Slovak University of Technology in Bratislava,
Faculty of Chemical and Food Technology,
Institute of Information Engineering, Automation and Mathematics

Tracking of Necessary Conditions of Optimality in Real-time Dynamic Optimisation of Batch Processes

Marián Podmajerský

Dissertation Thesis

Supervisor: Miroslav Fikar

2011

Acknowledgement

It would not have been possible to write this thesis without a tight collaboration with my colleagues. I would like to express my dearest gratitude to my direct supervisor prof. Miroslav Fikar who has taken care of my scientific activities since I was an early stage student. I am very grateful for his teaching guidance since my first encounter with Linux stuff as well as with LaTeX, Matlab, C, and much more. Furthermore, he allowed me to meet and visit the top level research groups for which I would like to express my sincere gratitude. Moreover, I owe a special thank to my consultant Dr. Benoit Chachuat who have introduced me to the subject and who have been valuable source of inspiration during the course of my studies. Especially, I would like to thank for his guidance throughout my research and for his patience with me. Without him, this thesis would not exists.

Many thanks goes to my former PhD colleague Juraj Vörös for instantaneous discussions and his support with my early times in the department. Michal Čižniar, Tomáš Hirmajer and Martin Herceg deserve a special thanks for the help with software implementation of dynamic optimisation methods.

I would also like to thank prof. Dominique Bonvin and prof. C. L. E. Schwartz for the opportunity to spend the time at their departments as a visiting researcher and for the welcoming environment at their departments. Throughout my studies I met a lot of remarkable people that have shared their research experiences. Thank you for that!

These acknowledgements would not be complete without thanking all the friends who have supported me during the study and who have genuinely enriched this times. The long list in short, I therefore wish to express my profound gratitude to them collectively. Scientific work is not devoid of administrative concerns which can be indescribably annoying. These concerns have been almost completely alleviated by the great work of Monika Mojžišová. Special thanks goes to the technician Stanislav Vagač for his instantaneous support with the technical stuff all over the department.

I would like to express my uttermost gratitude to my parents who have always encouraged me throughout my successes and failures as well as to my sister for her constant support. Finally, my the most profound gratitude goes to Silvia Holúbková for her immense generosity, for her enthusiastic, unconditional and sincere support in everything I commence and I commit to.

Marián Podmajerský
Bratislava, June 2011

Abstract

This work is deals with optimal control of batch processes in the presence of uncertainty. An integrated two-time-scale control is proposed, whereby a run-to-run adaptation strategy with adaptation of the terminal constraints is implemented at the slow time scale, and is integrated with a neighbouring-extremal controller that operates at the fast time scale and performs further on-line corrections. This control scheme is especially suitable for repeatable batch processes with the fast changes in process dynamics. In addition, this scheme can be easily realised in real batch processes as the required computational power is low. Particularly, the only computation performed in real-time at each sampling time is a solution of a linear two-point boundary value problem. By sacrificing a bit of accuracy, all the required controlled designs and an accompanying computations might by done off-line. In the presence of uncertainty, the necessary conditions of optimality no longer hold. The core idea is to use the so called NCO-tracking approach that pushes the gradients caused by an uncertainty to zero. Neighbouring-extremal controller is approximated controller, i.e. it is based on linearisation of the nominal solution. Because of a lower performance of such control solution in chemical applications, the need for a supplementary adaptation is obvious. Our proposed control scheme thus corrects approximated control by another control. Run-to-run adaptation strategy updates the model between batches according to the latest constraints measurements and re-optimises the nominal solution. This solution then provides the reference trajectories for neighbouring-extremal controller. The thesis describes essentials to understand the basic building blocks of the proposed control scheme. In particular, the first part introduces the nominal optimisation, i.e optimisation under ideal circumstances without the influence of the uncertainty. Next part discusses the efficient algorithms that deals with the uncertainty. The proposed control is verified on real process. It is shown that the integrated two-time-scale control scheme has faster convergence rate and better performance in comparison to the other tested approaches.

Abstrakt

Táto práca sa zaoberá optimálnym riadením vsádzkových procesov v prítomnosti neurčitostí. Navrhuje kombinované riadenie, ktoré operuje v dvoch časových škálach tak, aby koncové obmedzenia boli adaptované medzi jednotlivými vsádzkami (pomalá časová škála) a aby riadenie samotného vsádzkového procesu bolo opravované aproximovaným regulátorom (rýchla časová škála). Takáto riadiaca schéma je obzvlášť vhodná pre vsádzkové procesy, ktoré sa pravidelne opakujú a majú rýchle zmeny dynamiky. V neposlednom rade, takéto riadenie je v praxi ľahko realizovateľné, keďže náročnosť na výpočet je malá. V tomto prípade je jediným výpočtom, ktorý sa počíta v reálnom čase vo vzorkách periódy, lineárny dvojbodový hraničný problém. Za cenu nižšej presnosti môže byť celé riadenie navrhnuté a predpočítané vopred. V takom prípade, dvojbodový problém sa skonvertuje na maticový systém Ricattiho rovníc. Keďže v prítomnosti neurčitostí nie sú splnené nevyhnutné podmienky optimality (NPO), hlavnou myšlienkou je použiť prístup sledovania NPO. V tomto prístupe sú gradienty vzniknuté neurčitosťami usmerňované ku nule. Jeden zo spôsobov ako to dosiahnúť, je aproximovať riadenie linearizáciou optimálneho riešenia a použiť aktuálne merania stavových veličín. Avšak, takéto riadenie vykazuje nižšiu kvalitu riadenia chemických procesov pre ich veľmi nelineárne správanie. Nami navrhnutá riadiaca schéma preto koriguje aproximované riadenie ďalším riadením. Konkrétne, adaptovaním koncových obmedzení medzi jednotlivými vsádzkami podľa ich meraní. Podľa rozdielu medzi skutočnými a očakávanými hodnotami koncových obmedzení sa upraví model a vypočíta sa nové optimálne riadenie. Toto optimálne riadenie zároveň slúži ako referencia pre aproximovaný regulátor. Práca poskytuje teoretické minimum pre návrh a implementáciu navrhovanej riadiacej schémy. Konkrétne, prvá časť definuje optimalizačný problem a jeho riešenie za ideálnych okolností, t.j. bez vplyvu neurčitostí. Ďalšia časť navrhuje efektívne prístupy na riadenie vsádzkových procesov v prítomnosti neurčitostí. Záverečná časť overuje kvalitu kombinovaného riadenia na reálnom zariadení. Kombinované riadenie preukázalo, že rýchlejšie konverguje a má lepšie vlastnosti ako ostatné študované prístupy.

Contents

Introduction 1
Thesis Structure . . . 3
Main Goals . . . 5

I Dynamic Optimisation under Ideal Conditions 7

1 Nominal Optimisation 9
1.1 Introduction . . . 9
1.2 Optimisation Problem Statement . . . 10
1.2.1 Admissible Controls . . . 11
1.2.2 Process Model Equations . . . 11
1.2.3 Performance criterion . . . 11
1.2.4 Constraints . . . 13
1.2.5 Problem Formulation . . . 14

2 Necessary Conditions for Optimality 15
2.1 Problems without Constraints . . . 15
2.2 Problems with Constraints . . . 18
2.2.1 Terminal Constraints . . . 18
2.2.2 Path Constraints . . . 19

3 Numerical Optimisation Methods 23
3.1 Direct Optimisation Methods . . . 23
3.1.1 Sequential Approach . . . 24
3.1.2 Simultaneous Approach . . . 26
3.1.3 Other Direct Approaches . . . 29
3.2 Indirect Optimisation Methods . . . 29
3.2.1 Pontryagin's Formulation . . . 29
3.2.2 Hamilton-Jacobi-Bellman Formulation . . . 30
3.2.3 Boundary Condition Iteration – Single-Shooting . . . 30
3.2.4 Multiple Shooting . . . 32
3.2.5 Quasi-Linearisation or discretisation . . . 32
3.2.6 Control Vector Iteration . . . 33

4 Nominal Solution 35
4.1 Batch Reactor Control . . . 38
4.1.1 Unconstrained Case . . . 38
4.1.2 Constrained case . . . 39

