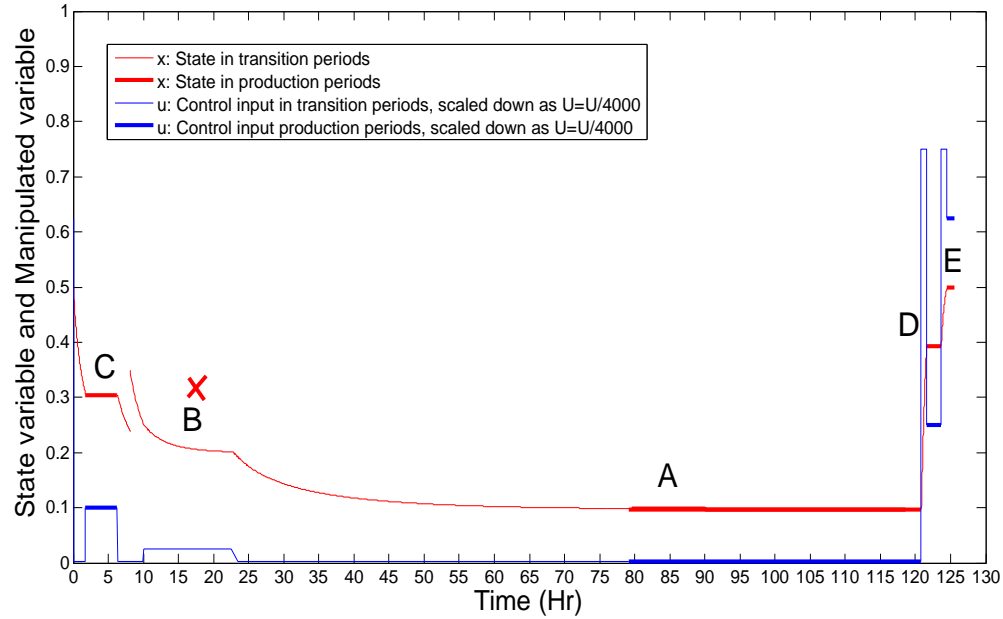
for state deviation is 0.01 mol/L in this case. That means a deviation greater than 0.01 mol/L is significant and closed loop implementation is required to deal with it. Applying these two strategies in the presence of significant state deviations, the process behaves differently and generates different objective values as described in the following.

#### State deviation occurs in transition periods

Following an open loop strategy in which the pre-calculated solution of scheduling and control is implemented during the whole process no matter if there is state deviation or not, when state deviation from 0.25 mol/L to 0.35 mol/L occurs at 8 hours, the current state information is not feedback to the controller, and control inputs remain the same as that of pre-calculated (Figure 2.6 and Table 2.4). However, the closed loop strategy reacts instantly to the deviation so at 8 hours, the process produces E instead of B (Figure 2.5 and Table 2.3).



**Figure 2.5 Closed loop implementation on CSTR, state deviation occurs in transition from C to B, new solution is generated as transition from C to E**



**Figure 2.6 Open loop implementation on CSTR, state deviation occurs in transition from C to B, original solution is implemented and product B is not produced**

**Table 2.3 Simultaneous scheduling and control results for closed loop implementation on CSTR, state deviation occurs in transition from C to B**

| Slot | Products | Transition Time | Production Time | Amount Produced |
|------|----------|-----------------|-----------------|-----------------|
| 1    | C        | 1.77            | 4.50            | 1254.89         |
| 2    | E        | 3.50            | 1.00            | 1251.47         |
| 3    | D        | 0.63            | 2.06            | 1251.47         |
| 4    | B        | 4.90            | 12.51           | 1001.18         |
| 5    | A        | 56.58           | 41.56           | 375.44          |

**Table 2.4 Simultaneous scheduling and control results for open loop implementation on CSTR, state deviation occurs in transition from C to B**

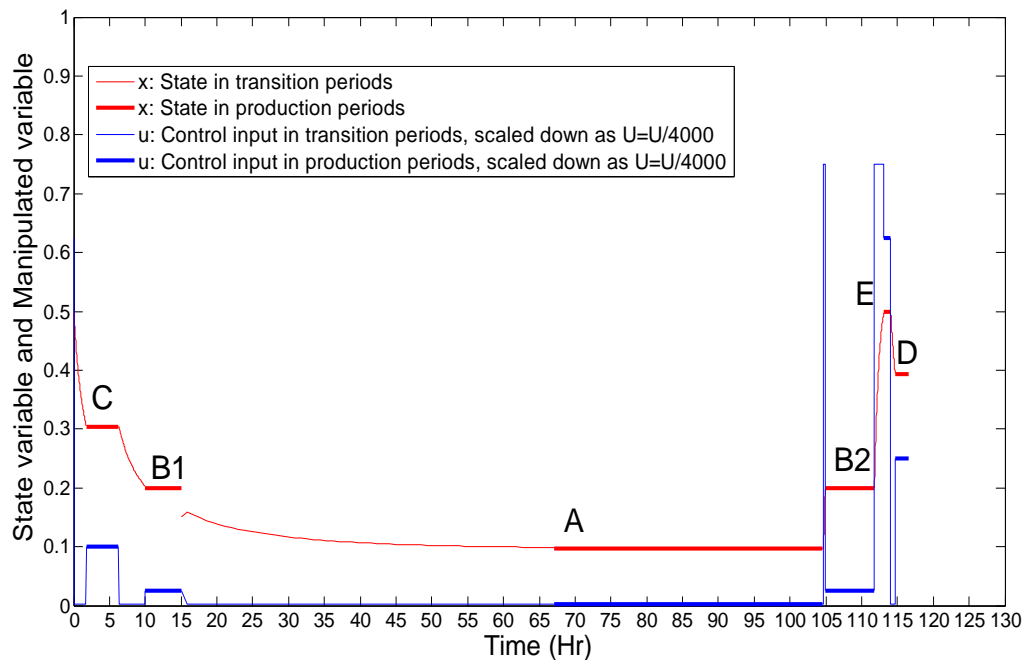
| Slot | Products | Transition Time | Production Time | Amount Produced |
|------|----------|-----------------|-----------------|-----------------|
| 1    | C        | 1.77            | 4.50            | 1254.89         |
| 2    | --       | 16.29           | 0               | 0               |
| 3    | A        | 56.58           | 41.68           | 376.47          |
| 4    | D        | 0.77            | 2.07            | 1256.89         |
| 5    | E        | 0.83            | 1.00            | 1251.47         |

**Table 2.5 Quantitatively comparison of open loop strategy and closed loop strategy on CSTR, state deviation occurs in transition from C to B**

| Solutions                          | No Deviation<br>Open loop strategy<br>(Figure 2.4) | With Deviation<br>Closed loop<br>strategy<br>(Figure 2.5) | With Deviation<br>Open loop<br>strategy<br>(Figure 2.6) |
|------------------------------------|--|---|---|
| Scheduling Solution                | C-B-A-D-E  | C-E-D-B-A   | C- $\times$ -A-D-E                                      |
| Cycle Time (Hr)                    | 125.489  | 129.012   | 125.489   |
| Revenue per Hour (\$)              | 5548.724   | 5387.180  | 4348.727  |
| Cost of Inventory per Hour (\$)    | 4184.573   | 4186.109  | 3536.573  |
| Cost of Raw Material per Hour (\$) | 1082.101   | 1083.134  | 1082.101  |
| Profit per Hour (\$)               | 282.04   | 117.937   | -269.947  |

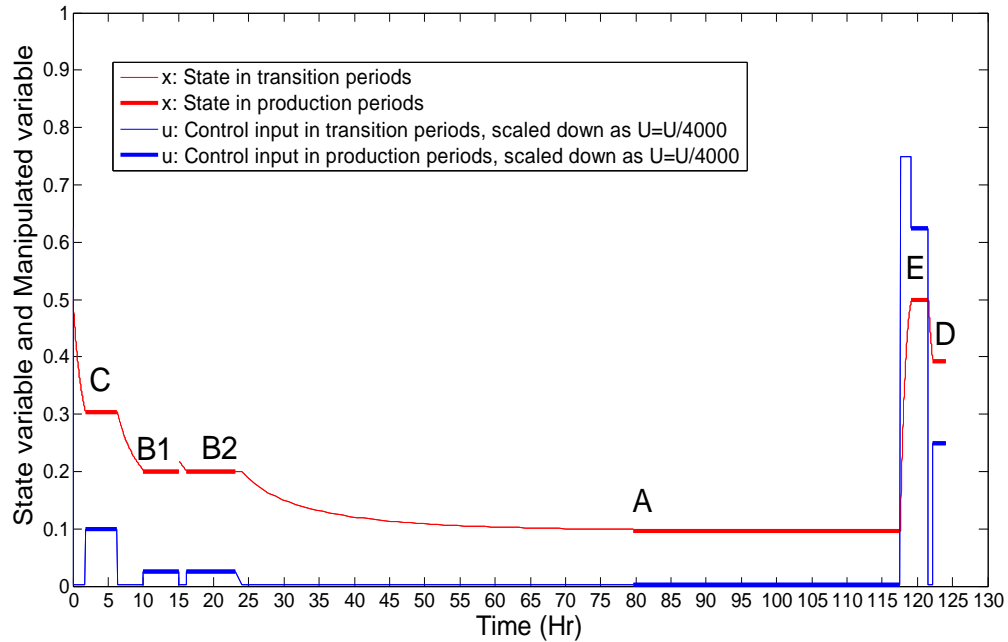
As shown in Table 2.5, the profit per hour obtained using open loop implementation is negative (i.e. -269.947) because product B is not produced (Figure 2.6). However, closed loop implementation guarantees a positive profit per hour (i.e. 117.937).

State deviation occurs in production periods

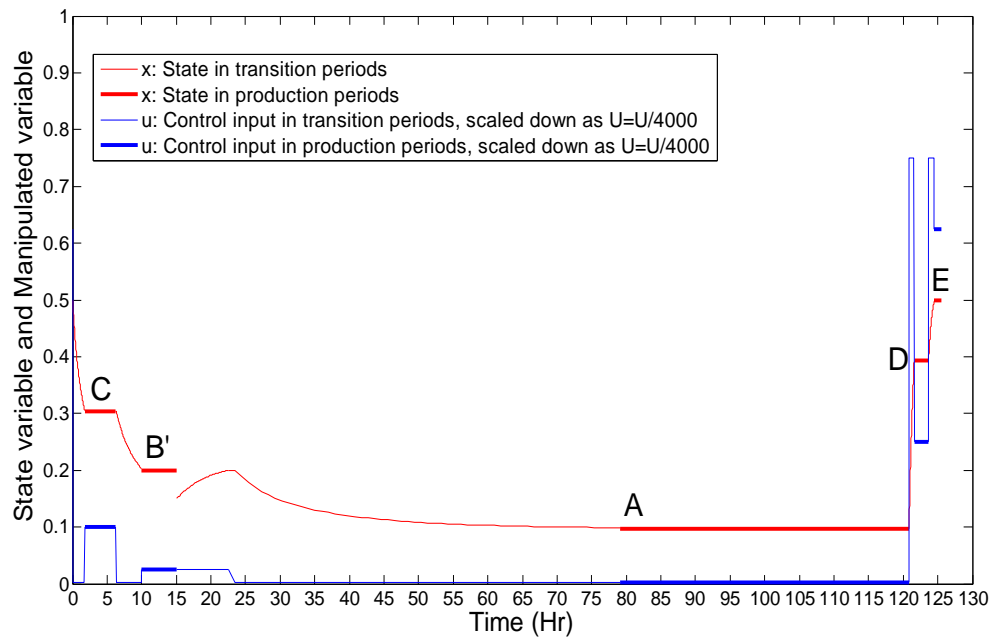


**Figure 2.7 Closed loop implementation, state deviation occurs in production period of B (Case 1), new solution is generated as transition from partial B to A**





**Figure 2.8 Closed loop implementation, state deviation occurs in production period of B (Case 2), B is retrieved right after the deviation**



**Figure 2.9 Open loop implementation, state deviation occurs in production period of B, original solution is implemented and product B is partially produced**

**Table 2.6 Simultaneous scheduling and control results for closed loop implementation on CSTR, state deviation occurs in production period of B (Case 1)**

| Slot | Products | Transition Time | Production Time | Amount Produced |
|------|----------|-----------------|-----------------|-----------------|
| 1    | C        | 1.77            | 4.50            | 1254.89         |

|   |    |       |       |         |
|---|----|-------|-------|---------|
| 2 | B1 | 3.74  | 4.45  | 356     |
| 3 | A  | 52.02 | 37.91 | 342.46  |
| 4 | B2 | 0.30  | 6.92  | 553.65  |
| 5 | E  | 1.40  | 1.00  | 1251.47 |
| 6 | D  | 0.63  | 1.88  | 1141.54 |

**Table 2.7 Simultaneous scheduling and control results for closed loop implementation on CSTR, state deviation occurs in production period of B (Case 2)**

| Slot | Products | Transition Time | Production Time | Amount Produced |
|------|----------|-----------------|-----------------|-----------------|
| 1    | C        | 1.77            | 4.50            | 1254.89         |
| 2    | B1       | 3.74            | 4.99            | 399.20          |
| 3    | B2       | 1.08            | 6.93            | 554.22          |
| 4    | A        | 56.58           | 37.95           | 342.82          |
| 5    | E        | 1.61            | 2.39            | 2989.89         |
| 6    | D        | 0.63            | 1.88            | 1141.16         |

**Table 2.8 Simultaneous scheduling and control results for open loop implementation on CSTR, state deviation occurs in production period of B**

| Slot | Products | Transition Time | Production Time | Amount Produced |
|------|----------|-----------------|-----------------|-----------------|
| 1    | C        | 1.77            | 4.50            | 1254.89         |
| 2    | B'       | 3.74            | 4.99            | 399.20          |
| 3    | A        | 64.14           | 41.68           | 376.47          |
| 4    | D        | 0.77            | 2.07            | 1256.89         |
| 5    | E        | 0.83            | 1.00            | 1251.47         |

**Table 2.9 Quantitatively comparison of open loop strategy and closed loop strategy, disturbance occurs in production period of B**

| Solutions                                | No Deviation<br>Open loop<br>strategy (Figure<br>2.4) | With Deviation<br>Closed loop<br>strategy case 1<br>(Figure 2.7) | With Deviation<br>Closed loop<br>strategy case 2<br>(Figure 2.8) | With Deviation<br>Open loop<br>strategy (Figure<br>2.9) |
|--|---|--|--|---|
| Scheduling<br>Solution                   | C-B-A-D-E   | C-B1-A-B2-E-D  | C-B1-B2-A-E-D  | C-B'-A-D-E  |
| Cycle Time<br>(Hr)                       | 125.489   | 116.522  | 124.052  | 125.489   |
| Revenue per<br>Hour (\$)                 | 5548.724  | 5672.269   | 7062.720   | 4825.900  |
| Cost of<br>Inventory per<br>Hour (\$)    | 4184.573  | 3992.450   | 5467.573   | 3791.318  |
| Cost of Raw<br>Material per<br>Hour (\$) | 1082.101  | 1366.946   | 1328.343   | 1082.101  |

|                      |        |         |         |         |
|----------------------|--------|---------|---------|---------|
| Profit per Hour (\$) | 282.04 | 312.873 | 266.804 | -12.561 |
|----------------------|--------|---------|---------|---------|

Figure 2.7 and Figure 2.8 demonstrate the response of closed loop implementation at the presence of significant state deviations that occur at 15 hours in the production period of B with different magnitudes (i.e. one is from 0.2 mol/L to 0.15 mol/L and the other from 0.2 mol/L to 0.22 mol/L). Figure 2.9 shows the behavior of open loop implementation when state is deviated from 0.2 mol/L to 0.15 mol/L. Open loop strategy generates negative profit per hour (i.e. -12.561) because B is partially produced (as shown in Figure 2.9 and Table 2.8) while the profit per hour of closed loop strategy is still high because product B is produced in the remaining slots. A large state deviation in Figure 2.7 results in production interruption and switching to different product, while a relative small deviation in Figure 2.8 results in continuation of the current production. It is interesting to note that when the deviation is large it is more costly to force a largely deviated state back to reference rather than taking advantage of the deviation and start with another product. The updated solution generated by the closed loop strategy reflects exactly this concept.

The open loop strategy fails because it carries out the pre-calculated manipulated variable on the real process regardless of the state deviation. When the state variable is deviated due to disturbances, the closed loop strategy forces the process to produce another product or take it back to steady value by implementing the updated control input. However, open loop strategy does not respond to this, making the current product slot less productive.

#### 2.4.2 Cyclic production of Multiproduct using Isothermal Tubular Reactor

In this case study five products (i.e. A, B, C, D, and E) are manufactured in a single multiproduct PFR following cyclic production. Their corresponding steady state conditions can be calculated under steady state assumption using Equation (2.54). Table 2.10 lists the steady state information (i.e. volumetric flow rate, concentration, conversion) and market information (i.e. demand, process cost, inventory cost). The steady state volumetric flow rate and concentration correspond

to the manipulated and controlled variables, respectively. Steady state information describes the feature of reactant during production stages, while the transition stages focus on the dynamic transition from one production stage to another, namely from one product to another. All the recipe and data of this case study are those used by Flores-Tlacuahuac and Grossmann (Flores-Tlacuahuac and Grossmann, 2010).

In an isothermal PFR, reaction  $2A \rightarrow B$  takes place with reaction rate  $-r = kC^2$ . One dimensional (i.e.  $z$  direction) mass transfer is assumed and both diffusion and convection is considered, resulting in the dynamic model in Equation (2.54) with initial condition in Equation (2.55) and boundary conditions in Equation (2.56). The dynamic model is composed of diffusion term ( $\partial^2 C / \partial z^2$ ), a convective term ( $\partial C / \partial z$ ), and a reaction term ( $-KC^2$ ).

Isothermal Tubular Reactor

$$\frac{\partial C}{\partial t} = D \frac{\partial^2 C}{\partial z^2} - v \frac{\partial C}{\partial z} - KC^2 \quad (2.54)$$

Initial conditions

$$C(z, 0) = C_i \quad (2.55)$$

Boundary conditions

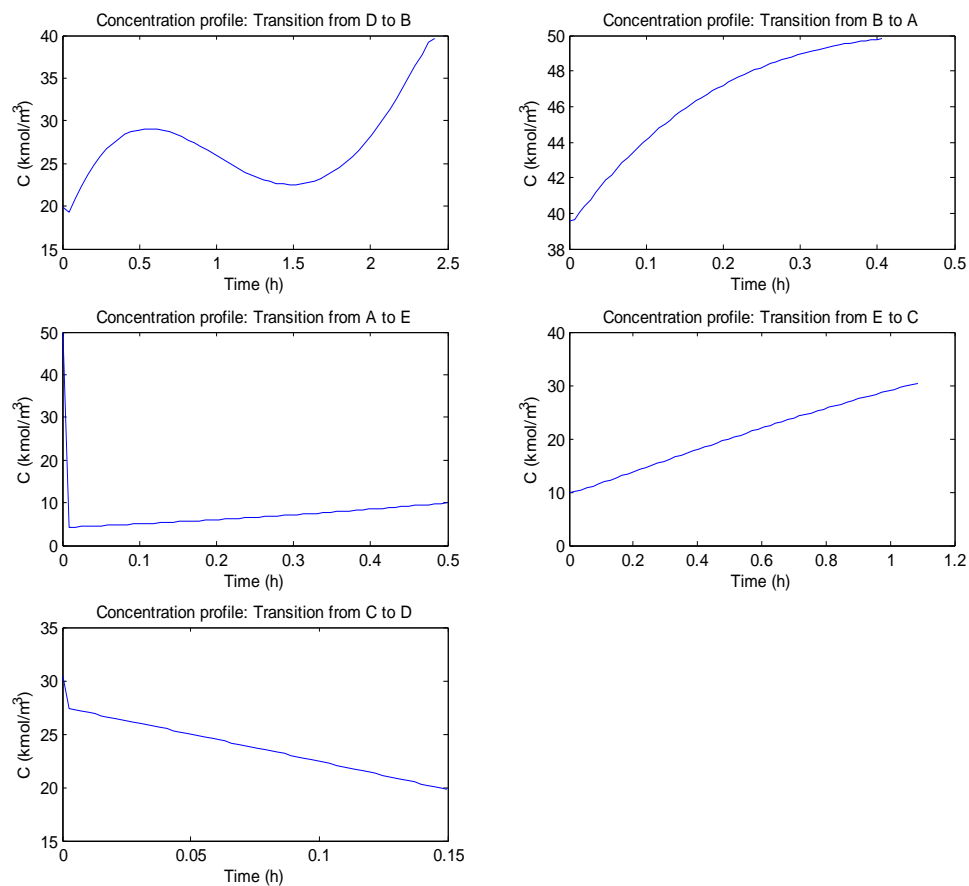
$$\begin{cases} C = C_f - \frac{D}{v} \frac{\partial C}{\partial z} & \text{at } z = 0 \\ \frac{\partial C}{\partial z} = 0 & \text{at } z = L \end{cases} \quad (2.56)$$

where  $C(z, t)$  is the concentration of component A at position  $z$  along longitudinal direction at time  $t$ .  $D$  and  $K$  are mass diffusivity and reaction constant, respectively; and  $v$  is the mass velocity within the reactor whose length equals  $L$ . The concentration of A at the end of PFR  $C_L$  is taken as state variable and the flow rate  $Q$  calculated by  $v$  times cross sectional area is the manipulated variable. A constraint on manipulated variable is enabled as  $0.01m^3 / s \leq Q \leq 2m^3 / s$ . The threshold for state deviation caused by disturbances is  $0.1kmol / m^3$ .

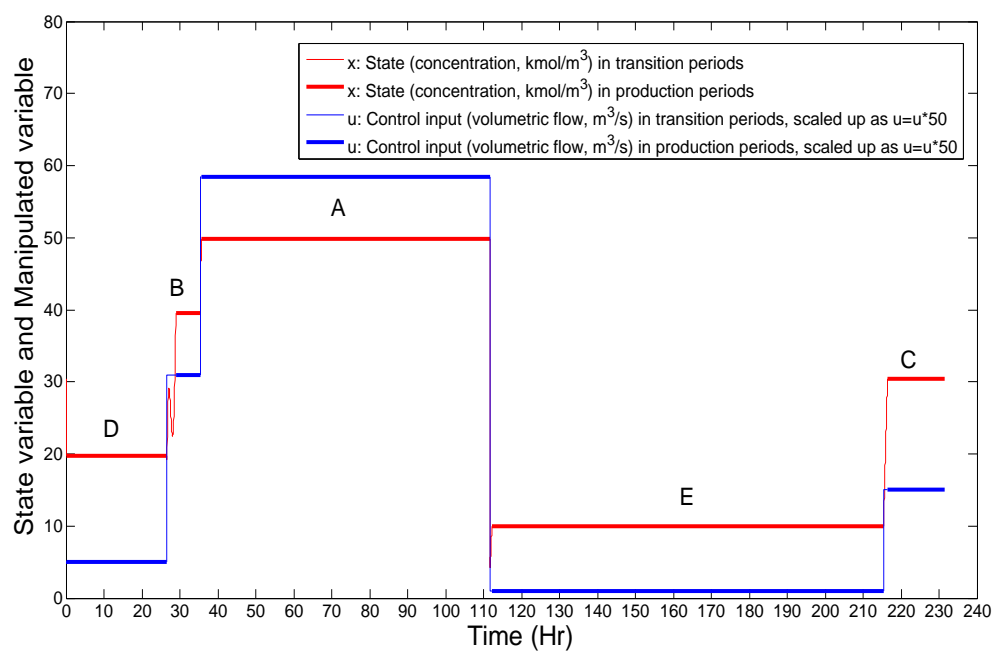
In principle, the optimization problem in this problem is similar to that of the first case study. The dynamic model is discretized and integrated into the scheduling problem to form a MINLP problem. However, since PFR is a distributed parameter system, Equation (2.54) is a partial differential equation and should be discretized both in space and time. The spatial discretization is realized by applying the method of lines (Appendix A) and the temporal discretization is addressed by implicit Runge-Kutta method (Equations (2.25)-(2.31)). After a two dimensional discretization, the integrated problem has 20130 variables and 25545 constraints. It takes approximately 300 seconds to solve using GAMS/SBB on an 8 CPUs server.

**Table 2.10 Five products with steady state information and market information with PFR**

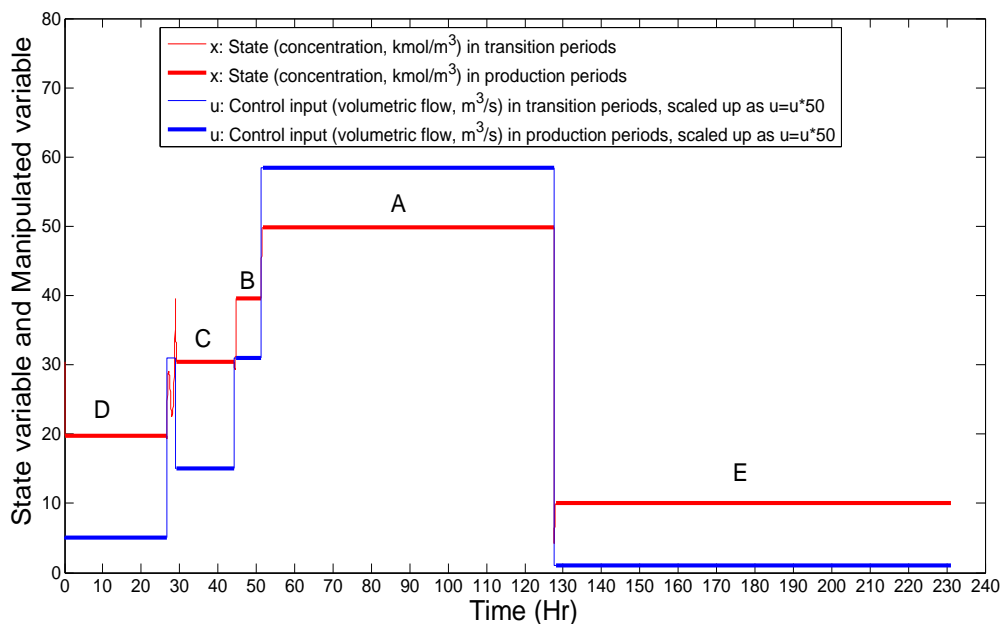
| product | Volumetric flow<br>( $m^3 / s$ ) | Concentration<br>( $kmol / m^3$ ) | Conversion fraction | Demand rate<br>( $kg / hour$ ) | Process cost<br>( $\$ / kg$ ) | Inventory cost<br>( $\$ / kg$ ) |
|---------|----------------------------------|-----------------------------------|---------------------|--------------------------------|-------------------------------|---------------------------------|
| A       | 1.169                            | 49.8                              | 0.5024              | 100                            | 620                           | 1.5                             |
| B       | 0.62                             | 39.6                              | 0.6036              | 90                             | 730                           | 1                               |
| C       | 0.302                            | 30.4                              | 0.6959              | 120                            | 750                           | 2                               |
| D       | 0.1                              | 19.8                              | 0.8019              | 80                             | 770                           | 3                               |
| E       | 0.02                             | 10                                | 0.9                 | 70                             | 790                           | 2.5                             |



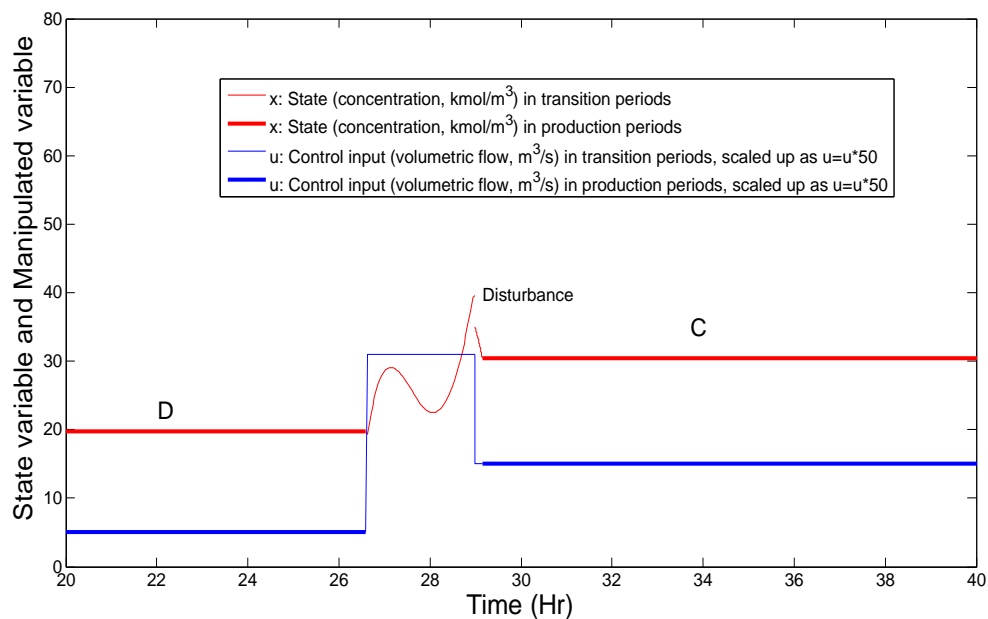
**Figure 2.10 Transition profile between products in PFR cyclic production**



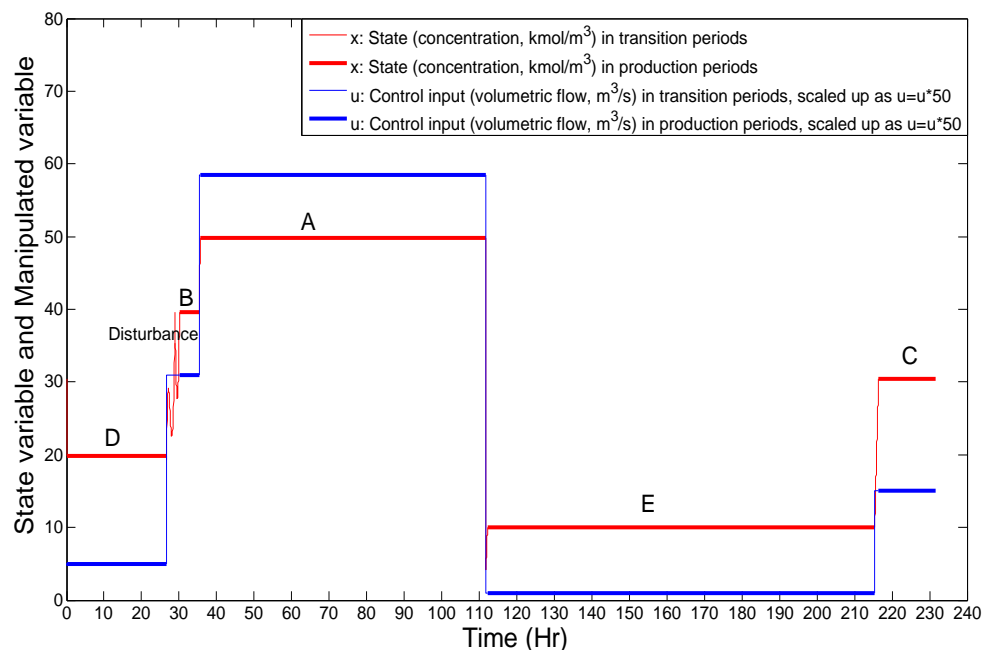
**Figure 2.11 Solution for the PFR original integrated problem, no state deviation**



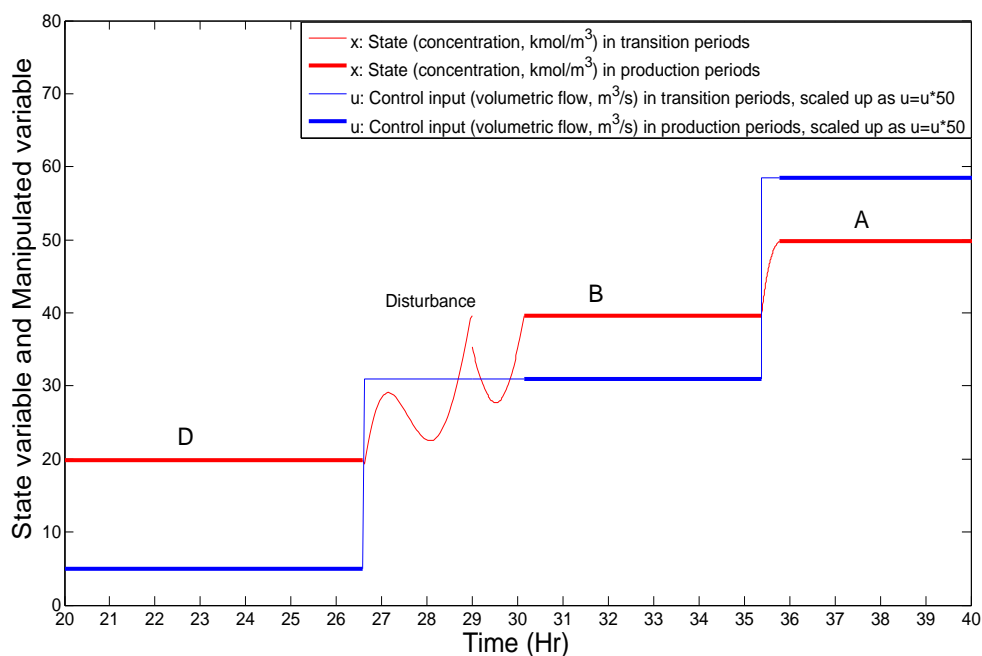
**Figure 2.12 Closed Loop Implementation, state deviation occurs at 29 hours in the transition from D to B, scheduling solution and control profile are updated as targeting C right after the deviation**



**Figure 2.13 enlarged illustration of the state deviation in Figure 2.12**



**Figure 2.14 Open Loop Implementation, state deviation occurs at 29 hours in the transition from D to B, scheduling solution and control profile are the same as the original solutions, the production period of B is shorter than the original one in Figure 2.11**



**Figure 2.15 An enlarged illustration of the deviation in Figure 2.14, production period of B is shortened due to the negligence of state deviation**

**Table 2.11 Simultaneous scheduling and control results for the original PFR problem, no state deviation, corresponding to Figure 2.11**



| Slot | Products | Transition Time | Production Time | Amount Produced |
|------|----------|-----------------|-----------------|-----------------|
| 1    | D        | 0.15            | 26.43           | 20162.70        |
| 2    | B        | 2.41            | 6.38            | 22681.52        |
| 3    | A        | 0.41            | 75.91           | 423580.00       |
| 4    | E        | 0.50            | 103.18          | 17641.72        |
| 5    | C        | 1.09            | 15.12           | 30235.16        |

**Table 2.12 Simultaneous scheduling and control results for closed loop implementation on PFR, state deviation occurs in transition from D to B, corresponding to Figure 2.12**

| Slot | Products | Transition Time | Production Time | Amount Produced |
|------|----------|-----------------|-----------------|-----------------|
| 1    | D        | 0.15            | 26.43           | 20162.70        |
| 2    | C        | 2.57            | 15.12           | 30235.16        |
| 3    | B        | 0.56            | 6.38            | 22681.52        |
| 4    | A        | 0.41            | 75.91           | 423580.00       |
| 5    | E        | 0.50            | 103.18          | 17641.72        |

**Table 2.13 Simultaneous scheduling and control results for open loop implementation on PFR, state deviation occurs in transition from D to B, corresponding to Figure 2.14**

| Slot | Products | Transition Time | Production Time | Amount Produced |
|------|----------|-----------------|-----------------|-----------------|
| 1    | D        | 0.15            | 26.43           | 20162.70        |
| 2    | B        | 3.56            | 5.23            | 18593.16        |
| 3    | A        | 0.41            | 75.91           | 423580.00       |
| 4    | E        | 0.50            | 103.18          | 17641.72        |
| 5    | C        | 1.09            | 15.12           | 30235.16        |

**Table 2.14 Quantitatively comparison of open loop strategy and closed loop strategy, state deviation occurs in transition from D to B**

| Solutions                          | No State Deviation<br>Open loop strategy<br>(Figure 12) | With State<br>Deviation<br>Closed loop strategy<br>(Figure 13,14) | With State Deviation<br>Open loop strategy<br>(Figure 15,16) |
|------------------------------------|---|---|--|
| Scheduling<br>Solution             | D-B-A-E-C   | D-C-B-A-E   | D-B-A-E-C  |
| Cycle Time (Hr)                    | 231.58  | 231.21  | 231.58   |
| Revenue per Hour<br>(\$)           | 1430675.00  | 1432964.48  | 1417787.44   |
| Cost of Inventory<br>per Hour (\$) | 289553.35   | 291852.31   | 289181.28  |

|                                    |           |           |           |
|------------------------------------|-----------|-----------|-----------|
| Cost of Raw Material per Hour (\$) | 376137.83 | 376179.23 | 376137.83 |
| Profit per Hour (\$)               | 764983.82 | 764932.94 | 752468.99 |

Solving the original integrated problem, we obtain the optimal transition between products as shown in Figure 2.10 and the scheduling solution in (Figure 2.11, Table 2.11). When significant state deviation from  $40\text{kmol} / \text{m}^3$  to  $35\text{kmol} / \text{m}^3$  occurs at 20 hours, (Figure 2.12, Table 2.12) present the updated solution at the point of deviation. Instead of producing B after D, the process produces C followed by B. In Figure 2.13, actually the manipulated variable takes advantage of state deviation by decreasing the feed flow rate to enable the production of C. The open loop performance with the presence of state deviation is shown in (Figure 2.14, Table 2.13). The production sequence remains the same but the production time of product B is decreased due to negligence of deviation, resulting in less revenue. We provide quantitative comparison of the solutions in Table 2.14, in which the profit per hour of closed loop strategy (\$764932.94) is slightly lower than the pre-calculated (\$764983.82). However, the profit per hour of open loop strategy (\$752468.99) is much lower than both of them.

## 2.5 Summary

In this study, we formulate a closed loop implementation of simultaneous scheduling and control and apply it to two case studies. We refer to the work of Flores-Tlacuahuac and Grossmann (Flores-Tlacuahuac and Grossmann, 2006) and Flores-Tlacuahuac and Grossmann (Flores-Tlacuahuac and Grossmann, 2010) in modeling the integrated problem as a MIDO problem. We discretize the dynamic model using implicit Runge-Kutta method which is more preferable than explicit Runge-Kutta method due to its high stability.

Because of the presence of disturbances in real processes, it is necessary to implement the simultaneous scheduling and control in a closed loop manner to eliminate the side effects of

disturbances. However, the mechanism of the joint effect on state variables brought by disturbances from multi sources is difficult to comprehend. In this work, we measure the state deviation by comparing the current state value with the reference value, and update the integrated solutions for the remaining part of the production cycle if the deviation exceeds a pre-defined threshold (i.e. quality bounds). Using the proposed approach, even though we cannot completely eliminate the side effects of the disturbances, we mitigate it as much as possible by fully making use of the current state information.

Two case studies offered validation that the closed loop strategy is effective in decreasing the influence of disturbances compared to open loop implementation. Quantitative results show that profit gain by closed loop implementation is much higher than that of open loop strategy.