II Dynamic Optimisation under Uncertain Conditions 43

5 Tracking of Necessary Conditions for Optimality 45

6 NCO-tracking Associated with Active Constraints 47
6.1 Switching Structure . . . 47
6.2 Fixed and Free Input Elements . . . 47
6.3 Linking the Variables to the NCO . . . 48
6.4 Two-stage Batch Reactor Control . . . 48
6.4.1 Problem Definition . . . 48
6.4.2 Nominal Solution . . . 49
6.4.3 Proposed Control Structures . . . 50
6.4.4 Results . . . 54

7 NCO Associated with Sensitivities 59
7.1 Non-singular NE Control without Terminal Constraints . . . 59
7.1.1 Problem Formulation . . . 59
7.1.2 Neighbouring-extremal Control . . . 59
7.1.3 Numerical Computation . . . 61
7.1.4 Van de Vusse Reaction . . . 63
7.2 Non-singular NE Control with Terminal Constraints . . . 66
7.2.1 Problem Formulation . . . 66
7.2.2 Necessary Conditions for Optimality . . . 66
7.2.3 Neighbouring-Extremal Control . . . 71
7.2.4 Numerical Computation of Neighbouring Feedback Control . . . 73
7.2.5 Shooting Method . . . 73
7.3 Run-to-run Constraint Adaptation . . . 75

8 NCO-tracking by Two-time-scale Control 77
8.1 Batch Reactor Control . . . 79
8.1.1 Results . . . 79
8.2 Semi-batch Reactor Control . . . 80
8.2.1 Problem Formulation . . . 83
8.2.2 Open-Loop Optimal Control . . . 83
8.2.3 Closed-Loop Optimal Control . . . 84

III Application 87

9 Two-tanks Connected in Series 89

9.1 Problem Formulation . . . 89

9.2 Open-loop Optimal Control . . . 90

9.3 Closed-loop Optimal Control . . . 91

9.4 Experimental Results . . . 94

10 Conclusions 97

References 106

Publication List 107

Curiculum Vitae 109

List of Abbreviations

DOP	–	Dynamic Optimisation Problem
NLP	–	Non-linear Problem
NE	–	Neighbouring-extremal
NCO	–	Necessary Conditions for Optimality
CA	–	Constraint Adaptation
TPBVP	–	Two-point Boundary Value Problem
ODE	–	Ordinary Differential Equations
DAE	–	Differential-algebraic Equations
PMP	–	Pontryagin's Minimum/Maximum Principle
HJB	–	Hamilton-Jacobi-Bellman
SQP	–	Successive Quadratic Programming
BCI	–	Boundary Condition Iteration
CVI	–	Control Vector Iteration
CVP	–	Control Vector Parametrisation
KKT	–	Karush-Kuhn-Tucker
MPC	–	Model Predictive Control
NMPC	–	Non-linear Model Predictive Control
LGC	–	Linear-Quadratic-Gaussian Control
IVP	–	Initial Value Problem
LQ	–	Linear-Quadratic
PI	–	Proportional-Integral
PID	–	Proportional-Integral Derivative
OC	–	Orthogonal Collocation
IS	–	Integrated two-times-scale control

Introduction

In general, controlled processes allow certain flexibility of operation within which the process can be handled. A control policy can then be heuristic approach based on knowledge and human observation, or more effective systematic approach based on feedback control and optimisation techniques. For instance, a reactor can operate with varying flow rate, temperature, current, heat duty, or pressure subject to physical constraints. The goal is to choose the best set of operating conditions that satisfy desired production objectives (e.g. steady-state, conversion rate, or final product concentration).

In addition, the controlled process is usually subject to large uncertainty during its operation. Common sources of uncertainty include measurement noise, inaccurate kinetic rate parameters, feed impurities, and fouling. These normally give a rise to a lower production quality and quantity along with operational constraint violations. In such a case, the optimal control has the ability not only to optimise the operating policy but as well as to mitigate the effect of uncertainty on process performance, especially in the presence of constraints (Kadam and Marquardt, 2007).

The solution of dynamic optimisation problem has been subject of interest centuries ago. Johann Bernoulli posed first dynamic optimisation problem, the Brachystochrone (greek *quickest*) problem (Tikhomirov, 1986), in 1696. The problem was to find the shape of a frictionless wire that causes a bead, initially at rest, to move under the influence of gravity to a specified point in minimum time. Bernoulli motivated the mathematicians of Europe and the problems of dynamic optimisation earned their interest.

In order to control processes, one needs the proper mathematical description of their behaviour. In general, mathematical models are usually inaccurate and they do not exactly describe a behaviour of a real process as some model variables cannot be determined at all. These unknown variables are considered as uncertainties and they may take a form of model mismatch, variations of the process parameters, and of the process disturbances. Hence, the optimal profiles (open-loop solution) computed for the modelled process lead the actual process into non-optimal operating conditions and constraint violations.

As the optimisation of dynamic processes heavily relies on mathematical models that are normally inaccurate, the need for a methodology that encompasses the lack of accuracy in models has motivated the development of optimisation schemes that can operate in presence of uncertainty.

Amongst the many approaches that have been proposed in the past, a popular one consists of adjusting the model according to the most recent measurements. Controller is then redesigned or optimisation is rerun in order to obtain input updates. Such optimisation is also called explicit since the model is used explicitly to compute the updated input.

The methodology can be observed in Linear-Quadratic-Gaussian (LGC) control (Zhou, Doyle, et al., 1995), adaptive control (Äström and Wittenmark, 1983, 1989), robust H_∞ loop-shaping (Doyle and Stein, 1981; McFarlane and Glover, 1989; Zhou, Doyle, et al., 1995), non-linear model predictive control (NMPC) with polytopic invariant sets (Cannon, Deshmukh, et al., 2003, 2004; Chen, Ballance, et al., 2001), and traditional NMPC (Abel and Marquardt, 1998; Allgöwer and Zheng, 2000; Garcia, Prett, et al., 1989).

Traditional NMPC implements a re-optimisation strategy and uses measurements to update the current state in the model. This strategy suffers from some important deficiencies: i) the re-optimisations may not be tractable in real-time; ii) the need of expensive hardware and software setup. Clearly, the time needed to re-optimise the system depends on both the problem complexity and the computing performance. A re-optimisation frequency that is too small may lead to performance loss, or worse constraint violations and instability, especially in chemical processes that exhibit fast process dynamic. These shortcomings have motivated development of advanced MPC techniques (Diehl, Bock, et al., 2002; Diehl, Gerhard, et al., 2008; Würth, Hannemann, et al., 2009a).

Another technique is the so-called explicit MPC approach (Bemporad, Morari, et al., 2002; Dominguez and Pistikopoulos, 2010; Kvasnica, 2009; Pistikopoulos, Georgiadis, et al., 2007a,b). Multi-parametric programming is used to pre-compute off-line all possible control actions for a given range of the state variables. The control inputs are then adjusted by simply selecting the control law that corresponds to the actual state of the process, as given by the latest measurements. Although this method can accommodate fast sampling times, its foremost limitation comes from the curse of dimensionality and from the quality of the linearisations. This currently limits the application of explicit MPC to problems having no more than a few state variables as well as piecewise linear dynamics.

Therefore, alternative (implicit) approaches that do not rely on re-optimisation are needed. The development of the calculus of variations, by many scientist of 18th and 19th century, allowed the derivation of necessary and sufficient conditions for a dynamic optimisation problem (also *optimal control*). However, these conditions indicate if a solution is optimal, but not necessarily how to find an optimal (or even improved) solution. In restricted cases, they can be used for finding an analytical solution. Numerical solutions were not attempted upon the advent of computers. In addition, an information about actual state of (non-) optimality can be used to recover the optimality loss. The implicit approach is known from the literature as a tracking of necessary conditions for optimality (NCO-tracking) (François, Srinivasan, et al., 2007; Srinivasan and Bonvin, 2004a, 2007; Srinivasan, Bonvin, et al., 2003b; Srinivasan, Visser, et al., 1997; Visser, Srinivasan, et al., 1999). The problem of NCO tracking can be divided into two main sub-problems in which it is associated with either active constraints or with sensitivities. Former research showed that the tracking of active constraints can be done using standard control tools. In opposite, the tracking of sensitivities is not so simple because the sensitivity part is not given explicitly by the state of the system.