### **Nomenclature:**

#### *Indices*

Products  $i$

Slots  $k$

Elements  $e$

States  $n$

Manipulated variables  $m$

#### *Parameters*

$N_p$  number of products

$N_s$  number of slots

$N_e$  number of elements within each slot

$N_s$  number of states

$N_u$  number of manipulated variables

$N_{spl}$  number of samples in the production cycle

$D_i$  demand rate for product  $i$

$C_i^p$  price of product  $i$

$C_i^s$  unit cost of inventory for product  $i$

$C^r$  unit cost of raw material

$\theta^{\max}$  upper bound on processing time

$X_i$  conversion

$F^o$  feed stream volumetric flow rate

$x_{\min}^n, x_{\max}^n$  minimum and maximum value of  $n^{\text{th}}$  state variable

$u_{\min}^n, u_{\max}^n$  minimum and maximum value of  $m^{\text{th}}$  manipulated variable

### *Decision Variables*

$\Phi$  profit rate

$\Phi_1$  revenue rate

$\Phi_2$  inventory cost rate

$\Phi_3$  raw material cost rate

$y_{ik}$  binary variable indicating assignment  
of product  $i$  to slot  $k$

$y'_{ik}$  auxiliary variable similar to  $y_{ik}$

$z_{ipk}$  binary variable indicating transition  
from product  $i$  to  $p$  in slot  $k$

$p_k$  process time at slot  $k$

$t_k^e$  ending time at slot  $k$

$t_k^s$  starting time at slot  $k$

$t_{ke}$  time point for each element in each slot

$u_{ke}^m$   $m^{\text{th}}$  manipulated variable at element  $e$  of slot  $k$

$x_{ke}^n$   $n^{\text{th}}$  state variable at element  $e$  of slot  $k$

$\bar{x}_k^n$  desired value of  $n^{\text{th}}$  state at slot  $k$

$\bar{u}_k^m$  desired value of  $m^{\text{th}}$  manipulated variable at slot  $k$

$x_{ss,i}^n$   $n^{\text{th}}$  steady state value of product  $i$

$u_{ss,i}^m$   $m^{\text{th}}$  steady manipulated value of product  $i$

$x_{in,k}^n$   $n^{\text{th}}$  state's initial value in slot  $k$

$u_{in,k}^m$   $m^{\text{th}}$  manipulated variable's initial value in slot  $k$

$h_k$  length of element in slot  $k$

$\theta_{ik}$  process time of product  $i$  at slot  $k$

$\theta_k^t$  transition time at slot  $k$

$\Theta_i$  total processing time of product  $i$

$K1, K2$  intermediate variables in implicit Runge-Kutta methods

$G_i$  production rate for product  $i$

$T_c$  total production cycle time

$W_i$  amount produced for product  $i$

## **Chapter 3 A decomposition approach for the solution of scheduling including process dynamics of continuous processes**

Integration of scheduling and dynamic optimization results in a large size MINLP which is computational very expensive. In order to achieve better computational performance, we explore the structure of the integrated optimization problem for continuous processes, and establish an efficient decomposition scheme based on the mathematical structure of the corresponding model. We decompose the scheduling problem into an optimization model for the production time and a separate one for production sequence. In this way, two sub-problems are formed. One is an MINLP dealing with the production sequence and control profile during transitions, and the other is an NLP dealing with the production time for each product considering the market demands.

We prove the separability of the resulting two sub-problems analytically. Unlike the Lagrangean decomposition approach, no iterations are needed in the proposed approach. Moreover, the computational complexity is significantly reduced compared to the simultaneous scheduling and dynamic optimization problem under varying demand. The main contribution of this work is that there is no need to solve the entire integrated problem for different demand specifications because the only decision variables that need to be updated are the production times. Results of case studies verify the feasibility and effectiveness of proposed approach in reducing the computational complexity of the integrated problem but also obtaining the optimal solution.

### **3.1 General integrated model**

The integrated model (2) follows the scheduling formulation introduced by Flores-Tlacuahuac and Grossmann (Flores-Tlacuahuac and Grossmann, 2006). Implicit Runge-Kutta is employed for the discretization of dynamic model (Zhuge and Ierapetritou, 2012).

To optimize process operations, we utilize the objective of profit maximization per unit time, which can be computed in equations (3.1)-(3.3).

$$\Phi = \Phi_1 - \Phi_2 \quad (3.1)$$

$$\Phi_1 = \sum_{i=1}^{N_p} \frac{C_i^p W_i}{T_c} \quad (3.2)$$

$$\begin{aligned} \Phi_2 &= (COST^{tr} + COST^{ss})/T_c \\ &= \left( \sum_{k=1}^{N_s} \sum_{e=1}^{N_e} C^r (u_{ke}^1 + \dots + u_{ke}^m) h_k \theta_k^t + \sum_{k=1}^{N_s} C^r (\bar{u}_k^{-1} + \dots + \bar{u}_k^{-m}) p_k \right) / T_c \end{aligned} \quad (3.3)$$

where  $\Phi_1, \Phi_2$  represent the rate of revenue and raw material cost, respectively.  $\Phi_2$  is composed of the cost during transition periods  $COST^{tr}$  and cost during steady state production periods  $COST^{ss}$ . The total revenue is calculated as the amount produced ( $W_i$ ) times the product price ( $C_i^p$ ). Raw material cost is composed of two parts, the raw material consumption during production periods and during the transition periods. Unlike the objective in (Flores-Tlacuahuac and Grossmann, 2006), we do not consider state variation in the objective, since we address a pure economic objective in this study. Another difference with (Flores-Tlacuahuac and Grossmann, 2006) is that we do not include inventory cost which can be considered at the planning level. We believe that incorporating inventory cost could complicate the decomposition addressed in this study, not only because the inventory cost could greatly add complexity to the objective but also because of the depleting strategy: delivering all the products at the end of the cycle or delivering the product once it is produced in the cycle. Detailed constraints of the integrated model are presented in Section 2.1.

By combining the objective functions described in (3.1)-(3.3) and the constraints at both scheduling level (2.7)-(2.24) and dynamic optimization level (2.27)-(2.33) along with the linking constraints (2.34)-(2.42) we obtain the optimization problem for the integrated model as shown in problem (3.4).

$$\begin{aligned}
& \max_{y_{ik}, z_{ipk}, x_{ke}, u_{ke}, \theta_k^t, \Theta_i, W_i, t_k^s, t_k^e, T_c} \Phi_1 - \Phi_2 \\
& s.t. \begin{cases} (2.7)-(2.24) \text{ constraints at scheduling level} \\ (2.27)-(2.33) \text{ constraints of dynamic optimization at control level} \\ (2.34)-(2.42) \text{ linking constraints} \end{cases} \quad (3.4)
\end{aligned}$$

Note that in the integrated problem (3.4) the production sequence and transitions are connected. For example, if the sequence is BDEAC, then the transitions from B to D, D to E, E to A, A to C, and C to A have to be determined. Thus the production sequence and transitions are both part of the solution of the integrated solution (i.e. the overall solution) and they are obtained simultaneously once the integrated problem is solved.

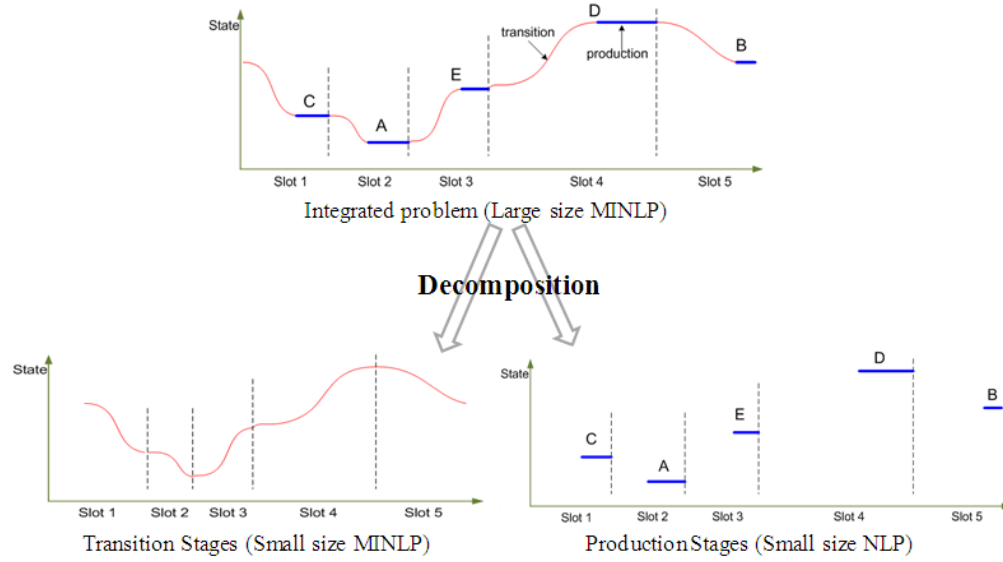
### 3.2 A decomposition approach with analytical proof

For the purpose of sensitivity analysis, we introduce an alternative form of objective as shown in Equation (3.5). It should be noted that the compact objective in Equation (3.5) is equivalent to the one in (3.1)-(3.3).

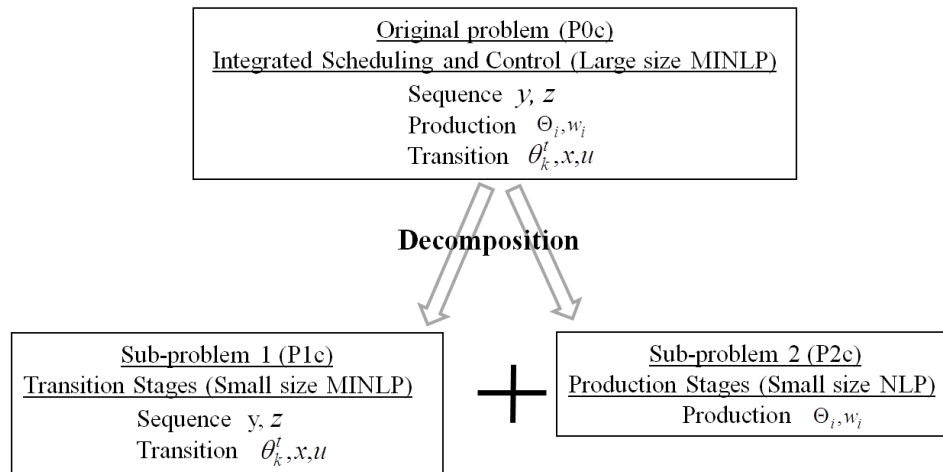
$$J = \frac{\sum_i P_i \Theta_i - f(y_{ik}, \theta_k^t)}{\sum_i \Theta_i + \sum_k \theta_k^t} \quad (3.5)$$

where  $\Theta_i$  is production time for product  $i$ , and  $\theta_k^t$  is the transition time at slot  $k$ .  $\sum_i P_i \Theta_i$  represents the profit gained from production and  $f(\bullet)$  is the total transition cost. Note that  $P_i = (C_i^p - C^r \cdot \text{material consumption rate})$ . Since production rate and material consumption rate are known in steady state, we group them together to make the expression more compact.  $P_i$  can be considered as a revenue coefficient.

As shown in Figure 3.1, the original solution can be decomposed into two sub-problems. The first problem involves the transition stages including sequences and transition profiles and the second one consists of the production stages. The corresponding optimization problems for these two parts are as shown in Figure 3.2.



**Figure 3.1 Decomposition for the solutions of integrated problem of continuous processes**



**Figure 3.2 Decomposition with variables for the optimization problem of continuous processes**

In Figure 3.2 sub-problem 1 deals with the sequence and transition times, and sub-problem 2 deals with production. By assigning a fixed value to the production time in the original integrated problem (3.4) we obtain sub-problem 1(3.6).



$$\begin{aligned}
& \max_{y_{ik}, z_{ipk}, x_{ke}, u_{ke}, \theta_k^t, t_k^s, t_k^e, T_c} \Phi_1 - \Phi_2 \\
& s.t. \left\{ \begin{array}{l} (2.7)-(2.13) \text{ constraints at scheduling level} \\ (2.17)-(2.24) \text{ constraints at scheduling level} \\ (2.27)-(2.33) \text{ constraints of dynamic optimization at control level} \\ (2.34)-(2.42) \text{ linking constraints} \\ \Theta_i = \Theta^0 \quad \text{assign production time for each product} \end{array} \right. \quad (3.6)
\end{aligned}$$

Solving (3.6) generates the optimal production sequence and optimal transition time. Those are then assigned in sub-problem 2 (3.7) which generates all the remaining decision variables in the integrated problem.

$$\begin{aligned}
& \max_{\Theta_i, W_i, t_k^s, t_k^e, T_c} \Phi_1 - \Phi_2 \\
& s.t. \left\{ \begin{array}{l} (2.14)-(2.24) \text{ constraints at scheduling level} \\ z_{ipk} = z_{ipk}^* \quad \text{assign production sequence} \\ y_{ik} = y_{ik}^* \quad \text{assign production sequence} \\ \theta_k^t = \theta_k^{t*} \quad \text{assign transition time for each slot} \\ x_{ke} = x_{ke}^* \quad \text{assign state profile} \\ u_{ke} = u_{ke}^* \quad \text{assign control profile} \end{array} \right. \quad (3.7)
\end{aligned}$$

where  $(z_{ipk}^*, y_{ik}^*, \theta_k^{t*}, x_{ke}^*, u_{ke}^*) \subset \{\text{solution set of problem (3.6)}\}$

Both sub problems correspond to much smaller problems compared to the original integrated problem. The following proof and case study results in Section 3.3 verify the proposed decomposition.

It should be noticed that the proposed decomposition approach does not lead to the solution of the scheduling and the dynamic optimization problem separately. Theorem 1 provides a proof that the solution of (3.6) and (3.7) is the same as the one from the integrated problem (3.4).

**Theorem 1:** The decomposition in Figure 3.2 does not affect the optimality of the integrated problem. i.e. solving the sub-problems (3.6) and (3.7) results in the same solution as the integrated problem (3.4).

Three lemmas are proposed to prove Theorem 1.

### 3.2.1 Lemma 1: The perturbation on production time does not affect the transitions

*Proof:*

The profit for each time slot is:

$$J = \frac{P\Theta}{\Theta + \theta^t} - \frac{f_{tc}(\theta^t)}{\Theta + \theta^t} = \frac{P\Theta - f_{tc}(\theta^t)}{\Theta + \theta^t} \quad (3.8)$$

where  $P$  is revenue coefficient introduced in Equation (3.5).  $\Theta$ ,  $\theta^t$  and  $f_{tc}(\theta^t)$  correspond to production time, transition time, and transition cost, respectively.

Assume that  $\theta^{t*}$  is the optimal transition time when the production time is  $\Theta$ . Then we have:

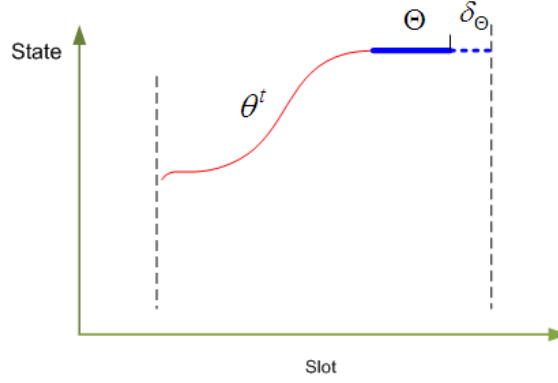
$$\frac{P\Theta - f_{tc}(\theta^{t*})}{\Theta + \theta^{t*}} \geq \frac{P\Theta - f_{tc}(\theta^t)}{\Theta + \theta^t} \quad (3.9)$$

The proposition that needs to be proved can be described as follows.

If there is a small perturbation on  $\Theta$  shown in Figure 3.3,  $\theta^{t*}$  still the optimal transition time which can be expressed mathematically as follows:

$$\frac{P(\Theta + \delta\Theta) - f_{tc}(\theta^{t*})}{(\Theta + \delta\Theta) + \theta^{t*}} \geq \frac{P(\Theta + \delta\Theta) - f_{tc}(\theta^t)}{(\Theta + \delta\Theta) + \theta^t} \quad (3.10)$$

The specific procedures to establish this proposition are provided in Appendix A.



**Figure 3.3** Perturbation on production does not affect transition within one slot

### 3.2.2 Lemma 2: The optimal transition time is obtained as the inferior of its feasible values

A property of  $\theta_k^{t*}$ :  $\theta_k^{t*}$  is obtained as the minimum value of feasible  $\theta_k^t$  obtained by solving problem (3.11).

$$\begin{aligned}
 q_k^{t*} &= \min_{u_k(t), t \in [0, q_k^t]} q_k^t \\
 &\left\{ \begin{array}{l} \dot{x}_k(t) = f(x_k(t), u_k(t)), t \in [0, q_k^t] \\ x_{in,k} = x_k(0) = \bar{x}_{k-1} \\ s.t. \left\{ \begin{array}{l} x_k(q_k^t) = \bar{x}_k \\ x_k(t) \in \mathbb{W}_x, t \in [0, q_k^t] \\ u_k(t) \in \mathbb{W}_u, t \in [0, q_k^t] \end{array} \right. \end{array} \right. \quad (3.11)
 \end{aligned}$$

*Proof:*

The overall objective of the whole production cycle is

$$J = \frac{\sum_i P_i \Theta_i - f_{tc}(y_{ik}, \theta_k^t)}{\sum_i \Theta_i + \sum_k \theta_k^t} \quad (3.12)$$

$$f_{tc}(y_{ik}, \theta_k^t) = \sum_i \sum_k C^r \int_0^{\theta_k^t} u_k y_{ik} dt = \sum_i \sum_p \sum_k C^r \int_0^{\theta_p^t} u_k z_{ipk} dt \quad (3.13)$$

and  $C^r$  is a coefficient for transition cost, for instance, the unit price for raw materials. The transition cost  $f_{tc}(y_{ik}, \theta_k^t)$  can be evaluated as the integral of the manipulated variable over the

transition duration  $\theta_k^t$  in all time slots. Binary variables  $y_{ik}$  and  $z_{ipk}$  indicate assignments of production in slots (2.7)-(2.13). Taking the partial derivative of  $J$  with respect to  $\theta_k^t$  when  $\Theta_i, \theta_{K-k}^t, y_{ik} (\forall i \in I, \forall k \in K)$  are fixed, lead to the following property for  $\theta_k^t$  i.e.  $\theta_k^{t*} \leq \theta_k^t$ . The details of the proof are provided in Appendix A.

With this property, we can calculate  $\theta_{ip}^{t*}$  before hand by solving optimization problem and calculate the corresponding transition cost from product  $i$  to product  $p$  at slot  $k$ . However, it involves significant computation since we need to solve problem (3.11) many times. For a cyclic production where  $n$  products are produced, the number of transitions would be  $p(n, 2) = n!/(n-2)!$ . Note that the transition from product A to B is not necessarily equivalent to the transition from B to A. It means that we have to solve problem (3.11) 20 times, so this will be computational impossible for a large number of products.

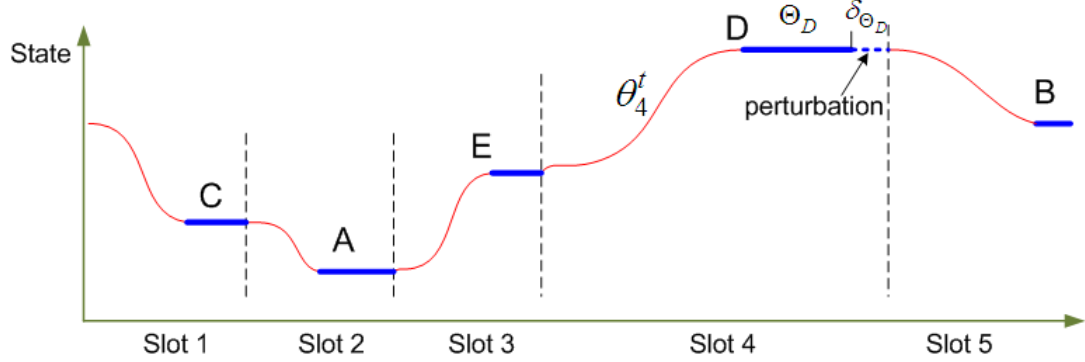
Note that the optimal transition time computed by problem (3.11) is independent of the product demands. It should be noted that which transitions exist in the scheduling solution is of course a function of the demand but the optimal transition times are not. For example, the transition from D to E can be found through problem (3.11) without knowing the demands of D or E. However, the transition from D to E may not exist in the overall solution if for example the optimal sequence is BCDAE.

### 3.2.3 Lemma 3: The perturbation on production time does not affect the production sequence

*Proof:*

Suppose  $z^*$  is the optimal sequence i.e.  $J(z^*) \geq J(z)$ , what we need to prove is that this inequality holds in the presence of perturbation in production. For example, the optimal sequence C-A-E-D-

B in is retained when  $\delta_{\Theta}$  presents. Following the steps in Appendix A, we obtain a sufficient condition (A.36) for this hypothesis.



**Figure 3.4 Perturbation on production does not affect all the transitions**

### 3.2.4 Solution procedures for the decomposition approach

Based on Lemmas 1, 2 and 3 we prove that the production does not affect the optimality of transitions and production sequence. Therefore Theorem 1 holds and the decomposition in Figure 3.2 is verified. Note that Lemma 1 and 3 are essential steps in the proof of the decomposition. In Lemma 1 we prove that the production does not affect the optimality of transition, and in Lemma 3 we prove the production does not affect sequence. Thus we can separate production variables from the integrated problem and achieve decomposition as shown in Figure 3.4.

In principle the transition time can be pre-computed through problem (3.11). However, it involves significant computation since we need to solve (3.11) repeatedly, as described in the paragraphs following (3.13). In order to avoid the huge computation burden, we propose a solution procedure that involves two steps for the integrated problem (3.4). Note that the proposed solution procedure is supported by Theorem 1 and Lemmas 1-3 in the proof.

Based on the above analysis for decomposition, we propose the following solution procedure for the integrated problem (3.4)

Step 1: Treat  $\Theta_i$  as constants and assign proper values for them. To get initial values of  $\Theta_i$  the demand constraints can be considered as equalities and for a typical production cycle  $T_c = 100$  hours,  $\Theta_i$  can be specified. Solve sub-problem (3.6) and obtain  $(z_{ipk}^*, y_{ik}^*, \theta_k^{t*}, x_{ke}^*, u_{ke}^*)$ .

Step 2: Substitute  $(z_{ipk}^*, y_{ik}^*, \theta_k^{t*}, x_{ke}^*, u_{ke}^*)$  into the original problem, which results in sub-problem (3.7) Solve it and obtain  $\Theta_i^*$ .

Since we have proved that production times do not affect production sequence or transition profiles, this solution approach will lead to the same solution as the integrated optimization problem.

### 3.3 Case studies

In order to demonstrate the feasibility of the decomposition approach, we studied two numerical cases, one for a single state CSTR and the other for methyl methacrylate (MMA) polymerization which involves multiple states. In each case study we investigate three scenarios, and discuss their results. The first scenario is to solve for the original integrated problem MINLP (problem (3.4)) using both local and global optimization solvers. In scenario 2 we apply the proposed decomposition approach and solve the sub-problems (3.6) and (3.7) sequentially following the procedure presented in Section 3.2.

In scenario 3 we solve for the scheduling and dynamic optimization problem in a conventional approach that does not involve integration of them. Given the estimation of transition time  $\tilde{\theta}_{pi}^t$  and transition cost  $\widetilde{COST}_{pi}^{tr}$ , scheduling problem (3.14) is solved under the constraints at scheduling level and linking constraints. Dynamic optimization problem (3.15) is then solved based on solution of the scheduling problem e.g.  $y_{ik}^*$ , The dynamic optimization problem minimizes the cost during transitions under the constraints at dynamic optimization level and

linking constraints. Note that the estimation of transition time  $\tilde{\theta}_{pi}^t$  is made empirically based on the knowledge of the process. A conservative estimation unnecessarily extends the transition period and thus undermines the profitability, while a reckless one could make the dynamic optimization problem infeasible *i.e.* a transition too short makes it impossible to drive the states to the desired steady states within the estimated time period. The transition cost from product  $p$  to  $i$   $\widetilde{COST}_{pi}^{tr}$  is calculated as the product of the average of initial and desired manipulated variables and the estimated transition time and the price *i.e.*  $\widetilde{COST}_{pi}^{tr} = C^r \cdot 0.5(u_{m,k} + \bar{u}_k) \cdot \tilde{\theta}_{pi}^t$ . This is a simplification of the standard form of transition cost in Equation (3.13). Both  $\tilde{\theta}_{pi}^t$  and  $\widetilde{COST}_{pi}^{tr}$  can be obtained beforehand. They present in problem (3.14) as parameters.

$$\begin{aligned}
 & \max_{y_{ik}, z_{ipk}, \theta_k^t, \Theta_i, W_i, t_k^s, t_k^e, T_c} \Phi_1 - \Phi_2 \\
 & s.t. \begin{cases} (2.7)-(2.24) \text{ constraints at scheduling level} \\ (2.34)-(2.39) \text{ linking constraints} \\ \theta_{pi}^t = \tilde{\theta}_{pi}^t \\ COST^{tr} = \sum_{i=1}^{N_p} \sum_{p=1}^{N_p} \widetilde{COST}_{pi}^{tr} z_{ipk}, \forall k \in K \end{cases} \quad (3.14)
 \end{aligned}$$

$$\begin{aligned}
 & \min_{x_{ke}, u_{ke}} \sum_{k=1}^{N_s} \sum_{e=1}^{N_e} C^r (u_{ke}^1 + \dots + u_{ke}^m) h_k \theta_k^t \\
 & s.t. \begin{cases} (2.27)-(2.33) \text{ constraints of dynamic optimization at control level} \\ (2.20), (2.34)-(2.42) \text{ linking constraints} \\ \theta_{pi}^t = \tilde{\theta}_{pi}^t \\ y_{ik} = y_{ik}^* \end{cases} \quad (3.15)
 \end{aligned}$$

The purposes of investigating three scenarios are to demonstrate the significance of integration in terms of boosting profitability (comparing scenario 1 and 3), and the advantage of the proposed decomposition approach in reducing computing time (comparing scenario 1 and 2).

### 3.3.1 Single state CSTR cyclic production

The specification of this case is the same as the one in Section 2.4.1. We solve the integrated problem for five production cycles in which demand for each product is varying. Table 3.1 shows the specific demands that we assign in five cycles and Figure 3.5 demonstrates the solutions for each cycle. It can be seen that the demands for all the five products are varying and correspondingly the productions are varying in order to satisfy the demands. Nevertheless, the transitions prior to the production stages are consistent but do not respond to the varying demands as shown in Figure 3.5. Obviously the optimal profit is varying with respect to cycles, due to the fluctuating demands. It's worth mentioning that all the cycles generate the same optimal sequence C-B-A-E-D, regardless of the varying demands.

Table 3.3 summarizes the problem size, computation time and the results of scheduling and dynamic optimization under all these three scenarios described above i.e. scenario 1 the original integration, scenario 2 the proposed approach and scenario 3 the conventional approach. All scenarios correspond to product demands of cycle 1 in Table 3.1. Problems in scenario 3 is solved based on the transition estimation provided by Table 3.2. Scenarios 1 and 2 result in the same scheduling and dynamic optimization solutions and the same profit, however scenario 2 (the proposed decomposition approach) takes much less time to compute it. Though the problem size of scenario 2 is only slightly smaller than the original integrated problem in scenario 1, the computational complexity for MINLP is significantly reduced. This is because decomposing the decision variables effectively changes the structure of the optimization problem without undermining the optimality of the integrated problem, as proved by the theorems in section 3.2.

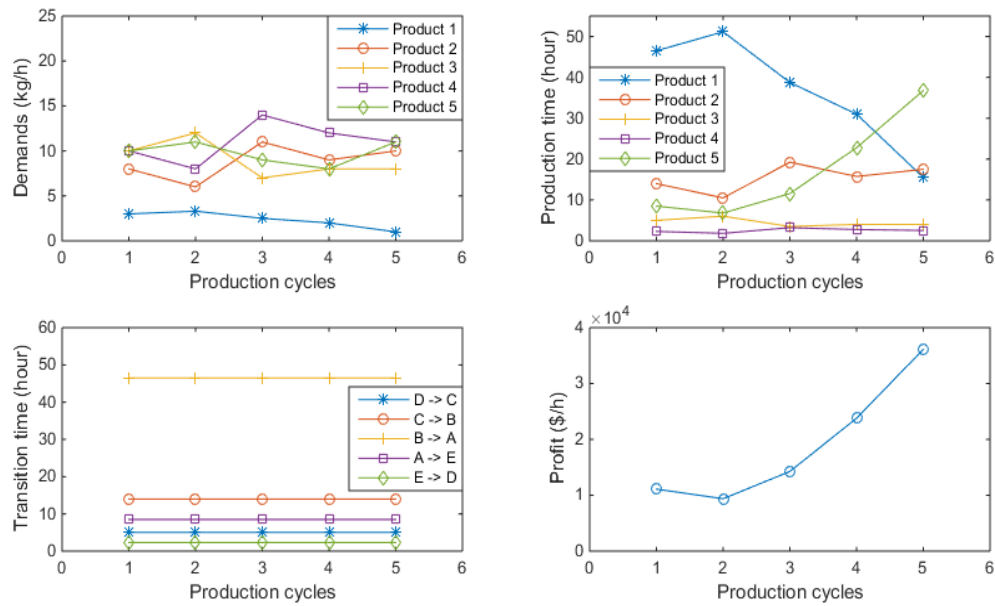
The comparison between scenario 1 and 3 demonstrates the significance of integration of scheduling and dynamic optimization. Scenario 3 follows the conventional approach where the integrated problem is not obtained. It results in much lower profit than the original integrated problem does. This is because the scheduling solutions obtained based on the estimation of



transition time and transition costs are not optimal in terms of the overall profit in the integrated problem. Note that all scenarios generate the same optimal cycle time 140 hours while solving the corresponding optimization problems under the constraints for the cycle time  $90 \leq T_c \leq 140$

**Table 3.1 Demands assignment in a single state CSTR cyclic production**

| Demands (kg/h) | Cycle 1 | Cycle 2 | Cycle 3 | Cycle 4 | Cycle 5 |
|----------------|---------|---------|---------|---------|---------|
| Product A      | 3       | 3.3     | 2.5     | 2       | 1       |
| Product B      | 8       | 6       | 11      | 9       | 10      |
| Product C      | 10      | 12      | 7       | 8       | 8       |
| Product D      | 10      | 8       | 14      | 12      | 11      |
| Product E      | 10      | 11      | 9       | 8       | 11      |



**Figure 3.5 Solutions for integrated problem at different cycles assigned with different demands**

**Table 3.2 Estimation of transition time (hour) and transition cost (\$) between products (single state CSTR), for instance, transition time from product C to B is 4 hours and the corresponding transition cost is \$10000**

| $(\bar{\theta}_{pi}^t, \overline{COST}_{pi}^{tr})$ | Product A   | Product B  | Product C    | Product D   | Product E    |
|--|-------------|------------|--------------|-------------|--------------|
| Product A  | (0, 0)      | (0.4, 220) | (0.5, 1030)  | (1.8, 9090) | (2.1, 26360) |
| Product B  | (60, 33000) | (0, 0)     | (0.4, 10000) | (0.8, 4400) | (1.8, 23400) |

|           |              |             |              |              |              |
|-----------|--------------|-------------|--------------|--------------|--------------|
| Product C | (62, 127100) | (4, 10000)  | (0, 0)       | (0.9, 6300)  | (1.5, 21750) |
| Product D | (65, 328250) | (6, 33000)  | (2.2, 15400) | (0, 0)       | (1.2, 21000) |
| Product E | (69, 865950) | (8, 104000) | (3.5, 50750) | (1.6, 28000) | (0, 0)       |

**Table 3.3 Results of single state CSTR cyclic production, comparison between three scenarios**

| Scenarios                             | Scenario 1: Original Integration Problem (3.4) |                                | Scenario 2: Decomposition approach |                            | Scenario 3: Conventional (no integration) |                             |
|---------------------------------------|--|--------------------------------|------------------------------------|----------------------------|---|-----------------------------|
| Problem type and solver in GAMS       | Problem (3.4), MINLP, SBB                      | Problem (3.4), MINLP, BARON    | Problem (3.6), MINLP, SBB          | Problem (3.7), NLP, CONOPT | Problem (3.14), MINLP, SBB                | Problem (3.15), NLP, CONOPT |
| Problem size (constraints, variables) | 4046, 2986                                     | 4046, 2986                     | 4041, 2981                         | 65, 31                     | 1011, 1116                                | 336, 244                    |
| CPU (s)                               | 49.3   | 527.9                          | 17.8                               | 0.32                       | 5.2                                       | 2.6                         |
| Production sequence                   | C-B-A-E-D                                      | C-B-A-E-D                      | C-B-A-E-D                          |                            | C-A-B-D-E                                 |                             |
| Cycle time (h)                        | 140  | 140                            | 140                                |                            | 140                                       |                             |
| Production time in slots (h)          | 5.02, 14.00, 46.49, 8.49, 2.31                 | 5.02, 14.00, 46.49, 8.49, 2.31 | 5.02, 14.00, 46.49, 8.49, 2.31     |                            | 5.02, 46.49, 14.0, 2.30, 4.27             |                             |
| Transition time in slots (h)          | 1.12, 3.74, 56.57, 1.61, 0.63                  | 1.12, 3.74, 56.57, 1.61, 0.63  | 1.12, 3.74, 56.57, 1.61, 0.63      |                            | 3.5, 62.0, 0.4, 0.8, 1.2                  |                             |
| Raw material cost (\$/h)              | 2353.67  | 2353.67                        | 2353.67                            |                            | 2218.30                                   |                             |
| Revenue (\$/h)                        | 13450.17                                       | 13450.17                       | 13450.17                           |                            | 8929.74                                   |                             |
| Profit (\$/h)                         | 11096.49                                       | 11096.49                       | 11096.49                           |                            | 6711.44                                   |                             |

### 3.3.2 MMA production with multiple states

The proposed decomposition approach is applied to an isothermal free radical polymerization process that produces methyl methacrylate (MMA) using azobisisobutyronitrile as initiator and toluene as the solvent. The reaction takes place in an isothermal CSTR at the temperature of 335 K. The kinetic model is described by (Mahadevan et al., 2002) and summarized in the following

equations. (3.16)-(3.19) present the dynamics of four state variables monomer concentration  $C_m$ , initiator concentration  $C_I$ , molar concentration of dead chains  $D_0$ , and mass concentration of dead chains  $D_1$ . (3.20) presents an auxiliary variable  $P_0$  and (3.21) calculates the output variable the molecular weight  $D_1/D_0$ . The manipulated variable is the initiator flow rate  $F_I$ . The associated parameters and state/manipulated variables are provided in Table 3.4 and Table 3.5, respectively. The process is illustrated in Figure 3.6.

$$\frac{dC_m}{dt} = -(k_p + k_{fm})P_0C_m + \frac{F(C_{m_{in}} - C_m)}{V} \quad (3.16)$$

$$\frac{dC_I}{dt} = -k_I C_I + \frac{F_I C_{I_{in}} - F C_I}{V} \quad (3.17)$$

$$\frac{dD_0}{dt} = (0.5k_{Td} + k_{Td})P_0^2 + k_{fm}P_0C_m - \frac{F D_0}{V} \quad (3.18)$$

$$\frac{dD_1}{dt} = M_m(k_p + k_{fm})P_0C_m - \frac{F D_1}{V} \quad (3.19)$$

$$P_0 = \left( \frac{2f^* k_I C_I}{k_{Td} + k_{Tc}} \right)^{0.5} \quad (3.20)$$

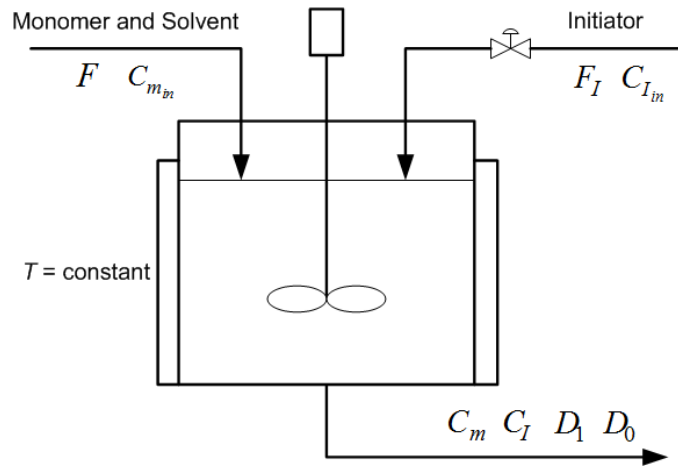
$$y = \frac{D_1}{D_0} \quad (3.21)$$

**Table 3.4 Parameters in the kinetic model of MMA polymerization in a CSTR**

|  |   |
|--|---|
| $T = 335 \text{ K}$                                      | Temperature of the reactor                      |
| $F = 10.0 \text{ m}^3/\text{h}$                          | Monomer flow rate                               |
| $V = 1.0 \text{ m}^3$                                    | Reactor volume                                  |
| $f^* = 0.58$   | Initiator efficiency                            |
| $k_p = 2.50 \times 10^6 \text{ m}^3/\text{kmol h}$       | Propagation rate constant                       |
| $k_{Td} = 1.09 \times 10^{11} \text{ m}^3/\text{kmol h}$ | Termination by disproportionation rate constant |
| $k_{Tc} = 1.33 \times 10^{10} \text{ m}^3/\text{kmol h}$ | Termination by coupling rate constant           |
| $C_{I_{in}} = 8.00 \text{ kmol}/\text{m}^3$              | Inlet initiator concentration                   |
| $C_{m_{in}} = 6.00 \text{ kmol}/\text{m}^3$              | Inlet monomer concentration                     |
| $k_{fm} = 2.45 \times 10^3 \text{ m}^3/\text{kmol h}$    | Chain transfer to monomer rate constant         |
| $k_I = 1.02 \times 10^{-1} \text{ h}^{-1}$               | Initiation rate constant                        |
| $M_m = 100.12 \text{ kg}/\text{kmol}$                    | Molecular weight of monomer                     |

**Table 3.5 Variables in the kinetic model of MMA polymerization in a CSTR**

|                              |   |
|------------------------------|---|
| $C_m$ (kmol/m <sup>3</sup> ) | State: Monomer concentration              |
| $C_I$ (kmol/m <sup>3</sup> ) | State: Initiator concentration            |
| $D_0$ (kmol/m <sup>3</sup> ) | State: Molar concentration of dead chains |
| $D_1$ (kg/m <sup>3</sup> )   | State: Mass concentration of dead chains  |
| $F_I$ (m <sup>3</sup> /h)    | Manipulate: Initiator flow rate           |
| $y = D_1/D_0$ (kg/kmol)      | Output: Molecular weight                  |

**Figure 3.6 MMA polymerization in a CSTR**

In this case study, five products are produced cyclically at steady state and each product grade corresponds to a value of molecular weight. The steady-state values of the state variables, the manipulated variables, and the output variables are provided in Table 3.6. Similarly to the first case study we again investigate three scenarios and compare their problem size, CPU time, scheduling and dynamic optimization solution as well as the overall profit as provided in Table 3.8.

Scenario 1 is solved using normal MINLP solver SBB and global optimal solver BARON. They result in similar overall profit. Scenario 2 is following the decomposition approach presented in section 3.2 and the solution procedure described in section 3.2.4. Scenario 3 is solved based on the estimated transition time and transition cost in Table 3.7. It can be observed that scenario 2

(the proposed decomposition approach) produces slight less profit 562.43 than the global optima 564.11 but consumes much less CPU time. However scenario 3 produces the least profit 542.56 due to the lack of integration.

**Table 3.6 Steady state information for each product of MMA production**

| Product | $C_m$<br>( $\text{kmol}/\text{m}^3$ ) | $C_I$<br>( $\text{kmol}/\text{m}^3$ ) | $D_0$<br>( $\text{kmol}/\text{m}^3$ ) | $D_1$<br>( $\text{kg}/\text{m}^3$ ) | $F_I$<br>( $\text{m}^3/\text{h}$ ) | $D_1/D_0$<br>( $\text{kg}/\text{kmol}$ ) | Demand<br>( $\text{m}^3/\text{h}$ ) | Price<br>( $\$/\text{m}^3$ ) |
|---------|---------------------------------------|---------------------------------------|---------------------------------------|-------------------------------------|------------------------------------|--|-------------------------------------|------------------------------|
| A       | 5.73                                  | 0.0366                                | 0.0007                                | 27.0324                             | 0.0463                             | 40084                                    | 0.4                                 | 170                          |
| B       | 5.63                                  | 0.0713                                | 0.0012                                | 37.0444                             | 0.0900                             | 31938                                    | 0.7                                 | 150                          |
| C       | 5.57                                  | 0.0984                                | 0.0015                                | 43.0516                             | 0.1242                             | 28293                                    | 1.1                                 | 130                          |
| D       | 5.46                                  | 0.1615                                | 0.0023                                | 54.0648                             | 0.2039                             | 23153                                    | 0.9                                 | 125                          |
| E       | 5.41                                  | 0.1963                                | 0.0028                                | 59.0708                             | 0.2479                             | 21294                                    | 0.8                                 | 110                          |

**Table 3.7 Estimation of transition time (hour) and transition cost (\$) between products (MMA case study), for instance, transition time from product C to B is 0.16h and the corresponding transition cost is \$85.68**

| $(\bar{\theta}_{pi}^t, \overline{COST}_{pi}^t)$ | Product A      | Product B      | Product C      | Product D      | Product E      |
|---|----------------|----------------|----------------|----------------|----------------|
| Product A                                       | (0, 0)         | (0.14, 47.70)  | (0.16, 68.20)  | (0.23, 143.86) | (0.28, 205.94) |
| Product B                                       | (0.12, 40.89)  | (0, 0)         | (0.14, 74.97)  | (0.19, 139.60) | (0.22, 185.84) |
| Product C                                       | (0.18, 76.72)  | (0.16, 85.68)  | (0, 0)         | (0.15, 123.03) | (0.19, 176.74) |
| Product D                                       | (0.28, 175.14) | (0.25, 183.68) | (0.17, 139.44) | (0, 0)         | (0.16, 180.72) |
| Product E                                       | (0.29, 213.29) | (0.27, 228.08) | (0.22, 204.65) | (0.15, 169.42) | (0, 0)         |

**Table 3.8 Results of MMA case study, comparison between three scenarios**

| Scenarios                             | Scenario 1: Original Integration Problem (3.4) |                             | Scenario 2: Decomposition approach |                            | Scenario 3: Sequential (no integration) |                             |
|---------------------------------------|--|-----------------------------|------------------------------------|----------------------------|---|-----------------------------|
| Problem type and solver in GAMS       | Problem (3.4), MINLP, SBB                      | Problem (3.4), MINLP, BARON | Problem (3.6), MINLP, SBB          | Problem (3.7), NLP, CONOPT | Problem (3.14), MINLP, SBB              | Problem (3.15), NLP, CONOPT |
| Problem size (constraints, variables) | 5430, 4035                                     | 5430, 4035                  | 5425, 4030                         | 65, 31                     | 592, 390                                | 1035, 759                   |

|                              |                              |                              |                              |      |                              |      |
|------------------------------|------------------------------|------------------------------|------------------------------|------|------------------------------|------|
| CPU (s)                      | 275.4                        | 1154.9                       | 129.1                        | 0.43 | 16.9                         | 24.4 |
| Production sequence          | C-B-E-D-A                    | C-E-D-A-B                    | C-B-E-D-A                    |      | C-A-D-E-B                    |      |
| Cycle time (h)               | 70                           | 70                           | 70                           |      | 70                           |      |
| Production time in slots (h) | 7.7, 9.7, 5.6, 6.3, 40.0     | 7.7, 5.6, 6.3, 40.0, 9.80    | 7.7, 9.7, 5.6, 6.3, 40.0     |      | 7.7, 40.0, 6.3, 5.6, 9.4     |      |
| Transition time in slots (h) | 0.10, 0.10, 0.12, 0.07, 0.27 | 0.06, 0.10, 0.07, 0.27, 0.08 | 0.10, 0.10, 0.12, 0.07, 0.27 |      | 0.14, 0.18, 0.23, 0.16, 0.27 |      |
| Raw material cost (\$/h)     | 960.97                       | 960.85                       | 960.97                       |      | 974.22                       |      |
| Revenue (\$/h)               | 1523.41                      | 1524.96                      | 1523.41                      |      | 1516.78                      |      |
| Profit (\$/h)                | 562.43                       | 564.11                       | 562.43                       |      | 542.56                       |      |

### 3.4 Summary

In this work, we propose a decomposition approach for the integrated scheduling and dynamic optimization for continuous cyclic production. Although the main focus is in reaction processes since this is the example used for demonstration, the approach is general to handle other processes that have the same basic behavior. The decomposition approach of continuous process was applied to the CSTR cyclic production with general kinetic models. Note that the decomposition for continuous case is feasible when certain condition (A.36) is satisfied. Through sensitivity analysis we prove that for continuous processes the production sequence is retained when demands are varying. This property is of particular importance in real applications where the demands are varying for different production cycles, since there is no need to update production sequence for the future cycles and the production can be updated by solving the sub-problem regarding production periods only. Results of case studies verify this property and also show that the decomposition approaches generate the same results as the integrated approach but using less computation time.

## Nomenclature:

### *Indices*

Products  $i$   
 Slots  $k$   
 Elements  $e$   
 States  $n$   
 Manipulated variables  $m$

### *Sets*

Product set  $I$   
 Slot set  $K$   
 Element set  $E$   
 State set  $S_x$   
 Manipulated variable set  $S_u$

### *Constants*

$N_p$  number of products  
 $N_s$  number of slots  
 $N_e$  number of elements within each slot  
 $N_x$  number of states  
 $N_u$  number of manipulated variables  
 $N_{spl}$  number of samples in the production cycle  
 $D_i$  demand rate for product  $i$   
 $C_i^p$  price of product  $i$   
 $C^r$  unit cost of raw material  
 $\theta^{\max}$  upper bound on processing time  
 $X_i$  conversion in steady state of continuous processes  
 $F^o$  feed stream volumetric flow rate  
 $x_{\min}^n, x_{\max}^n$  minimum and maximum value of  $n^{\text{th}}$  state variable  
 $u_{\min}^n, u_{\max}^n$  minimum and maximum value of  $m^{\text{th}}$  manipulated variable

### *Common variables*

$x$  state variables  
 $u$  manipulated variables  
 $q$  output variables  
 $y$  binary variable indicating assignment of production in slots  
 $z$  binary variable indicating transfer between products  
 $J$  objective function  
 $T_c$  cycle time  
 $W$  amount of product  
 $\tau_k^e$  ending time at slot  $k$   
 $\tau_k^s$  starting time at slot  $k$

*Continuous case*

$\Phi$  profit

$\Phi_1$  revenue

$\Phi_2$  cost

$COST^{tr}$  cost during transition period

$\overline{COST}_{pi}^{tr}$  estimation of transition cost from product  $p$  to  $i$

$COST^{ss}$  cost during steady state production

$\Theta$  production time of a product

$\theta^t$  transition time between products

$p_k$  process time at slot  $k$

$t_{ke}$  time point for each element in each slot

$u_{ke}^m$   $m$ th manipulated variable at element  $e$  of slot  $k$

$x_{ke}^n$   $n$ th state variable at element  $e$  of slot  $k$

$\bar{x}_k^n$  desired value of  $n^{\text{th}}$  state at slot  $k$

$\bar{u}_k^m$  desired value of  $m^{\text{th}}$  manipulated variable at slot  $k$

$x_{ss,i}^n$   $n^{\text{th}}$  steady state value of product  $i$

$u_{ss,i}^m$   $m^{\text{th}}$  steady manipulated value of product  $i$

$x_{in,k}^n$   $n^{\text{th}}$  state's initial value in slot  $k$

$u_{in,k}^m$   $m^{\text{th}}$  manipulated variable's initial value in slot  $k$

$h_k$  length of element in slot  $k$

$\theta_{ik}$  process time of product  $i$  at slot  $k$

$\theta_k^t$  transition time at slot  $k$

$\theta_{pi}^t$  transition time from product  $p$  to  $i$   $\tilde{\theta}_{pi}^t$  estimation of transition time from product  $p$  to  $i$

$f_{tc}$  transition cost

$\Theta_i$  total processing time of product  $i$

$K1, K2$  intermediate variables in implicit Runge-Kutta methods

$G_i$  production rate for product  $i$

$T_c$  total production cycle time

$W_i$  amount produced for product  $i$

$P$  revenue coefficient



## **Chapter 4 Integration of scheduling and control for batch processes using multi-parametric model predictive control (mp-MPC)**

### **4.1 Introduction**

In this study we propose to use Multi-Parametric Model Predictive Control (mp-MPC) (Bemporad et al., 2000) in the integration of scheduling and control, since it is capable in generating the control solution fast in the presence of disturbance by evaluating the function of pre-determined explicit solution. Model Predictive Control (MPC) is an online optimization technique based on a receding horizon mode (Lee, 2011). At each sample point the current state and output are measured and a constrained optimization problem is solved over a future time horizon to generate the optimal future control strategy. After the first control strategy is implemented to the process, the procedure is repeated in the following time point with the horizon moving forward. mp-MPC generates the control law as a set of explicit functions of state variables, via multi-parametric programming (Pistikopoulos, 2009). The explicit control law can then be obtained offline and the online optimization is reduced to simple function evaluations. Therefore mp-MPC results in faster application of MPC in larger scale problems.

In order to enable the integration of scheduling and control using this idea, we first linearize the non-linear dynamics and obtain the piece wise affine (PWA) approximation model. Then we apply the Multi-Parametric Toolbox (MPT) (M. Kvasnica et al., 2004) and obtain the explicit control solutions as functions of state variables. We then transform the explicit solutions into explicit linear constraints and incorporate them into the constraints of scheduling and obtain the integrated problem. We apply this approach in two batch processes, and the results demonstrate the feasibility and efficiency of the proposed methodology.

## 4.2 Modeling the integration of scheduling and control for batch processes

As outlined in the introduction section, scheduling and control are modeled simultaneously in this study. The dynamic behavior of batch processes is incorporated into the constraints of scheduling problem to form an integrated problem.

The scheduling problems for batch processes typically involve decisions associated with equipment assignment, amount of material used in each task, task sequence and task starting/ending times. The constraints are mainly composed of precedence constraints, duration of tasks, mass balance in units and demand fulfillment. Because infinite amount of raw material is assumed in our case studies, there are no constraints on raw material availability. A general mathematical model of the scheduling problem can be concisely presented as follows:

$$\max_{w, y, V, T} J(w, y, V, T) \quad (4.1)$$

$$s.t. \begin{cases} g_1(w, y, V, T) \geq \text{demand} \\ g_2(y) \geq \text{Constant} \\ V_{in} = V_{out} \\ T_f = T_i + T_s \end{cases} \quad (4.2)$$

where the vectors  $w$  and  $y$  stand for task and equipment assignment with respect to time slots or event points, respectively;  $V$  is the vector of amount of material used in different tasks; and  $T$  is the vector of processing times for tasks. The first constraint is to satisfy the demands and the second represents the task and unit assignments at slots or event points. The other two sets of constraints correspond to mass balances and duration constraints for each task or unit. The objective  $J(w, y, V, T)$  is to maximize the profit over a given time horizon.

The control problem focuses on the dynamic profile of tasks in batch processes. Constraints involve the material and energy balances, initial and end values as well as bounds of state and manipulated variables in each task, during processing. Optimal control strategies are desirable

since they generate transient profiles of state variables which are economically preferable. An appropriate control strategy is effective in improving the conversion or selectivity as well as saving raw materials and utility cost. A general form of the optimal control problem is as follows:

$$\max_{u(t)} J(x(t), u(t), q(t)) \quad (4.3)$$

$$s.t. \begin{cases} \dot{x}(t) = f(x(t), u(t), V, t) \\ q(t) = h(x(t), u(t), V, t) \\ (x(t), u(t), q(t)) \in \Omega_{xuq} \\ 0 \leq t \leq T \end{cases} \quad (4.4)$$

where  $x(t)$  is the vector of state variables such as concentration, temperature and reaction rate;  $u(t)$  is the set of manipulated variables such as heating or cooling flow rate;  $q(t)$  is the set of output (*i.e.* controlled variables) like conversion. Note that all these variables are changing with respect to time.  $\Omega_{xuq}$  is the set of bounds for  $x(t)$ ,  $u(t)$  and  $q(t)$ . Since safety is a priority, we enforce bound constraints for state variables e.g. temperature cannot exceed the upper bound. Vector  $V$  is the amount of raw materials involved in the reactions; and  $T$  is the processing time which can be either a variable or a known value provided by the scheduling level.

With simultaneous modeling, dynamic models are incorporated into the constraints of scheduling problem giving rise to the following integrated model:

$$\max_{w, y, V, T, u(t)} J(w, y, V, T, x(t), u(t), q(t)) \quad (4.5)$$

$$s.t. \begin{cases} g_1(w, y, V, T) \geq \text{demand} \\ g_2(y) \geq \text{Constant} \\ V_{in} = V_{out} \\ T_f = T_i + T_s \\ \dot{x}_i(t) = f_i(x_i(t), u_i(t), V_i, t) \\ q_i(t) = h_i(x_i(t), u_i(t), V_i, t) \\ (x_i(t), u_i(t), q_i(t)) \in \Omega_{xuq} \\ 0 \leq t \leq T_i \end{cases} \quad (4.6)$$

where index  $i$  represents time slots or event points, depending on whether time slots formulation or event point formulation is adopted in the scheduling problem,  $V_i$  and  $T_i$  are the linking variables between the scheduling and control problems. The integration of scheduling and control levels results in an overall optimization problem incorporating all the constraints and an objective function that takes into account all the decision variables in scheduling and control level, resulting in a better overall performance.

### **4.3 Simultaneous scheduling and control incorporating mp-MPC**

With integration of scheduling and control, the resulting MINLP is generally computational very expensive. In the literature in order to achieve better computational performance, Terrazas-Moreno et al. (Terrazas-Moreno et al., 2008a) applied Lagrangean decomposition to the integrated model and developed an iterative strategy between a master scheduling problem and a primal control problem. Their approach is effective in lowering computation time and achieving optimality of the integrated problem. Chu and You (Chu and You, 2013a) used generalized Bender decomposition to solve the MIDO of the integrated problem and observed significant reduction of computation time. In chapter 3 we explore the structure of the integrated optimization problem for continuous processes cyclic production, and establish an efficient decomposition scheme in which the production sequence could be separated from the integrated problem when certain conditions are satisfied.

As described in the previous section, the online implementation of integrated scheduling and control requires a repetitive solution of the resulting MINLP. In this study we propose to use mp-MPC in order to not only reduce the complexity of the integrated problem but also reduce the calculation in online application.

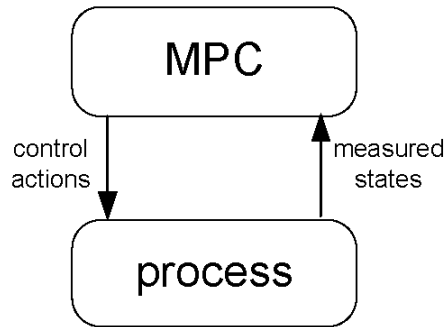
A brief summary of the parametric programming is provided here which serves as a basis for mp-MPC. Problem (4.7) describe a parametric programming problem which obtains the objective  $z$  as a function of parameters  $x$ , and Equation (4.8) describes the corresponding critical regions.

$$\begin{aligned}
 z(x) &= \min_u f(u, x) \\
 \text{s.t. } g(u, x) &\leq 0 \\
 x &\in \Omega_x \\
 u &\in \Omega_u
 \end{aligned} \tag{4.7}$$

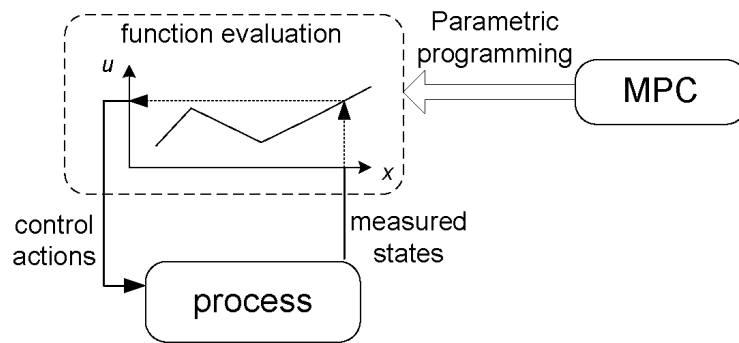
$$\begin{aligned}
 &\begin{cases} z(x) = z^i(x) \\ u(x) = u^i(x) \end{cases} \quad \text{when } x \in \Omega_x^i \\
 &\bigcup_i \Omega_x^i = \Omega_x \\
 &\Omega_x^i \cap \Omega_x^j = \emptyset
 \end{aligned} \tag{4.8}$$

The solution set of parametric programming includes the objective and decision variables as a function of the parameters and the partition of the space of parameters. A certain partition produces critical regions where the optimal solutions are valid. According to the features of the objective and constraints the parametric programming problem can be categorized as mp-LP, (Gal and Nedoma, 1972), (Gal, 1975) mp-MIQP (Dua et al., 2002), mp-MILP (Dua and Pistikopoulos, 2000), mp-MINLP (Dua and Pistikopoulos, 1999), mp-NLP (Domínguez et al., 2010).

MPC is widely recognized as it's capable to repetitively solve the optimization problem that accounts for a future horizon in online application (Figure 4.1) (Lee, 2011). Mp-MPC on the other hand parameterizes the states and solves the optimization problem in MPC using parametric programming, and performs function evaluation in online application (Figure 4.2) (Pistikopoulos, 2009), (Kouramas et al., 2011). In a sense mp-MPC transfers the online computation into offline computation, effectively reducing the computational burden in online application.



**Figure 4.1** The working mode of conventional MPC



**Figure 4.2** The working mode of mp-MPC

In this work we apply mp-MPC in the online implementation for the simultaneous solution of scheduling and control, and solve the mp-MPC problem using MPT toolbox (M. Kvasnica et al., 2004). As shown in Figure 4.3 the dynamic optimization at the control level is solved offline using MPT toolbox. The obtained explicit control solutions are incorporated into scheduling problem, resulting in a simplified integrated problem i.e. a MINLP whose nonlinearity only present in the objective.

Specifically four steps are involved in implementing this approach.

Step 1: Linearize the original dynamic model using PWA approximation, which is a common approximation for Lipschitz continuous nonlinear dynamics (Sontag, 1981), (Azuma et al., 2006).

Step 2: Solve the control problem for the derived PWA using MPT toolbox and obtain the explicit solutions for the control problem.

Step 3: Transform the explicit solutions into explicit linear constraints by introducing additional variables.

Step 4: Incorporate the constraints obtained in Step 3 into the constraints of scheduling problem and build an overall economic objective, which is calculated as the revenue minus raw material and utility costs.

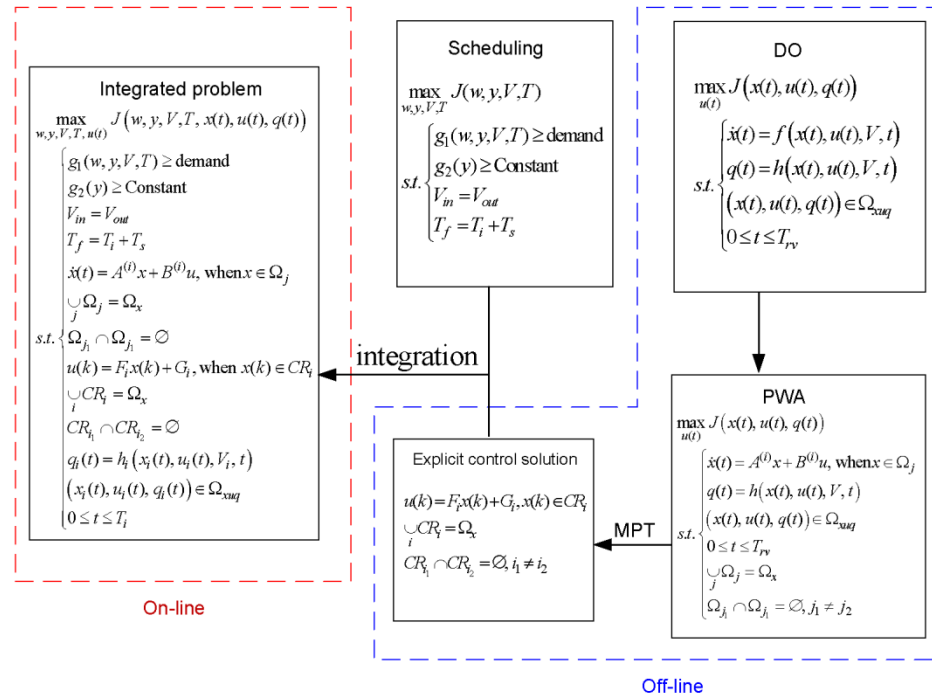
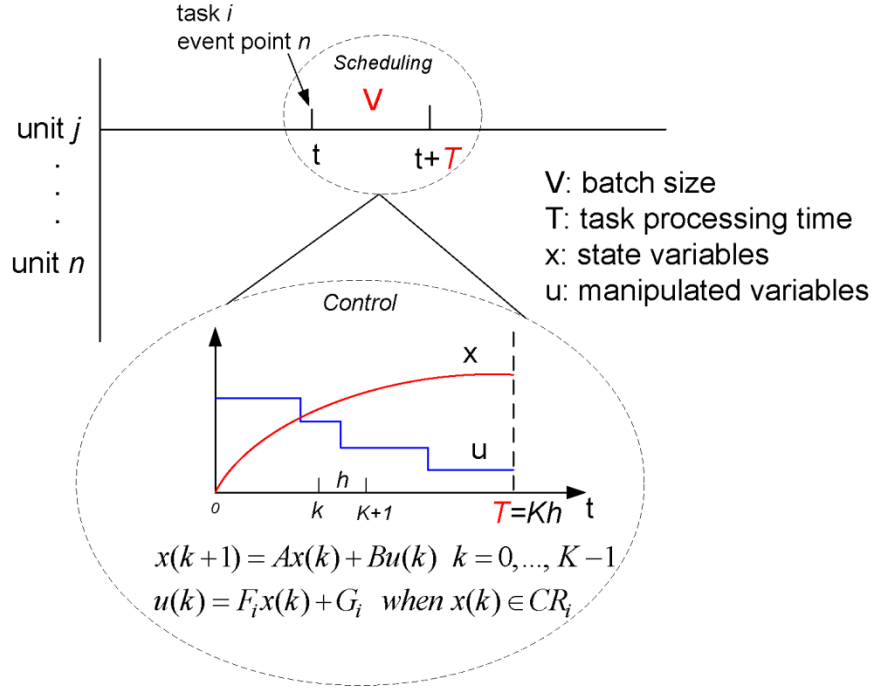


Figure 4.3 A scheme of integrated scheduling and control using MPT toolbox

#### 4.4 Detailed integrated model of scheduling and control incorporating mp-MPC

Figure 4.4 demonstrates how the scheduling and control level are connected and how they are integrated. Let's assume that task  $i$  is executed in unit  $j$  at event point  $n$ . The dynamic behavior of the manipulated variables  $u$  and state variables  $x$  are the focus of the control problem. It can be

observed that the amount of material  $V$  and processing time  $T$  are shared as common variables at scheduling and control levels. Therefore it is essential to integrate these two levels and handle the shared information simultaneously.



**Figure 4.4 Demonstration of the integration using event point based scheduling formulation**

#### 4.4.1 Constraints at scheduling level

The scheduling model follows the event point based formulation proposed by Ierapetritou and Floudas (Ierapetritou and Floudas, 1998). The model involves the constraints that are described below.

##### Allocation Constraints

For each event point  $n$ , only one task can take place in unit  $j$  if unit  $j$  is suitable for task *i.e.*

$i \in I_j$   $y(j,n)=1$  indicates that unit  $j$  is utilized at event point  $n$ . If  $y(j,n)=0$ , then all  $w(i,n)$  are forced to be zero *i.e.* neither task nor unit is assigned in event point  $n$ .



$$\sum_{i \in I_j} w(i, n) = y(j, n), \quad \forall j \in J, n \in N \quad (4.9)$$

#### Capacity Constraints

The material undertaken in the unit should be greater than the minimum requirement of material and meanwhile less than the capacity of the unit. For a trivial case, if  $w(i, n)=0$ , then  $B(i, j, n)=0$ , and the following inequality still hold.

$$V_{ij}^{\min} w(i, n) \leq V(i, j, n) \leq V_{ij}^{\max} w(i, n), \quad \forall i \in I, j \in J_i, n \in N \quad (4.10)$$

#### Storage Constraints

The intermediate material at any state stored in storage tank at event point  $n$  is limited by the tank capacity. In the case that unlimited intermediate storage (UIS) is assumed,  $ST(s)^{\max}$  goes to infinity.

$$ST(s, n) \leq ST(s)^{\max}, \quad \forall s \in S, n \in N \quad (4.11)$$

#### Material Balance

The amount of material of state  $s$  at event point  $n$  is equal to the sum of the material amount at the previous event point and the amount produced at the previous event point subtracted by the amount delivered to the market and amount consumed by the unit at the current event point.

$$\begin{aligned} ST(s, n) = ST(s, n-1) - d(s, n) + \sum_{i \in I_s} \rho_{si}^p \sum_{j \in J_i} V(i, j, n-1) \\ - \sum_{i \in I_s} \rho_{si}^c \sum_{j \in J_i} V(i, j, n), \quad \forall s \in S, n \in N \end{aligned} \quad (4.12)$$

#### Duration Constraints

The processing time of task  $i$  at unit  $j$  is calculated as the sum of a fixed term  $\alpha_{ij}$  and a varying term  $\beta_{ij} V(i, j, n)$  which is linearly increasing with the amount of material.

$$T^f(i, j, n) = T^s(i, j, n) + \alpha_{ij}w(i, n) + \beta_{ij}V(i, j, n),$$

$$\forall i \in I, j \in J_i, n \in N$$
(4.13)

When the process dynamics are integrated, the processing time of tasks ( $T_{rv}$ ) should be variables as shown in (4.14) rather than a proportional term in (4.13)

$$T^f(i, j, n) = T^s(i, j, n) + \alpha_{ij}w(i, n) + T_{rv}(i, j, n)y(j, n),$$

$$\forall i \in I, j \in J_i, n \in N$$
(4.14)

Sequence Constraints: same task in the same unit

Task  $i$  starting at event point  $n+1$  should start after the end of the same task processed at unit  $j$  but started at event point  $n$ . If  $w(i, j) = y(i, j) = 0$ , (4.15) is trivially satisfied.

$$T^s(i, j, n+1) \geq T^f(i, j, n) - H(2 - w(i, n) - y(j, n)),$$

$$\forall i \in I, j \in J_i, n \in N, n \neq N$$
(4.15)

$$T^s(i, j, n+1) \geq T^s(i, j, n), \forall i \in I, j \in J_i, n \in N, n \neq N$$
(4.16)

$$T^f(i, j, n+1) \geq T^f(i, j, n), \forall i \in I, j \in J_i, n \in N, n \neq N$$
(4.17)

Sequence Constraints: different tasks in the same unit

Task  $i$  starting at event point  $n+1$  in unit  $j$  should start after the end of the other tasks processed at unit  $j$  but started at the prior event point.

$$T^s(i, j, n+1) \geq T^f(i', j, n) - H(2 - w(i', n) - y(j, n)),$$

$$\forall i \in I, i' \in I, i \neq i', j \in J, n \in N, n \neq N$$
(4.18)

Sequence Constraints: different tasks in different units

Tasks starting at event point  $n+1$  should start after the end of tasks starting at event point  $n$ , for whichever units the tasks are processed in.

$$T^s(i, j, n+1) \geq T^f(i', j', n) - H(2 - w(i', n) - y(j', n)),$$

$$\forall j, j' \in J, i \in I_j, i' \in I_j, i \neq i', n \in N, n \neq N \quad (4.19)$$

#### Sequence Constraints: Completion of Previous Tasks

Task should start after all the previous tasks in the same unit are finished.

$$T^s(i, j, n+1) \geq \sum_{n' \in N, n' < N} \sum_{i' \in I_j} (T^f(i', j, n') - T^s(i', j, n')),$$

$$\forall i \in I, j \in J_i, n \in N, n \neq N \quad (4.20)$$

#### Time Horizon Constraints

The starting time and ending time for each task should be within the time horizon.

$$T^f(i, j, n) \leq H, \forall i \in I, j \in J_i, n \in N \quad (4.21)$$

$$T^s(i, j, n) \leq H, \forall i \in I, j \in J_i, n \in N \quad (4.22)$$

#### **4.4.2 Constraints at control level**

The control strategy in batch processes involves generating time-varying profiles of manipulated and state variables. For example, when a batch reactor is in operation temperature plays an important role in affecting reaction rate and conversion. In order to achieve high conversion as well as reduce the heating or cooling utility cost, a comprehensive objective involving the process yield and utility cost is usually adopted.

A first principle model based on material and energy balances describes the system dynamic of batch process. The state variables include concentration, temperature, and pressure; the manipulated variables are heating or cooling flow rate, and feeding flow rate in semi-batch processes. Following is a general form of such a model.

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ q(t) &= h(x(t), u(t)) \\ x(0) &= x_0, u(0) = u_0 \end{aligned} \quad (4.23)$$

where  $x$  represents the vector of state variables,  $q$  is the output, and  $u$  is the vector of manipulated variables.

During batch operations the process variables are undergoing transient state. Therefore, unlike the continuous case in which an optimal constant set-point is obtained, the objective in batch operation is to determine the optimal transient profile which maximizes or minimizes an economic performance including revenue, material cost and utility cost.

A general form of objective function is

$$\max_{u(t)} J = \Phi(x(t_f)) + \int_0^{t_f} L(x(t), u(t)) dt \quad (4.24)$$

where  $J$  is the overall profit to be maximized,  $t_f$  is final time; and  $\Phi$  is the profit at the final time point. The integral term is a comprehensive form involving the profit due to production and the utility cost with respect to the entire transient period.

Constraints including the process dynamics (4.23) as well as safety constraints including bounds for temperature and other important state variables (4.25), and unit operation specifications such as the maximum valve position:

$$\begin{aligned} x_{\min} &\leq x(t) \leq x_{\max} \\ u_{\min} &\leq u(t) \leq u_{\max} \end{aligned} \quad (4.25)$$

#### 4.4.3 The linking variables and constraints

After solving the mp-MPC problem we obtain the state transitions at each sample step Equations (4.26)-(4.28) and the manipulated variables as linear functions of the states Equations (4.29)-(4.31). These constraints describe the dynamic profiles of manipulated variables and state variables.

#### PWA approximation of the kinetic model

$$\begin{aligned} x(k+1) &= A_j x(k) + B_j u(k) + C_j, \\ \text{if } x(k) &\in \Omega_j, k \in K = \{1, 2, \dots, N_K\} \end{aligned} \quad (4.26)$$

where

$$\Omega_j = \{x : V_j x \leq W_j\}, j \in J = \{1, 2, \dots, N_J\} \quad (4.27)$$

and  $\Omega_j$  satisfies the following conditions which enforce that  $\{\Omega_1, \Omega_2, \dots, \Omega_{N_J}\}$  represents a complete and non-overlapping partition of  $\Omega_x$ .

$$\begin{aligned} \bigcup_j \Omega_j &= \Omega_x \\ \Omega_{j_1} \cap \Omega_{j_2} &= \emptyset, j_1, j_2 \in J \text{ and } j_1 \neq j_2 \end{aligned} \quad (4.28)$$

Explicit control solutions obtained by solving mp-MPC problem using the MPT toolbox

$$u(k) = F_i x(k) + G_i, \text{ if } x(k) \in CR_i, k \in K \quad (4.29)$$

where

$$CR_i = \{x : H_i x \leq K_i\}, i \in I = \{1, 2, \dots, N_I\} \quad (4.30)$$

and  $CR_i$  satisfies the following constraints.

$$\begin{aligned} \bigcup_i CR_i &= \Omega_x \\ CR_{i_1} \cap CR_{i_2} &= \emptyset, i_1, i_2 \in I \text{ and } i_1 \neq i_2 \end{aligned} \quad (4.31)$$

Equations (4.26)-(4.31) are implicitly linear, because the linear functions are valid in certain regions. We need to transform them into explicit linear constraints Equations (4.32)-(4.37).

Binary variables  $y1$  are introduced to select the critical region in (4.30). If  $y1_i = 1$  this means that  $x$  is located in region  $i$  while if  $y1_i = 0$   $x$  is not at this region.  $M$  is a big positive number which relaxes constraints inequality (4.32) when  $y1_i = 0$ . Constraint (4.33) enforces that only one region is valid for a certain pair of  $H_i, K_i$ .

$$-M(1 - y1_i) + H_i x(k) \leq K_i \quad (4.32)$$

$$\sum_i y1_i = 1 \quad (4.33)$$

Using these binary variables, (4.29) is transformed into the following constraint:

$$F_i x(k) + G_i - M(1 - y1_i) \leq u(k) \leq F_i x(k) + G_i + M(1 - y1_i) \quad (4.34)$$

Similarly variables  $y2$  are introduced to select the region (4.27) where the specific linearization is valid

$$-M(1 - y2_j) + V_j x(k) \leq W_j \quad (4.35)$$

$$\sum_j y2_j = 1 \quad (4.36)$$

(4.26) is then transformed into the following:

$$\begin{aligned} A_j x(k) + B_j u(k) + C_j - M(1 - y2_j) &\leq x(k+1) \leq \\ A_j x(k) + B_j u(k) + C_j + M(1 - y2_j) \end{aligned} \quad (4.37)$$

Since the dynamic period is discretized, the processing time  $T_{rv}$  can be calculated in Equation (4.38) where  $h$  is the step length, and  $N_k$  is the total number of steps.

$$T_{rv} = N_k h \quad (4.38)$$

Both manipulated and state variables are confined by their corresponding bounds in inequalities (4.39).

$$\begin{aligned} u_{low} &\leq u \leq u_{up} \\ x_{low} &\leq x \leq x_{up} \end{aligned} \quad (4.39)$$

Note that all the derived constraints at control level and the constraints at scheduling level are linear. Thus the integrated problem is simplified from the original MINLP.

#### 4.4.4 The objective of the integrated problem

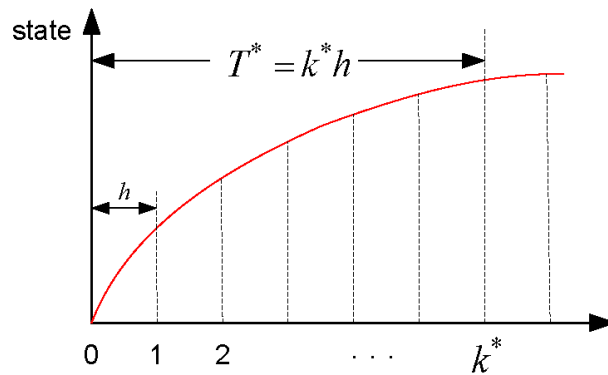
The objective of the integrated problem is to maximize the overall profit by obtaining the optimal scheduling and control solutions. Thus the objective corresponds to revenue minus raw material and utility cost as follows.

$$J = C^P V X(k) - C^r V - C^u V \sum_{k'=1}^{k-1} u(k') \quad (4.40)$$

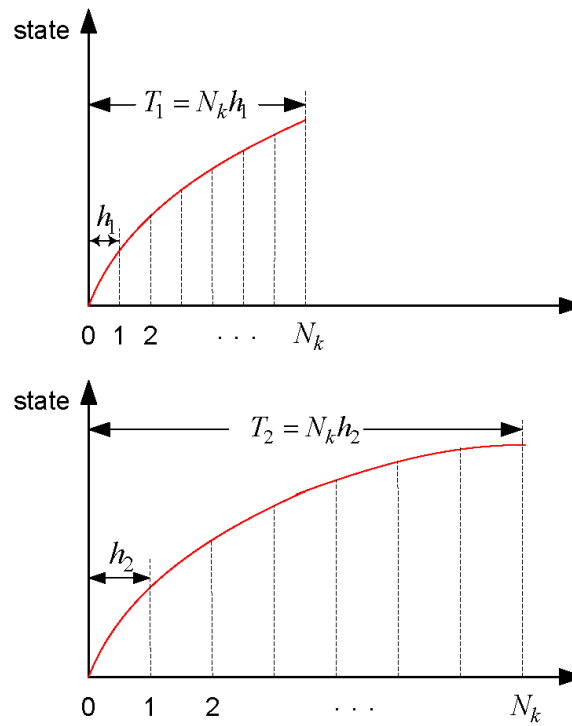
where  $X$  is the conversion,  $u$  is the scaled temperature corresponding to utility consumption (Nie et al., 2012),  $C^P$ ,  $C^r$  and  $C^u$  are the product price, raw material price and utility price, respectively.

The dynamic model is discretized with fixed step size as shown in Figure 4.5, and in the control problem the optimal number of steps is determined. Note that the index  $k$  in (4.40) is a decision variable and  $X$  and  $u$  are defined over  $k$ , making  $k$  an implicit decision variable. In order to eliminate this complexity we introduce an alternative formulation given by Equation (4.41) that eliminates index  $k$  from the objective using varying step size  $h$  and fixed number of steps  $N_k$ , as shown in Figure 4.6. The number of sample points  $N_k$  is constant and the step size  $h$  and the processing time  $T_l$  are variables. Figure 4.6 demonstrates two cases with different processing times but identical number of sample steps. It can be observed that a varying step size enables a flexible processing time with a constant sample size. Therefore solving the integrated problem generates the optimal step size and the corresponding processing time.

$$J = C^P V X(N_k) - C^r V - C^u V \sum_{k'=1}^{N_k-1} u(k') \quad (4.41)$$



**Figure 4.5 Discretization with fixed step size**



**Figure 4.6 Discretization with varying step size but fixed number of steps**

## 4.5 Case studies

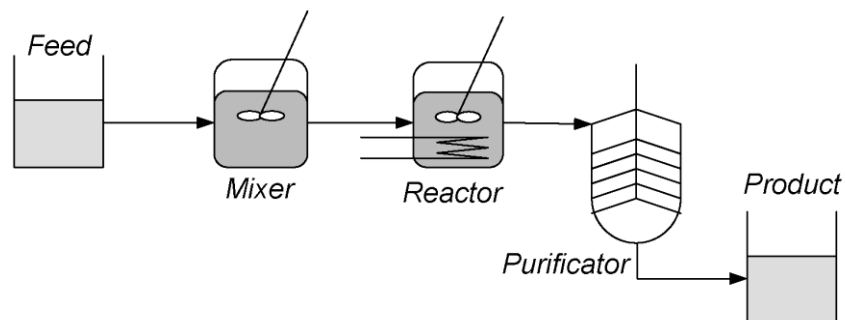
In this section we consider two case studies that have been extensively studied in the scheduling literature and compare the results of the integrated methodology presented above with the approach that considers the scheduling and control problems separately. In this later approach, we solve the scheduling problem first and then solve the control problem based on the solution of the scheduling problem. The overall objective for the optimization problem is the maximization of profit, which is defined as the revenue from selling the products minus raw material cost, equipment cost, and utility cost. The recipe and data set of the case studies are the ones used at Ierapetritou and Floudas (Ierapetritou and Floudas, 1998).