The very first methods of reference seeking called “extremal control” or “self-optimising control” (Blackman, 1962; Morosanov, 1957; Ostrovskii, 1957; Pervozvanskii, 1960) were developed back in the 1950–1960s. In the last decades, treating of the uncertainties regained significant popularity. This renewed interest is possible due advances in instrumentation, i.e. through the availability of measurements and of a computational power.

This work presents a two-time-scale approach, where a run-to-run adaptation strategy (Bonvin, Srinivasan, et al., 2006) with adaptation of the terminal constraint (Marchetti, Chachuat, et al., 2007) is implemented at the slow time scale (outer loop) and is integrated with a (constrained) neighbouring-extremal (NE) controller (Bryson and Ho, 1975) that operates at the fast time scale (inner loop). A similar two-time-scale control was previously proposed by (Würth, Hannemann, et al., 2009a; Zavala and T. 2009) where NMPC is implemented at slow time scale and a first-order sensitivity update is performed at the fast time scale. (Gros, 2007) introduced the combination of NMPC and NE control. The NE control provided fast input updates in inner loops, while the optimal control trajectories are updated by NMPC in outer loops.

The novelty of this work is the development of a two-level scheme for batch processes. Analogous to the adaptation of initial conditions for each receeding horizon in NMPC, terminal constraints are adapted here between each batch based on the mismatch between their predicted and measured values. Then, the entire batch operation is re-optimised between the runs. In order to reject disturbances within each run and at the same time to promote feasibility and optimality, a NE controller is considered here as the inner loop. The theory of NE control, which has been developed over the last 4-5 decades to avoid the costly re-optimisation of (fast) dynamic systems, is indeed well-suited for batch process control. The integration between the outer- and inner-loops occurs naturally since the NE controllers are recalculated after each run based on the solution to the outer-loop optimisation problem. The resulting integrated two-time-scale optimisation scheme thus offers promise to enhance performance and tractability.

Thesis Structure

Part I introduces the problem of dynamic optimisation under ideal conditions and its various forms. This part also provides the overview of the current state in the literature. Chapter 2 postulates the necessary conditions of optimality for the problems of dynamic optimisation. These conditions represent the foundation of the real-time dynamic optimisation for the batch processes. Next, the computational principles of nominal optimisation are explained, in particular, the direct and indirect numerical methods are discussed in Chapter 3. Possible implementations of direct methods have been published in:

— M. Čižniar, M. Podmajerský, T. Hirmajer, M. Fikar, M. A. Latifi (2009). “Global optimization for parameter estimation of differential-algebraic systems”. In: *Chemical Papers* **63**(3), pp. 274–283

— M. Podmajerský, M. Čižniar, T. Hirmajer, M. Fikar, M. A. Latifi (2007). “Recent Developments in Dynopt Package”. In: *Proceedings of the 16th International Conference Process Control '07*. Slovak University of Technology in Bratislava, 032f.pdf

Finally, the Chapter 4 demonstrates step-by-step procedure to obtain a nominal solution.

Part II discusses the real-time optimisation scheme that uses separate controllers to meet the optimality conditions and to deliver optimal control policy for real batch processes. This part also presents the main contributions of the thesis. Chapter 5 explains the NCO-tracking control scheme that directly incorporates the NCO in the control design. The following Chapter 6 and Chapter 7 are devoted to NCO-tracking associated with active constraints and associated with sensitivities. Secondly, an extension of these two approaches results in NCO-tracking by two-time-scale control for constrained problems and it is proposed in Chapter 8. The results in these chapters correspond to those given as:

NCO-tracking by standard PID controllers. Published in:

- M. Podmajerský and M. Fikar (2008). “Measurement-based Run-to-run Optimisation of Hybrid Two-stage Reactor System”. In: *Proceedings of the 8th International Scientific - Technical Conference Process Control 2008*. Kouty nad Desnou, Czech Republic: University of Pardubice, C025a–1–C025a–11

NCO-tracking by NE controller. Published in:

- M. Podmajerský and M. Fikar (2009a). “On-Line Neighbouring-Extremal Controller Design for Setpoint-Transition in Presence of Uncertainty”. In: *AT&P Journal Plus*(2), pp. 77–83
- M. Podmajerský and M. Fikar (2010). “Measurement-based Run-to-run Optimisation of Hybrid Two-stage Reactor System”. In: *Selected Topics in Modelling and Control*. **6**. Slovak University of Technology Press, pp. 44–51

NCO-tracking by two-time-scale control. Published in:

- M. Podmajerský, B. Chachuat, M. Fikar (2011a). “Integrated Two-Time-Scale Scheme for Real-time Optimisation of Batch Processes”. In: *Proc. of the 18th IFAC World Congress*. Milano, Italy
- M. Podmajerský and M. Fikar (2011). “Real-time Dynamic Optimisation by Integrated Two-Time-Scale Scheme”. In: *Proceedings of the 18th International Conference on Process Control 11*. Slovak University of Technology in Bratislava. Štrbské Pleso, Slovakia
- M. Podmajerský, B. Chachuat, M. Fikar (2011b). “Measurement-based Optimisation of Batch Processes using an Integrated Two-Time-Scale Scheme”. In: *Optimization and Engineering*. (Submitted)
- M. Podmajerský, B. Chachuat, M. Fikar (2011c). “Run-to-run Optimisation of Batch Processes with In-batch Controller”. In: *Selected Topics on Constrained and Nonlinear Control. Preprints*. STU Bratislava - NTNU Trondheim, pp. 337–342

Part III presents the application of the proposed control approaches introduced in the Part II to a laboratory device. The level control of two connected tanks with liquid interaction is considered to illustrate the advantages of the proposed NCO-tracking control scheme with integrated controllers in comparison to NCO-tracking schemes with standalone controllers. The results have been published in:

— M. Podmajerský, B. Chachuat, M. Fikar (2011a). “Integrated Two-Time-Scale Scheme for Real-time Optimisation of Batch Processes”. In: *Proc. of the 18th IFAC World Congress.* Milano, Italy

— M. Podmajerský and M. Fikar (2011). “Real-time Dynamic Optimisation by Integrated Two-Time-Scale Scheme”. In: *Proceedings of the 18th International Conference on Process Control 11.* Slovak University of Technology in Bratislava. Štrbské Pleso, Slovakia

— M. Podmajerský, B. Chachuat, M. Fikar (2011b). “Measurement-based Optimisation of Batch Processes using an Integrated Two-Time-Scale Scheme”. In: *Optimization and Engineering.* (Submitted)

— M. Podmajerský, B. Chachuat, M. Fikar (2011c). “Run-to-run Optimisation of Batch Processes with In-batch Controller”. In: *Selected Topics on Constrained and Nonlinear Control. Preprints.* STU Bratislava - NTNU Trondheim, pp. 337–342

Main Goals

The prime aim of this thesis is to develop and experimentally examine a control scheme that is able to control batch processes with respect to the terminal requirements, in the optimal way, and in the on-line fashion. It is also required that a such control scheme is model-based and it respects the conditions of optimality, it is robust enough to deal with various forms of an uncertainty (measurement noise, model mismatch, perturbations), it can handle fast process dynamic and finally, with low implementation costs (without unnecessary re-optimisations). This objective is exploited in the three phases: the optimisation of the modelled process, the control design, and at last the practical application of the proposed solution in laboratory conditions.

In detail, the objectives can be summarised as follows:

- Transform the open-loop nominal solution into a closed-loop control structure by decoupling into several NCO parts that are further adapted by standalone control schemes.

- Implement a simple NCO-tracking control for batch processes with the use of standard control tools, e.g. PID controllers.

- Extent the simple NCO-tracking control for unconstrained and constrained batch processes via advanced self-optimising controllers. Where, in the presence of uncertainty, the optimality loss of the unconstrained problems cancels out the neighbouring-extremal control within the batch. The optimality loss of the constrained problems is handled by the run-to-run controller that updates the variables between the batches. In addition the optimality loss of the constrained problems can be also eliminated by extended NE controller.

- Design the NCO-tracking control scheme that improves the control policy of the constrained problems with lower complexity but with maximum of the performance.

- The intention of the practical part is to apply the proposed control scheme in laboratory conditions and to experimentally verify its performance. The results shall discuss difficulties regarding an implementation, suitability, applicability, and performance of the proposed methods.