### 4.5.1 A simple batch process

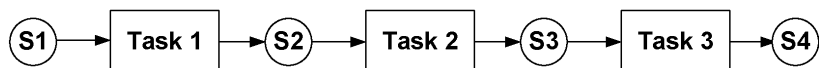
As shown in Figure 4.7 a batch process produces a single product through three consecutive processing stages i.e. mixing, reaction and separation. The process is represented by the state-task



network (STN) in Figure 4.8. Information regarding the capacity of units, suitability, price of material is provided in Table 4.1. In this case No Intermediate Storage (NIS) and Zero Wait (ZW) policies are assumed.



**Figure 4.7** Flow sheet of a simple batch process



**Figure 4.8** State task network of a simple batch process

**Table 4.1** Data for unit specification and market information of example 1: a simple batch process

| Unit              | Capacity (L)         | Suitability        | Mean processing time (h) |
|-------------------|----------------------|--------------------|--------------------------|
| Mixer             | 100                  | task 1             | 4.5                      |
| Reactor           | 75                   | task 2             | 3.0                      |
| Purificator       | 50                   | task 3             | 1.5                      |
| State and Utility | Storage capacity (L) | Initial amount (L) | Price (\$/L)             |
| state 1 (Raw)     | unlimited            | unlimited          | 0.2                      |
| state 2           | 100                  | 0.0                | 0.0                      |
| state 3           | 100                  | 0.0                | 0.0                      |
| state 4 (Product) | unlimited            | 0.0                | 1.0                      |
| Utility           | unlimited            | 0.0                | 0.05                     |

In this study a simplified reaction kinetic is assumed.  $u$  is a scaled temperature which is defined to substitute the rate constant and (4.42) is an approximation of the kinetic model (Nie et al., 2012). The utility consumption is equal to an integral of  $u$  with respect to time. In this case the reaction is assumed to be endothermic, so higher temperature drives the concentration of reactants to decrease fast, as indicated in (4.42).

$$\frac{dx}{dt} = -ux \quad (4.42)$$

As mentioned in section “Simultaneous scheduling and control incorporating mp-MPC” we transform the nonlinear model into PWA as follows.

Let  $f(u, x) = dx/dt$ , and linearize it at  $x = x_0, u = u_0$

$$\begin{aligned} \frac{dx}{dt} &= f(u, x) \\ &= f(u_0, x_0) + (u - u_0)f_u(u_0, x_0) + (x - x_0)f_x(u_0, x_0) \\ &= -x_0u - u_0x + u_0x_0 \end{aligned} \quad (4.43)$$

$dx/dt = (x(k+1) - x(k))/h$ , where  $h$  is the sample step. Thus we obtain

$$\frac{x(k+1) - x(k)}{h} = -x_0u - u_0x + u_0x_0 \quad (4.44)$$

leading to the following equation

$$x(k+1) = (1 - u_0h)x(k) - x_0hu(k) + u_0x_0h \quad (4.45)$$

which pertains to the general form as follows:

$$x(k+1) = Ax(k) + Bu(k) + C \quad (4.46)$$

In this case we linearize the dynamic around seven points:  $u_0=1, x_0=[3, 2.5, 2, 1.5, 1, 0.5, 0]$ .

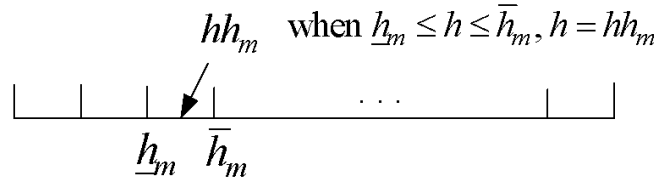
Handle the nonlinearity brought by  $h$

For a control problem with fixed sample step  $h$ , (4.45) corresponds to a linear problem. However, in this study we consider the unit processing time as a variable ( $T_{rv}$  in (4.14) and  $T_1, T_2$  in Figure 4.6) and since we use fixed number of sample points (Figure 4.6), the sample step is also a variable which transforms (4.46) to a nonlinear equation.

From (4.45) and (4.46) we obtain

$$\begin{aligned} A &= 1 - u_0 h \\ B &= -x_0 h \\ C &= u_0 x_0 h \end{aligned} \tag{4.47}$$

In order to avoiding the nonlinearity in (4.46), we discretize  $h$  as shown in Figure 4.9 and introduce binary variables to represent the selection of  $h$ .



**Figure 4.9 Discretization of step size  $h$**

(4.48) and (4.49) represent the selection of  $h$  in the discretized segment.

$$-M(1 - y_{3_m}) + H3_m h \leq K3_m \tag{4.48}$$

$$\sum_m y_{3_m} = 1 \tag{4.49}$$

Therefore (4.37) is transformed into (4.50).

$$\begin{aligned} (1 - u_0 h h_m) x(k) - x_{0j} h h_m u(k) + u_0 x_{0j} h h_m - M(1 - y_{2_j}) - M(1 - y_{3_m}) \leq \\ x(k+1) \leq (1 - u_0 h h_m) x(k) - x_{0j} h h_m u(k) + u_0 x_{0j} h h_m + M(1 - y_{2_j}) + M(1 - y_{3_m}) \end{aligned} \tag{4.50}$$

Handle the nonlinearity in (4.14)

Since in (4.14) both  $T_{rv}$  and  $y$  are decision variables, the last term corresponds to a bilinear term.

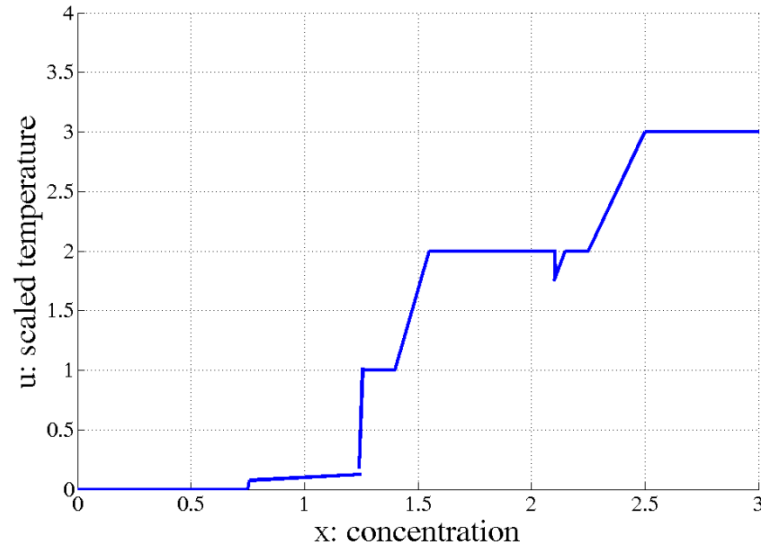
In order to eliminate the nonlinearity we introduce an alternative form as follows:

$$T^f(i, j, n) = T^s(i, j, n) + \alpha_{ij}w(i, n) + T_{rv}(i, j, n) \quad (4.51)$$

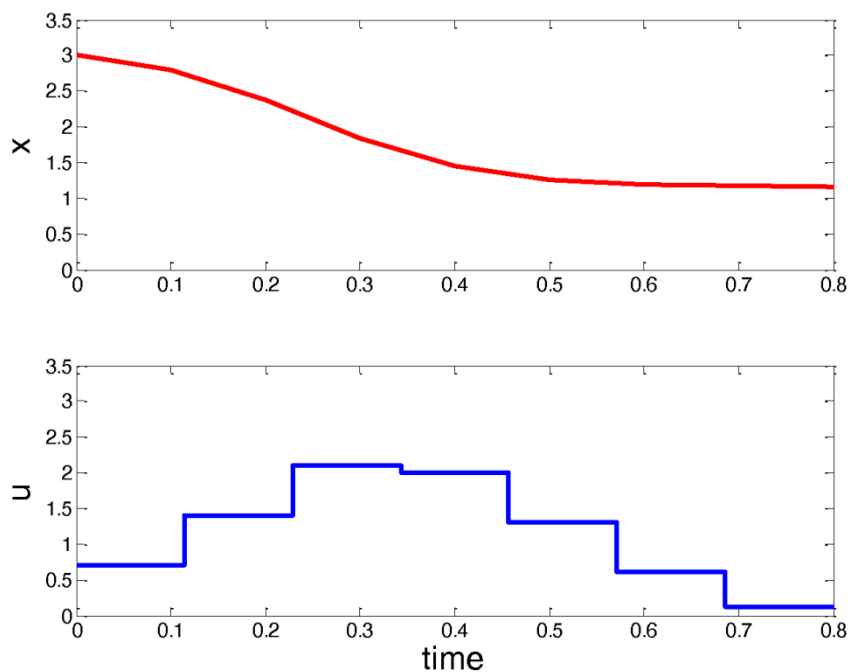
$$\underline{T}y(j, n) \leq T_{rv}(i, j, n) \leq \bar{T}y(j, n) \quad (4.52)$$

Where  $y(j, n) = 1$ , unit  $j$  is selected for processing and  $T_{rv}$  is subject to the lower and upper bounds. Otherwise the unit is not selected and the processing time should be zero.

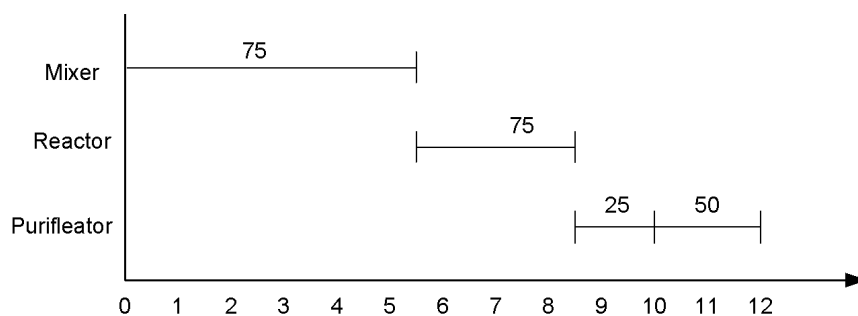
The explicit control solutions generated by mp-MPC are shown in Figure 4.10. The integrated problem (MINLP) has 2582 variables and 17050 constraints and it takes 218.71s to solve using GAMS/SBB on a 3.0GHz CPU/1.0GB RAM PC. Solving the integrated problem we obtain the dynamic profile at control level (Figure 4.11) and the scheduling solution in Figure 4.12.



**Figure 4.10 Explicit solution for mp-MPC**



**Figure 4.11** Dynamic profile of the reactor.  $x$  represents the concentration of raw material and  $u$  represents the scaled temperature



**Figure 4.12** Scheduling solution for example 1 a simple batch process

#### 4.5.2 A more complex batch process

As shown in Figure 4.13 this problem's STN represents a process that is capable of producing two products through five processing stages: heating, reaction 1, 2 and 3, and separation of product 2 from impure E. The material flow in this process indicates that two cascade reactions (reaction 1 and 2) are involved in producing product 1. It is reasonable to assume that the overall

conversion is calculated by multiplying the conversions of the related reactions. The problem time horizon is 8 hours. Equipment specification and price information are provided in Table 4.2.

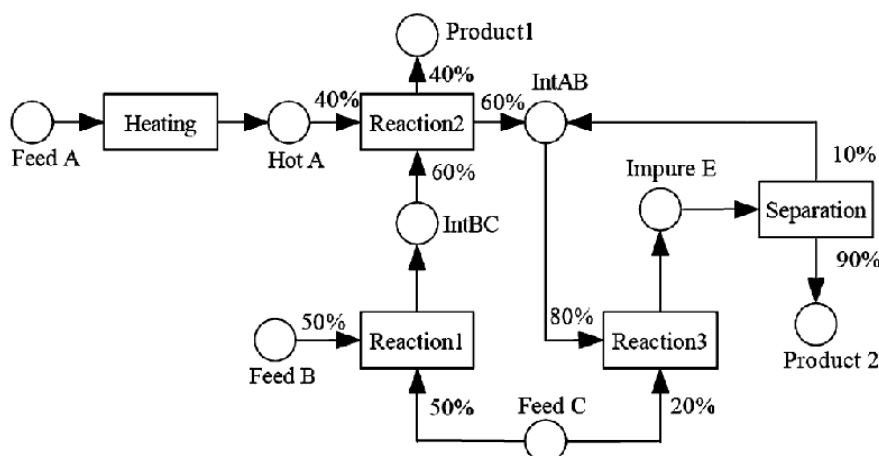


Figure 4.13 Flow sheet for a complex batch process

Table 4.2 Data for unit specification and market information of example 2: a complex batch process

| Unit              | Capacity ( <i>L</i> )         | Suitability                 | Mean processing time ( <i>h</i> ) |
|-------------------|-------------------------------|-----------------------------|-----------------------------------|
| Heater            | 100                           | Heating                     | 1.0                               |
| reactor 1         | 50                            | Reaction 1,2,3              | 2.0, 2.0, 1.0                     |
| reactor 2         | 80                            | Reaction 1,2,3              | 2.0, 2.0, 1.0                     |
| distillation      | 200                           | separation                  | 1 for product 2, 2 for Int AB     |
| State and utility | Storage capacity ( <i>L</i> ) | Initial amount ( <i>L</i> ) | Price (\$/ <i>L</i> )             |
| feed A            | unlimited                     | unlimited                   | 0.2                               |
| feed B            | unlimited                     | unlimited                   | 0.2                               |
| feed C            | unlimited                     | unlimited                   | 0.2                               |
| hot A             | 100                           | 0.0                         | 0.0                               |
| Int AB            | 200                           | 0.0                         | 0.0                               |
| Int BC            | 150                           | 0.0                         | 0.0                               |
| impure E          | 200                           | 0.0                         | 0.0                               |
| product 1         | unlimited                     | 0.0                         | 5.0                               |

|           |           |     |      |
|-----------|-----------|-----|------|
| product 2 | unlimited | 0.0 | 1.0  |
| utility   | unlimited | 0.0 | 0.05 |

Compared to the previous case, this process features cascade reactions and multiple raw materials multiple products. Thus a product of conversions  $VX_1X_2$  is present in the objective (4.53).

$$J = C^{P_1} V_1 X_1(N_k) X_2(N_k) + C^{P_2} V_3 X_3(N_k) - \sum_i C^{r_i} V_i - C^u \sum_j V_j \sum_{k'} u_j(k') \quad (4.53)$$

When building the integrated model in order to make the problem computationally tractable without losing essential parts of the proposed modeling approach, we made the following assumptions in this study.

Assumption 1:

A generalized kinetic model that relates scaled temperature ( $u$ ) and conversion ( $X$ ).

A first order reaction kinetics is assumed:

$$\frac{dC}{dt} = -uC \quad (4.54)$$

Using the relation between concentration and conversion  $C = C_0(1 - X)$ , we obtain:

$$\frac{dX}{dt} = u(1 - X) \quad (4.55)$$

We simplify the specific kinetic of the all the reactions and use (4.55) as a general form. The amounts of product 1 and product 2 are then given by  $VX_1X_2$  and  $VX_3$ , respectively. Utility amount is equal to  $Vu$ .

Assumption 2:

The conversions for the tasks, which belong to the same reaction but are executed at different reactors are equal. For example, the conversion for tasks 2 and 3 are equal. This avoids mixing the material with different conversion in storage IntBC.

Assumption 3:

In order to satisfy the products' quality specification, we assign lower bounds for the conversion of reaction 1, 2 and 3:  $X_1 > 0.6$ ,  $X_2 > 0.5$  and  $X_3 > 0.3$ .

In this case study we investigate four scenarios, and compare their detailed results.

Scenario 1: The integrated problem (MINLP) involving the nonlinear objective in (4.53) and the linear constraints shown in formula (4.56). This is the proposed methodology where mp-MPC is incorporated in this study.

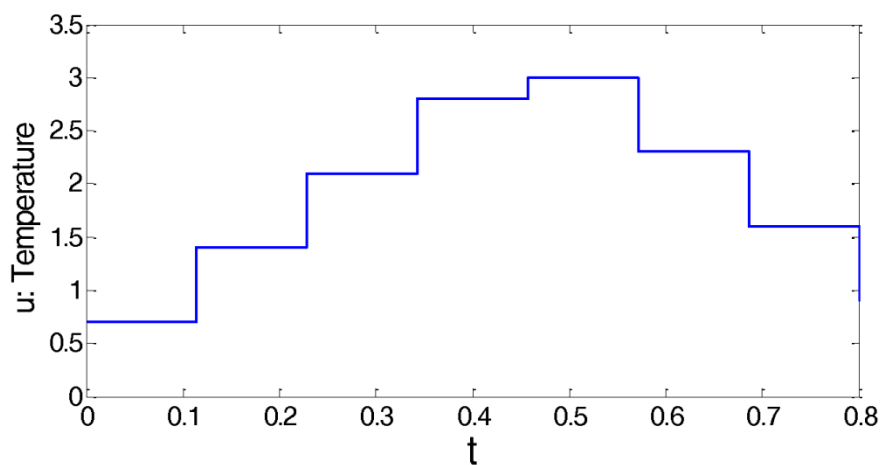
$$\begin{aligned}
 \max_{\substack{w,y,V,d,ST,T^s,T^f \\ T_{rv},x,u,X,y1,y2}} J = & C^{P_1} V_1 X_1(N_k) X_2(N_k) + C^{P_2} V_3 X_3(N_k) \\
 & - \sum_{i \in I_R} C^{r_i} V_i - C^u \sum_j V_j \sum_{k'} u_j(k') \\
 \text{s.t. } & \begin{cases} (4.9)-(4.12) \text{ scheduling constraints} \\ (4.15)-(4.22) \text{ scheduling constraints} \\ (4.32)-(4.39) \text{ explicit MPC} \\ (4.48)-(4.50) \text{ handle the nonlinearity in (4.45)} \\ (4.51)-(4.52) \text{ handle the nonlinearity in (4.14)} \end{cases}
 \end{aligned} \tag{4.56}$$

Scenario 2: Directly apply the explicit control solution produced by mp-MPC to the scheduling solution generated by the pure scheduling problem in (Ierapetritou and Floudas, 1998). In the pure scheduling problem involving constraints (4.9)-(4.13) and (4.15)-(4.22), the objective is to maximize the throughput of the process with empirical estimation of the processing time in (4.13). The optimization problem for scenario 2 is presented in (4.57).

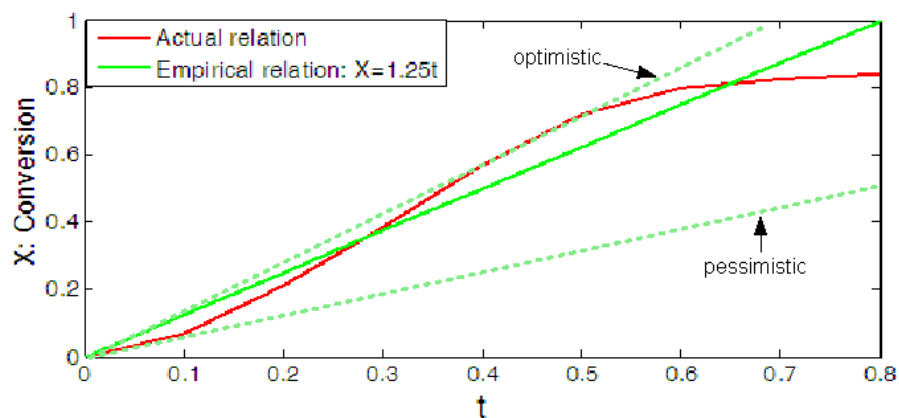
$$\begin{aligned}
 \max_{w,y,V,d,ST,T^s,T^f} J = & \sum_n d(s = \text{"product1"}, n) \\
 & + \sum_n d(s = \text{"product2"}, n) \\
 \text{s.t. } & \begin{cases} (4.9)-(4.13) \text{ scheduling constraints} \\ (4.15)-(4.22) \text{ scheduling constraints} \end{cases}
 \end{aligned} \tag{4.57}$$

Scenario 3: Solve the control problem using mp-MPC and apply the obtained explicit control solutions to the dynamic and obtain the operation conditions (Figure 4.14-Figure 4.16). The approximated relations are incorporated into the scheduling constraints.

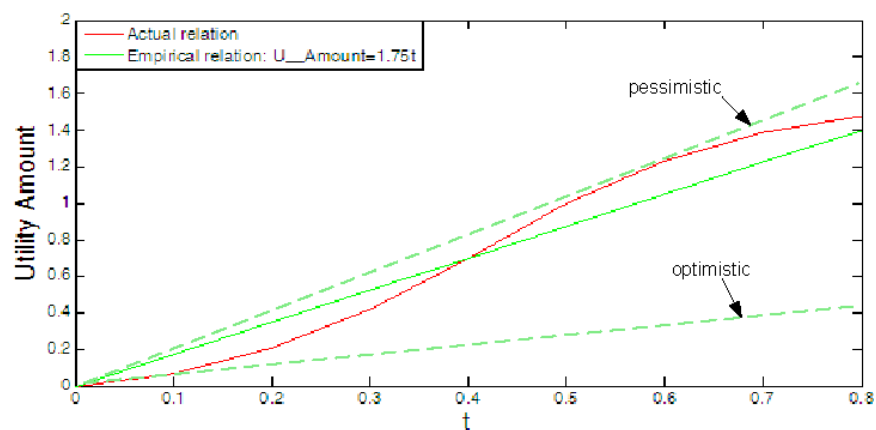




**Figure 4.14** Temperature profile obtained by applying the explicit control solution to the reaction



**Figure 4.15** Conversion profile and the approximation (empirical relation)



**Figure 4.16** Utility amount profile and the approximation (empirical relation)

From Figure 4.15 we obtain the empirical relation between reaction time and conversion: the required reaction time is proportional to the desired conversion.

$$T_{rv} = \beta X \quad (4.58)$$

Thus we have the following time constraint which replaces the original constraints in (4.13)

$$T^f = T^s + \alpha + \beta X \quad (4.59)$$

By combining the empirical relations in Figure 4.15 and Figure 4.16, we obtain the empirical relation between utility amount and the conversion: the utility consumption is proportional to the desired conversion.

$$\text{Utility amount} = \int_0^{T_{rv}} u(t) dt = \gamma X \quad (4.60)$$

Based on the above, we have the optimization problem for scenario 3.

$$\begin{aligned} \max_{w,y,V,d,ST,T^s,T^f,X} \quad & J = C^{P_1} V_1 X_1 X_2 + C^{P_2} V_3 X_3 \\ & - \sum_{i \in I_R} C^{\bar{n}} V_i - C^u \sum_{i \in I_R} V_i \gamma X_i \\ \text{s.t.} \quad & \begin{cases} T^f = T^s + \alpha + \beta X & \text{empirical time constraint} \\ (4.9)-(4.12) & \text{scheduling constraints} \\ (4.15)-(4.22) & \text{scheduling constraints} \end{cases} \end{aligned} \quad (4.61)$$

This is a small scale MINLP problem which involves a nonlinear objective function and linear constraints. The formulation in (4.61) realizes an implicit integration (compared to the explicit integration in scenario 1) of scheduling and control since it incorporates empirical relations of the operation conditions obtained in the control problem. Note that the implicit integration does not bring extra constraints to the scheduling problem but replaces constraint (4.13) with (4.59).

Scenario 4: In this scenario we build an integrated problem (MIDO) and discretize it into a MINLP using implicit Runge-Kutta method (Frank et al., 1985) which has a general form in equations (4.62) and (4.63).

$$x(k+1) = x(k) + b_1 K1 + b_2 K2 + \dots + b_s K_s \quad (4.62)$$

where

$$K_i = hf \left( t(k) + c_i h, x(k) + \sum_{j=1}^s a_{ij} K_j \right) \quad (4.63)$$

Here  $a$ ,  $b$ ,  $c$  are parameters and  $K$  is the intermediate variable.  $x$  is the state variable here it represents conversion. The dynamic model  $f$  follows the kinetic model in (4.55). We use Hammer-Hollingsworth with Butcher tableau as follows:

$$\begin{bmatrix} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ & b_1 & b_2 & \dots & b_s \end{bmatrix} = \begin{bmatrix} \frac{3-\sqrt{3}}{6} & \frac{1}{4} & \frac{1}{4} - \frac{\sqrt{3}}{6} \\ \frac{3+\sqrt{3}}{6} & \frac{1}{4} + \frac{\sqrt{3}}{6} & \frac{1}{4} \\ & \frac{1}{2} & \frac{1}{2} \end{bmatrix} \quad (4.64)$$

Thus we obtain the following discretization of the dynamic model:

$$\dot{x}(k, n) = f(x(k, n), u(k, n)), \forall k \in [1, \dots, N_k], n \in N \quad (4.65)$$

$$\begin{aligned} K1(k, n) &= f(t(k, n) + 0.2113h(n), x(k, n) + h(n)(0.25K1(k, n) - 0.0387K2(k, n)), u(k, n)) \\ \forall k \in [1, \dots, N_k], n \in N \end{aligned} \quad (4.66)$$

$$\begin{aligned} K2(k, n) &= f(t(k, n) + 0.7887h(n), x(k, n) + h(n)(0.5387K1(k, n) + 0.25K2(k, n)), u(k, n)) \\ \forall k \in [1, \dots, N_k], n \in N \end{aligned} \quad (4.67)$$

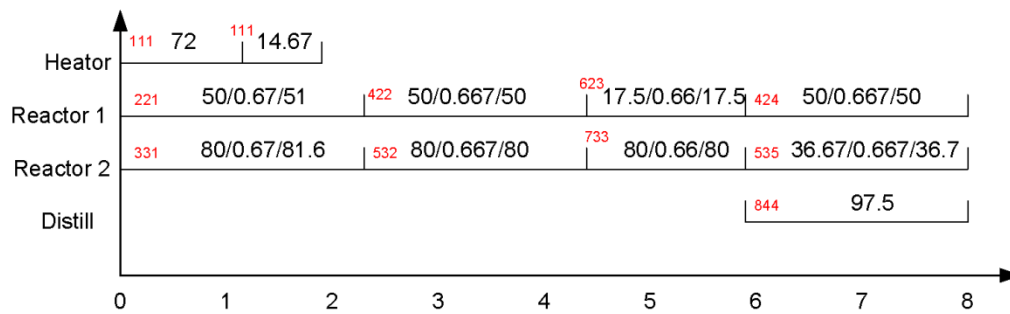
$$h(n) = \frac{T_{rv}(n)}{N_k} \quad (4.68)$$

$$x(k+1, n) = x(k, n) + h(n)(0.5K1(k, n) + 0.5K2(k, n)), \forall k \in [1, \dots, N_k - 1], n \in N \quad (4.69)$$

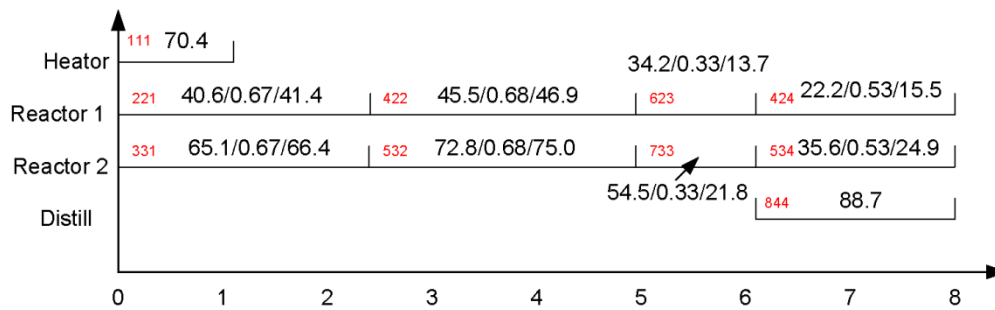
The first-order derivatives of the state variables at each step are calculated using Equation (4.65) and  $k$  represents sample steps in the transient duration. Through the calculation of intermediate variables in equations (4.66) and (4.67), and the sample size in (4.68) the state of the next step is obtained by Equation (4.69). Optimization problem for this scenario is presented in (4.70).

$$\begin{aligned}
& \max_{w,y,V,d,ST,T^s,T^f} \quad J = C^{P_1} V_1 X_1(N_k) X_2(N_k) + C^{P_2} V_3 X_3(N_k) \\
& \quad - \sum_{i \in I_R} C^h V_i - C^u \sum_j V_j \sum_{k'} u_j(k') \\
& \text{s.t.} \quad \begin{cases} (4.9)-(4.12) \text{ scheduling constraints} \\ (4.14)-(4.22) \text{ scheduling constraints} \\ (4.65)-(4.69) \text{ discretization using implicit Runge-Kutta method} \end{cases}
\end{aligned} \tag{4.70}$$

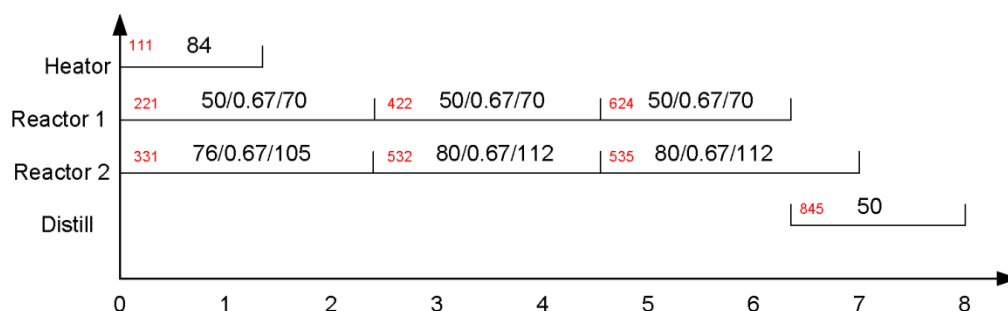
Scheduling solutions and detailed results of all scenarios are presented in Figure 4.17-Figure 4.21 and Table 4.3. Note that in the sequential solution procedures of scenario 1, the optimization problem in (4.56) corresponds to MILP with the objective function being linearized using (4.78) and (4.79), whereas the optimization problem in (4.56) is a MINLP. For scenario 3, the optimization problem in (4.61) with the objective linearized using (4.72), (4.78) and (4.79) is a MILP, and the optimization problem in (4.61) corresponds to MINLP. In scenario 4 we solve problem (4.70) using MINLP solver DICOPT and also the global optimizer BARON.



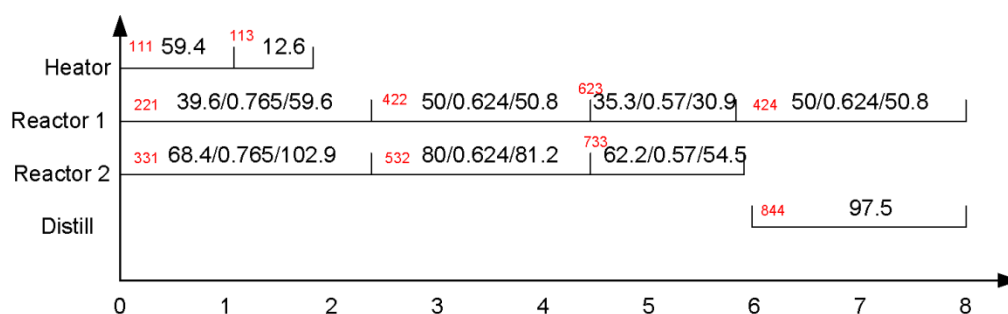
**Figure 4.17 Scenario 1, scheduling solution for MINLP. Red number represents the index of task-unit-eventpoint (ijn) and black number represents the (amount of material/conversion/utility consumption)**



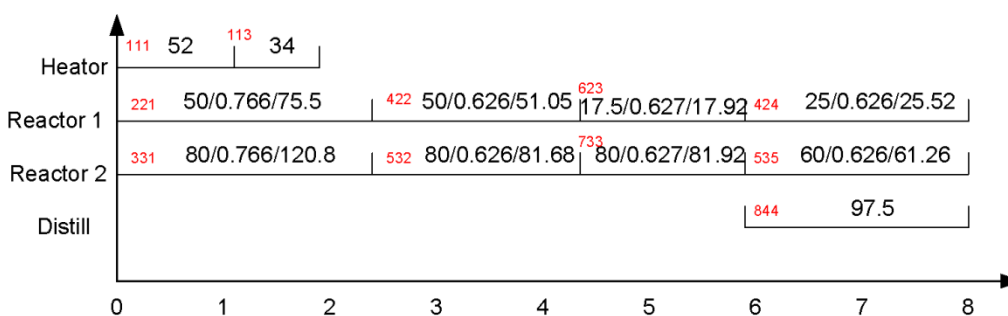
**Figure 4.18 Scenario 2, apply control to pre-obtained scheduling solution. Red number represents the index of task-unit-event point (ijn) and black number represents the (amount of material/conversion/utility consumption)**



**Figure 4.19 Scenario 3, implicit integration based on the recipe obtained in control problem. Red number represents the index of task-unit-event point (ijn) and black number represents the (amount of material/conversion/utility consumption)**



**Figure 4.20 Scenario 4, original integrated problem MIDO discretized into a MINLP using implicit RK method, solved using GAMS/DICOPT. Red number represents the index of task-unit-event point (ijn) and black number represents the (amount of material/conversion/utility)**



**Figure 4.21 Scenario 4, original integrated problem MIDO discretized into a MINLP using implicit RK method, solved using GAMS/BARON. Red number represents the index of task-unit-event point (ijn) and black number represents the (amount of material/conversion/utility)**

**Table 4.3 Comparison of the quantitative results for four scenarios of the complex batch process case study**

| 4 scenarios          | Scenario 1:<br>Integrated problem (MINLP) nonlinearity only in the objective                  | Scenario 2:<br>Apply control to obtained scheduling (Directly apply explicit control solution) | Scenario 3:<br>Implicit integration based on the recipe obtained in control problem (small MINLP) | Scenario 4:<br>original MIDO discretized into MINLP using implicit RK4 method, nonlinearities in the objective and constraints |   |
|----------------------|---|--|---|--|---|
| Problem size         | 16039 variables, 107324 constraints   | N/A  | 653 variables, 1284 constraints   | 5932 variables, 8495 constraints   |   |
| Solver               | warm start solving procedures: solve MILP using CPLEX, and then solve MINLP using DICOPT, 76s | N/A (no solver is needed)  | warm start solving procedures: solve MILP using CPLEX, and then solve MINLP using DICOPT, 14s     | DICOPT 157.6s  | BARON 4804.9s, maximum CPU time is set to 6000s |
| Product 1 Amount (L) | 38.71   | 29.77  | 37.70   | 34.38  | 41.25   |
| Product 2 Amount (L) | 58.50   | 26.34  | 30.15   | 50.0   | 55.03   |
| Revenue (\$)         | 252.06  | 175.19   | 218.65  | 221.89   | 261.23  |
| Raw A Amount (L)     | 86.67   | 70.44  | 84.0  | 72.0   | 86.0  |
| Raw B Amount (L)     | 65.0  | 52.84  | 63.0  | 54.0   | 85.0  |
| Raw C Amount (L)     | 84.50   | 70.58  | 73.0  | 73.50  | 84.50   |
| Raw Cost (\$)        | 47.23   | 38.77  | 44.0  | 39.90  | 47.10   |
| Utility Amount (L)   | 445.47  | 305.60   | 540.40  | 430.76   | 515.60  |
| Utility Cost (\$)    | 22.27   | 15.28  | 27.02   | 21.54  | 25.78   |
| Profit (\$)          | 182.55  | 121.14   | 147.63  | 160.45   | 188.35  |

It can be observed that scenario 2 generates significantly lower profit. This is due to the lack of integration of scheduling and control. In scenario 2 the scheduling problem and control are solved sequentially and thus it cannot achieve the overall optimum solution since the obtained

scheduling solutions based on which the control problem is solved may not be optimal for the control level. As shown in Figure 4.18, slot 623 and 733 which correspond to reaction 3 correspond to smaller processing time compared to scenario 1, resulting in less production of product 2 (see the amount of product 2 in Table 4.3) and thus less revenue.

It can also be observed that the profit of scenario 3 is lower than scenario 1 but higher than scenario 2. The implicit integration of scheduling and control in scenario 3 partially shares information between scheduling and control levels since it approximates the nonlinear relations using linear functions (Figure 4.15 and Figure 4.16), so it performs better than scenario 2 where no integration is performed but worse than the explicit integration in scenario 1. However, the problem size of scenario 3 is significantly smaller than scenario 1 and thus requires much less computation time.

Table 4.3 also presents the detailed results for scenario 4 where we build a MIDO for the integrated problem and discretize it into a MINLP using implicit Runge-Kutta method. There is a remarkable comparison between scenario 1 and scenario 4, where they both represent the integrated problem but they differ significantly in problem size, CPU time and profit. Though scenario 1 (the proposed approach) has much larger problem size, it needs much less time to compute compared to DICOPT and BARON solver in scenario 4. This is because of the problem formulation since all the constraints are linear in scenario 1, while constraints (4.14), (4.65)-(4.69) in scenario 4 are nonlinear. Scenario 1 produces slightly lower profit than BARON does while its solution time is reduced in nearly two orders of magnitude. Apparently DICOPT in scenario 4 obtains a local optimum solution since in this case the profit found is 15% lower than the one using a global optimization approach.

The dash lines in Figure 4.15 and Figure 4.16 show two extreme cases of scenario 3 that corresponds to an optimistic and a pessimistic case. In the optimistic case the approximated

conversion profile is always above the true profile and the utility amount is lower than the true one, while in pessimistic case the approximation follows an opposite trend. Solving problem (4.61) for both cases results in profit \$140.29 and \$136.5, respectively. Both of them have lower profit than scenario 3 because they use worse approximation (the distance between the linear profile and the true profile is greater) as shown in Figure 4.15 and Figure 4.16. Note that the profit associated with scenario 3 is obtained by incorporating the solution of problem (4.61)-(4.65) and calculating the real profit using the objective in (4.56) *i.e.* the profit is the one that realized in the real case. This ensures a fair comparison between the different scenarios.

The proposed formulation for simultaneous scheduling and control results corresponds to a MINLP which involves linear constraints and a nonlinear objective function. To improve the computation efficiency we linearize the bilinear term in the objective using a simple first order Taylor expansion (4.71) and thus transforming the problem to a MILP.

In case 1 we linearize the term  $V \cdot X(N_k)$  at  $(V_0, X_0)$  following (4.72) and obtain a linear objective function (4.73). We chose different points where the bilinear term is expanded and solve the corresponding MILP, and summarize the results in Table 4.4.

$$f(u, x) \approx f(u_0, x_0) + (u - u_0) \frac{\partial f}{\partial u}(u_0, x_0) + (x - x_0) \frac{\partial f}{\partial x}(u_0, x_0) \quad (4.71)$$

$$VX \approx V_0 X_0 + (V - V_0) X_0 + (X - X_0) V_0 = V_0 X + X_0 V - V_0 X_0 \quad (4.72)$$

$$J_0 = C^P (V_0 X(N_k) + VX_0(N_k) - V_0 X_0(N_k)) - C^r V - C^u \sum_{k=1}^{N_k-1} u(k') \quad (4.73)$$

**Table 4.4 Results of the MINLP and the derived MILP (case 1)**

|        | MINLP         | MILP<br>$V_0=50$<br>$X_0=0.5$ | MILP<br>$V_0=60$<br>$X_0=0.4$ | MILP<br>$V_0=70$<br>$X_0=0.6$ | MILP<br>$V_0=70$<br>$X_0=0.7$ | MILP<br>$V_0=40$<br>$X_0=0.7$ | MILP<br>$V_0=80$<br>$X_0=0.6$ |
|--------|---------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|-------------------------------|
| Solver | SBB/<br>IPOPT | Cplex                         | Cplex                         | Cplex                         | Cplex                         | Cplex                         | Cplex                         |



|                       |       |       |       |       |       |       |       |
|-----------------------|-------|-------|-------|-------|-------|-------|-------|
| ProductAmount (L)     | 53.56 | 45.33 | 43.67 | 51.0  | 51.0  | 51.0  | 45.33 |
| Revenue (\$)          | 53.56 | 45.33 | 43.67 | 51.0  | 51.0  | 51.0  | 45.33 |
| RawMaterialAmount (L) | 78.0  | 68.0  | 64.0  | 75.0  | 75.0  | 75.0  | 68.0  |
| RawCost (\$)          | 15.60 | 13.60 | 12.80 | 15.0  | 15.0  | 15.0  | 13.60 |
| UtilityAmount (L)     | 42.32 | 40.0  | 41.16 | 20.8  | 20.8  | 20.8  | 40.0  |
| UtilityCost (\$)      | 2.12  | 2.0   | 2.06  | 1.04  | 1.04  | 1.04  | 2.0   |
| Profit (\$)           | 35.84 | 29.73 | 28.80 | 34.96 | 34.96 | 34.96 | 29.73 |

Note that the MILP is solved with a different objective (4.73) compared to the one of the MINLP (4.40). However, the results in Table 4.4 are calculated following the same equations as follows:

$$\text{Revenue} = \text{ProductAmount} * \text{ProductPrice} \quad (4.74)$$

$$\text{RawCost} = \text{RawMaterialAmount} * \text{RawPrice} \quad (4.75)$$

$$\text{UtilityCost} = \text{UtilityAmount} * \text{UtilityPrice} \quad (4.76)$$

$$\text{Profit} = \text{Revenue} - \text{RawCost} - \text{UtilityCost} \quad (4.77)$$

In case 2 through linearization of the bilinear term in the objective (4.78) and (4.79) we obtain a linear objective function and an overall MILP problem. The derived MILP has the same problem size as the MINLP and takes around 18s to solve using GAMS/CPLEX. The results of the MILP with four linearization points are provided in Table 4.5. The best linearization point is  $V_1^0 = V_3^0 = 50$ ,  $X_1^0 = X_2^0 = X_3^0 = 0.6$  where the corresponding result \$179.96 is slightly lower than the profit of the MINLP which is \$182.55.

$$V_1 X_1 X_2 = V_1^0 X_1^0 X_2 + V_1^0 X_2^0 X_1 + X_1^0 X_2^0 V_1 - 2V_1^0 X_1^0 X_2^0 \quad (4.78)$$

$$V_3 X_3 = V_3^0 X_3 + X_3^0 V_3 - V_3^0 X_3^0 \quad (4.79)$$

**Table 4.5 Results of the MINLP and the derived MILP (case 2)**

|                     | MINLP                                   | MILP<br>$V_1^0 = V_3^0 = 60,$<br>$X_1^0 = X_2^0 = 0.5$<br>$X_3^0 = 0.5$ | MILP<br>$V_1^0 = V_3^0 = 60,$<br>$X_1^0 = X_2^0 = 0.6$<br>$X_3^0 = 0.6$ | MILP<br>$V_1^0 = V_3^0 = 50,$<br>$X_1^0 = X_2^0 = 0.6$<br>$X_3^0 = 0.6$ | MILP<br>$V_1^0 = V_3^0 = 50,$<br>$X_1^0 = X_2^0 = 0.5$<br>$X_3^0 = 0.5$ |
|---------------------|---|---|---|---|---|
| Solver              | CPLEX for MILP,<br>DICOPT for MINLP 76s | CPLEX, 17s  | CPLEX, 19s  | CPLEX, 18s  | CPLEX, 18s  |
| Product1 Amount (L) | 38.71                                   | 37.15   | 37.77   | 38.31   | 37.45   |
| Product2 Amount (L) | 58.5                                    | 55.82   | 50.14   | 57.38   | 47.61   |
| Revenue (\$)        | 252.06                                  | 241.55  | 239.02  | 248.93  | 234.84  |
| Raw A Amount (L)    | 86.67                                   | 89.07   | 82.4  | 83.1  | 96.1  |
| Raw B Amount (L)    | 65.0                                    | 75.3  | 78.63   | 62.07   | 72.36   |
| Raw C Amount (L)    | 84.5                                    | 80.23   | 61.12   | 85.13   | 91.79   |
| Raw Cost (\$)       | 47.23                                   | 48.92   | 44.43   | 46.04   | 52.05   |
| Utility Amount (L)  | 445.47                                  | 484.0   | 413.8   | 459   | 349.6   |
| Utility Cost (\$)   | 22.27                                   | 24.2  | 20.69   | 22.95   | 17.48   |
| Profit (\$)         | 182.55                                  | 168.43  | 173.87  | 179.96  | 165.31  |

Results for both cases show that the MILP generates lower profit than the MINLP. This is because the MILP fails to capture the optimum of the original objective. However, the MILP requires significantly less computation time. This helps to achieve a balance between computation complexity and optimality, since the linearization greatly simplifies the problem while keeping the solution close to the optimum.

## 4.6 Summary

In this study we model scheduling and control problem for batch processes simultaneously using mp-MPC. The reason that we adopt mp-MPC is that it generates control signal instantly by function evaluation and thus greatly reduce computation complexity when control level is integrated with scheduling level. To the authors' knowledge, this is the first attempt in this area to explore the possibility and feasibility of applying mp-MPC in control and scheduling problem. The main contribution of this study is that we propose a framework which is capable to transform explicit control solution generated by mp-MPC into explicit linear constraints, and incorporate with the linear constraints at scheduling level. This results in an integrated problem that involves linear constraints and a nonlinear objective. Results of case studies demonstrate that the integration achieves much high profit compared to the sequential approach.

### Nomenclature:

#### *Indices*

$i$  tasks

$j$  units

$s$  states

$n$  event points

$k$  sample steps

#### *Sets*

$I$  tasks

$I_j$  tasks which can be performed in unit  $j$

$I_s$  tasks which process state  $s$

$J_i$  units which are suitable to perform task  $i$

$I_R$  set of reaction tasks

$N$  event points within the time horizon

$S$  states

#### *Parameters*

$V_{ij}^{\min}$  minimum amount of material required to process task  $i$  in unit  $j$

$V_{ij}^{\max}$  capacity of unit  $j$  when processing task  $i$

$ST(s)^{\max}$  storage capacity for state  $s$

$\alpha_{ij}$  coefficient of constant part of processing time of task  $i$  in unit  $j$

$\beta_{ij}$  coefficient of variable part of processing time of task  $i$  in unit  $j$

$H$  time horizon

$C^p, C^r, C^u$  price of product, raw material and utility

$M$  big positive number

$N_k$  number of discretization point

### *Decision Variables*

$w(i, n)$  binary variable assign task  $i$  at event point  $n$

$y(i, n)$  binary variable assign unit  $j$  at event point  $n$

$V(i, j, n)$  amount of material undertaking task  $i$  in unit  $j$  at event point  $n$

$d(s, n)$  amount of state  $s$  sold at event point  $n$

$ST(s, n)$  amount of state  $s$  at event point  $n$

$T^s(i, j, n)$  starting time of task  $i$  in unit  $j$  at event point  $n$

$T^f(i, j, n)$  ending time of task  $i$  in unit  $j$  at event point  $n$

$T_{rv}(i, j, n)$  processing time of task  $i$  in unit  $j$  at event point  $n$

$X$  conversion

$h$  step size

$x$  state variables such as concentration

$u$  scaled temperature corresponding to utility consumption

$A, B, C$  coefficients in PWA model

$V, W$  coefficients of the inequalities corresponding to the PWA

$F, G$  coefficients in the explicit solution

$H, K$  coefficients of the inequalities describing the critical regions

$CR$  critical region

$y_1$  binary variables selecting the critical regions

$y_2$  binary variables selecting the valid regions of PWA

$K1, K2$  intermediate variables in implicit RK method

$\Omega$  domain of variables

## **Chapter 5 An integrated framework for scheduling and control using fast Model Predictive Control**

### **5.1 Introduction**

Online integration of scheduling and control requires updating operation solutions for both scheduling and control levels at real time in the presence of disturbance, thus the online integration requires a repetitive solution of the integrated problem at each time interval (Zhuge and Ierapetritou, 2012). To reduce the computation complexity of the integrated problem, researchers proposed to use Lagrangean Decomposition (Terrazas-Moreno et al., 2008a) and Bender Decomposition (Chu and You, 2013b) (Chu and You, 2013a) (Nie et al., 2014) to converge to the optimal solution. Chu and You (Chu and You, 2014a) models the integrated problem using game theory and efficiently solve the resulting bi-level optimization problem. Zhuge and Ierapetritou (Zhuge and Ierapetritou, 2014) utilizes mp-MPC to handle the control problem and incorporation the explicit control solution with the scheduling level, resulting in a MILP which simplifies the integrated problem.

Model Predictive Control (MPC) is an online optimization technique that involves repetitively solution of an optimization problem over a future time horizon. To reduce the computation burden of conventional MPC, multi-parametric Model Predictive Control (mp-MPC) was proposed to solve for the explicit control law offline and thus the online optimization is reduced to simple function evaluations (Pistikopoulos, 2009). However, as the problem size increases in terms of state dimension and prediction horizon, the number of polyhedral regions in the state partition increases exponentially and this limits the applicability of mp-MPC to small and medium-sized control problems (Richter et al., 2012).

Fast MPC on the other hand is capable in handling large scale problems, and therefore can be used to facilitate the efficient integration of scheduling and control of large-scale processes. Fast MPC for linear systems transforms the MPC problem into a convex quadratic programming

problem, for which efficient nonlinear programming methods and computational tools can be used to speed up the computation by exploring the problem structure. Among the existing work in the literature three solution approaches can be identified: active set method (Ferreau et al., 2008), interior point method (Rao et al., 1998) (Wang and Boyd, 2010), and Fast gradient method (Richter et al., 2009) (Richter et al., 2012). These methods are specifically elaborated in section 5.4.1. Zavala et al. (Zavala et al., 2008) developed algorithms for fast nonlinear MPC that is based on sensitivity calculation for the Karush–Kuhn–Tucker (KKT) conditions of the NLP derived from differential and algebraic equations (DAE) of the nonlinear MPC. This approach was further investigated and applied in large scale industrial processes in Lopez-Negrete et al (Lopez-Negrete et al., 2013).

In this study we propose a cascade control strategy that involves two control loops for the online integration of scheduling and control (Figure 5.1). In the outer loop we approximate the original process dynamics using a piece-wise affine (PWA) model and incorporate it with the scheduling level. This leads to an integrated problem that is subject to linear constraints. The primary MPC at the outer loop solves the integrated problem and generates both the production scheduling and the control solution. However, only the scheduling solution is transferred to the inner loop where the secondary MPC (a fast MPC) treats the scheduling solution as parameters and computes the exact control solution online. Note that these two loops correspond to different models and different time scales. The outer loop uses the integrated model and solves it in the period of days or weeks while the inner loop uses the process dynamic model and updates the solution in seconds. Essentially the inner loop is solved much more frequently than the out loop does. Following this approach the primary MPC is able to achieve an overall optimality for both scheduling and control levels efficiently, and the secondary MPC is able to respond quickly to the local disturbance. Note that the proposed approach is applicable in a dynamic market environment as the outer loop incorporates the market information such as demand and price. The proposed

framework solves an integrated model to guarantee the overall optimality and employs fast MPC to respond timely to process disturbances. While solving the integrated problem the latest market information such as material price and product demands are incorporated into the integrated problem, and the scheduling solutions are updated accordingly.

When disturbance is detected (state deviation from the reference is measured), the state information is feedback to the inner fast MPC or outer integrated problem. A threshold is introduced to determine the feedback. If the state deviation is less than the threshold, the state is feedback to inner fast MPC and locally treated by fast MPC. Otherwise if the state deviation is large (higher than the threshold) it is feedback to the integrated problem and the scheduling solution is updated accordingly. The threshold is an empirical value and it is determined to avoid the unnecessary changes of the scheduling solution when disturbances are small enough that can be handled efficiently in the control level, whereas updates in the scheduling solution are considered if significant disturbance occurs.

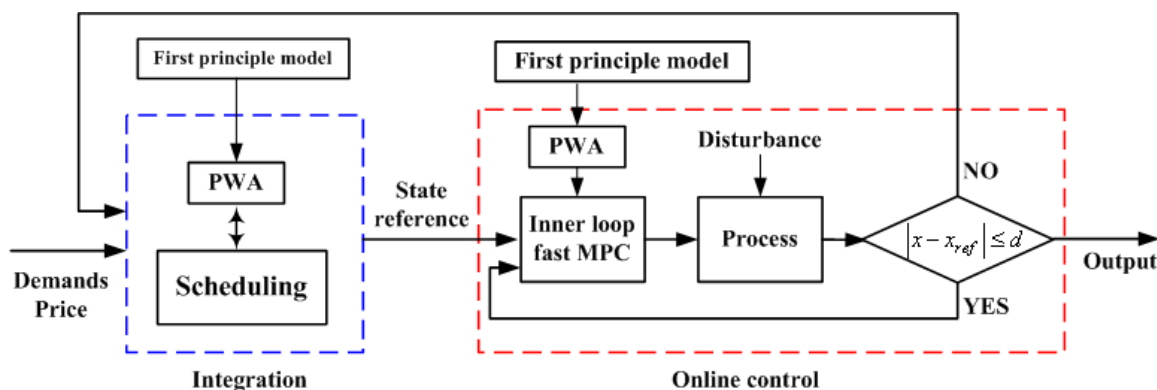


Figure 5.1 Cascade strategy for the online integration of scheduling and control

## 5.2 Integration of scheduling and control based on (Piece-Wise Affine) PWA model

The main challenge of integration of scheduling and control for chemical processes is that the integrated problem cannot be solved as fast as it is required for the control level but also that the

scheduling solution should not be adjusted all the time. Since scheduling and control levels correspond to different dynamics and different time scales, an appropriate integration has to capture the process behavior well enough at both scheduling and control levels, while maintaining the appropriate level of integration (Engell and Harjunkski, 2012). Moreover, the integrated problem should not be very computational expensive thus the integrated solution could be updated timely when disturbance occurs. Typically an integration results in a MIDO which is discretized into a MINLP using collocation point method (Flores-Tlacuahuac and Grossmann, 2006) (Chu and You, 2012) (Nie et al., 2012) or implicit RK4 method (Zhuge and Ierapetritou, 2012). However, the nonlinearity in MINLP poses significant difficulty in the computation method of solving the integrated optimization problem.

In control practice a standard way to handle the nonlinear dynamics is to linearize the nonlinear model around operating points. However, the linearized model is valid only if the process operates at the vicinity of operating points. It is just an approximation of the actual process dynamics when process experiences transitions where the states are far from the operation points (e.g. steady state points). In contrast, PWA systems have shown to be an effective approach in dealing with nonlinear systems (Johansson, 2003) (Rodrigues and How, 2003) (Rodrigues and Boyd, 2005). The basic idea of PWA system is that the nonlinear dynamics can be approximated by a collection of distinct linear (or affine) dynamic approximations with associated regions of validity. Compared to standard linear model, PWA composes a group of linear models and therefore it is capable to address the process dynamics at the entire state domain. PWA models eliminate the nonlinearity while retaining high approximation accuracy.

### **5.2.1 A brief review of the PWA identification techniques**

There are two types of techniques regarding PWA model identification. One is clustering technique where PWA models are obtained by processing K-means algorithm on input-output data. Ferrari-Trecate et al (Ferrari-Trecate et al., 2003) combined use of clustering, linear



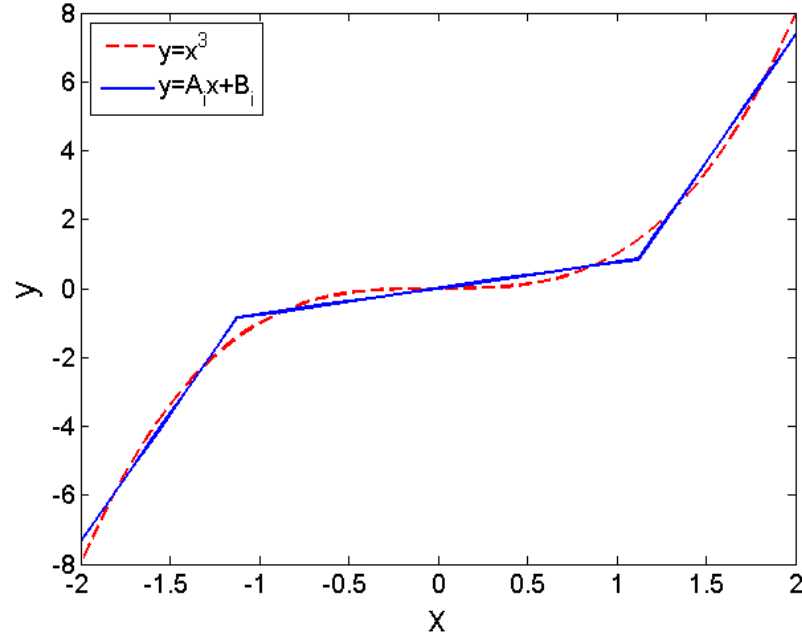
identification, and pattern recognition techniques to identify both the affine sub-models and the polyhedral partition of the domain. Magnani and Boyd (Magnani and Boyd, 2009) proposed a heuristic method for fitting a convex piecewise linear function to a given set of data. It uses K-means algorithm for clustering, and requires the piecewise linear function to be convex. Researchers (Buchan et al., 2013) (Gegúndez et al., 2008) (Nakada et al., 2005) (Ohlsson and Ljung, 2011) (Roll et al., 2004) (Vasak et al., 2006) employ K-means clustering technique or variations derived from that. The downside of this approach is that there is no guarantee that the union of regions obtained by clustering is able to cover the whole area of the original domain without gaps where the model is actually defined.

The other type of PWA identification technique is optimization based PWA approximation. This technique applies to the cases where the original nonlinear functions are available. Stevek et al. (Števek et al., 2012) proposed a two-stage optimization-based approach. At the first stage, they fit the input/output data using multivariable nonlinear functions that are represented as a sum of products of functions in single variables. They employ neural networks with pre-defined basis functions to obtain the nonlinear functions. At the second stage they obtain the PWA approximation for the nonlinear functions obtained in the first stage using the approaches in Kvasnica et al. (Michal Kvasnica et al., 2011). Kvasnica et al. (Michal Kvasnica et al., 2011) handles one-dimensional functions. They treat the partition of the domain and the corresponding linear model as decision variables and build an optimization problem (5.1) to minimize the error between the PWA and the original functions. For example, PWA approximation  $\tilde{f}(x)$  for a nonlinear function  $f(x) = x^3$  is obtained through solving problem (5.1) where  $A$  and  $B$  are coefficients in PWA and  $r$  is the partition of the  $x$  domain. Note that continuity is enforced by the third constraint. The solutions are presented in formula (5.2) and Figure 5.2.

$$\min_{A_i, B_i, r_i} \int_{\underline{x}}^{\bar{x}} (f(x) - \tilde{f}(x))^2 dx$$

$$s.t. \begin{cases} \tilde{f}(x) = A_i x + B_i & \text{if } x \in [r_{i-1}, r_i] \\ \underline{x} = r_1 \leq \dots \leq r_{N-1} = \bar{x} \\ A_i r_i + B_i = A_{i+1} r_i + B_{i+1} \end{cases} \quad (5.1)$$

$$\tilde{f}(x) = \begin{cases} 7.431x + 7.495, & -2 \leq x < -1.123 \\ 0.757x, & -1.123 \leq x < 1.123 \\ 7.431x - 7.459, & 1.123 \leq x < 2 \end{cases} \quad (5.2)$$



**Figure 5.2 Example of one-dimension PWA approximation**

For two and multiple dimensional functions, Kvasnica et al. (Michal Kvasnica et al., 2011) investigated two particular cases:

Case 1: if the function  $f(x_1, x_2, \dots, x_n)$  is separable, i.e.  $f(x_1, x_2, \dots, x_n) = f_1(x_1) + f_2(x_2) + \dots + f_n(x_n)$ , then apply one-dimensional approximation for each term.

Case 2: if the function  $f(x_1, x_2, \dots, x_n)$  is in a form of product i.e.  $f(x_1, x_2, \dots, x_n) = f_1(x_1)f_2(x_2)\dots f_n(x_n)$ , then introduce new variables to transform the product into a

summation form in case 1. For example, let  $y_1 = x_1 + x_2$  and  $y_2 = x_1 - x_2$ , we have  $x_1 x_2 = (y_1^2 - y_2^2)/4$ .

However, for general multiple dimension functions that cannot be put into the form of case 1 and 2, Kvasnica et al. (Michal Kvasnica et al., 2011) did not provide any approximation approaches. Julian et al. (Julian et al., 1999) explored a uniform triangular-shape partition for the domain and approximated a nonlinear function by adjusting the grid size of the partition. Zavieh and Rodrigues (Zavieh and Rodrigues, 2013) used heuristic selection of linearization points where a new linearization point is determined by the intersection of tangential lines associated with the old linearization points. Casselman and Rodrigues (Casselmann and Rodrigues, 2009) used a set of linearization points and Voronoi partitions in the approximation. Sontag (Sontag, 1981) discussed the stability, controllability, and observability in PWA systems. However, the existing work in PWA identification cannot handle a general form of nonlinear model.

### 5.2.2 A PWA identification technique using optimization methods

To overcome the shortcomings of the optimization based PWA identification described in section 5.2.1, we propose a PWA identification technique that is able to handle general forms of nonlinear function. In 2-dimensional cases we use rectangular partition of the domain of function

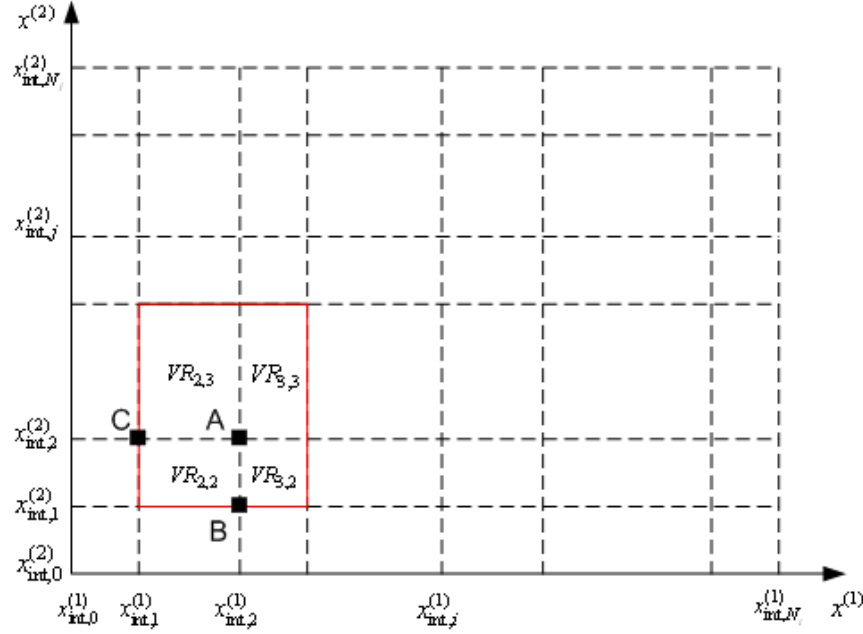
$f(x^{(1)}, x^{(2)})$  (Figure 5.3). Index  $i \in I = \{1, 2, \dots, N_i\}$  is associated with  $x^{(1)}$ , and  $j \in J = \{1, 2, \dots, N_j\}$  is associated with  $x^{(2)}$ . Intermediate points  $x_{\text{int},i}^{(1)}$  and  $x_{\text{int},j}^{(2)}$  are introduced as partitions for  $x^{(1)}$  and  $x^{(2)}$ , respectively. Therefore we have constraints:

$$x_{\text{int},0}^{(1)} \leq x_{\text{int},1}^{(1)} \leq \dots \leq x_{\text{int},N_i}^{(1)} \text{ and } x_{\text{int},0}^{(2)} \leq x_{\text{int},1}^{(2)} \leq \dots \leq x_{\text{int},N_j}^{(2)}, \text{ where } x_{\text{int},0}^{(1)} \text{ and } x_{\text{int},0}^{(2)}$$

are the lower bounds of  $x^{(1)}$  and  $x^{(2)}$  and  $x_{\text{int},N_i}^{(1)}$  and  $x_{\text{int},N_j}^{(2)}$  are the upper bounds.  $N_i$  and

$N_j$  are the number of intermediate points for  $x^{(1)}$  and  $x^{(2)}$ . The sub-index ‘‘int’’ represents

‘intermediate points’.



**Figure 5.3 Rectangular partition of the domain of two-dimensional functions**

Define the Valid Regions (VR) and the PWA

$$VR_{i,j} = \{ (x^{(1)}, x^{(2)}) \mid x_{int,i-1}^{(1)} \leq x^{(1)} \leq x_{int,i}^{(1)}, x_{int,j-1}^{(2)} \leq x^{(2)} \leq x_{int,j}^{(2)} \} \quad (5.3)$$

$$\hat{f}_{i,j}(x^{(1)}, x^{(2)}) = a_{i,j}x^{(1)} + b_{i,j}x^{(2)} + c_{i,j}, \text{ if } (x^{(1)}, x^{(2)}) \in VR_{i,j} \quad (5.4)$$

Equation (5.3) determines that point  $(x^{(1)}, x^{(2)})$  belongs to a certain valid region and Equation (5.4) provides the Linear Time Invariant (LTI) associated with the regions.

Constraints:

The profile of PWA should be continuous at the boundaries of valid regions. In other words, the function values of adjacent LTIs are the same at the boundaries. Note that if the continuity at the intersection points is established, then the continuity at the segment defined by the intersection points is established as well. For example, at point A in Figure 5.3, the LTIs of the four adjacent valid regions should have the same value. This gives rise to the constraints in Equation (5.5).

$$\hat{f}_{i,j}(x_{int,i}^{(1)}, x_{int,j}^{(2)}) = \hat{f}_{i+1,j}(x_{int,i}^{(1)}, x_{int,j}^{(2)}) = \hat{f}_{i,j+1}(x_{int,i}^{(1)}, x_{int,j}^{(2)}) = \hat{f}_{i+1,j+1}(x_{int,i}^{(1)}, x_{int,j}^{(2)}) \quad (5.5)$$

More specifically,

$$\begin{aligned}
 & a_{i,j}x_{\text{int},i}^{(1)} + b_{i,j}x_{\text{int},j}^{(2)} + c_{i,j} \\
 &= a_{i+1,j}x_{\text{int},i}^{(1)} + b_{i+1,j}x_{\text{int},j}^{(2)} + c_{i+1,j} \\
 &= a_{i,j+1}x_{\text{int},i}^{(1)} + b_{i,j+1}x_{\text{int},j}^{(2)} + c_{i,j+1} \\
 &= a_{i+1,j+1}x_{\text{int},i}^{(1)} + b_{i+1,j+1}x_{\text{int},j}^{(2)} + c_{i+1,j+1}
 \end{aligned} \tag{5.6}$$

#### Optimization problem:

The proposed PWA identification approach can be modeled as an optimization problem with decision variables that include the partition of the domain  $x_{\text{int},i}^{(1)}, x_{\text{int},j}^{(2)}$  and the coefficients of LTI  $a_{i,j}, b_{i,j}, c_{i,j}$ , and constraints that specify the continuity constraints in (5.6). The objective of the optimization problem is to minimize the sum of squared error between the PWA and the original function as provided in problem (5.7).

$$\begin{aligned}
 & \min_{x_{\text{int},i}^{(1)}, x_{\text{int},j}^{(2)}, a_{i,j}, b_{i,j}, c_{i,j}} \sum_i \sum_j \left( \hat{f}(x_{\text{int},i}^{(1)}, x_{\text{int},j}^{(2)}) - f(x_{\text{int},i}^{(1)}, x_{\text{int},j}^{(2)}) \right)^2 \\
 & \text{s.t. (5.3) - (5.6)}
 \end{aligned} \tag{5.7}$$

The optimization problem of identifying PWA from a 3-dimensional function is provided in Appendix.

#### Complexity analysis

The decision variables of problem (5.7) include the coefficients of PWA  $(a_{i,j,k}, \dots, b_{i,j,k}, \dots, c_{i,j,k}, \dots, d_{i,j,k}, \dots)$  and the intermediate points of each dimension  $(x_{\text{int},i}^{(1)}, x_{\text{int},j}^{(2)}, x_{\text{int},k}^{(3)}, \dots)$ . If there is an additional dimension, there would be a new coefficient in PWA, a new dimension sub-index for each coefficient and a new group of intermediate points. Therefore the number of coefficients of PWA and the number of intermediate points are increasing in the order of  $o(N(N+1))$  and  $o(N)$ ,

respectively, where  $N$  is the number of dimension. However, the number of constraints is increasing exponentially with respect to the number of dimensions, since the number of adjacent valid regions is increasing in the order of  $o(2^N)$  and the associated linear functions all have the same value at the intersection point (the continuity constraints). For example for a 2-dimensional function, there are 4 continuity constraints and for a 3-dimensional function, there are 8 continuity constraints.

### 5.3 Integrated problem incorporating PWA systems

In this study we address continuous cyclic production for which the scheduling constraints are adopted from the work of Flores-Tlacuahuac and Grossmann (Flores-Tlacuahuac and Grossmann, 2006). The constraints at control level are built via the PWA identification proposed in section 5.2.2. The PWA model is incorporated into the scheduling constraints by introducing extra binary variables in a similar way as the previous work in mp-MPC (Zhuge and Ierapetritou, 2014). Since the out loop MPC has a fixed horizon, cycle time  $T_c$  is not a decision variable. Thus the objective is linear and the integrated problem is a MILP.

#### Constraints at scheduling level

##### Product assignment

$$\sum_{s=1}^{N_s} y_{p,s} = 1, \quad \forall p \in S_p \quad (5.8)$$

$$\sum_{p=1}^{N_p} y_{p,s} = 1, \quad \forall s \in S_s \quad (5.9)$$

where  $y_{p,s}$  are binary variables indicating the assignment of each product in each slot.  $y_{p,s} = 1$  if product  $p$  is assigned to slot  $s$ , otherwise  $y_{p,s} = 0$ . Equations (5.8) and (5.9) imply that only one product is produced in one slot and each product can only be produced in one slot.

### Production and demands

$$W_p = G_p \Theta_p, \forall p \in S_p \quad (5.10)$$

$$W_p \geq D_p, \forall p \in S_p \quad (5.11)$$

Inequality (5.11) implies that the production amount of each product  $w_p$ , which is a product of production rate  $G_p$  and production time  $\Theta_p$  (equation (5.10)), should satisfy the demand  $D_p$  in current production cycle.

### Production time

$$\Theta_{p,s} \leq \Theta_{\max} y_{p,s}, \forall p \in S_p, s \in S_s \quad (5.12)$$

$$\Theta_p = \sum_{s=1}^{N_s} \Theta_{p,s}, \forall p \in S_p \quad (5.13)$$

$$\Theta_s = \sum_{p=1}^{N_p} \Theta_{p,s}, \forall s \in S_s \quad (5.14)$$

Inequality (5.12) implies that the production time  $\Theta_{p,s}$  is less than the maximum duration allowed for producing product  $p$  in slot  $s$ . In Equations (5.13) and (5.14),  $\Theta_p, \Theta_s$  define the production time of product  $p$  and slot  $s$ , respectively.

### Timing constraints

$$t_1^s = 0 \quad (5.15)$$

$$t_s^e = t_s^s + \Theta_s + \theta_s^t, \forall s \in S_s \quad (5.16)$$

$$t_s^s = t_{s-1}^e, \forall s \in (S_s - \{1\}) \quad (5.17)$$

$$t_s^e \leq T_c, \forall s \in S_s \quad (5.18)$$

Equation (5.15) initializes the starting time of the first slot. As described by (5.16), the ending point  $t_s^e$  is equal to the sum of the starting point  $t_s^s$ , the production  $\Theta_s$  and transition times  $\theta_s^t$  of the current slot. Equation (5.17) enforces that the starting point in slot  $s$  equals the ending point of the previous slot to ensure continuity between slots. Inequality (5.18) indicates that all slots should end before the end of the cycle.

### Constraints at control level

#### Selection of LTI in PWA,

The LTI described by equation (5.19) is valid when state  $x_{s,k}$  is in the region of validity  $\Omega_{xi}$ . We transform this implicit requirement into explicit constraints (equations (5.20)-(5.23)) by introducing binary variables  $y1_{s,k,i}$ . If  $y1_{s,k,i} = 1$ , the region of validity  $\Omega_{xi}$  is selected by (5.20) and thus the LTI associated with coefficients  $A_i, B_i, C_i$  is selected through the constraints (5.21) and (5.22). Equation (5.23) indicates that only one LTI is selected at each sample step.

$$x_{s,k+1} = A_i x_{s,k} + B_i u_{s,k} + C_i, \text{ if } x_{s,k} \in \Omega_{xi} = \{x : V_i x \leq W_i\} \quad (5.19)$$

$$-M(1 - y1_{s,k,i}) + V_i x_{s,k} \leq W_i, \quad \forall s \in S_s, k \in S_k, i \in S_i \quad (5.20)$$

$$x_{s,k+1} \geq A_i x_{s,k} + B_i u_{s,k} + C_i - M(1 - y1_{s,k,i}), \quad \forall s \in S_s, i \in S_i, k \in (S_k - \{N_k\}) \quad (5.21)$$

$$x_{s,k+1} \leq A_i x_{s,k} + B_i u_{s,k} + C_i + M(1 - y1_{s,k,i}), \quad \forall s \in S_s, i \in S_i, k \in (S_k - \{N_k\}) \quad (5.22)$$

$$\sum_i y1_{s,k,i} = 1, \quad \forall s \in S_s, k \in S_k \quad (5.23)$$

#### Bounds for state and manipulated variables

$$x_{\min} \leq x_{s,k} \leq x_{\max}, \quad \forall s \in S_s, k \in S_k \quad (5.24)$$

$$u_{\min} \leq u_{s,k} \leq u_{\max}, \quad \forall s \in S_s, k \in S_k \quad (5.25)$$

$$-\Delta_u \leq u_{s,k+1} - u_{s,k} \leq \Delta_u, \quad \forall s \in S_s, k \in (S_k - \{N_k\}) \quad (5.26)$$



Inequalities (5.24) confine the states using lower and upper bounds while the constraints (5.25) and (5.26) introduce lower/upper bounds for the manipulated variables and their increments, respectively.

#### Determination of the end of transitions and evaluation of the transition times

In the integration of scheduling and control for continuous processes, the control action applies to the transitions between products that are distinguished by their steady states. The time points and state values at the beginning and end of the transitions are important variables in connecting scheduling and control levels. In practice, the transitions naturally start from the steady state of the previous product but the end point and the duration of transition are unknown. Therefore in the integrated model we need to build constraints to determine the end of the transition and calculate the transition time accordingly.

As shown schematically in Figure 5.4, there is a need to determine whether the process states governed by the PWA model reach the end of transition or not. Thus in this section we develop the constraints required to enable this. Constraints (5.27) to (5.30) enforce that the transient states achieve the steady state values when transitions end.

Due to the existence of some margins around the set point, if the state is between the lower bound  $\bar{x}_s - x_{\text{margin}}$  and the upper bound  $\bar{x}_s + x_{\text{margin}}$ , it is viewed as meeting the set point (steady state point  $\bar{x}_s$ ). In order to determine the end of transition we need to identify the point that lies within the bounds while the previous point is outside the bounds. Two types of binary variables are involved in determining the end of transition  $y4_{s,k}$  and  $y5_s$ .  $y4_{s,k}$  indicates whether the end of transition state is reached or not while  $y5_s$  indicate if the state right before the end of transition is above or below the bounds. More specifically, if  $y4_{s,k}=1$ , step  $k$  is selected as the end of transition and the state at step  $k$  is within the bounds. If the state at step  $k-1$  is above the upper

bound  $\bar{x}_s + x_{\text{margin}}$  ,  $y5_s = 1$  whereas if it is below the lower bound  $\bar{x}_s - x_{\text{margin}}$  ,  $y5_s = 0$  .

Conceptually,  $y5_s = 0$  means state increases to steady state during transition (as shown in Figure 5.4) and  $y5_s = 1$  means state decreases to steady state.

Inequality (5.27) is based on the definition of  $y4_{s,k}$  and (5.28) ensures that only one of the step points is selected as the end of the transition.

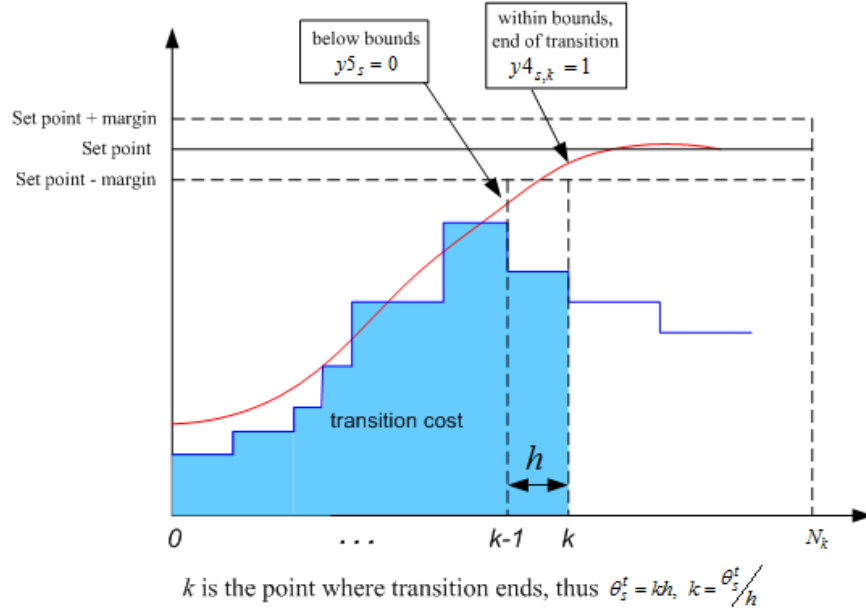
$$\begin{aligned} \bar{x}_s - x_{\text{margin}} - M(1 - y4_{s,k}) \leq x_{s,k} \leq \bar{x}_s + x_{\text{margin}} + M(1 - y4_{s,k}), \\ \forall s \in S_s, k \in S_k \end{aligned} \quad (5.27)$$

$$\sum_k y4_{s,k} = 1, \quad \forall s \in S_s \quad (5.28)$$

If  $y4_{s,k} = 0$ , the end of transition is not reached and constraints (5.27), (5.29) and (5.30) are relaxed. If  $y4_{s,k} = 1$  step  $k$  is the end of transition, so constraint (5.27) enforces that the state at step  $k$  ( $x_{s,k}$ ) must be within the quality bounds, and constraint (5.29) enforces that the state at step  $k-1$  ( $x_{s,k-1}$ ) must be below the lower bound  $\bar{x}_s - x_{\text{margin}}$  or above the upper bound  $\bar{x}_s + x_{\text{margin}}$ . The transition time is determined as  $kh$  by constraints (5.30).

$$\begin{aligned} \bar{x}_s + x_{\text{margin}} - M(2 - y4_{s,k} - y5_s) \leq x_{s,k-1} \leq \bar{x}_s - x_{\text{margin}} \\ + M(1 - y4_{s,k} + y5_s), \quad \forall s \in S_s, k \in (S_k - \{1\}) \end{aligned} \quad (5.29)$$

$$kh - M(1 - y4_{s,k}) \leq \theta_s^t \leq kh + M(1 - y4_{s,k}), \quad \forall s \in S_s, k \in S_k \quad (5.30)$$



**Figure 5.4** The end of the transition locates at the point inside the bounds with the prior one locating outside the bounds

Constraints that link scheduling and control level

Initial values and steady state values are linked by  $y_{p,s}$

$$x_{in,s} = \sum_{p=1}^{N_p} x_{ss,p} y_{p,s-1}, \forall s \in (S_s - \{1\}) \quad (5.31)$$

$$x_{in,1} = \sum_{p=1}^{N_p} x_{ss,p} y_{p,N_s} \quad (5.32)$$

$$x_{s,1} = x_{in,s}, \forall s \in S_s \quad (5.33)$$

Equations (5.31) and (5.32) calculate the initial state value at each slot. Equation (5.33) defines the initial state value  $x_{in,s}$  as the value of the first sample step.

$$u_{in,s} = \sum_{p=1}^{N_p} u_{ss,p} y_{p,s-1}, \forall s \in (S_s - \{1\}) \quad (5.34)$$

$$u_{in,1} = \sum_{p=1}^{N_p} u_{ss,p} y_{p,N_s} \quad (5.35)$$

$$u_{s,1} = u_{in,s}, \quad \forall s \in S_s \quad (5.36)$$

Equations (5.34)-(5.36) compute the initial value for manipulated variables following the similar way as shown in equations (5.31)-(5.33). Note that  $x_{s,1}$  and  $u_{s,1}$  are also present in the dynamic PWA in equation (5.19). They are linked to the scheduling variables  $(x_{ss,p}, u_{ss,p}, y_{p,s})$  through (5.31)-(5.36). Thus constraints (5.31)-(5.36) demonstrate how scheduling and control are linked.