Part I

Dynamic Optimisation under Ideal Conditions

Chapter 1

Nominal Optimisation

To comprehend the advanced control approach via real-time dynamic optimisation the basic building blocks have to be recalled, first. This part briefly overviews mathematical background around which is this work built. Particularly, the importance of the chapter dedicated to the optimality conditions is accented throughout the thesis as it provides methodology for real-time optimisation of batch processes.

Within this part, the following topics are reviewed:

- Nominal optimisation problem: objective function, constraints, and dynamic model;
- Nominal solution of the problem by: Optimal Control Theory, Pontryagin's Maximum Principle (PMP), direct and indirect numerical optimisation methods;
- Necessary conditions of optimality.

1.1 Introduction

Dynamic optimisation of processes can generally be characterised as an optimisation of time-varying functions. The goal is to find a continuous time-dependent function that fits best some given criterion. Similarly, this can be expressed as a search for a continuous control function that influences the evolution of a state function in each time instant in such a way that it optimises the objectives. The former refers to calculus of variations and the latter refers to optimal control problems.

In the calculus of variations, the main goal is to choose a set of continuously differentiable functions $\boldsymbol{x}(t)$, $t_0 \leq t \leq t_1$ that optimise the following criterion:

$$\begin{aligned} &\min \int_{t_0}^{t_1} L(\dot{\boldsymbol{x}}(t), \boldsymbol{x}(t), t)\mathrm{d}t \\ \text{s.t.} \quad &\boldsymbol{x}(t_0) = \boldsymbol{x}_0. \end{aligned}$$

It refers to the problem of finding optimal trajectories, arcs and surfaces. These problems came from physics and geometry and they have been posed by Newton, Galileo, Huygens, and by the others. Later, the mathematicians such as Bernoulli, Leibnitz, Euler and Lagrange, founded the solution for such problems.

Optimal control refers to the problem where the evolution of the one set of variables dictates the dynamic behaviour of the other set of variables through the set of ordinary differential equations (ODE) or the set of differential-algebraic equations (DAE). Indeed, in such problems, the unknown variables are separated into the states and controls. It is obvious that current and future states evolve accordingly to chosen control actions. The problem of optimal control is an extension of the calculus of variations and it can be transformed into the comparable problem of choosing a set of continuously differential functions $\boldsymbol{u}(t)$, $t_0 \leq t \leq t_1$ to satisfy the criterion and dynamic constraints:

$$\begin{aligned} &\min \int_{t_0}^{t_1} L(\boldsymbol{u}(t), \boldsymbol{x}(t), t)\mathrm{d}t \\ &\text{s.t.} \quad \dot{\boldsymbol{x}}(t) = F(\boldsymbol{u}(t), \boldsymbol{x}(t), t); \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0. \end{aligned}$$

A precise mathematical formulation of optimal control problems follows in Section 1.2.

A systematic approach to choose the best set of optimal operating conditions that lead to a desired state in optimal manner and with respects the constraints, is as follows:

Step 1 to develop a mathematical model for the process under consideration

Step 2 to develop a mathematical realisation for the objective function that needs to be minimised

Step 3 to run an optimisation in order to compute the best set of operating conditions that minimise the performance index subject to the process constraints

Step 4 to implement the optimal operating condition to the real process

Clearly, optimal control relies heavily on process model. If the mathematical model is accurate enough then the direct implementation of optimal solution steers the real process in optimal way. To reduce the level of uncertainty in models, one tendency is to increase the model complexity so as it predicts better evolution of the real process. The optimisation of large models leads to large-scale optimisation schemes that might be difficult to solve on the one side, on the other side, they require prohibitive amount of work to describe the real dynamic within certain accuracy. If the process dynamic changes or if some process variables varies, the model needs to be adjusted and re-optimised, accordingly. Second tendency is to work with simple models, and to deal with the uncertainty by alternative optimisation approach. This part is devoted to optimisation under ideal circumstances, also known as nominal optimisation. The optimisation under uncertainty is explained in the Part II.

At last, as regular optimal control problems are too complex to be solved analytically, numerical methods provide solution of such problems. Several classes of algorithms are presented in Chapter 3.

1.2 Optimisation Problem Statement

The formulation of an optimal control problem follows Bryson and Ho, 1975; Chachuat, 2007; Troutman, 1995; Víteček and Vítečková, 2002 and is closely presented in the following sections. In particular, the class of admissible controls is addressed in Section 1.2.1, the mathematical representation of batch processes in Section 1.2.2, the objective criterion is specified in Section 1.2.3. Finally, the constraints are stated in Section 1.2.4.

1.2.1 Admissible Controls

Let us assume that process exists and its dynamic behaviour is characterised at any time instant by the state variables. Further, let us assume that this process can be controlled, i.e. the future states of the process are affected by positions of controller. These controller positions are represented by a finite number of points $\boldsymbol{u} = (u_1, u_2, \ldots, u_{n_u}) \in \mathcal{R}^{n_u}, n_u \geq 1$, specifically by the vector of control variables. The values that these variables can encompass are usually restricted to certain control region $\mathcal{U} \in \mathcal{R}^{n_u}$, in the majority of optimal control problems. For example, the control region $\mathcal{U}$ may have a shape of hypercube:

$$u_i^L \leq u_i \leq u_i^R, \quad i = 1, \ldots, n_u. \tag{1.1}$$

The physical meaning is obvious. The bounded control region represents real limitations, e.g. actuator limitations, the fuel in the tank, the volume, voltage, temperature, etc. Throughout the thesis, every function $u(t)$, defined in time domain $[t_0, t_1]$ and normally also defined within the range of control region $\mathcal{U}$, represents a continuous control.

1.2.2 Process Model Equations

In optimal control, the process model plays very important role. It is non-trivial work to obtain the simplest mathematical representation of the process that accurately predicts the physical system reaction to any control from a set of all admissible controls. This work is restricted to the class of problems for which the dynamic constraints involve ordinary differential equations (ODE). The problems in which dynamic behaviour is described by differential-algebraic equations (DAE) are not treated here.

Consequently, a dynamical system is denoted by a set of ordinary differential equations (ODE) of the form:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}(\boldsymbol{x}(t), \boldsymbol{u}(t), t); \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0. \tag{1.2}$$

Here, $\boldsymbol{F} \in \mathcal{R}^{n_x}$ is a smooth vector of differential equations with a priori known initial conditions $\boldsymbol{x}_0 \in \mathcal{R}^{n_x}$. It is assumed that the gradient $\boldsymbol{F}_{\boldsymbol{x}}(t) := \frac{\partial f}{\partial x}(\boldsymbol{x}(t), \boldsymbol{u}(t), t)$ of $\boldsymbol{F}$ exists. The initial conditions are constant and they are not function of any parameters. $t \in \mathcal{R}$ represents the time (independent variable). If time t does not appear explicitly in $\boldsymbol{F}$ then the system is so-called autonomous. The control (also input or manipulated) variables are given by the control vector $\boldsymbol{u}(t) \in \mathcal{U}$. Dynamical behaviour of the process in whole time domain is characterised by vector of state variables $\boldsymbol{x}(t) \in \mathcal{R}^{n_x}$. The response of the system $\boldsymbol{x}(\boldsymbol{x}_0, \boldsymbol{u}(t), t)$ is then given by solving (1.2) for the initial condition $\boldsymbol{x}_0$ and corresponding control signals.

1.2.3 Performance criterion

The performance criterion (also called cost functional, objective, optimisation criterion, or simply cost) is a quantitative measure of system performance. It corresponds, for example, to achieving a desired product quality/quantity at the most economical cost, or to maximising a product yield.

In general, the performance criterion $\mathcal{J} \in [t_0, t_1]$ is mathematically expressed in one of the three forms:

Lagrange form

$$\mathcal{J} := \int_{t_0}^{t_1} L(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\mathrm{d}t \tag{1.3a}$$

Mayer form

$$\mathcal{J} := \phi(\boldsymbol{x}(t_1), t_1) \tag{1.3b}$$

Bolza form

$$\mathcal{J} := \phi(\boldsymbol{x}(t_1), t_1) + \int_{t_0}^{t_1} L(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\mathrm{d}t \tag{1.3c}$$

where

$\mathcal{J} \in \mathcal{R}$ represents the overall cost functional;

$\phi \in \mathcal{R}$ represents the component of the performance criterion at the final time;

$L \in \mathcal{R}$ represents the Lagrangian;

$\boldsymbol{x}(t) \in \mathcal{R}^{n_x}$ represents the vector of state variables;

$\boldsymbol{u}(t) \in \mathcal{R}^{n_u}$ represents the vector of control variables.

It is assumed that the Lagrangian $L(\boldsymbol{x}(t), \boldsymbol{u}(t), t)$ is defined, it is continuous as has continuous partial derivatives $L_{\boldsymbol{x}}$. Moreover, the final time t_1 may be considered as a fixed or free variable in the optimisation problem.

Note that all the three forms of the cost functional are interchangeable and can be derived one from another:

- Initially, Lagrange form can be reduced to Mayer form by introducing a new state variable $x_L(t)$, by augmenting the state vector $\hat{\boldsymbol{x}}(t) := (x_1(t), x_2(t), \ldots, x_n(t), x_L(t))^T$ and by a supplementary differential equation:

  $$\dot{x}_L(t) = L(\boldsymbol{x}(t), \boldsymbol{u}(t), t); \quad x_L(t_0) = 0. \tag{1.4}$$

  Subsequently, the cost functional (1.3a) takes the Mayer form (1.3b) with $\phi(\hat{\boldsymbol{x}}(t_1), t_1) := x_L(t_1)$.

- Next, Mayer form can be reduced to Lagrange form by defining a new state variable $x_L(t)$, the augmented state vector $\hat{\boldsymbol{x}}(t) := (x_1(t), x_2(t), \ldots, x_n(t), x_L(t))^T$, and by an additional differential equation

$$\dot{x}_L(t) = 0; \quad x_L(t_0) = \frac{\phi(\boldsymbol{x}(t_1), t_1)}{t_1 - t_0}. \tag{1.5}$$

  The functional (1.3b) is transformed into Lagrange form (1.3a) with $L(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t), t) := x_L(t)$.

- Finally, the previous two transformations can be used to rewrite Bolza form (1.3c) into the one of Mayer or Lagrange form, respectively. See that Mayer form is a special Bolza form with $L(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t), t) := 0$. Additionally, Bolza form with component $\phi(\hat{\boldsymbol{x}}(t_1), t_1) := 0$ is equal to Lagrange form.

1.2.4 Constraints

The process is usually subject to several constraints given by physical properties that restrict the range of values assumed by both the control and state variables. They results from safety, ecological, or operability requirements, such as limits on temperature, concentration, NO_x and CO_2 emissions, or re-prioritising of production. In optimal control problems usually occur either point-wise or path constraints. Point-wise constraints arise from selectivity or performance requirements. For instance, the concentration of a reactant in the final product at the end of batch may be constrained to be at least as a required value for its marketability or further usage in other reactions. Path constraints normally arise from safety or actuator limitations. Flow rate, temperature, and concentration are typical path constraints as they are limited by physical realisation, e.g. positive values of temperature. All these constraints can be equalities or inequalities.

Point-wise Constraints The common form of these constraints in optimal control problems are terminal constraints, i.e. point-wise constraints defined at final time t_1. An inequality terminal constraint is of the form:

$$\boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \leq \mathbf{0} \tag{1.6}$$

Path Constraints This type of constraints restricts the range of values of control and state variables over the entire time interval $[t_0, t_1]$ or any time subinterval. Path constraint could be of form:

$$\boldsymbol{\rho}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) \leq \mathbf{0}; \quad \text{for } t \in [t_0, t_1] \tag{1.7}$$

In general, a distinction is made between path constraints depending explicitly on the control variables:

$$\boldsymbol{u}^L \leq \boldsymbol{u}(t) \leq \boldsymbol{u}^U; \quad \text{for } t \in [t_0, t_1] \tag{1.8}$$

or depending only on the state variables:

$$\boldsymbol{x}^L \leq \boldsymbol{x}(t) \leq \boldsymbol{x}^U; \quad \text{for } t \in [t_0, t_1] \tag{1.9}$$

In general, the previous conditions are also referred as box constraints, i.e. the control or state variables can take values between its given lower and upper bounds.

Note that only the path constraints for control variables $\boldsymbol{\rho}(\boldsymbol{u}(t), t) \leq \mathbf{0}$ are considered further.

1.2.5 Problem Formulation

Regrouping of previous parts posed from Section 1.2.1 to Section 1.2.4, the optimisation problem becomes:

$$
\begin{aligned}
\min_{\boldsymbol{u}} \mathcal{J} = \phi(\boldsymbol{x}(t_1)) + \int_{t_0}^{t_1} L(\boldsymbol{x}, \boldsymbol{u}) \mathrm{d}t \qquad & (1.10) \\
\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}), \quad t_0 \leq t \leq t_1 \qquad & (1.11) \\
\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \qquad & (1.12) \\
\boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \leq \mathbf{0} \qquad & (1.13) \\
\boldsymbol{\rho}(\boldsymbol{u}(t), t) \leq \mathbf{0}. \qquad & (1.14)
\end{aligned}
$$

Note that the problem of an optimal control (1.14) may not have a solution. Whether this solution exists or not, one may investigate if an admissible control can be found in such a way that it respects the constraints. This solution is then feasible and the system is controllable, i.e. it express the ability of the system to move from any initial state at initial time to any desired state at final time. Indeed, if the solution is not feasible, then an admissible control for the optimisation problem does not exist.

To actually find an optimal control for a considered problem, one needs algebraic conditions that are necessary or sufficient for optimality. These conditions are an instrument to diminish a small number of candidates for an optimal control. In Chapter 2, applications of variational methods are studied in order to obtain necessary and sufficient conditions of optimality for problems with and without constraints.

Chapter 2

Necessary Conditions for Optimality

A set of conditions which have to be necessarily satisfied by any optimal control is closely introduced in this chapter. In many cases, these conditions allow to sort out several control profiles, often even a single control profile. It is clear that if exists an optimal control, there is relevant chance to select one from plenty of candidates.

In the first section, the necessary conditions of optimality are exploited for optimal control problems without restricted control and state variables. More general optimal control problems with additional constraints are considered later, in the second section.

2.1 Problems without Constraints

Let us assume the simplest optimal control problem without path and end-point constraints with fixed initial time t_0 and fixed terminal time t_1 as follows:

$$\min_{\boldsymbol{u}} \mathcal{J} = \int_{t_0}^{t_1} L(\boldsymbol{x}, \boldsymbol{u}) \mathrm{d}t \tag{2.1}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}); \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{2.2}$$

The response $\boldsymbol{x}(t), t_0 \leq t \leq t_1$ to a control function $\boldsymbol{u}(t), t_0 \leq t \leq t_1$ is provided by solving the initial value problem (2.2) with initial condition $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$ and in time domain $[t_0, t_1]$. Remind that for sake of simplicity, further, the time functions (e.g. $\boldsymbol{x}(t)$) will be used without (t) notation after initial definition.

Subsequently, let us assume that there exists an optimal control $\boldsymbol{u}^*$ and that for all other controls $\boldsymbol{u}$ hold:

$$\mathcal{J}[\boldsymbol{u}] \geq \mathcal{J}[\boldsymbol{u}^*] \tag{2.3}$$

In some cases, the optimal control $\boldsymbol{u}^*$ might not exist even though it satisfies condition (2.3). It need to be emphasis that the proof of existence of an optimal control is difficult to find. However, the necessary conditions may single out a non-empty set of candidates, but an optimal control still might not exist for the problem.

In order to derive the conditions that must be necessarily satisfied at the optimal point, it is assumed that the optimal control $\boldsymbol{u}^*$ exists. These conditions follow from small changes in functional (2.1) and from small changes $\delta\boldsymbol{u}$ in optimal control $\boldsymbol{u}^*$. Then, the variation of state variables $\delta\boldsymbol{x}$ are implicitly related to variations of control variables $\delta\boldsymbol{u}$ through the non-linear differential equations (2.2).