The desired values and steady state values are linked by  $y_{p,s}$

$$\bar{x}_s = \sum_{p=1}^{N_p} x_{ss,p} y_{p,s}, \quad \forall s \in S_s \quad (5.37)$$

$$\bar{u}_s = \sum_{p=1}^{N_p} u_{ss,p} y_{p,s}, \quad \forall s \in S_s \quad (5.38)$$

Equation (5.37) and (5.38) compute the desired state values and desired manipulated variables at each slot.

#### Optimization problem

To achieve economically optimal operations of chemical processes, we utilize the objective of maximizing profit per unit time, which can be calculated as follows. Profit per unit time ( $\Phi$ ) = (Revenue – Raw material cost – Utility cost)/Cycle time

$$\Phi = \Phi_1 - \Phi_2 - \Phi_3 \quad (5.39)$$

$$\Phi_1 = \sum_{p=1}^{N_p} \frac{P_p W_p}{T_c} \quad (5.40)$$

$$\Phi_2 + \Phi_3 = \left( \sum_{s=1}^{N_s} \sum_{k=1}^{\theta_s^t / h_s} (P_r u_{1s,k} + P_u u_{2s,k}) h_s + \sum_{s=1}^{N_s} (P_r \bar{u}_{1s} + P_u \bar{u}_{2s}) \Theta_s \right) \frac{1}{T_c} \quad (5.41)$$

where  $\Phi_1$ ,  $\Phi_2$  and  $\Phi_3$  represent the revenue rate, raw material cost rate and utility cost respectively. Note that for continuous processes raw material cost and utility cost (heating/cooling) can be combined in (5.41) where  $u_1$  represents raw material feeding flow rate and  $u_2$  represents utility flow rate. The total revenue is given as the amount produced ( $W_p$ ) times the product price ( $P_r$ ). Raw material cost is composed of two parts, the raw material consumption during production periods (the second term in the parenthesis of (5.41)) and during the transition periods (the first term in the parenthesis of (5.41)). Utility cost is calculated following the similar way of raw material cost. Note that the raw material consumed during the transition is the shaded area of Figure 4 which can be calculated as

$$\int_{t=0}^{t=\theta_s^t} u(t) dt \approx \sum_{k=1}^{k=\theta_s^t / h_s} u_{s,k} h_s \quad (5.42)$$

Combining the objective functions described in (5.39)-(5.41) and the constraints at both scheduling level (5.8)-(5.18), control level (5.19)-(5.30) and the linking constraints (5.31)-(5.38) we obtain the optimization model for the integrated problem as shown in (5.43).

$$\begin{aligned} & \max_{y_{p,s}, x_{s,k}, u_{s,k}, \theta_s^t, \Theta_p, W_p, t_s^s, t_s^e, T_c} \Phi_1 - \Phi_2 - \Phi_3 \\ \text{s.t.} \quad & \begin{cases} (5.8)-(5.18) \text{ constraints at scheduling level} \\ (5.19)-(5.30) \text{ constraints at control level} \\ (5.31)-(5.38) \text{ linking constraints} \end{cases} \end{aligned} \quad (5.43)$$

## 5.4 Fast MPC for PWA systems

### 5.4.1 The role of fast MPC

The MPC problem for LTI systems with quadratic objective corresponds to a quadratic programming (QP) problem. Fast MPC solves the QP problem online to satisfactory optimality criteria, whereas if mp-MPC is employed the calculations are done offline.

One of the shortcomings of mp-MPC however, is that the number of critical regions grows exponentially with respect to the size of MPC problem (e.g. number of input and output, prediction horizon). In contrast, fast MPC can solve large scale problem online. Another advantage of fast MPC is that fast MPC is capable to perform Real Time Optimization (RTO) as described by problem (5.44) where the reference  $\bar{x}_k$  (set point) is a continuous function rather than a fixed value. However, mp-MPC treats the reference trajectory as parameters and the parameter space (the critical regions) is exponentially increasing with respect to the length of prediction horizon (the length of reference track). This makes it impossible to apply mp-MPC in cases with long prediction horizon.

$$\begin{aligned} \min_{u_k} \quad & \sum_{k=1}^{N-1} \left( \|x_{k+1} - \bar{x}_{k+1}\|_2^{Q_k} + \|u_k - \bar{u}_k\|_2^{R_k} \right) + \|x_{k+N} - \bar{x}_{k+N}\|_2^{Q_N} \\ \text{s.t.} \quad & \begin{cases} x_1 = x^0 \\ x_{k+1} = Ax_k + Bu_k + C, \quad k = 1, \dots, N-1 \\ x_{\min} \leq x_k \leq x_{\max}, \quad k = 1, \dots, N \\ u_{\min} \leq u_k \leq u_{\max}, \quad k = 1, \dots, N \end{cases} \end{aligned} \quad (5.44)$$

The general methodology of fast MPC for linear systems is to transform the MPC problem into a convex quadratic programming, apply nonlinear programming methods and speed up the computation by exploring the problem structure. The detail solution approaches can be categorized as, interior point method (barrier method), active set method and fast gradient method. Using interior point method the MPC problem is built as a QP and the inequality constraints of

QP are treated with a log barrier and added to the objective (Rao et al., 1998) (Wang and Boyd, 2008). The resulting problem is then solved using Newton's method with the barrier coefficient adjusted in each iteration till convergence. When active set method is applied to MPC problem (Ferreau et al., 2008), equality problems are defined by active set of the original QP and updated when different active constraints are considered in the equality problems. Thus the equality problems are solved sequentially and the iterations are terminated if all KKT conditions are satisfied. Fast gradient method (Nesterov, 1983) has also been applied in solving the linear quadratic MPC problem where iteration algorithms with derived lower iteration bounds are developed (Richter et al., 2009) (Richter et al., 2012). Recently a primal-dual interior point method is applied in solving MPC problems (A. Domahidi et al., 2012) where the authors proposed an efficient solution method for the KKT conditions derived from the linear MPC problem and thus speed up the interior point method.

#### 5.4.2 Fast MPC for PWA systems

Note that the above mentioned fast MPC strategies are targeting linear systems. In the following we propose an algorithm of implementing fast MPC on nonlinear systems (problem (5.45)). The basic idea is to firstly derive the PWA model from nonlinear dynamics, and then iteratively locate the LTI in PWA and implement fast MPC at the local LTI so as to quickly drive the states to the next sample step.

$$\begin{aligned}
 & \min_{x(t), u(t)} \int_{t=0}^{t=t_H} (x^T Q x + u^T R u) dt + x_N^T Q x_N \\
 & s.t. \begin{cases} x(0) = x^0 \\ \dot{x} = f(x, u) \\ x_{\min} \leq x(t) \leq x_{\max}, \quad 0 \leq t \leq t_H \\ u_{\min} \leq u(t) \leq u_{\max}, \quad 0 \leq t \leq t_H \end{cases} \quad (5.45)
 \end{aligned}$$

Step 1: Transform nonlinear dynamics  $\dot{x} = f(x, u)$  into PWA using the proposed optimization method in section 5.2.2.

$$PWA = \bigcup_i LTI^{(i)} \quad (5.46)$$

$$\begin{aligned} LTI^{(i)}: & x(k+1) = A_i x(k) + B_i u(k) + C_i \\ & \text{if } x(k) \in \Omega_i = \{x : V_i x \leq W_i\}, i \in S_i = \{1, 2, \dots, N_i\} \\ & \text{where } \bigcup_i \Omega_i = \Omega_x \text{ and } \Omega_{i_1} \cap \Omega_{i_2} = \emptyset \text{ if } i_1 \neq i_2 \end{aligned} \quad (5.47)$$

Step 2: Set the initial state and initial manipulated variables  $(x^0, u^0)$

Step 3: Locate the corresponding LTI for current states

$$\begin{aligned} & \text{if } x \in \Omega_i = \{x : V_i x \leq W_i\}, i \in S_i = \{1, 2, \dots, N_i\}, \\ & \text{then select } LTI^{(i)} : x(k+1) = A_i x(k) + B_i u(k) + C_i \end{aligned} \quad (5.48)$$

This step can be computationally very expensive especially when  $x$  and  $u$  are of high dimensions.

If an enumeration method is employed someone has to loop over all the LTIs to determine the position of  $(x(k+1), u^*(k))$ , leading to complexity that is increasing in the order of  $o\left(\prod_i N_i\right)$  In

order to lower the complexity we employ binary search method (Tøndel et al., 2003) (Jones et al., 2006) with much lower computational complexity  $o\left(\sum_i \lg(N_i)\right)$ .

When the dimension is high and the number of discretization points is large, binary search is significantly faster. This is shown in (Table 5.1) with an example of searching for a LTI in a 3-dimensional PWA with varying number of discretization points (10, 100 and 200).

**Table 5.1 Comparison of enumeration and binary search based on their CPU time used in searching for a LTI in a 3-dimensional PWA**

| Number of discretization | $N_1=N_2=N_3=10$ | $N_1=N_2=N_3=100$ | $N_1=N_2=N_3=200$ |
|--------------------------|------------------|-------------------|-------------------|
| Binary search CPU time   | 0.016s           | 0.036s            | 0.047s            |
| Enumeration CPU time     | 0.016s           | 1.922s            | 14.266s           |



Step 4: Solve the MPC problem (5.49) for LTI using toolbox FORCES (Alexander Domahidi, 2012) which is based on primal-dual interior-point method (A. Domahidi et al., 2012), and obtain  $u^*(k)$ . Primal-dual interior-point method efficiently solves for the KKT conditions derived from the linear MPC problem and thus speed up the interior point method.

$$\begin{aligned} \min_{u_k} \quad & \sum_{k_p=1}^{N-1} \left( \|x_{k+k_p} - \bar{x}_{k+k_p}\|_2^{Q_{k_p}} + \|u_{k+k_p-1} - \bar{u}_{k+k_p-1}\|_2^{R_{k_p}} \right) \\ & + \|x_{k+N} - \bar{x}_{k+N}\|_2^{Q_N} \\ \text{s.t.} \quad & \begin{cases} x_k = x^0 \\ x_{k+k_p} = A_i x_{k+k_p-1} + B_i u_{k+k_p-1} + C_i, \quad k_p = 1, \dots, N \\ x_{\min} \leq x_{k+k_p} \leq x_{\max}, \quad k_p = 1, \dots, N \\ u_{\min} \leq u_{k+k_p} \leq u_{\max}, \quad k_p = 1, \dots, N \end{cases} \end{aligned} \quad (5.49)$$

Step 5: Evaluate the state transfer using the following equation:

$$x(k+1) = A_i x(k) + B_i u^*(k) + C_i \quad (5.50)$$

Step 6:  $k=k+1$ , go to step 3.

### 5.4.3 Stability analysis

Input-output stability implies Bounded Input-Bounded Output (BIBO) stability i.e. in a well behaved system a bounded input should result in a bounded output (Safonov, 1980). Since PWA is a discrete system and the stability is relevant to the sample step, we analyze the stability in the context of z-transform and derive the necessary conditions for a stable PWA. The necessary conditions provide a reference when deciding the sample step.

A general form of PWA  $x(k+1) = Ax(k) + Bu(k) + C$  where  $x(k) \in \mathbb{R}^{n \times 1}$   $u(k) \in \mathbb{R}^{m \times 1}$   $A \in \mathbb{R}^{n \times n}$

$B \in \mathbb{R}^{n \times m}$   $C \in \mathbb{R}^{n \times 1}$  has an equivalent form

$$x'(k+1) = Ax'(k) + Bu'(k) \quad (5.51)$$

where  $x'(k) = x(k) + M_1$  and  $u'(k) = u(k) + M_2$ .  $M_1$  and  $M_2$  can be obtained through

$$(A-I)M_1 + BM_2 = C.$$

Applying  $z$  transform to (5.51) yields

$$(I - Az^{-1})X(z) = Bz^{-1}U(z) \quad (5.52)$$

Thus we obtain a transfer function

$$\frac{X(z)}{U(z)} = (I - Az^{-1})^{-1} Bz^{-1} \quad (5.53)$$

The necessary conditions for BIBO stability of (5.52) are that the absolute value of all the diagonal terms of  $A$  should be less than one:

$$|A_{ii}| < 1, 1 \leq i \leq n \quad (5.54)$$

## 5.5 Case studies

In order to demonstrate the feasibility of the proposed PWA identification and solve the integrated problem using PWA and fast MPC, we studied two numerical cases, a SISO CSTR and a MIMO CSTR continuous production.

### 5.5.1 Single-Input Single-Output (SISO) CSTR

The data for the first case study involving a CSTR is taken from Flores-Tlacuahuac and Grossmann (Flores-Tlacuahuac and Grossmann, 2006). The reaction  $3R \xrightarrow{k} P$  takes place in an isothermal CSTR with reaction rate  $-r_R = kC_R^3$ , while products A, B, C, D and E, which are differentiated by their concentration  $C_R$  (Table 5.2), are manufactured in a cyclic mode (Figure 5.5 and Figure 5.6). The basic dynamic model of the process is shown in Equation (5.55).

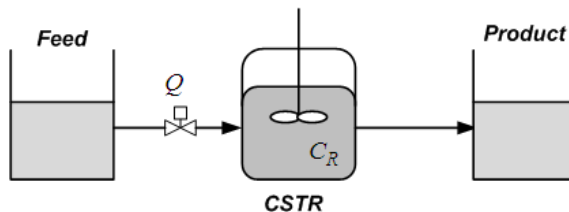
$$\frac{dC_R}{dt} = \frac{Q}{V}(C_0 - C_R) + r_R \quad (5.55)$$

where  $C_0$  is feed stream concentration,  $Q$  is the feed flow rate (i.e. manipulated variable), and

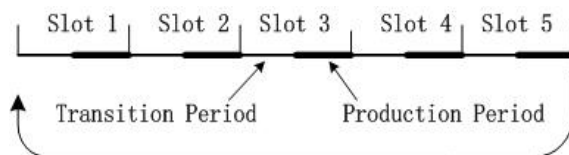
$C_R$  is the concentration of the raw material in the outflow (i.e. state variable). Using  $u$  and  $x$  to

represent the manipulated and state variable respectively leads to the following:

$$\frac{dx}{dt} = \frac{u}{5000} (1 - x) - 2x^3 \quad (5.56)$$



**Figure 5.5 CSTR cyclic production, feeding flow rate is the manipulated variable and raw material concentration is the state variable**



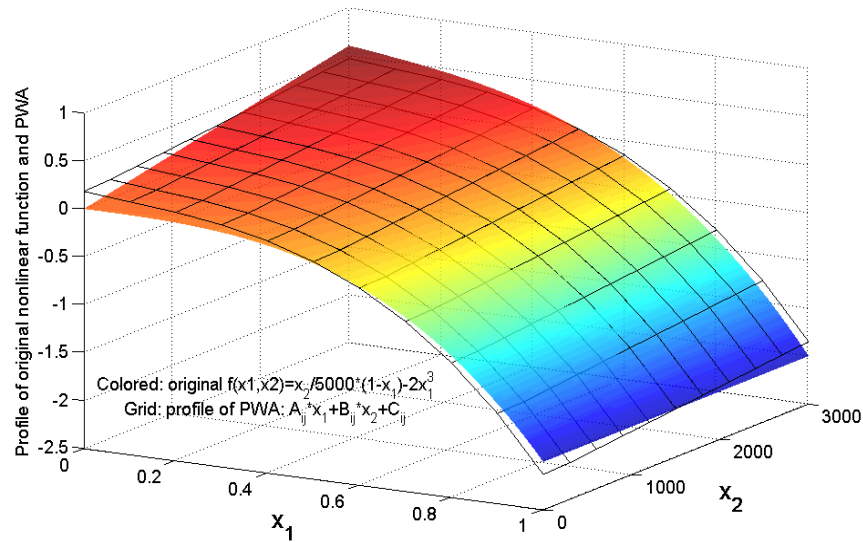
**Figure 5.6 CSTR cyclic production, each slot is composed of a transition period and a production period**

**Table 5.2 Steady state information of a CSTR with cyclic production**

| Product | $Q(L / hour)$ | $C_R(mol / L)$ | Demand rate<br>( $kg / hour$ ) | Product price<br>( $\$ / kg$ ) |
|---------|---------------|----------------|--------------------------------|--------------------------------|
| A       | 10            | 0.0967         | 3                              | 200                            |
| B       | 100           | 0.2            | 8                              | 150                            |
| C       | 400           | 0.3032         | 10                             | 130                            |
| D       | 1000          | 0.393          | 10                             | 125                            |
| E       | 2500          | 0.5            | 10                             | 120                            |

### PWA identification

Following the PWA identification approach presented in section 5.2.2, we transform the nonlinear dynamics in (5.56) into PWA by solving problems (5.57) using GAMS/IPOPT. Using 10 discretization points for  $x$  and  $u$  results in the solution of problem (5.57) in 3.4 CPU sec. The profile of the original function and PWA are shown in Figure 5.7 where the smooth and colored sphere represents the original nonlinear dynamic and the grid represents the PWA.



**Figure 5.7 Profiles of the nonlinear dynamic and the PWA**

#### Solving the integrated problem

In this case study we assign five slots, five products, 20 sample steps during transitions. Therefore  $N_p=N_S=5$ ,  $N_k=20$ . We solve the integrated problem using GAMS/SBB and summarize the results in Table 5.3.

**Table 5.3 Results of the integrated problem of SISO CSTR case**

|                                       |                                     |
|---------------------------------------|-------------------------------------|
| Number of variables                   | 1601                                |
| Number of binary variables            | 1225                                |
| Number of constraints                 | 4542                                |
| CPU time (s)                          | 334.64                              |
| Relative Gap                          | 1.19%                               |
| Optimal sequence                      | D-E-A-B-C                           |
| Cycle time                            | 116.52                              |
| Transition time in slot 1 to 5 (hour) | 0.600, 1.000, 2.200, 0.500, 0.600   |
| Production time in slot 1 to 5 (hour) | 0.198, 0.880, 44.282, 13.750, 2.511 |
| Revenue (\$)                          | 4639.03                             |
| Raw material cost (\$)                | 1174.48                             |

|             |         |
|-------------|---------|
| Profit (\$) | 3464.55 |
|-------------|---------|

It takes 334.64 sec to solve the integrated problem using SBB solver in GAMS. The optimal production sequence is D-E-A-B-C. The scheduling solution including transition time, production time for each product and the economic performance such as revenue, cost and profit are provided in Table 5.3.

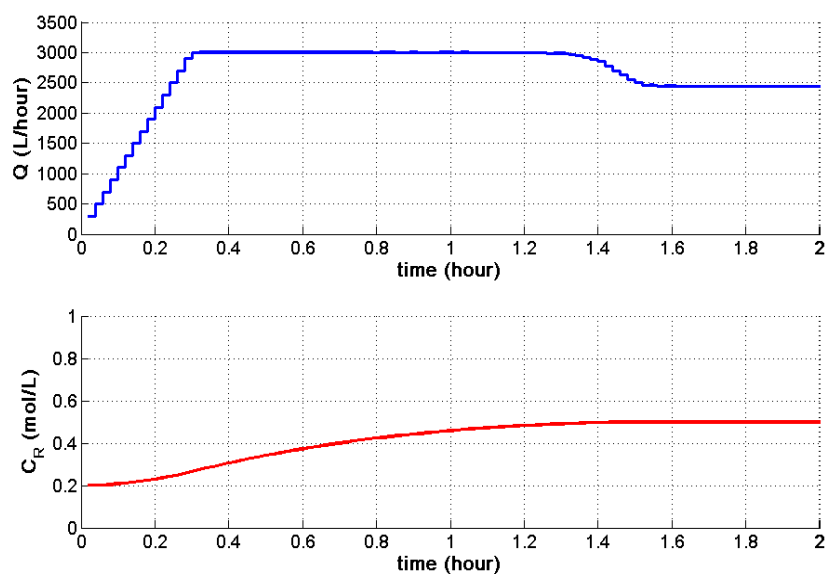
#### Implementation of fast MPC

We implement fast MPC on the obtained PWA system following the algorithm presented in section 5.4.2. The original control problem is a nonlinear MPC problem (5.57), which is transformed through the PWA identification method.

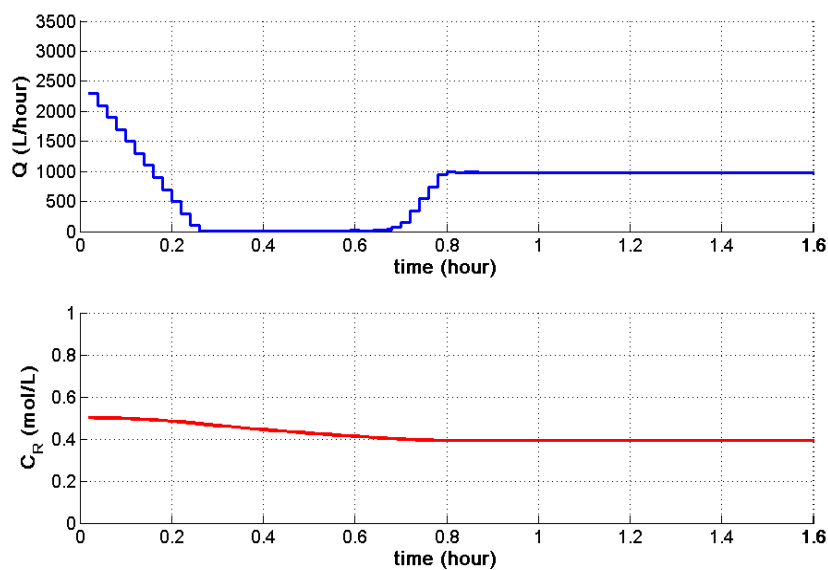
$$\begin{aligned}
 \min_u J &= \int_0^{t_H} |x - x_{ref2}| dt \\
 s.t. \quad &\begin{cases} \frac{dx}{dt} = \frac{u}{5000} (1 - x) - 2x^3 \\ x(0) = x_{ref1} \\ 0 \leq x(t) \leq 1, \quad 0 \leq t \leq t_H \\ 0 \leq u(t) \leq 3000, \quad 0 \leq t \leq t_H \\ \Delta u \leq 200 \end{cases}
 \end{aligned} \tag{5.57}$$

The PWA of this case has 100 *LTIs*. It takes 0.0636 CPU sec to solve for the entire transition using a sample step of 0.02 hour, and 0.000279 CPU sec for the MPC problem over the local LTI at one sample step. The calculations were performed using a PC of 1.86GHz/4GB RAM. Figure 5.8 and Figure 5.9 illustrate two examples of transition (transition from product 2 to 5 and transition from product 5 to 4). Both figures demonstrate the profiles of manipulated variables and state variables. Note that the manipulated variables are subject to a bound of 200L/hour that represents the maximum increasing or decreasing rate within one sample interval. In Figure 5.8 the manipulated variable feeding flow rate  $Q$  increases up to its maximum value 3000 L/hour to drive the concentration of raw material  $C_R$  from one steady state 0.2 mol/L to another steady state 0.5 mol/L in a short time period. Note that  $Q$  decreases to the corresponding steady state value of

2500 L/hour before  $C_R$  is approaching 0.5 mol/L. This is due to the involvement of MPC that helps to avoid the overshoot. Similarly in Figure 5.9 the feeding flow rate  $Q$  decreases to its minimum value 10 L/hour to speed up the transition and then adjusts to 1000 L/hour to catch the following production period where the steady state value of  $C_R$  is 0.393 mol/L.



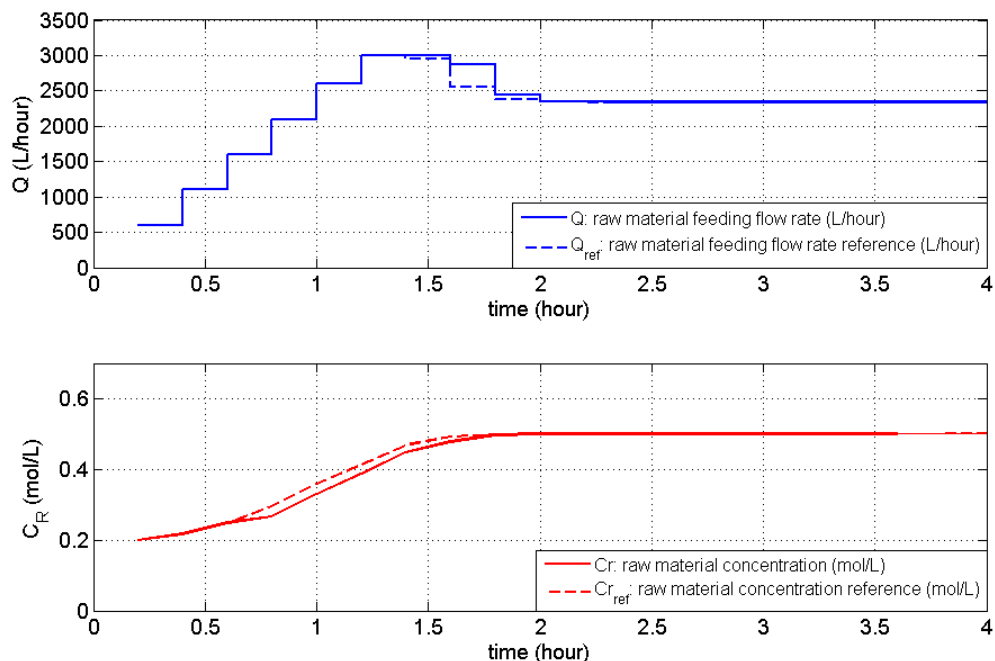
**Figure 5.8 Transition profile from product 2 to product 5 in SISO CSTR, obtained by fast MPC**



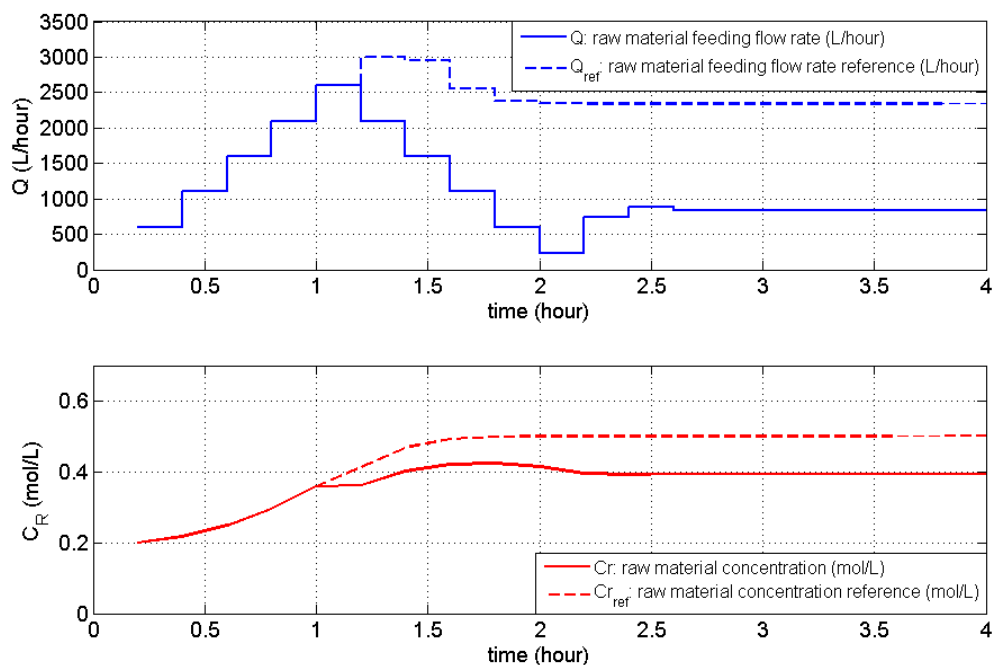
**Figure 5.9 Transition profile from product 5 to product 4 in SISO CSTR, obtained by fast MPC**

### The effects of threshold at the state feedback in the proposed framework

As describe in the introduction section a threshold of state deviation is employed to determine whether the state information is fed back to the inner fast MPC or outer integrated problem so that the scheduling solution can be updated. As we test many different values of the threshold we find that a state deviation less than 0.08 mol/L would not disrupt the scheduling solution and thus could be handled by the fast MPC, while in the case of a state deviation greater than that it's more profitable to update the scheduling solution. Therefore we set up the threshold as 0.08 mol/L. Figure 5.10 and Figure 5.12 demonstrate two examples of the responses to minor disturbance at transtion and production periods where the scheduling solutions are not interrupted. In Figure 5.10 the disturbance occurs at 0.7 h with magnitude 0.04 mol/L. Since this is a minor disturbance the scheduling solution of the integrated problem is not updated *i.e.* the process continues with the current transition to product 5. In Figure 5.12 the disturbance occurs at steady state with magnitude 0.05 mol/L and the control action adjusts the deviated state back to the steady state. Figure 5.11 however, demonstrates a significant disturbance at transitions and thus the scheduling solution is updated *i.e.* the process goes to another production stage. In this case the disturbance occurs at 1.2 h with magnitute 0.09 mol/L. This is a significant deviation and the scheduling solution is updated in order to keep the optimality of the overall operations. So product 4 is produced next instead of product 5.

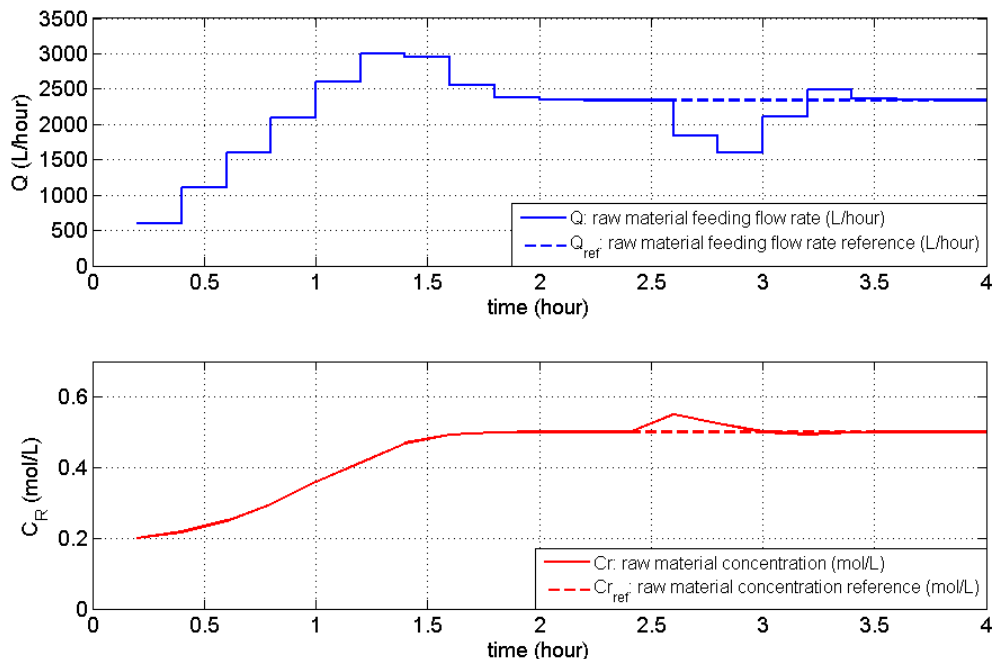


**Figure 5.10 Disturbance with magnitude -0.04 (less than threshold 0.08) at time 0.7 h (in transition), handled by fast MPC, scheduling solution does not change**



**Figure 5.11 Disturbance with magnitude -0.09 (great than threshold 0.08) at time 1.2 h (in transition), steady state value (scheduling solution) updated by integrated problem, process goes to product 4 (SS=0.393) instead of product 5 (SS=0.5)**

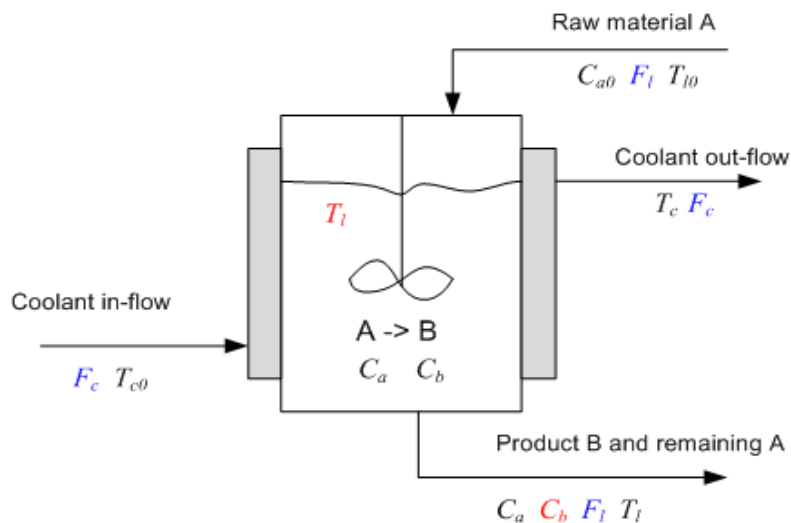




**Figure 5.12 Disturbance with magnitude +0.05 (less than threshold 0.08) at time 2.6 h (in production), handled by fast MPC, scheduling solution does not change**

### 5.5.2 Multi-Input Multi-Output (MIMO) CSTR

In this case study we investigate a MIMO CSTR cyclic production where reaction  $A \rightarrow B$  takes place with first order reaction rate:  $r = -kC_a$  (Figure 5.13). Specifications of this process is adopted from Camacho and Bordons Alba (Camacho and Bordons Alba, 2007). The manipulated variables are  $F_l$  (flow rate of the liquid) and  $F_c$  (flow rate of the coolant). States variables are  $C_b$  (concentration of product B) and  $T_l$  (liquid temperature). Three steady states are involved in the cyclic production (Table 5.4). The detailed process specifications are provided in Table 5.5. The utility price for the coolant  $F_c$  is  $\$3/\text{m}^3$ , and the raw material price is  $\$10/\text{m}^3$ . The model derived from mass and energy balance are represented by Equations (5.58) and (5.59), respectively.