The system response related to the control:

$$\boldsymbol{u} = \boldsymbol{u}^* + \delta\boldsymbol{u} \tag{2.4}$$

is then given by

$$\boldsymbol{x} = \boldsymbol{x}^* + \delta\boldsymbol{x}. \tag{2.5}$$

The dynamic constraints (2.2) then vary accordingly

$$\delta\dot{\boldsymbol{x}} = \frac{\partial \boldsymbol{F}^*}{\partial \boldsymbol{x}^*}\delta\boldsymbol{x} + \frac{\partial \boldsymbol{F}^*}{\partial \boldsymbol{u}^*}\delta\boldsymbol{u}. \tag{2.6}$$

Note that superscript $*$ indicates that the corresponding quantity is evaluated along the extremal path $\boldsymbol{u}^*, t_0 \leq t \leq t_1$, and corresponding states $\boldsymbol{x}^*, t_0 \leq t \leq t_1$.

These variations cause variation of objective functional $\delta\mathcal{J}$. If this variation is continuous and variations of controls are unbounded (i.e. $\delta\boldsymbol{u}$ has a real positive or negative values) then $\delta\mathcal{J} := 0$ is necessary condition for the extreme. The variation of the objective functional can be also expressed as:

$$\delta\mathcal{J} = \int_{t_0}^{t_1} \left[\left(\frac{\partial L}{\partial \boldsymbol{x}}\right)^T \delta\boldsymbol{x} + \left(\frac{\partial L}{\partial \boldsymbol{u}}\right)^T \delta\boldsymbol{u} \right] \mathrm{d}t. \tag{2.7}$$

Next, let us augment the functional variation $\delta\mathcal{J}$ by encompassing the dynamic constraints via a vector of non-determined adjoint variables $\boldsymbol{\lambda} \in \mathcal{R}^{n_x}$. At first, let us define the adjoint variable $\boldsymbol{\lambda}(t)$ and to rewrite (2.6) in the form:

$$\boldsymbol{\lambda}^T\delta\dot{\boldsymbol{x}} = \boldsymbol{\lambda}^T\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\delta\boldsymbol{x} + \boldsymbol{\lambda}^T\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}\delta\boldsymbol{u}. \tag{2.8}$$

Consequent integration of (2.8) within the interval $[t_0, t_1]$ results in:

$$\int_{t_0}^{t_1} \left[\boldsymbol{\lambda}^T\delta\dot{\boldsymbol{x}} + \boldsymbol{\lambda}^T\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\delta\boldsymbol{x} + \boldsymbol{\lambda}^T\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}\delta\boldsymbol{u} \right] \mathrm{d}t = 0. \tag{2.9}$$

Combination of (2.9) and (2.7) thus provides another expression for $\delta\mathcal{J}$:

$$\begin{aligned}\delta\mathcal{J} = &\int_{t_0}^{t_1} \left[\left(\frac{\partial L}{\partial \boldsymbol{x}} + \boldsymbol{\lambda}^T\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}\right)\delta\boldsymbol{x} + \left(\frac{\partial L}{\partial \boldsymbol{u}} + \boldsymbol{\lambda}^T\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}\right)\delta\boldsymbol{u} \right] \mathrm{d}t - \\ &- \int_{t_0}^{t_1} \left(\boldsymbol{\lambda}^T\delta\dot{\boldsymbol{x}}\right)\mathrm{d}t.\end{aligned} \tag{2.10}$$

Integral $\int_{t_0}^{t_1}\left(\boldsymbol{\lambda}^T\delta\dot{\boldsymbol{x}}\right)\mathrm{d}t$ by per-partes gives $\left[\boldsymbol{\lambda}^T\delta\boldsymbol{x}\right]_{t_0}^{t_1} - \int_{t_0}^{t_1}\left(\dot{\boldsymbol{\lambda}}^T\delta\boldsymbol{x}\right)\mathrm{d}t$, so that (2.10) becomes:

$$\begin{aligned}\delta\mathcal{J} = &\int_{t_0}^{t_1} \left[\left(\frac{\partial L}{\partial \boldsymbol{x}} + \boldsymbol{\lambda}^T\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} + \dot{\boldsymbol{\lambda}}^T\right)\delta\boldsymbol{x} + \left(\frac{\partial L}{\partial \boldsymbol{u}} + \boldsymbol{\lambda}^T\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}\right)\delta\boldsymbol{u} \right] \mathrm{d}t + \\ &+ \boldsymbol{\lambda}^T\delta\boldsymbol{x}\Big|_{t=t_0} - \boldsymbol{\lambda}^T\delta\boldsymbol{x}\Big|_{t=t_1}.\end{aligned} \tag{2.11}$$

Further, it is more convenient to define the Hamiltonian $\mathcal{H}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}, t)$:

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{u}, t) = L(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda}^T \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}). \tag{2.12}$$

Incorporating Hamiltonian into (2.11), the variation of cost function is following:

$$\delta \mathcal{J} = \int_{t_0}^{t_1} \left[\left(\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}} + \dot{\boldsymbol{\lambda}}^T \right) \delta \boldsymbol{x} + \left(\frac{\partial \mathcal{H}}{\partial \boldsymbol{u}} \right) \delta \boldsymbol{u} \right] \mathrm{d}t + \boldsymbol{\lambda}^T \delta \boldsymbol{x} \bigg|_{t=t_0} - \boldsymbol{\lambda}^T \delta \boldsymbol{x} \bigg|_{t=t_1}. \tag{2.13}$$

NCO directly implies from (2.13). It is required that the first-time variation of the objective functional $\delta \mathcal{J}$ is zero at optimum. Therefore, all terms in brackets need to be zero.

The first bracket becomes zero when $\boldsymbol{\lambda}(t)$ is a result of the following ODE:

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}} \tag{2.14}$$

and the last term is zero if

$$\boldsymbol{\lambda}(t_1) = \boldsymbol{0}, \tag{2.15}$$

Subsequently, the necessary condition for optimal control for zero initial variation of state vector $\delta \boldsymbol{x}(t_0) = \boldsymbol{0}$ must satisfy:

$$\delta \mathcal{J} = \int_{t_0}^{t_1} \left[\left(\frac{\partial \mathcal{H}}{\partial \boldsymbol{u}} \right) \delta \boldsymbol{u} \right] \mathrm{d}t = 0. \tag{2.16}$$

Finally, the necessary conditions for optimality of unconstrained problems can be summarised:

- the necessity for the optimal states is obtained by differentiation of (2.12):

$$\dot{\boldsymbol{x}} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}}; \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}), \quad t_0 \leq t \leq t_1 \tag{2.17a}$$

- the necessity for the optimal controls must satisfy (2.16):

$$\boldsymbol{0} = \frac{\partial \mathcal{H}}{\partial \boldsymbol{u}}; \quad \boldsymbol{0} = \frac{\partial L}{\partial \boldsymbol{u}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}}, \quad t_0 \leq t \leq t_1 \tag{2.17b}$$

- the necessity for the optimal adjoints reads:
  - the adjoint variables:

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}}; \quad \dot{\boldsymbol{\lambda}} = -\frac{\partial L}{\partial \boldsymbol{x}} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}, \quad t_0 \leq t \leq t_1 \tag{2.17c}$$

  - terminal condition for adjoint variables:

$$\boldsymbol{\lambda}(t_1) = \boldsymbol{0} \tag{2.17d}$$

In the calculus of variations, the equations (2.17a), (2.17b), (2.17c), (2.17d) are also known as the *Euler-Lagrange equations*. Note that the triple $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{u}^*)$ gives a local minimum of $\mathcal{J}$ if $\boldsymbol{u}^*(t)$ is a stationary point of the Hamiltonian function evaluated at $\boldsymbol{x}^*(t)$ and $\boldsymbol{\lambda}^*(t)$, at each $t \in [t_0, t_1]$.

These equations also give necessary conditions both for a minimisation and for a maximisation problem. In a minimisation problem, $\boldsymbol{u}^*(t)$ minimises $\mathcal{H}(\boldsymbol{x}^*(t), \boldsymbol{\lambda}^*(t), \boldsymbol{u}(t), t)$, that the additional necessary condition $\mathcal{H}_{\boldsymbol{uu}}(\boldsymbol{x}^*(t), \boldsymbol{\lambda}^*(t), \boldsymbol{u}^*(t), t) \geq 0$ also holds. On the other side, in a maximisation problem, $\boldsymbol{u}^*(t)$ maximises $\mathcal{H}(\boldsymbol{x}^*(t), \boldsymbol{\lambda}^*(t), \boldsymbol{u}(t), t)$ in that way that $\mathcal{H}_{\boldsymbol{uu}}(\boldsymbol{x}^*(t), \boldsymbol{\lambda}^*(t), \boldsymbol{u}^*(t), t) \leq 0$ holds as well. The positive definite matrix $\mathcal{H}_{\boldsymbol{uu}}$ denotes the second partial derivative of Hamiltonian $\mathcal{H}$ with respect to the control variables $(\partial^2 \mathcal{H} / \partial \boldsymbol{u}^2)$.

2.2 Problems with Constraints

In the previous section, necessary conditions for unconstrained problems with fixed final time were presented. However, common and real-world optimal control problems feature end-point state conditions and also the control actions are limited on the whole time interval $t \in [t_0, t_1]$.

2.2.1 Terminal Constraints

In this section the problems with end-point constraints of the form $\boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \leq \boldsymbol{0}$ and with specified terminal time t_1 are considered. Besides end-point constraints, the terminal cost $\phi(\boldsymbol{x}(t_1))$ is considered to augment the original cost functional. In overall the problem reads:

$$\min_{\boldsymbol{u}} \mathcal{J} = \phi(\boldsymbol{x}(t_1)) + \int_{t_0}^{t_1} L(\boldsymbol{x}, \boldsymbol{u}) \mathrm{d}t \tag{2.18}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}); \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{2.19}$$

$$\boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \leq \boldsymbol{0}. \tag{2.20}$$

To obtain necessary conditions of optimality for such problem, one needs to add the terminal constraints into the cost functional via vector of Lagrange multipliers $\boldsymbol{\nu} \in \mathcal{R}^{n_\psi}$, with n_ψ as a number of constraints, as follows:

$$\mathcal{J} = \Phi(\boldsymbol{x}(t_1), \boldsymbol{\nu}, t_1) + \int_{t_0}^{t_1} \left[L(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\lambda}^T \left(\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) - \dot{\boldsymbol{x}} \right) \right] \mathrm{d}t \tag{2.21}$$

where

$$\Phi(\boldsymbol{x}(t_1), \boldsymbol{\nu}, t_1) = \phi(\boldsymbol{x}(t_1)) + \boldsymbol{\nu}^T (\boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1)). \tag{2.22}$$

Similarly as in previous section, the first-time variation of the augmented cost functional (2.21) equal to zero gives necessary condition for extreme:

$$\begin{aligned}\delta\mathcal{J} &= \left(\frac{\partial\Phi}{\partial\boldsymbol{x}} - \boldsymbol{\lambda}\right)\bigg|_{t=t_1} \delta\boldsymbol{x}\bigg|_{t_1} \\ &+ \int_{t_0}^{t_1} \left[\left(\frac{\partial L}{\partial\boldsymbol{x}} + \boldsymbol{\lambda}^T\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{x}} + \dot{\boldsymbol{\lambda}}\right)\delta\boldsymbol{x} + \left(\frac{\partial L}{\partial\boldsymbol{u}} + \boldsymbol{\lambda}^T\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{u}}\right)\delta\boldsymbol{u}\right]\mathrm{d}t \\ &= \left(\frac{\partial\Phi}{\partial\boldsymbol{x}} - \boldsymbol{\lambda}\right)\bigg|_{t=t_1} \delta\boldsymbol{x}\bigg|_{t_1} + \int_{t_0}^{t_1} \left[\left(\frac{\partial\mathcal{H}}{\partial\boldsymbol{x}} + \dot{\boldsymbol{\lambda}}\right)\delta\boldsymbol{x} + \left(\frac{\partial\mathcal{H}}{\partial\boldsymbol{u}}\right)\delta\boldsymbol{u}\right]\mathrm{d}t \end{aligned} \tag{2.23}$$

from which results following NCOs:

- the necessity for the optimal states:

$$\dot{\boldsymbol{x}} = \frac{\partial\mathcal{H}}{\partial\boldsymbol{\lambda}}; \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x},\boldsymbol{u}), \quad t_0 \le t \le t_1 \tag{2.24}$$

- the necessity for the optimal controls:

$$\boldsymbol{0} = \frac{\partial\mathcal{H}}{\partial\boldsymbol{u}}; \quad \boldsymbol{0} = \frac{\partial L}{\partial\boldsymbol{u}} + \boldsymbol{\lambda}^T\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{u}}, \quad t_0 \le t \le t_1 \tag{2.25}$$

- the necessity for the optimal adjoints reads:

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial\mathcal{H}}{\partial\boldsymbol{x}}; \quad \dot{\boldsymbol{\lambda}} = -\frac{\partial L}{\partial\boldsymbol{x}} - \boldsymbol{\lambda}^T\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{x}}, \quad t_0 \le t \le t_1 \tag{2.26}$$

  with terminal condition:

$$\boldsymbol{\lambda}(t_1) = \frac{\partial\Phi}{\partial\boldsymbol{x}}\bigg|_{t_1}; \quad \boldsymbol{\lambda}(t_1) = \left(\frac{\partial\phi}{\partial\boldsymbol{x}} + \boldsymbol{\nu}^T\frac{\partial\boldsymbol{\psi}}{\partial\boldsymbol{x}}\right)\bigg|_{t_1} \tag{2.27}$$

  and conditions for Lagrange multipliers (KKT conditions):

$$0 = \nu_k\psi_k, \quad k = 1,\ldots,n_\psi \tag{2.28}$$

$$0 \le \nu_k, \tag{2.29}$$

  where Lagrange multiplier $\nu_k = 0, k \in \{1,...,n_\psi\}$ denotes inactive terminal constraint and $\nu_k > 0, k \in \{1,...,n_\psi\}$ denotes active terminal constraint.

2.2.2 Path Constraints

Finally, in this section the last class of problems is considered. It is very common that some real-world problems require to handle physical limits of the input actions while still demanding certain state values at terminal time. The following problem is considered:

$$\min_{\boldsymbol{u}} \mathcal{J} = \phi(\boldsymbol{x}(t_1)) + \int_{t_0}^{t_1} L(\boldsymbol{x},\boldsymbol{u})\mathrm{d}t \tag{2.30}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x},\boldsymbol{u}); \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{2.31}$$

$$\boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \le \boldsymbol{0} \tag{2.32}$$

$$\boldsymbol{\rho}(\boldsymbol{u}(t), t) \le \boldsymbol{0}; \quad t_0 \le t \le t_1. \tag{2.33}$$

The path control constraints can be expressed in a form of functional:

$$\varphi_j(\boldsymbol{u}(t),t) = \int_{t_0}^{t_1} \rho_j(\boldsymbol{u}(t),t)\mathrm{d}t \tag{2.34}$$

where $j = 1,\ldots,n_\rho$, and n_ρ is a number of path constraints. Next, they can be adjoined to an original cost function through a vector of Lagrange multipliers $\boldsymbol{\mu} \in \mathcal{R}^{n_\rho}$ to form an augmented Lagrange cost:

$$\overline{\mathcal{J}} = \mathcal{J} + \sum_{j=1}^{n_\rho} \mu_j^T \varphi_j, \quad j = 1,\ldots,n_\rho. \tag{2.35}$$

what equals to following:

$$\begin{aligned}\overline{\mathcal{J}} =& \Phi(\boldsymbol{x}(t_1),\boldsymbol{\nu},t_1)+\\ &+\int_{t_0}^{t_1} \left[L(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{\lambda}^T\left(\boldsymbol{F}(\boldsymbol{x},\boldsymbol{u}) - \dot{\boldsymbol{x}}\right)\right]\mathrm{d}t+\\ &+\sum_{j=1}^{n_\rho} \mu_j^T \int_{t_0}^{t_1} \rho_j(\boldsymbol{u}(t),t)\mathrm{d}t\end{aligned} \tag{2.36}$$

with:

$$\Phi(\boldsymbol{x}(t_1),\boldsymbol{\nu},t_1) = \phi(\boldsymbol{x}(t_1)) + \boldsymbol{\nu}^T(\boldsymbol{\psi}(\boldsymbol{x}(t_1),t_1)). \tag{2.37}$$