**Figure 5.13 A MIMO CSTR process**

**Table 5.4 Products at different steady states for MIMO CSTR case study**

| Steady state   | $C_b$<br>(mol/L) | $T_l$<br>(K) | $F_l$<br>(m <sup>3</sup> /h) | $F_c$<br>(m <sup>3</sup> /h) | Demand<br>(L/h) | Price<br>(\$/L) | Inventory<br>(\$/h) |
|----------------|------------------|--------------|------------------------------|------------------------------|-----------------|-----------------|---------------------|
| SS1: produce A | 1                | 290          | 1829                         | 774                          | 110             | 17              | 1.7                 |
| SS2: produce B | 1.3              | 310          | 1266                         | 264                          | 120             | 25              | 2                   |
| SS3: produce C | 2                | 330          | 610                          | 132                          | 70              | 32              | 1.8                 |

**Table 5.5 Process specifications of MIMO CSTR**

|  |                        |
|--|------------------------|
| $k$ reaction constant                      | 26/h                   |
| $H$ heat of reaction                       | 70000 kJ/kmol          |
| $\rho_l$ liquid density                    | 800 kg/m <sup>3</sup>  |
| $\rho_c$ coolant density                   | 1000 kg/m <sup>3</sup> |
| $C_{pl}$ specific heat of liquid           | 3 kJ/(kg K)            |
| $C_{pc}$ specific heat of coolant          | 4.19 kJ/(kg K)         |
| $V_l$ tank volume                          | 24 m <sup>3</sup>      |
| $T_{l0}$ liquid entering temperature       | 283 K                  |
| $T_{c0}$ coolant in-flow temperature       | 273 K                  |
| $T_c$ coolant out-flow temperature         | 303 K                  |
| $C_{a0}$ initial concentration of reactant | 4 mol/L                |

$$\frac{d(V_l C_b)}{dt} = V_l k(C_{a0} - C_b) - F_l C_b \quad (5.58)$$

$$\begin{aligned} \frac{d(V_l \rho_l C_{pl} T_l)}{dt} = & F_l \rho_l C_{pl} T_{l0} - F_l \rho_l C_{pl} T_l + F_c \rho_c C_{pc} (T_{c0} - T_c) \\ & + V_l k(C_{a0} - C_b) H \end{aligned} \quad (5.59)$$

Identifying the PWA from (5.58) and (5.59) leads to equations (5.60) and (5.61).

$$C_b(k+1) = [A_{11} \ A_{12}] \begin{bmatrix} C_b(k) \\ T_l(k) \end{bmatrix} + [B_{11} \ B_{12}] \begin{bmatrix} F_l(k) \\ F_c(k) \end{bmatrix} + C_1 \quad (5.60)$$

$$T_l(k+1) = [A_{21} \ A_{22}] \begin{bmatrix} C_b(k) \\ T_l(k) \end{bmatrix} + [B_{21} \ B_{22}] \begin{bmatrix} F_l(k) \\ F_c(k) \end{bmatrix} + C_2 \quad (5.61)$$

That can be represented in a matrix format as follows:

$$\begin{bmatrix} C_b(k+1) \\ T_l(k+1) \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} C_b(k) \\ T_l(k) \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} F_l(k) \\ F_c(k) \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \quad (5.62)$$

The obtained PWA is composed of three LTIs as shown in (5.63)-(5.65).

$$\begin{aligned} \begin{bmatrix} C_b(k+1) \\ T_l(k+1) \end{bmatrix} = & \begin{bmatrix} -0.0164 & 0 \\ -7.4115 & 0.2377 \end{bmatrix} \begin{bmatrix} C_b(k) \\ T_l(k) \end{bmatrix} + \begin{bmatrix} -0.0004 & 0 \\ -0.0029 & -0.0218 \end{bmatrix} \begin{bmatrix} F_l(k) \\ F_c(k) \end{bmatrix} \\ & + \begin{bmatrix} 1.7788 \\ 250.7210 \end{bmatrix}, \text{ if } 0 \leq C_b < 1.15 \end{aligned} \quad (5.63)$$

$$\begin{aligned} \begin{bmatrix} C_b(k+1) \\ T_l(k+1) \end{bmatrix} = & \begin{bmatrix} 0.2181 & 0 \\ -7.4115 & 0.4722 \end{bmatrix} \begin{bmatrix} C_b(k) \\ T_l(k) \end{bmatrix} + \begin{bmatrix} -0.0005 & 0 \\ -0.0113 & -0.0218 \end{bmatrix} \begin{bmatrix} F_l(k) \\ F_c(k) \end{bmatrix} \\ & + \begin{bmatrix} 1.7025 \\ 193.2533 \end{bmatrix}, \text{ if } 1.15 \leq C_b < 1.6 \end{aligned} \quad (5.64)$$

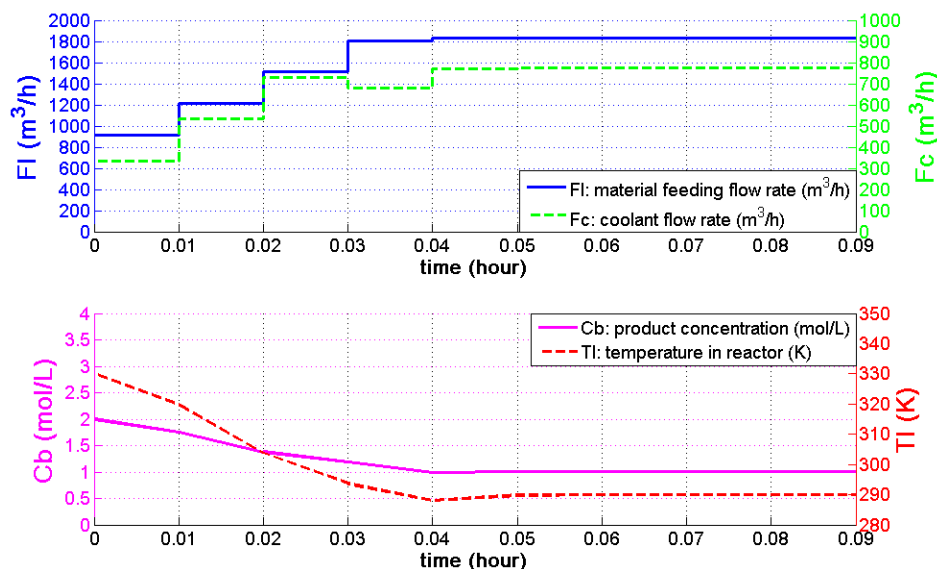
$$\begin{aligned} \begin{bmatrix} C_b(k+1) \\ T_l(k+1) \end{bmatrix} = & \begin{bmatrix} 0.4918 & 0 \\ -7.4115 & 0.7459 \end{bmatrix} \begin{bmatrix} C_b(k) \\ T_l(k) \end{bmatrix} + \begin{bmatrix} -0.0008 & 0 \\ -0.0196 & -0.0218 \end{bmatrix} \begin{bmatrix} F_l(k) \\ F_c(k) \end{bmatrix} \\ & + \begin{bmatrix} 1.5247 \\ 113.5021 \end{bmatrix}, \text{ if } C_b \geq 1.6 \end{aligned} \quad (5.65)$$

**Table 5.6 Scheduling solutions for the integrated problem of MIMO CSTR case**

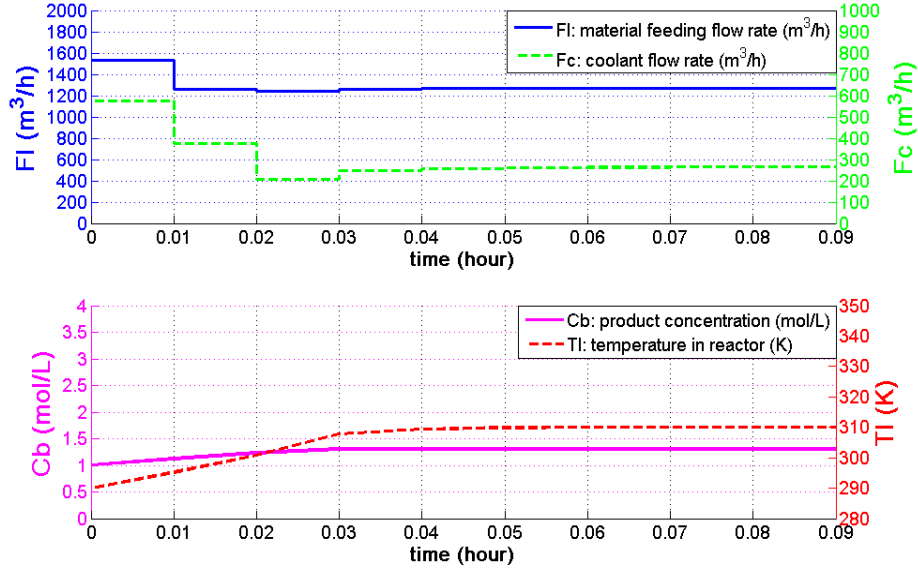
|                         |                               |
|-------------------------|-------------------------------|
| Number of variables     | 555                           |
| Number of constraints   | 2712                          |
| Solver and CPU time (s) | DICOPT, 14.1s on 3.9GHz/4GRAM |
| Relative Gap            | 0.00%                         |
| Optimal sequence        | C-A-B                         |
| Cycle time (h)          | 10                            |

|                                       |                  |
|---------------------------------------|------------------|
| Transition time in slot 1 to 3 (hour) | 0.1, 0.15, 0.07  |
| Production time in slot 1 to 3 (hour) | 1.15, 0.60, 7.93 |
| Revenue (\$)                          | 29211.73         |
| Raw material cost(\$)                 | 12222.46         |
| Utility cost (\$)                     | 829.03           |
| Inventory cost (\$)                   | 3513.85          |
| Profit(\$)                            | 12646.39         |

The solution of the integrated problem is provided in Table 5.6. It takes 14.1 sec to solve the integrated problem which maximizes the profit using DICOPT solver in GAMS. The optimal production sequence is C-A-B. The transition time, production time for each product and the economic performance such as revenue, cost and profit are provided in Table 5.6. It takes 0.028s for the fast MPC to obtain the entire transitions (sample step is 0.02 hour) and 0.00045s for the MPC problem over the local LTI at one sample step, using a PC of 1.86GHz/4GB RAM. Figure 5.14 and Figure 5.15 illustrate two examples of transition (transition from steady state 3 to 1 and transition from steady state 1 to 2).



**Figure 5.14 Transition profile from steady state 3 to steady state 1 in MIMO CSTR, obtained by fast MPC**



**Figure 5.15 Transition profile from steady state 1 to steady state 2 in MIMO CSTR, obtained by fast MPC**

#### A comparison between fast MPC and mp-MPC

Here we compare the performance of fast MPC and mp-MPC in terms of algorithms and computational complexity. We present solution algorithms of implementing fast MPC ((5.45)-(5.50) in section 5.4.2) and mp-MPC ((5.67)-(5.71)) on nonlinear systems, and their detailed computation time as summarized in Table 5.7.

MPC problem for nonlinear systems (The original problem)

$$\begin{aligned}
 & \min_{x(t), u(t)} \int_{t=0}^{t=t_H} (x^T Q x + u^T R u) dt + x_N^T Q x_N \\
 & s.t. \begin{cases} x(0) = x^0 \\ \dot{x} = f(x, u) \\ x_{\min} \leq x(t) \leq x_{\max}, \quad 0 \leq t \leq t_H \\ u_{\min} \leq u(t) \leq u_{\max}, \quad 0 \leq t \leq t_H \end{cases}
 \end{aligned} \tag{5.66}$$

#### Algorithms of implementing mp-MPC on nonlinear systems

Step 1: Transfer nonlinear dynamic  $\dot{x} = f(x, u)$  into PWA using the proposed optimization based method

$$PWA = \bigcup_i LTI^{(i)} \tag{5.67}$$

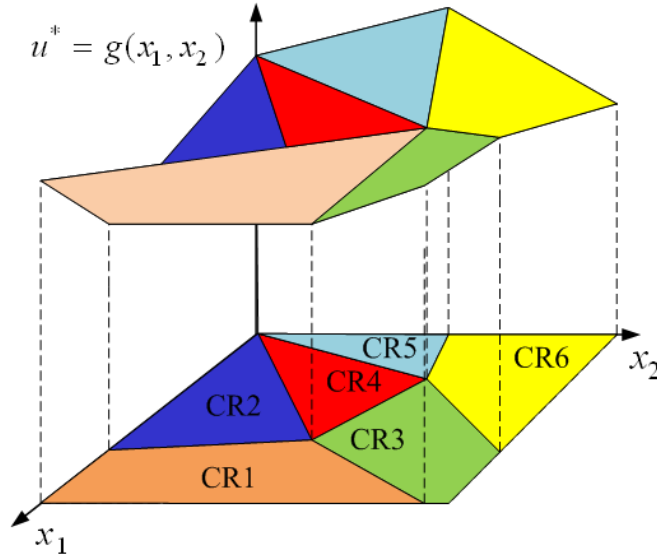
$$\begin{aligned}
LTI^{(i)}: x(k+1) &= A_i x(k) + B_i u(k) + C_i \\
&\text{if } x(k) \in \Omega_i = \{x : V_i x \leq W_i\}, i \in S_i = \{1, 2, \dots, N_i\} \\
&\text{where } \bigcup_i \Omega_i = \Omega_x \text{ and } \Omega_{i_1} \cap \Omega_{i_2} = \emptyset \text{ if } i_1 \neq i_2
\end{aligned} \tag{5.68}$$

Step 2: Solve the following mp-MPC problem offline for the PWA system obtained in step 1.

$$\begin{aligned}
\min_{u_k} \quad & \sum_{k_p=1}^{N-1} \left( \|x_{k+k_p} - \bar{x}_{k+k_p}\|_2^{Q_{k_p}} + \|u_{k+k_p-1} - \bar{u}_{k+k_p-1}\|_2^{R_{k_p}} \right) \\
& + \|x_{k+N} - \bar{x}_{k+N}\|_2^{Q_N} \\
s.t. \quad & \begin{cases} x_k = x^0 \\ x_{k+k_p} = A_i x_{k+k_p-1} + B_i u_{k+k_p-1} + C_i, \quad k_p = 1, \dots, N \\ x_{\min} \leq x_{k+k_p} \leq x_{\max}, \quad k_p = 1, \dots, N \\ u_{\min} \leq u_{k+k_p} \leq u_{\max}, \quad k_p = 1, \dots, N \end{cases}
\end{aligned} \tag{5.69}$$

Step 3: Set the initial state and initial manipulated variables  $(x^0, u^0)$

Step 4: Locate the corresponding critical region (CR) (Figure 5.16) for current states and reference states in the prediction horizon, using binary searching method (Jones et al., 2006) (Tøndel et al., 2003).



**Figure 5.16** Locate the critical regions and evaluate the explicit solution obtained by mp-MPC

Step 5: Evaluate the manipulated variables as a function of states and obtain the optimal control inputs.

$$\begin{aligned} \text{if } x(k) \in CR_j = \{x : H_j x \leq K_j\}, j \in J = \{1, 2, \dots, N_j\} \\ \text{then } u^*(k) = F_j x(k) + G_j \end{aligned} \quad (5.70)$$

Step 6: Locate the corresponding LTI for the current states using binary searching method.

Step 7: State transfer

$$\begin{aligned} \text{if } x(k) \in \Omega_i = \{x : V_i x \leq W_i\}, i \in S_i = \{1, 2, \dots, N_i\} \\ \text{then } x(k+1) = A_i x(k) + B_i u^*(k) + C_i \end{aligned} \quad (5.71)$$

Step 8:  $k=k+1$ , go to step 4.

Note that mp-MPC and fast MPC share step 1. In mp-MPC, step 1, 2 are solved off-line, and steps 4 through 8 are calculated online in a loop manner. In fast MPC, step 1 is solved off-line, and Steps 3 through 6 are calculated online in a loop manner.

When comparing the computation complexity of fast MPC and mp-MPC, we focus on the online steps. Table 5.7 presents the detailed CPU time for each online step of fast MPC and mp-MPC.

**Table 5.7 Analysis of the computation time for fast MPC and mp-MPC, online steps only**

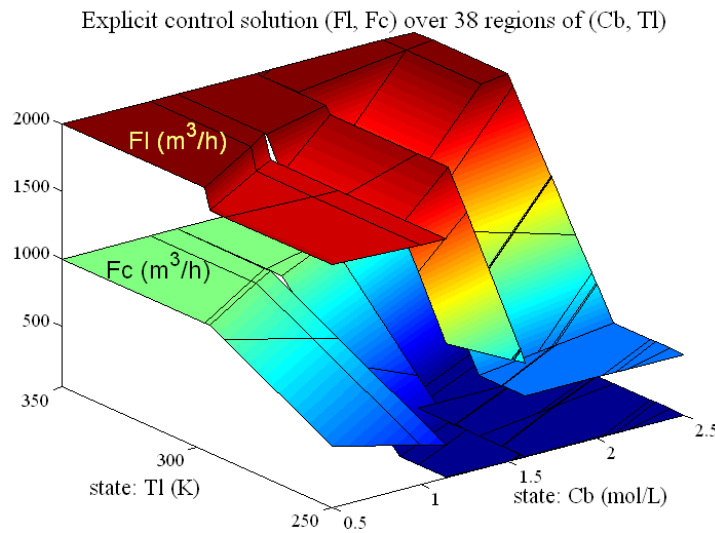
| CPU time | fast MPC   | mp-MPC  |
|----------|--|---|
| Step 3   | Locate LTI: $t_{LTI} = \varepsilon \sum_{m=1}^{N_x} \lg(N_m)$                    | N/A   |
| Step 4   | Solve QP using primal-dual interior-point method: $t_{QP}$                       | Locate CR:<br>$t_{CR} = \varepsilon \left( \sum_{m=1}^{N_x} \lg(N_m) + \sum_{n=1}^{N_H} \lg(N_n) \right)$                   |
| Step 5   | State transfer: $t_{ST}$   | Function evaluation: $t_{FE}$   |
| Step 6   | Move to next sample step: $t_{MO}$   | Locate the LTI: $t_{LTI} = \varepsilon \sum_{m=1}^{N_x} \lg(N_m)$   |
| Step 7   | N/A  | State transfer: $t_{ST}$  |
| Step 8   | N/A  | Move to next sample step: $t_{MO}$  |
| total    | $t_{fastMPC} = \varepsilon \sum_{m=1}^{N_x} \lg(N_m) + t_{QP} + t_{ST} + t_{MO}$ | $t_{mpMPC} = \varepsilon \left( 2 \sum_{m=1}^{N_x} \lg(N_m) + \sum_{n=1}^{N_H} \lg(N_n) \right) + t_{FE} + t_{ST} + t_{MO}$ |

**Table 5.8 Notations associated with Table 5.7**

|       |   |
|-------|---|
| $N_x$ | number of states dimension                          |
| $N_m$ | number of discretization point for the $m$ th state |

|               |   |
|---------------|---|
| $N_H$         | prediction horizon  |
| $N_n$         | number of discretization point for the $n$ th reference state in the prediction horizon |
| $\mathcal{E}$ | unit time in binary searching method  |
| $t_{LTI}$     | time consumed in locating $LTI$   |
| $t_{CR}$      | time consumed in locating critical regions  |
| $t_{ST}$      | time consumed in state transfer   |
| $t_{MO}$      | time consumed in moving to the next sample step   |
| $t_{FE}$      | time consumed in function evaluation  |
| $t_{fastMPC}$ | time consumed in implanting fast MPC in one sample step                                 |
| $t_{mpMPC}$   | time consumed in implanting mp-MPC in one sample step                                   |

We employ YALMIP toolbox (Löfberg, 2004) to solve the MPC problem in (5.69) and obtain the explicit control solution shown in Figure 5.17 where both manipulated variables  $F_I$  and  $F_C$  are functions of state variables  $C_b$  and  $T_I$ . Then we implement the explicit solution following steps 4 to 8 in the algorithm of implementing mp-MPC and obtain the same transition profiles as those obtained by fast MPC. Table 5.9 summarizes the computation time of fast MPC and mp-MPC in solving for one sample step as well as the whole transition period. As shown in the table, mp-MPC requires more computational time. This is consistent with the fact presented in Table 5.7 that mp-MPC takes more time due to its repeatedly locating the critical regions.



**Figure 5.17** Explicit solution for the control problem in transition from product 3 to 1 of MIMO CSTR case



**Table 5.9 Comparison of CPU time in implementing fast MPC and mp-MPC using a 1.86GHz/4G RAM PC**

| CPU time          | fast MPC | mp-MPC |
|-------------------|----------|--------|
| One sample step   | 0.00045s | 0.016s |
| Entire transition | 0.028s   | 0.437s |

## 5.6 Summary

In this study we build a novel framework for the integration of scheduling and control. In the framework, we proposed a heuristic determination of the threshold feedback value. For example in the case of SISO CSTR, we tested many cases of disturbances with different magnitudes and found that when the magnitude of disturbance was great than 0.09, there was a high probability that the scheduling solution should be updated in order to keep the optimality of the process. When the disturbance was less than 0.05, it's very likely that scheduling solution remains the same. Therefore, there is no exact threshold for the feedback. In this study we choose an empirical value between 0.05 and 0.09 to avoid unnecessarily computing the integrated problem and meanwhile guarantee the optimality when disturbance occurs.

This framework aims to simultaneously consider the scheduling and control problem and facilitate the online applications where disturbances are efficiently handled. In this framework a PWA model is identified from the process's first principle model and incorporated within the scheduling level, leading to an integrated model. More specifically, the PWA is obtained using optimization method that minimizes the error between the original nonlinearity and the proposed PWA composed of a group of local *LTIs*. Note that the PWA is incorporated as linear constraints with the scheduling level. This eliminates the nonlinear constraints brought by traditional integration that uses collocation point discretization, and thus reduces the computation complexity of the integrated problem. Results show that fast MPC computes the control problem much faster than mp-MPC does, though both of them perform in the speed of msec.

## Nomenclature:

### Index and sets

$i$  LTI in PWA

$s$  slot

$p$  product

$k$  sample points

$N_p$  number of products

$N_s$  number of slots

$N_k$  number of sample steps

$N_i$  number of polytopes in PWA

### Parameters

$Ma$  big positive number

$h$  sample step

$G_p$  production rate of product  $p$

$D_p$  demand of product  $p$

$P_p$  price of product  $p$

$P_r$  price of raw material

$\Theta_{\max}$  maximum production time

$A_i$  coefficient of LTI

$B_i$  coefficient of LTI

$C_i$  coefficient of LTI

$V_i$  coefficient of polytope

$W_i$  coefficient of polytope

$x_{\min}$  lower bound of state variables

$x_{\max}$  upper bound of state variables

$u_{\min}$  lower bound of manipulated variables

$u_{\max}$  upper bound of manipulated variables

$\Delta_u$  maximum increment of manipulated variables

$x_{\text{margin}}$  the margin of quality bounds

### Variables

$x_{s,k}$  state variable at sample step  $k$  in slot  $s$

$u_{s,k}$  manipulated variable at sample step  $k$  in slot  $s$

$x_{ss,p}$  steady state value of product  $p$

$u_{ss,p}$  steady manipulated value of product  $p$

$x_{in,s}$  initial state value in slot  $s$

$u_{in,s}$  initial manipulated value in slot  $s$

$\bar{x}_s$  desired state value in slot  $s$ (the set point)

$\bar{u}_s$  desired manipulated value in slot  $s$ (the set point)

$y_{p,s}$  binary variable indicating assignment of product  $p$  to slot  $s$

$y1_{s,k,i}$  binary variable indicating the selection of LTI in PWA

$y4_{s,k}$  binary variable indicating the end of transitions, equal to 1 if step  $k$  is the end of transition

$y41_{s,k}$  auxiliary binary variable for  $y4_{s,k}$ , equal to one if state is below the lower bound

$y42_{s,k}$  auxiliary binary variable for  $y4_{s,k}$ , equal to one if state is above the upper bound

$\Theta_s$  production time in slot  $s$

$\Theta_p$  production time of product  $p$

$\Theta_{p,s}$  production time of product  $p$  in slot  $s$

$\theta'_s$  transition time in slot  $s$

$t_s^s$  starting time of slot  $s$

$t_s^e$  ending time of slot  $s$

$T_c$  total production cycle time

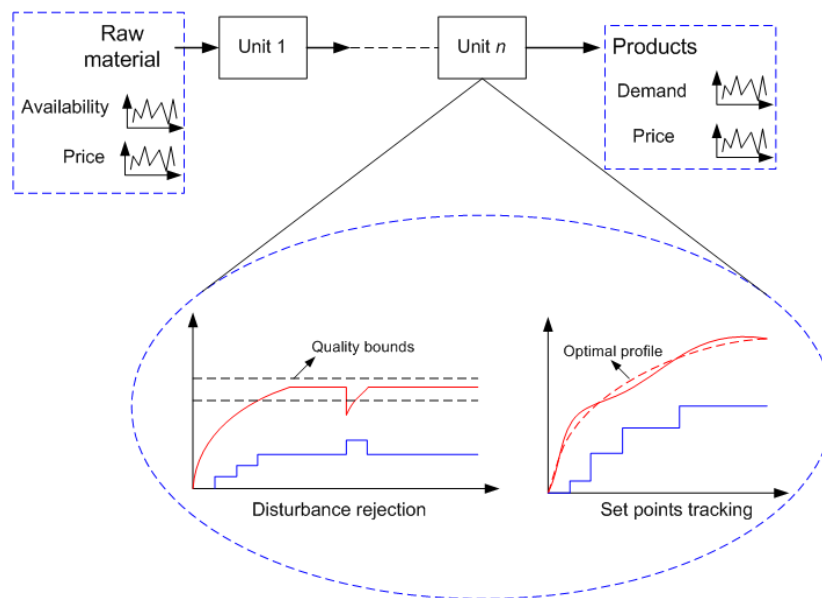
$W_p$  amount produced for product  $p$

$VR$  valid region where the LTI of PWA is valid

## Chapter 6 Dealing with uncertainties in the integration of scheduling and control

### 6.1 Uncertainty in process operations, origin and models

Uncertainty in process operations can originate from both internal and external of the process (Figure 6.1). In general, it can be classified as follows due to the source of its origin and the nature of uncertainty (Pistikopoulos, 1995). (1) model-inherent uncertainty (model mismatch), such as mass/heat transfer coefficients and kinetic constants; (2) internal disturbance, such as the disturbance on feeding flow rate or feeding flow composition, and heating/cooling flow rate; (3) external disturbance, mainly referred as market uncertainty such as demand and price fluctuation; (4) discrete uncertainty, such as equipment availability, machine breakdown and operational personnel absence.



**Figure 6.1 Origin of uncertainty in process operations**

The model of uncertainty can be categorized into two types, based on the availability of the information about the uncertainty. One type is bounded form that includes the lower and upper bounds of the uncertainty. This form applies to the cases where the probability distribution of the uncertainty is not available but the bound information can be obtained according to physical

limits or historical observations. The other type of model of uncertainty is probability distribution function. This type of model can be used if the random behavior of the uncertainty is fully known.

## **6.2 Approaches of dealing with uncertainty**

There are two types of approaches in dealing with uncertainty: preventive approach and reactive approach. With preventive approach the decisions at scheduling and control levels are generated prior to the occurrence of disturbance. Specifically, the model of uncertainty is incorporated into scheduling and control problems when they are solved. Typical techniques used in preventive approach are stochastic programming (when probability distribution of uncertainty is available) and robust optimization (only bound form of uncertainty is available) (Li and Ierapetritou, 2008b).

Using reactive approach the solutions at scheduling and control levels are obtained based on nominal models that have no consideration of uncertainty. The solutions are then implemented and updated in response to the occurrence of uncertainty. In fact, this corresponds to a closed loop implementation where the latest state information is fed-back to the scheduler or controller and the new solutions are generated accordingly. However, the reactive approach involves expensive computation, since the uncertainty can occur frequently and the original optimization problems have to be solved repeatedly. In order to improve the efficiency of reactive approach, parametric programming is proposed to parametrize the uncertainty and generate the explicit solutions offline. (Li and Ierapetritou, 2008a) (Pistikopoulos, 2009) Following this approach, reactive scheduling and parametric MPC are obtained as explicit functions of the states. When the explicit solutions are implemented in real applications, the online computing only involves function evaluation which needs much less computing time.

### 6.3 Robust MPC for control problems

In this study, we focus on the uncertainty at control level and employ robust MPC to handle uncertainty. Problem (6.1) presents a general form of robust programming where a worst case of the minimization problem is considered.

$$\begin{aligned} \min_{u, \delta} \quad & f(u, \delta) \\ \text{s.t.} \quad & g(u, \delta) \leq 0 \end{aligned} \tag{6.1}$$

where  $u$  are decision variables and  $\delta$  stands for the uncertainty. Problem (6.1) is equivalent to (6.2) that introduces an auxiliary variable  $\psi$  and ensures that  $\psi$  is a cap of the objective function in any case of uncertainty.

$$\begin{aligned} \min_{u, \delta} \quad & \psi \\ \text{s.t.} \quad & \begin{cases} \psi \geq f(u, \delta), \quad \forall \delta \in \Omega_\delta \\ g(u, \delta) \leq 0 \end{cases} \end{aligned} \tag{6.2}$$

#### 6.3.1 Robust MPC formulation for PWA systems

We first obtain PWA approximation to the nonlinear systems based on the optimization method presented in section 5.2.2, than apply robust MPC to the PWA system. Problem (6.3) describes a conventional MPC formulation while (6.4) presents a robust MPC formulation. Both of them minimize the quadratic error between the predictive states and the reference states. The last term of the objective function ensures the stability of the process i.e. state at the end of horizon  $x_{k+N}$  should be closed to the reference.  $x^0$  is the initial state in the prediction horizon. States and manipulated variables are confined by upper and lower bounds. In the formulations we assume additive uncertainty in the process dynamic.

$$\begin{aligned}
\min_{u_k} \quad & \sum_{k_p=1}^{N-1} \left( \left\| x_{k+k_p} - \bar{x}_{k+k_p} \right\|_2^{Q_{k_p}} + \left\| u_{k+k_p-1} - \bar{u}_{k+k_p-1} \right\|_2^{R_{k_p}} \right) \\
& + \left\| x_{k+N} - \bar{x}_{k+N} \right\|_2^{Q_N} \\
s.t. \quad & \begin{cases} x_k = x^0 \\ x_{k+k_p} = A_i x_{k+k_p-1} + B_i u_{k+k_p-1} + C_i \delta_{k+k_p-1} + D_i, \quad k_p = 1, \dots, N \\ x_{\min} \leq x_{k+k_p} \leq x_{\max}, \quad k_p = 1, \dots, N \\ u_{\min} \leq u_{k+k_p} \leq u_{\max}, \quad k_p = 1, \dots, N \end{cases}
\end{aligned} \tag{6.3}$$

$$\begin{aligned}
\min_{u_k} \max_{\delta} \quad & \sum_{k_p=1}^{N-1} \left( \left\| x_{k+k_p} - \bar{x}_{k+k_p} \right\|_2^{Q_{k_p}} + \left\| u_{k+k_p-1} - \bar{u}_{k+k_p-1} \right\|_2^{R_{k_p}} \right) \\
& + \left\| x_{k+N} - \bar{x}_{k+N} \right\|_2^{Q_N} \\
s.t. \quad & \begin{cases} x_k = x^0 \\ x_{k+k_p} = A_i x_{k+k_p-1} + B_i u_{k+k_p-1} + C_i \delta_{k+k_p-1} + D_i, \quad k_p = 1, \dots, N \\ x_{\min} \leq x_{k+k_p} \leq x_{\max}, \quad k_p = 1, \dots, N \\ u_{\min} \leq u_{k+k_p} \leq u_{\max}, \quad k_p = 1, \dots, N \end{cases}
\end{aligned} \tag{6.4}$$

According to the following Theorem (Bertsekas, 1999), min-max problem in (6.4) can be transformed into a min problem in (6.5).

Let  $\mathbb{C}$  be a closed convex set and let  $f : \mathbb{C} \rightarrow \mathbf{R}$  be a convex function. Then if  $f$  attains a maximum over  $\mathbb{C}$ , it attains a maximum at some extreme point of  $\mathbb{C}$ .

Note that the inner max problem (6.4) is a QP where the objective function is convex and the feasible region is a convex set. Therefore  $\delta$  should attain its extreme value provided in Table 6.1 and problem (6.4) can be transformed into (6.5), as the way of transforming (6.1) into (6.2).

**Table 6.1 Possible combinations of future uncertainty, and the associated possible states**

| Possible states,<br>$m$ scenarios | Future uncertainty<br>$k_p = 1$ | Future uncertainty<br>$k_p = 2$ | Future uncertainty<br>$k_p = 3$ | Future uncertainty<br>$k_p = 4$ | Future uncertainty<br>$k_p = 5$ |
|-----------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| $x_{k+k_p}^{(1)}$                 | $\bar{\delta}$                  | $\bar{\delta}$                  | $\bar{\delta}$                  | $\bar{\delta}$                  | $\bar{\delta}$                  |
| $x_{k+k_p}^{(2)}$                 | $\underline{\delta}$            | $\underline{\delta}$            | $\underline{\delta}$            | $\underline{\delta}$            | $\underline{\delta}$            |

|                   |                      |                      |                      |                      |                      |
|-------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| ...               | ...                  | ...                  | ...                  | ...                  | ...                  |
| $x_{k+k_p}^{(m)}$ | $\underline{\delta}$ | $\underline{\delta}$ | $\underline{\delta}$ | $\underline{\delta}$ | $\underline{\delta}$ |

Note that in Table 6.1  $m = 2^N$ , where N is the prediction horizon.

$$\begin{aligned}
 & \min_{u_k} \varphi \\
 & \left\{ \begin{aligned}
 & x_k^{(1)} = x^0 \\
 & \varphi \geq \sum_{k_p=1}^{N-1} \left( \left\| x_{k+k_p}^{(1)} - \bar{x}_{k+k_p} \right\|_2^{Q_{k_p}} + \left\| u_{k+k_p-1} - \bar{u}_{k+k_p-1} \right\|_2^{R_{k_p}} \right) + \left\| x_{k+N}^{(1)} - \bar{x}_{k+N} \right\|_2^{Q_N} \\
 & x_{k+1}^{(1)} = A_i x_k^{(1)} + B_i u_k + C_i \bar{\delta} + D_i \\
 & x_{k+2}^{(1)} = A_i x_{k+1}^{(1)} + B_i u_{k+1} + C_i \bar{\delta} + D_i \\
 & x_{k+3}^{(1)} = A_i x_{k+2}^{(1)} + B_i u_{k+2} + C_i \bar{\delta} + D_i \\
 & \dots \\
 & x_{k+5}^{(1)} = A_i x_{k+4}^{(1)} + B_i u_{k+4} + C_i \bar{\delta} + D_i \\
 & x_{\min} \leq x_{k+k_p}^{(1)} \leq x_{\max}, \quad k_p = 1, \dots, 5 \\
 & u_{\min} \leq u_{k+k_p} \leq u_{\max}, \quad k_p = 1, \dots, 5 \\
 & \dots \\
 & x_k^{(2)} = x^0 \\
 & \varphi \geq \sum_{k_p=1}^{N-1} \left( \left\| x_{k+k_p}^{(2)} - \bar{x}_{k+k_p} \right\|_2^{Q_{k_p}} + \left\| u_{k+k_p-1} - \bar{u}_{k+k_p-1} \right\|_2^{R_{k_p}} \right) + \left\| x_{k+N}^{(2)} - \bar{x}_{k+N} \right\|_2^{Q_N} \\
 & x_{k+1}^{(2)} = A_i x_k^{(2)} + B_i u_k + C_i \bar{\delta} + D_i \\
 & x_{k+2}^{(2)} = A_i x_{k+1}^{(2)} + B_i u_{k+1} + C_i \bar{\delta} + D_i \\
 & x_{k+3}^{(2)} = A_i x_{k+2}^{(2)} + B_i u_{k+2} + C_i \bar{\delta} + D_i \\
 & \dots \\
 & x_{k+5}^{(2)} = A_i x_{k+4}^{(2)} + B_i u_{k+4} + C_i \bar{\delta} + D_i \\
 & x_{\min} \leq x_{k+k_p}^{(2)} \leq x_{\max}, \quad k_p = 1, \dots, 5 \\
 & u_{\min} \leq u_{k+k_p} \leq u_{\max}, \quad k_p = 1, \dots, 5 \\
 & \vdots
 \end{aligned} \right. \quad (6.5)
 \end{aligned}$$

In the following we implement robust MPC and conventional MPC on a LTI system (6.6) and (6.7) and compare their performance in tracking the reference.

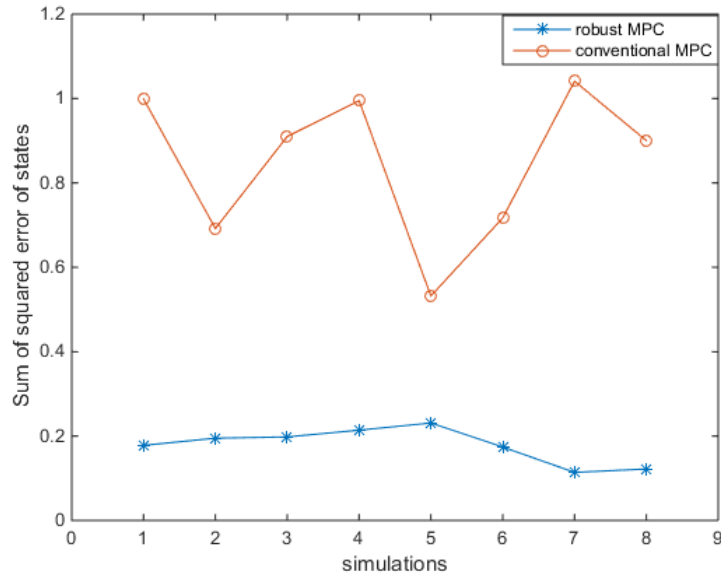
$$x(k+1) = \begin{bmatrix} 1.1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 1 & 3 \\ 0.5 & -2 \end{bmatrix} u(k) + \delta \quad (6.6)$$



$$\begin{aligned}
x_{desired} &= [2; -1] \\
x_{initial} &= [2; -1] \\
x_{min} &= [-5; -5] \\
x_{max} &= [5; 5] \\
u_{min} &= [-3; -3] \\
u_{max} &= [3; 3] \\
\delta &\sim Uniform(0, 0.25) \\
N &= 3
\end{aligned} \tag{6.7}$$

We simulated the process using robust MPC (6.5) and conventional MPC (6.3) for 8 times. In each simulation, there are 20 sample points. The criterion for the performance of robust MPC and conventional MPC is the sum of the squared error over the whole simulation span. Figure 6.2 shows that the *SumSquareError* of conventional MPC is much higher for all simulations.

$$SumSquareError = \sum_{k=1}^{20} (x_k - x_{desired})^2 \tag{6.8}$$



**Figure 6.2 Performance comparison between robust MPC and conventional MPC**

### 6.3.2 A general formulation of robust MPC for nonlinear systems

In this section we develop solution procedures of robust MPC for nonlinear systems. Following is a general form of robust MPC for nonlinear system.

$$\begin{aligned} \min_{u(t)} \max_{\delta} \int_{t=0}^{t=t_H} (x^T Q x + u^T R u) dt + x_N^T P x_N \\ \text{s.t.} \begin{cases} x(0) = x^0 \\ \dot{x} = f(x, u, \delta) \\ x_{\min} \leq x(t) \leq x_{\max}, \quad 0 \leq t \leq t_H \\ u_{\min} \leq u(t) \leq u_{\max}, \quad 0 \leq t \leq t_H \end{cases} \end{aligned} \quad (6.9)$$

where  $\delta$  is a random variable confined by lower and upper bounds  $\underline{\delta} \leq \delta \leq \bar{\delta}$ .

#### Procedures of applying robust MPC to nonlinear systems

Step 1: Treat the uncertainty  $\delta$  as a variable and transform nonlinear dynamics  $\dot{x} = f(x, u, \delta)$  into  $PWA = \bigcup_i LTI^{(i)}$

$$\begin{aligned} LTI^{(i)}: x(k+1) &= A_i x(k) + B_i u(k) + C_i \delta(k) + D_i \\ \text{if } x(k) &\in \Omega_i = \{x : V_i x \leq W_i\}, i \in S_i = \{1, 2, \dots, N_i\} \\ \text{where } \bigcup_i \Omega_i &= \Omega_x \text{ and } \Omega_{i_1} \cap \Omega_{i_2} = \emptyset \text{ if } i_1 \neq i_2 \end{aligned} \quad (6.10)$$

Now we obtain the robust MPC formulation for PWA systems (6.4).

Step 2: Set the initial states  $x^0$

Step 3: Locate the corresponding LTI for current states

$$\begin{aligned} \text{if } x \in \Omega_i = \{x : V_i x \leq W_i\}, i \in S_i = \{1, 2, \dots, N_i\}, \\ \text{then select } LTI^{(i)} : x(k+1) &= A_i x(k) + B_i u(k) + C_i \delta(k) + D_i \end{aligned} \quad (6.11)$$

Step 4: Solve problem (6.5) and obtain the current control solution.

Step 5: Do state transfer using the following equation:

$$x(k+1) = A_i x(k) + B_i u^*(k) + C_i \delta(k) + D_i \quad (6.12)$$

Step 6:  $k=k+1$ , go to step 3.

## Chapter 7 Conclusions and future perspectives

The proposed closed loop implementation works in a similar manner as the Model Predictive Control. In the proposed approach the integrated model of scheduling and control is regarded as the predictive model, and the remaining part of the production cycle is regarded as the prediction horizon. At each sample step, state deviation is determined and new solutions are generated in the case of large deviation exceeding the quality bounds by solving the integrated optimization problem based on the predictive model. Unlike traditional MPC, the solutions here are not only manipulated variables in transition periods but also scheduling solutions in production periods.

Note that in the integration of scheduling and dynamic optimization the dynamic optimization problem is formulated at the control level of process operations. In this study the integrated problem is solved offline and implemented in an open-loop manner. What we present in this paper is an solution approach that reduces the computation complexity of the integrated problem. It should be emphasized that the decomposition approach proposed in this study is not simply decomposing the integrated problem into the scheduling and dynamic optimization problems, though these two problems are integrated as one. In other words, solving the integrated problem in a decomposition scheme does not change the nature of integration. The problem that is addressed is indeed, the integrated scheduling and dynamic optimization problem and the decomposition is proposed as a way to improve the computational efficiency of the solution approach. The fact that the transition periods are proved independent of the production in continuous cyclic production decreases the computation complexity of the integrated scheduling and dynamic optimization problem. This facilitates the integration with planning level, since the only linking variables between planning and integrated scheduling and dynamic optimization are the productions. Based on this, our future efforts are towards the modeling and optimization of simultaneous planning, scheduling and control.

To reduce the computation burden of the integrated problem and facilitate the online applications, we propose a framework which is capable to transform explicit control solution generated by mp-MPC into explicit linear constraints, and incorporate with the linear constraints at scheduling level. This results in an integrated problem that involves linear constraints and a nonlinear objective. Currently we use PWA system in solve mp-MPC using MPT toolbox. We believe that the progress in developing mp-MPC for nonlinear dynamic would benefit the direct integration with a nonlinear system.

This study demonstrates that mp-MPC builds linear constraints for control level and thus effectively reduces the complexity of the integrated problem involving scheduling and control levels. Since the integration of planning and scheduling levels results in a MILP (Li and Ierapetritou, 2009) (Li and Ierapetritou, 2010) we believe an integration of planning, scheduling and control would also lead to linear constraints, if mp-MPC is incorporated. Based on this, our future efforts are towards enterprise wide optimization.

To facilitate the online application, we use fast MPC to compute the optimal control actions and eliminate the effects of disturbances in a timely manner. Previous investigation of mp-MPC in solving the control problem has shown that the number of critical regions in the explicit solution of mp-MPC increases exponentially with respect to problem size in terms of number of dimensions of state and manipulated variables and the length of prediction horizon. To overcome the dimensionality problem, we propose to use fast MPC in this study. We develop the algorithms of implementing fast MPC and mp-MPC and compare their computing performance. Results show that fast MPC computes the control problem much faster than mp-MPC does, though both of them perform in the speed of msec. To the authors' knowledge, this is the first attempt in this area to explore the possibility and feasibility of applying fast-MPC in simultaneous scheduling and control problem. The future efforts are towards the application of fast MPC in large scale systems.

## Appendix A (Supporting Material for the decomposition proof in section 3.2)

Lemma 1: The perturbation on production time does not affect the transition for any specific time slot.

*Continued proof:*

Rearranging Inequality (9), we obtain:

$$P\Theta\theta^t - \Theta f_{tc}(\theta^{t*}) - \theta^t f_{tc}(\theta^{t*}) + \Theta f_{tc}(\theta^t) - P\Theta\theta^{t*} + \theta^{t*} f_{tc}(\theta^t) \geq 0 \quad (\text{A.1})$$

Rearranging Inequality (10) we obtain:

$$\begin{aligned} & [P\Theta\theta^t - \Theta f_{tc}(\theta^{t*}) - \theta^t f_{tc}(\theta^{t*}) + \Theta f_{tc}(\theta^t) - P\Theta\theta^{t*} + \theta^{t*} f_{tc}(\theta^t)] \\ & + [P\delta\Theta\theta^t - \delta\Theta f_{tc}(\theta^{t*}) - P\delta\Theta\theta^{t*} + \delta\Theta f_{tc}(\theta^t)] \geq 0 \end{aligned} \quad (\text{A.2})$$

Comparing (A.1) and (A.2), we can establish a sufficient condition so that (A.2) holds.

$$P\theta^t - f_{tc}(\theta^{t*}) - P\theta^{t*} + f_{tc}(\theta^t) \geq 0 \quad (\text{A.3})$$

which can be rearranged as follows

$$\frac{f_{tc}(\theta^t) - f_{tc}(\theta^{t*})}{\theta^t - \theta^{t*}} \geq -P \quad (\text{A.4})$$

Assume  $f_{tc}(\bullet)$  is continuous on interval  $[0, \theta'_{\max}]$  and differentiable on interval  $(0, \theta'_{\max})$ , apply

Lagrange's Mean Value Theorem, there exists a point  $\theta^{t'}$  in  $(\theta^t, \theta^{t*})$  if  $\theta^t < \theta^{t*}$  or  $(\theta^{t*}, \theta^t)$  if  $\theta^{t*} < \theta^t$  such that:

$$f'_{tc}(\theta^{t'}) = \frac{f_{tc}(\theta^t) - f_{tc}(\theta^{t*})}{\theta^t - \theta^{t*}} \quad (\text{A.5})$$

Based on (A.5) we obtain a sufficient condition to establish (A.4).

$$\min_{\theta^t \in (0, \theta_{\max}^t)} f_{tc}'(\theta^t) \geq -P \quad (\text{A.6})$$

Since  $f_{tc}(\theta^t) = C^r \int_0^{\theta^t} u \, dt$  (This is a simplified version of Equation (13) addressing only one transition), we have

$$\frac{df_{tc}(\theta^t)}{d\theta^t} = C^r u(\theta^t) > 0 \quad (\text{A.7})$$

Apparently it verifies (A.6), and as a result, (A.2)-(A.5) all hold. Thus the proposal in (10) is proved. Q.E.D.

**Lemma 2:** a property of  $\theta_k^{t*}$ :  $\theta_k^{t*}$  is obtained as the minimum value of feasible  $\theta_k^t$

*Continued proof:*

$$\left. \frac{\partial J}{\partial \theta_k^t} \right|_{\Theta_i, \theta_{-k}^t, y_{ik}} = \frac{-\left. \frac{\partial f_{tc}(y_{ik}, \theta_k^t)}{\partial \theta_k^t} \right|_{y_{ik}, \theta_{-k}^t} \left( \sum_i \Theta_i + \sum_k \theta_k^t \right) - \left( \sum_i P_i \Theta_i - f_{tc}(y_{ik}, \theta_k^t) \right)}{\left( \sum_i \Theta_i + \sum_k \theta_k^t \right)^2} \quad (\text{A.8})$$

As has been proved before

$$\left. \frac{\partial f_{tc}(y_{ik}, \theta_k^t)}{\partial \theta_k^t} \right|_{y_{ik}, \theta_{-k}^t} = C^r u(\theta_k^t) \geq 0 \quad (\text{A.9})$$

Note that  $u$  is a function of *time*, and  $u(t)$  is valid when  $t \in [0, \theta_k^t]$ .  $u$  does not depend on  $\theta_k^t$ , instead,  $u$  can be any value in the range of  $[u_{\min}, u_{\max}]$ .  $u(\theta_k^t)$  represents the value of  $u$  at  $t = \theta_k^t$ . It does not correspond to a function of  $\theta_k^t$ .

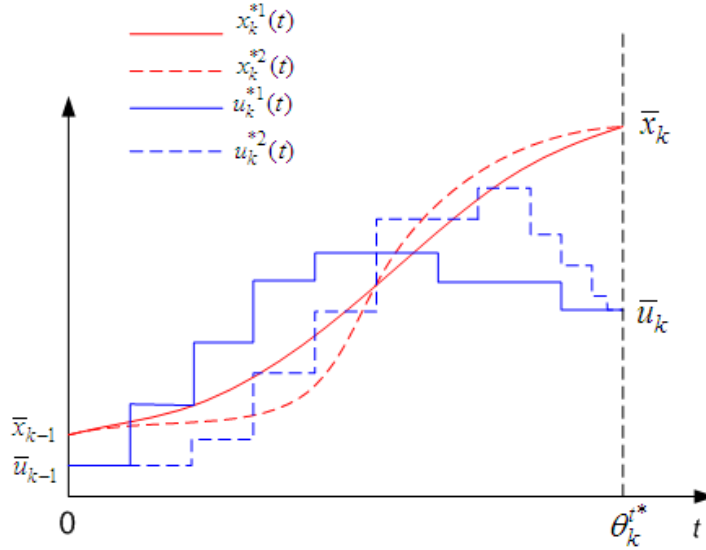
It is assumed that  $\sum_i P_i \Theta_i - f_{tc}(y_{ik}, \theta_k^t) \geq 0$  i.e. the process is profitable.

Thus the numerator of (A.8) is negative, and we have

$$\left. \frac{\partial J}{\partial \theta_k^t} \right|_{\Theta_i, \theta_{-k}^t, y_{ik}} \leq 0 \quad (\text{A.10})$$

which implies that  $\theta_k^{t*}$  should be the minimum value of the feasible region of  $\theta_k^t$  i.e.  $\theta_k^{t*} \leq \theta_k^t$ .

Figure A.1 illustrates a transition from slot  $k-1$  to slot  $k$ , where two dynamic profiles  $(x_k^{*1}(t), u_k^{*2}(t))$ ,  $t \in [0, \theta_k^{t*}]$  and  $(x_k^{*2}(t), u_k^{*2}(t))$ ,  $t \in [0, \theta_k^{t*}]$  correspond to one optimal transition duration  $\theta_k^{t*}$ . If a one-to-one mapping between optimal transition time and optimal transition profile can be proved (i.e.  $x_k^{*1}(t) = x_k^{*2}(t)$  and  $u_k^{*1}(t) = u_k^{*2}(t)$ ,  $\forall t \in [0, \theta_k^{t*}]$ ), we only need to investigate the optimal transition time rather than the whole transition profile, which effectively simplify the proof. In the following through (A.11)-(A.27) we prove the one-to-one mapping between optimal transition time and optimal transition profile i.e. the case shown in Figure A.1 does not exist.



**Figure A.1: Two dynamic profiles correspond to one optimal transition duration**

We need to prove that

$$\theta_k^{t*} \Leftrightarrow (u_k^*(t), x_k^*(t)), t \in [0, \theta_k^{t*}] \quad (\text{A.11})$$

A general formula for the optimal control problem in formula (11) is

$$\begin{aligned} \max_{u(t), t \in [0, \tau]} J &= L(x(\tau), u(\tau)) + \int_0^\tau \Phi(x(t), u(t)) dt \\ \text{s.t.} \quad &\begin{cases} \dot{x}(t) = f(x(t), u(t)), t \in [0, \tau] \\ x(0) = x^0 \\ x(\tau) = x^1 \\ u(t) \in \Omega_u, t \in [0, \tau] \\ x(t) \in \Omega_x, t \in [0, \tau] \end{cases} \end{aligned} \quad (\text{A.12})$$

Define Hamiltonian  $H$  as

$$H[x(t), u(t), \lambda(t), t] = \Phi(x(t), u(t), t) + \lambda^T(t) f(x(t), u(t), t) \quad (\text{A.13})$$

Assume  $u^*(t), t \in [0, \tau^*]$  is the optimal solution for (A.12) and  $x^*(t)$  is the corresponding trajectory. According to Pontryagin Maximum Principle (Pontryagin et al., 1962), the necessary conditions for  $u^*(t)$  to be the optimum are (A.14)-(A.16):

$$\dot{x}^*(t) = \frac{\partial}{\partial \lambda} H(x^*(t), u^*(t), \lambda^*(t)) \quad (\text{A.14})$$

$$\dot{\lambda}^*(t) = -\frac{\partial}{\partial x} H(x^*(t), u^*(t), \lambda^*(t)) \quad (\text{A.15})$$

$$\begin{aligned} H[x^*(t), u^*(t), \lambda^*(t)] &\geq H[x^*(t), u(t), \lambda^*(t)], \\ \forall u(t) \in \Omega_u, 0 \leq t \leq \tau^* \end{aligned} \quad (\text{A.16})$$

Applying Equations (A.13)-(A.16) to problem (11) we obtain Hamiltonian

$$H[x(t), u(t), \lambda(t), t] = 1 + \lambda(t) f(x(t), u(t), t) \quad (\text{A.17})$$

and the necessary conditions (A.18)-(A.20).



$$\dot{\lambda}^*(t) = -\lambda^*(t) \frac{\partial f}{\partial x} \quad (\text{A.18})$$

$$\frac{\partial H}{\partial u} = \lambda^*(t) \frac{\partial f}{\partial u} = 0 \quad (\text{A.19})$$

$$1 + \lambda^*(t) f(x^*(t), u^*(t), t) \geq 1 + \lambda^*(t) f(x^*(t), u(t), t), \quad \forall u(t) \in \Omega_u, 0 \leq t \leq \theta_k^* \quad (\text{A.20})$$

Here we assume  $\lambda^*(t) \neq 0$ , otherwise Equations (A.18)-(A.20) are always satisfied and  $u^*(t)$  is free in  $\Omega_u$ , which is abnormal. If  $u^*$  could attain any value in the feasible region, this becomes a trivial solution of Equations (A.18)-(A.20), which is not complete nor correct in the real case *i.e.*  $u$  must be governed and be able to drive the state to the following production stage. This point can also be verified by checking the Hamilton–Jacobi–Bellman equation in the following. In addition, we assume that  $\partial f / \partial u \neq 0$  *i.e.* the dynamic is affected by control input. Therefore condition Equation (A.19) is not satisfied, which indicates that there does not exist a stationary trajectory for *u*. *i.e.*  $u^*(t)$  could locate at the boundary or non-differentiable point. Since  $u_{\min} \leq u(t) \leq u_{\max}$ ,  $0 \leq t \leq \theta_k^t$  we infer that the optimal  $u$  must be obtained at the boundary *i.e.*  $u(t) = u_{\min}$ ,  $0 \leq t \leq \theta_k^t$  or  $u(t) = u_{\max}$ ,  $0 \leq t \leq \theta_k^t$ .

Here we prove the one-to-one mapping (A.11) through checking the Hamilton–Jacobi–Bellman equations (Kirk, 1970) (A.21) and (A.22).

$$\frac{\partial V(x, t)}{\partial t} + \min_u \left\{ \Phi(x, u) + \frac{\partial V(x, t)}{\partial x} f(x, u) \right\} = 0 \quad (\text{A.21})$$

where

$$\begin{aligned}
V(x, t) = & \min_{u(t'), t' \in [t, \tau]} L(x(\tau), \tau) + \int_t^\tau \Phi(x(t'), u(t'), t') dt' \\
s.t. \left\{ \begin{aligned} & \dot{x}(t') = f(x(t'), u(t')), t' \in [t, \tau] \\ & x(t) = x \\ & x(\tau) = x^1 \\ & u(t') \in \Omega_u, t' \in [t, \tau] \\ & x(t') \in \Omega_x, t' \in [t, \tau] \end{aligned} \right. \quad (A.22)
\end{aligned}$$

Note that in formula (A.22) and (A.24) we use  $t'$  in order to differentiate it from  $t$ . Both  $t'$  and  $t$  represent time.

In the case of problem (11) we have HJB equations (A.23) and (A.24).

$$\frac{\partial V(x, t)}{\partial t} + \min_u \left\{ 1 + \frac{\partial V(x, t)}{\partial x} f(x, u) \right\} = 0, \quad \forall t \in [0, \theta_k^t] \quad (A.23)$$

$$\begin{aligned}
V(x, t) = & \min_{u(t'), t' \in [t, \theta_k^t]} \int_t^{\theta_k^t} dt' \\
s.t. \left\{ \begin{aligned} & \dot{x}(t') = f(x(t'), u(t')), t' \in [t, \theta_k^t] \\ & x(t) = x \\ & x(\theta_k^t) = \bar{x} \\ & u(t') \in \Omega_u, t' \in [t, \theta_k^t] \\ & x(t') \in \Omega_x, t' \in [t, \theta_k^t] \end{aligned} \right. \quad (A.24)
\end{aligned}$$

Propose that there are two optimal control profiles  $u^{*1}(t')$  and  $u^{*2}(t')$  satisfying

$$u^{*1}(t') \neq u^{*2}(t'), \quad \exists t' \in [t, \theta_k^t] \quad (A.25)$$

If there exist  $t' \in [t, \theta_k^t]$  such that  $u^{*1}(t') \neq u^{*2}(t')$ , we say that the profiles of  $u^{*1}(t')$  and  $u^{*2}(t')$  are different over region  $[t, \theta_k^t]$ .

Plugging  $u^{*1}(t')$  and  $u^{*2}(t')$  into (A.23) yields

$$\frac{\partial V(x, t)}{\partial t} + \left\{ 1 + \frac{\partial V(x, t)}{\partial x} f(x, u^{*1}) \right\} = 0, \quad \forall t \in [0, \theta_k^t] \quad (A.26)$$

$$\frac{\partial V(x, t)}{\partial t} + \left\{ 1 + \frac{\partial V(x, t)}{\partial x} f(x, u^{*2}) \right\} = 0, \quad \forall t \in [0, \theta_k^t] \quad (\text{A.27})$$

Note that both  $x$  and  $u$  are trajectories with respect to time.  $f$  is a compound function and therefore  $f$  is also a trajectory with respect to time. If there exists  $t \in [0, \theta_k^t]$  such that  $u^{*1}(t) \neq u^{*2}(t)$ , the resulting  $f$  trajectories would be different, given the same state trajectories. In other words, there exists  $t \in [0, \theta_k^t]$  such that  $f(x(t), u^{*1}(t)) \neq f(x(t), u^{*2}(t))$ . Note that when we compare two functions  $f(x(t), u^{*1}(t))$  and  $f(x(t), u^{*2}(t))$  we are comparing their profiles over the whole region of  $t$ . Even though their profiles could overlap at some points, the profiles are considered different over the whole region. Therefore, the only way to make both Equation (A.26) and (A.27) hold is that  $\partial V(x, t)/\partial x = 0$  i.e. the state at  $t$  does not affect  $V(x, t)$ . However, this is not true. Suppose  $x(t) = \bar{x}$ , then obviously  $V(x, t) = 0$  given by (A.24). But if  $x(t) \neq \bar{x}$ , we have  $V(x, t) > 0$  according to (A.24). This fact contradicts  $\partial V(x, t)/\partial x = 0$ . Thus the proposition in inequality (A.25) is not established and there is unique optimal control solution  $u^*$ . Therefore  $\theta_k^{t*} \Leftrightarrow (u_k^*, x_k^*)$  is proved.

Through (A.21)-(A.27) we introduce a contradiction to prove that (A.25) does not hold. In other words, the optimal control profile is unique. Based on the above derivation, we infer that optimal control input for problem (11) is unique and the corresponding optimal state path is unique as well. In other words, there is a one-to-one mapping between optimal transition time and optimal dynamic profile. Thus it's reasonable to investigate the optimal transition time rather than dynamic profile, for the sake of simplification. Q.E.D.

**Lemma 3:** The optimal production sequence is not affected by production perturbation

*Continued proof:*

Given that

$$J^* = \frac{\sum_i P_i \Theta_i - f_{tc}(\theta_{ip}^{t*}, z_{ipk}^*)}{\sum_i \Theta_i + \sum_i \sum_p \sum_k \theta_{ip}^{t*} z_{ipk}^*} \quad (\text{A.28})$$

Substituting (A.28) into  $J^*(z^*) \geq J(z)$  yields

$$J^*|_{\Theta_i} = \frac{\sum_i P_i \Theta_i - f_{tc}(\theta_{ip}^{t*}, z_{ipk}^*)}{\sum_i \Theta_i + \sum_i \sum_p \sum_k \theta_{ip}^{t*} z_{ipk}^*} \geq J|_{\Theta_i} = \frac{\sum_i P_i \Theta_i - f_{tc}(\theta_{ip}^{t*}, z_{ipk})}{\sum_i \Theta_i + \sum_i \sum_p \sum_k \theta_{ip}^{t*} z_{ipk}} \quad (\text{A.29})$$

We need to prove that

$$J^*(\theta_{ip}^{t*}, z_{ipk}^*)|_{\Theta_i + \delta_{\Theta_i}} \geq J(\theta_{ip}^{t*}, z_{ipk})|_{\Theta_i + \delta_{\Theta_i}} \quad (\text{A.30})$$

Substituting (A.28) into (A.30) yields

$$\frac{\sum_i P_i \Theta_i + P_i \delta_{\Theta_i} - f_{tc}(\theta_{ip}^{t*}, z_{ipk}^*)}{\sum_i \Theta_i + \delta_{\Theta_i} + \sum_i \sum_p \sum_k \theta_{ip}^{t*} z_{ipk}^*} \geq \frac{\sum_i P_i \Theta_i + P_i \delta_{\Theta_i} - f_{tc}(\theta_{ip}^{t*}, z_{ipk})}{\sum_i \Theta_i + \delta_{\Theta_i} + \sum_i \sum_p \sum_k \theta_{ip}^{t*} z_{ipk}} \quad (\text{A.31})$$

After fraction manipulations, we obtain a sufficient condition for (A.31).

$$P_i \sum_i \sum_p \sum_k \theta_{ip}^{t*} z_{ipk}^* - P_i \sum_i \sum_p \sum_k \theta_{ip}^{t*} z_{ipk} + f_{tc}(\theta_{ip}^{t*}, z_{ipk}) - f_{tc}(\theta_{ip}^{t*}, z_{ipk}^*) \geq 0 \quad (\text{A.32})$$

Prove (A.32):

$$\text{Let } T_c = \sum_i \Theta_i + \sum_i \sum_p \sum_k \theta_{ip}^{t*} z_{ipk}^*, \quad T_c^* = \sum_i \Theta_i + \sum_i \sum_p \sum_k \theta_{ip}^{t*} z_{ipk}, \quad f_{tc} = f_{tc}(\theta_{ip}^{t*}, z_{ipk}),$$

$$f_{tc}^* = f_{tc}(\theta_{ip}^{t*}, z_{ipk}^*)$$

Rearranging (A.29), we obtain

$$f_{tc} \geq \frac{f_{tc}^* T_c - \sum_i P_i \Theta_i (T_c - T_c^*)}{T_c^*} \quad \forall z \in \Omega_z \quad (\text{A.33})$$

Rearranging (A.32), we obtain

$$f_{tc} \geq f_{tc}^* - P_i (T_c - T_c^*) \quad \forall z \in \Omega_z, \quad (\text{A.34})$$

Now the problem is transformed into: given (A.33), prove (A.34).

Obviously a sufficient condition to establish (A.34) is

$$\frac{f_{tc}^* T_c - \sum_i P_i \Theta_i (T_c - T_c^*)}{T_c^*} \geq f_{tc}^* - P_i (T_c - T_c^*) \quad \forall z \in \Omega_z \quad (\text{A.35})$$

Rearranging (A.35), we obtain

$$(f_{tc}^* + P_i T_c^* - \sum_i P_i \Theta_i) (T_c - T_c^*) \geq 0 \quad \forall z \in \Omega_z \quad (\text{A.36})$$

Thus, (A.36) is a sufficient condition to verify the hypothesis (A.30) and (A.31), i.e. the optimality is preserved in the presence of production variation.

(A.36) consists of two terms. The first term (the one in the first parentheses) is calculated as the revenue gained by dedicating the whole production cycle to one product plus all the transition cost minus the nominal revenue. This term will be greater than zero in most common cases but there can be some extreme cases (price  $P_i$  is extremely low) where this term can be negative. The second term is positive since we proved that the optimal transition time is obtained as the minimum value given any production sequence. Therefore (A.36) is satisfied in most cases.

In our case studies,  $f_{tc}^* + P_i T_c^* - \sum_i P_i \Theta_i \geq 0$ ,  $T_c^* \leq T \quad \forall z \in \Omega_z$ , therefore (A.36) is satisfied. Q.E.D.

## Bibliography

Allgor, R. J. and Barton, P. I. (1999). Mixed-integer dynamic optimization I: problem formulation. *Computers & Chemical Engineering*, 23, 567-584.

Azuma, S., Imura, J., and Sugie, T. (2006). Lebesgue Piecewise Affine Approximation of Nonlinear Systems and Its Application to Hybrid System Modeling of Biosystems. *Proceedings of 45th IEEE Conference on Decision and Control*, 2128-2133.

Bemporad, A., Bozinis, N. A., Dua, V., Morari, M., Pistikopoulos, E. N., and Sauro, P. (2000), Model predictive control: A multi-parametric programming approach, in *Computer Aided Chemical Engineering (Volume 8: Elsevier)*, 301-306.

Bertsekas, D. P. (1999). *Nonlinear Programming*: Athena Scientific.

Bhatia, T. and Biegler, L. T. (1996). Dynamic Optimization in the Design and Scheduling of Multiproduct Batch Plants. *Industrial & Engineering Chemistry Research*, 35, 2234-2246.

Bonvin, D. (2006). Control and optimization of batch processes. *Control Systems, IEEE*, 26, 34-45.

Buchan, A. D., Haldane, D. W., and Fearing, R. S. (2013). Automatic identification of dynamic piecewise affine models for a running robot. *Proceedings of 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, 5600-5607.

Camacho, E. F. and Bordons Alba, C. (2007). *Model Predictive Control*. London: Springer.

Casselman, S. and Rodrigues, L. (2009). A New methodology for piecewise affine models using Voronoi partitions. *Proceedings of The 48th IEEE Conference on Decision and Control*, held jointly with the 28th Chinese Control Conference, Shanghai, 3920-3925.

Chu, Y. and You, F. (2012). Integration of scheduling and control with online closed-loop implementation: Fast computational strategy and large-scale global optimization algorithm. *Computers & Chemical Engineering*, 47, 248-268.

Chu, Y. and You, F. (2013a). Integrated scheduling and dynamic optimization of complex batch processes with general network structure using a generalized benders decomposition approach. *Industrial & Engineering Chemistry Research*, 52, 7867-7885.

Chu, Y. and You, F. (2013b). Integration of production scheduling and dynamic optimization for multi-product CSTRs: Generalized Benders decomposition coupled with global mixed-integer fractional programming. *Computers & Chemical Engineering*, 58, 315-333.

Chu, Y. and You, F. (2013c). Integration of Scheduling and Dynamic Optimization of Batch Processes under Uncertainty: Two-Stage Stochastic Programming Approach and Enhanced Generalized Benders Decomposition Algorithm. *Industrial & Engineering Chemistry Research*, 52, 16851-16869.

Chu, Y. and You, F. (2014a). Integrated scheduling and dynamic optimization by stackelberg game: bilevel model formulation and efficient solution algorithm. *Industrial & Engineering Chemistry Research*, 53, 5564-5581.

Chu, Y. and You, F. (2014b). Moving horizon approach of integrating scheduling and control for sequential batch processes. *AIChE Journal*, 60, 1654–1671.

De Nicolao, G., Magni, L., Scattolini, R., Garulli, A., and Tesi, A. (1999), Robustness of receding horizon control for nonlinear discrete-time systems, in *Robustness in identification and control* (Lecture Notes in Control and Information Sciences, 245; Berlin / Heidelberg: Springer), 408-421.

Domahidi, A., Zraggen, A. U., Zeilinger, M. N., Morari, M., and Jones, C. N. (2012). Efficient interior point methods for multistage problems arising in receding horizon control. *Proceedings of Decision and Control (CDC), IEEE 51st Annual Conference on*, 668-674.

Domahidi, A. (2012), *FORCES: Fast Optimization for Real-time Control on Embedded Systems*, <<http://forces.ethz.ch>>.

Domínguez, L. F., Narciso, D. A., and Pistikopoulos, E. N. (2010). Recent advances in multiparametric nonlinear programming. *Computers & Chemical Engineering*, 34, 707-716.

Dua, V. and Pistikopoulos, E. N. (1999). Algorithms for the Solution of Multiparametric Mixed-Integer Nonlinear Optimization Problems. *Industrial & Engineering Chemistry Research*, 38, 3976-3987.

Dua, V. and Pistikopoulos, E. (2000). An Algorithm for the Solution of Multiparametric Mixed Integer Linear Programming Problems. *Annals of Operations Research*, 99, 123-139.

Dua, V., Bozinis, N. A., and Pistikopoulos, E. N. (2002). A multiparametric programming approach for mixed-integer quadratic engineering problems. *Computers & Chemical Engineering*, 26, 715-733.

Engell, S. and Harjunkski, I. (2012). Optimal operation: Scheduling, advanced control and their integration. *Computers & Chemical Engineering*, 47, 121-133.

Feather, D., Harrell, D., Lieberman, R., and Doyle, F. J. (2004). Hybrid approach to polymer grade transition control. *AIChE Journal*, 50, 2502-2513.

Ferrari-Trecate, G., Muselli, M., Liberati, D., and Morari, M. (2003). A clustering technique for the identification of piecewise affine systems. *Automatica*, 39, 205-217.

Ferreau, H. J., Bock, H. G., and Diehl, M. (2008). An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18, 816-830.

Flores-Tlacuahuac, A. and Biegler, L. T. (2005). A robust and efficient mixed-integer non-linear dynamic optimization approach for simultaneous design and control. *Computer Aided Chemical Engineering*, 20, 67-72.

Flores-Tlacuahuac, A. and Grossmann, I. E. (2006). Simultaneous Cyclic Scheduling and Control of a Multiproduct CSTR. *Industrial & Engineering Chemistry Research*, 45, 6698-6712.

Flores-Tlacuahuac, A. and Grossmann, I. E. (2010). Simultaneous Cyclic Scheduling and Control of Tubular Reactors: Single Production Lines. *Industrial & Engineering Chemistry Research*, 49, 11453-11463.

- Floudas, C. A. and Lin, X. (2004). Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review. *Computers & Chemical Engineering*, 28, 2109-2129.
- Frank, R., Schneid, J., and Ueberhuber, C. W. (1985). Stability Properties of Implicit Runge-Kutta Methods. *SIAM Journal on Numerical Analysis*, 22, 497-514.
- Gal, T. and Nedoma, J. (1972). Multiparametric Linear Programming *Management Science*, 18, 406-422.
- Gal, T. (1975). Rim Multiparametric Linear Programming. *Management Science*, 21, 567-575.
- Gegúndez, M. E., Aroba, J., and Bravo, J. M. (2008). Identification of piecewise affine systems by means of fuzzy clustering and competitive learning. *Engineering Applications of Artificial Intelligence*, 21, 1321-1329.
- Harjunkski, I., Nystrom, R., and Horch, A. (2009). Integration of scheduling and control--Theory or practice? *Computers & Chemical Engineering*, 33, 1909-1918.
- Ierapetritou, M. G. and Floudas, C. A. (1998). Effective Continuous-Time Formulation for Short-Term Scheduling. 1. Multipurpose Batch Processes. *Industrial & Engineering Chemistry Research*, 37, 4341-4359.
- Johansson, M. (2003). *Piecewise Linear Control Systems*. Berlin: Springer-Verlag.
- Jones, C. N., Grieder, P., and Raković, S. V. (2006). A logarithmic-time solution to the point location problem for parametric linear programming. *Automatica*, 42, 2215-2218.
- Julian, P., Desages, A., and Agamennoni, O. (1999). High-level canonical piecewise linear representation using a simplicial partition. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 46, 463-480.
- Kirk, D. E. (1970). *Optimal control theory: an introduction*. London: Prentice-Hall.
- Kouramas, K. I., Faísca, N. P., Panos, C., and Pistikopoulos, E. N. (2011). Explicit/multi-parametric model predictive control (MPC) of linear discrete-time systems by dynamic and multi-parametric programming. *Automatica*, 47, 1638-1645.
- Kvasnica, M., Grieder, P., and Baoti, M. (2004), Multi-Parametric Toolbox (MPT), <<http://control.ee.ethz.ch/~mpt/>>.
- Kvasnica, M., Szucs, A., and Fikar, M. (2011). Automatic Derivation of Optimal Piecewise Affine Approximations of Nonlinear Systems. *Proceedings of 18th IFAC World Congress*, Milano, Italy, 8675-8680.
- Lee, J. H. (2011). Model predictive control: Review of the three decades of development. *International Journal of Control, Automation and Systems*, 9, 415-424.
- Li, Z. and Ierapetritou, M. G. (2008a). Reactive scheduling using parametric programming. *AIChE Journal*, 54, 2610-2623.
- Li, Z. and Ierapetritou, M. G. (2008b). Robust Optimization for Process Scheduling Under Uncertainty. *Industrial & Engineering Chemistry Research*, 47, 4148-4157.



- Li, Z. and Ierapetritou, M. G. (2009). Integrated production planning and scheduling using a decomposition framework. *Chemical Engineering Science*, 64, 3585-3597.
- Li, Z. and Ierapetritou, M. G. (2010). Production planning and scheduling integration through augmented Lagrangian optimization. *Computers & Chemical Engineering*, 34, 996-1006.
- Löfberg, J. (2004). YALMIP : A Toolbox for Modeling and Optimization in MATLAB. *Proceedings of Computer Aided Control System Design, 2004 IEEE International Symposium on, Taipei, Taiwan*.
- Lopez-Negrete, R., D'Amato, F. J., Biegler, L. T., and Kumar, A. (2013). Fast nonlinear model predictive control: Formulation and industrial process applications. *Computers & Chemical Engineering*, 51, 55-64.
- Magnani, A. and Boyd, S. (2009). Convex piecewise-linear fitting. *Optimization and Engineering*, 10, 1-17.
- Mahadevan, R., Doyle, F. J., and Allcock, A. C. (2002). Control-relevant scheduling of polymer grade transitions. *AIChE Journal*, 48, 1754-1764.
- Maravelias, C. T. (2012). General framework and modeling approach classification for chemical production scheduling. *AIChE Journal*, 58, 1812-1828.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36, 789-814.
- Mendez, C. A., Cerda, J., Grossmann, I. E., Harjunkski, I., and Fahl, M. (2006). State-of-the-art review of optimization methods for short-term scheduling of batch processes. *Computers & Chemical Engineering*, 30, 913-946.
- Mishra, B. V., Mayer, E., Raisch, J., and Kienle, A. (2005). Short-Term Scheduling of Batch Processes. A Comparative Study of Different Approaches. *Industrial & Engineering Chemistry Research*, 44, 4022-4034.
- Mitra, K., Gudi, R. D., Patwardhan, S. C., and Sardar, G. (2009). Resiliency Issues in Integration of Scheduling and Control. *Industrial & Engineering Chemistry Research*, 49, 222-235.
- Mouret, S., Grossmann, I. E., and Pectiaux, P. (2011). Time representations and mathematical models for process scheduling problems. *Computers & Chemical Engineering*, 35, 1038-1063.
- Nakada, H., Takaba, K., and Katayama, T. (2005). Identification of piecewise affine systems based on statistical clustering technique. *Automatica*, 41, 905-913.
- Nesterov, Y. (1983). A method for solving a convex programming problem with convergence rate  $O(1/k^2)$ . *Soviet Mathematics Doklady*, 27, 372-376.
- Nie, Y., Biegler, L. T., and Wassick, J. M. (2012). Integrated scheduling and dynamic optimization of batch processes using state equipment networks. *AIChE Journal*, 58, 3416-3432.

- Nie, Y., Biegler, L. T., Villa, C. M., and Wassick, J. M. (2014). Discrete time formulation for the integration of scheduling and dynamic optimization. *Industrial & Engineering Chemistry Research*, DOI: 10.1021/ie502960p,
- Nystrom, R. H., Franke, R., Harjunkski, I., and Kroll, A. (2005). Production campaign planning including grade transition sequencing and dynamic optimization. *Computers & Chemical Engineering*, 29, 2163-2179.
- Nystrom, R. H., Harjunkski, I., and Kroll, A. (2006). Production optimization for continuously operated processes with optimal operation and scheduling of multiple units. *Computers & Chemical Engineering*, 30, 392-406.
- Ohlsson, H. and Ljung, L. (2011). Identification of Piecewise Affine Systems Using Sum-of-Norms Regularization. *Proceedings of 18th IFAC World Congress, Milano, Italy*, 6640-6645.
- Padhiyar, N., Bhartiya, S., and Gudi, R. D. (2006). Optimal Grade Transition in Polymerization Reactors: A Comparative Case Study. *Industrial & Engineering Chemistry Research*, 45, 3583-3592.
- Pistikopoulos, E. N. (1995). Uncertainty in process design and operations. *Computers and Chemical Engineering*, 19, S553-S563.
- Pistikopoulos, E. N. (2009). Perspectives in multiparametric programming and explicit model predictive control. *AIChE Journal*, 55, 1918-1925.
- Pontryagin, L. S., Boltyanskil, V. G., Gamkrelidge, R. V., and Mishchenko, E. F. (1962). *The mathematical theory of optimal processes*. New York: Interscience.
- Rao, C. V., Wright, S. J., and Rawlings, J. B. (1998). Application of Interior-Point Methods to Model Predictive Control. *Journal of Optimization Theory and Applications*, 99, 723-757.
- Richter, S., Jones, C. N., and Morari, M. (2009). Real-time input-constrained MPC using fast gradient methods. *Proceedings of Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference.*, 7387-7393.
- Richter, S., Jones, C. N., and Morari, M. (2012). Computational Complexity Certification for Real-Time MPC With Input Constraints Based on the Fast Gradient Method. *Automatic Control, IEEE Transactions on*, 57, 1391-1403.
- Rodrigues, L. and How, J. P. (2003). Synthesis of piecewise-affine controllers for stabilization of nonlinear systems. *Proceedings of 42nd IEEE Conference on Decision and Control*, 2071-2076.
- Rodrigues, L. and Boyd, S. (2005). Piecewise-affine state feedback for piecewise affine slab systems using convex optimization. *Systems & Control Letters*, 835-853.
- Roll, J., Bemporad, A., and Ljung, L. (2004). Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40, 37-50.
- Safonov, M. G. (1980). *Stability and Robustness of Multivariable Feedback Systems*. Cambridge, MA: MIT Press.

Sontag, E. D. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, 26, 346-358.

Števek, J., Szűcs, A., Kvasnica, M., Fikar, M., and Kozák, Š. (2012). Two steps piecewise affine identification of nonlinear systems. *Archives of Control Sciences*, 22, 371-388.

Sundaram, B. S., Upreti, S. R., and Lohi, A. (2005). Optimal Control of Batch MMA Polymerization with Specified Time, Monomer Conversion, and Average Polymer Molecular Weights. *Macromolecular Theory and Simulations*, 14, 374-386.

Terrazas-Moreno, S., Flores-Tlacuahuac, A., and Grossmann, I. E. (2007). Simultaneous cyclic scheduling and optimal control of polymerization reactors. *AIChE Journal*, 53, 2301-2315.

Terrazas-Moreno, S., Flores-Tlacuahuac, A., and Grossmann, I. E. (2008a). Lagrangean heuristic for the scheduling and control of polymerization reactors. *AIChE Journal*, 54, 163-182.

Terrazas-Moreno, S., Flores-Tlacuahuac, A., and Grossmann, I. E. (2008b). Simultaneous design, scheduling, and optimal control of a methyl-methacrylate continuous polymerization reactor. *AIChE Journal*, 54, 3160-3170.

Tøndel, P., Johansen, T. A., and Bemporad, A. (2003). Evaluation of piecewise affine control via binary search tree. *Automatica*, 39, 945-950.

Vasak, M., Klanjcic, D., and Peric, N. (2006). Piecewise affine identification of MIMO processes. *Proceedings of Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, Munich*, 1493-1498.

Wang, Y. and Boyd, S. (2008). Fast model predictive control using online optimization. *Proceedings of Proceedings of the 17th IFAC World Congress, Seoul, Korea*.

Wang, Y. and Boyd, S. (2010). Fast Model Predictive Control Using Online Optimization. *Control Systems Technology, IEEE Transactions on*, 18, 267-278.

Zavala, V. M., Laird, C. D., and Biegler, L. T. (2008). Fast implementations and rigorous models: Can both be accommodated in NMPC? *International Journal of Robust and Nonlinear Control*, 18, 800-815.

Zavieh, A. and Rodrigues, L. (2013). Intersection-based piecewise affine approximation of nonlinear systems. *Proceedings of 21st Mediterranean Conference on Control & Automation, Chania*, 640-645.

Zhuge, J. and Ierapetritou, M. G. (2012). Integration of Scheduling and Control with Closed Loop Implementation. *Industrial & Engineering Chemistry Research*, 51, 8550-8565.

Zhuge, J. and Ierapetritou, M. G. (2014). Integration of scheduling and control for batch processes using multi-parametric model predictive control. *AIChE Journal*, 60, 3169-3183.