Accordingly, Hamiltonian reads:

$$\overline{\mathcal{H}}(\boldsymbol{x},\boldsymbol{\lambda},\boldsymbol{u},\boldsymbol{\mu},t) = L(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{\lambda}^T\boldsymbol{F}(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{\mu}^T\boldsymbol{\rho}(\boldsymbol{u}(t),t) \tag{2.38}$$

Next, the necessary conditions of optimality can be derived analogically to previous problem, i.e. from the first-time variation of cost functional $\delta\overline{\mathcal{J}}$:

$$\begin{aligned}\delta\overline{\mathcal{J}} =& \left(\frac{\partial\Phi}{\partial\boldsymbol{x}} - \boldsymbol{\lambda}\right)\Bigg|_{t=t_1} \delta\boldsymbol{x}\Bigg|_{t_1} +\\ &+\int_{t_0}^{t_1}\left[\left(\frac{\partial L}{\partial\boldsymbol{x}} + \boldsymbol{\lambda}^T\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{x}} + \dot{\boldsymbol{\lambda}}\right)\delta\boldsymbol{x} + \left(\frac{\partial L}{\partial\boldsymbol{u}} + \boldsymbol{\lambda}^T\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{u}} + \boldsymbol{\mu}^T\frac{\partial\boldsymbol{\rho}}{\partial\boldsymbol{u}}\right)\delta\boldsymbol{u}\right]\mathrm{d}t\\ =& \left(\frac{\partial\Phi}{\partial\boldsymbol{x}} - \boldsymbol{\lambda}\right)\Bigg|_{t=t_1} \delta\boldsymbol{x}\Bigg|_{t_1}\\ &+\int_{t_0}^{t_1}\left[\left(\frac{\partial\overline{\mathcal{H}}}{\partial\boldsymbol{x}} + \dot{\boldsymbol{\lambda}}\right)\delta\boldsymbol{x} + \left(\frac{\partial\overline{\mathcal{H}}}{\partial\boldsymbol{u}}\right)\delta\boldsymbol{u}\right]\mathrm{d}t.\end{aligned} \tag{2.39}$$

Resulting NCOs then are following:

- the necessity for the optimal states:

$$\dot{\boldsymbol{x}} = \frac{\partial\overline{\mathcal{H}}}{\partial\boldsymbol{\lambda}}; \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x},\boldsymbol{u}), \quad t_0 \le t \le t_1 \tag{2.40}$$

- the necessity for the optimal controls:

$$\mathbf{0} = \frac{\partial \overline{\mathcal{H}}}{\partial \boldsymbol{u}}; \quad \mathbf{0} = \frac{\partial L}{\partial \boldsymbol{u}} + \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{u}}, \quad t_0 \leq t \leq t_1 \tag{2.41}$$

- the necessity for the optimal adjoints reads:

$$\dot{\boldsymbol{\lambda}} = -\frac{\partial \overline{\mathcal{H}}}{\partial \boldsymbol{x}}; \quad \dot{\boldsymbol{\lambda}} = -\frac{\partial L}{\partial \boldsymbol{x}} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}}, \quad t_0 \leq t \leq t_1 \tag{2.42}$$

  with terminal condition:

$$\boldsymbol{\lambda}(t_1) = \left.\frac{\partial \Phi}{\partial \boldsymbol{x}}\right|_{t_1}; \quad \boldsymbol{\lambda}(t_1) = \left.\left(\frac{\partial \phi}{\partial \boldsymbol{x}} + \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\right)\right|_{t_1} \tag{2.43}$$

  and conditions for Lagrange multipliers (KKT conditions):

$$0 = \mu_j \rho_j, \quad j = 1, \ldots, n_\rho \tag{2.44}$$

$$0 = \nu_k \psi_k, \quad k = 1, \ldots, n_\psi \tag{2.45}$$

$$0 \leq \mu_j, \tag{2.46}$$

$$0 \leq \nu_k. \tag{2.47}$$

Note that each Lagrange multiplier may have a zero value $\mu_j = 0,\ j \in \{1, ..., n_\rho\}$ along an interior arc, or a non-zero value $\mu_j > 0,\ j \in \{1, ..., n_\rho\}$ along a boundary arc. Also note that Lagrange multiplier $\nu_k = 0, k \in \{1, ..., n_\psi\}$ denotes inactive terminal constraint and $\nu_k < 0, k \in \{1, ..., n_\psi\}$ denotes active terminal constraint.

Chapter 3

Numerical Optimisation Methods

Numerical optimisation methods for dynamic problems (Srinivasan, Bonvin, et al., 2003a; Vemuri, 2004) address two major issues:

Solution of system equations – an initial value problem (IVP) can be solved by numeric integration, or discretised at multiple time instants and converted to regular non-linear programming (NLP).

Formulation of optimisation strategy – problem of constrained dynamic optimisation is transformed into static NLP optimisation formulation which can be solved via common NLP strategies; in addition, the gradients for NLP solver are obtained by computing sensitivity or adjoint variables.

Most methods differ from the one to the other in the approach to address these two issues. Two main methods can be distinguished based on solving strategy:

1. Direct optimisation methods

2. Indirect optimisation methods

These methods are discussed further.

3.1 Direct Optimisation Methods

In this class of methods, inputs are parametrised by a finite set of variables. The resulting system is then coupled with optimisation algorithms such as Successive Quadratic Programming (Agrawal and Fabien, 1999; Edgar, Soderstrom, et al., 2001; Jacoby, Kowalik, et al., 1972; Ray and Szekely, 1973; Reklaitis, Ravindran, et al., 1983) to find the optimal set of variables. Depending on a way used to solve IVP, two unique approaches have been reported in the literature: the sequential approach and the simultaneous approach, respectively. Both methods are explained in the two following sections.

3.1.1 Sequential Approach

Sequential approach is also referred to Control Vector Parametrisation in literature (Edgar and Himmelblau, 1988; Guntern, Keller, et al., 1998; Ray, 1981) and can be found in a variety of chemical process applications (Ishikawa, Natori, et al., 1997; Mujtaba and Macchietto, 1997; Sorensen, Macchietto, et al., 1996; Vassiliadis, Sargent, et al., 1994a,b). The main idea behind it is to parametrise the continuous controls using a finite set of decision variables. Typically, a piece-wise constant approximation over equally spaced time intervals is chosen for the inputs (Vassiliadis, Sargent, et al., 1994a,b). Consequently, the general NLP solver iteratively optimises an objective function by a choosing the control variables and by respecting algebraic constraints. The sequential method is of the feasible path type method, i.e. in every iteration, the solution of IVP problem remains feasible while optimising performance index. This leads to a robust solution procedure if feasible initial conditions for all variables are provided.

The gradient for the cost function and for constraints with respect to the all optimised variable is estimated by the one of the following two: (i) by sensitivity equations of the system which are integrated together with the process equations, or (ii) by adjoint variables that have to be integrated backwards. The sensitivity equations are found by differentiating of right sides of the process $\boldsymbol{F}$ with respect to the time invariant parameters and variables from discretised inputs (Dovi and Reverberi, 1993; Storen and Hertzberg, 1995). The obvious advantage of sequential approach is that it leads to a very efficient computation of the gradient. The gradient computed by adjoint variables is less accurate over gradient directly expressed by equations because the states are approximated during backward integration. In opposite, with increasing number of discretised intervals, the advantage lies on the side of adjoints because of computing advantage over sensitivity equations: (i) in case of sensitivity equations, large number of ODEs needs to be solved as every discretised interval adds one differential equation to process ODEs; (ii) in case of adjoints, the number of integrated differential equations does not depend on number of discretised intervals, simply it is two times process equations plus an additional equation per another optimised variable and per the constraint, as reported in (Dovi and Reverberi, 1993). Several efficient optimisation algorithms for sequential methods can be found in (Biegler and Hughes, 1983; Lau and Ulrichson, 1992; Storen and Hertzberg, 1995). Majority of algorithms update cost function according to the gradients and reformulate the original problem into a quadratic program. The general algorithm for sequential approach can be described as follows:

Problem:

$$
\begin{aligned}
&\min_{\boldsymbol{u}} \mathcal{J}(\boldsymbol{x}, \boldsymbol{u}) \\
&\quad \text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}), \quad t_0 \leq t \leq t_1 \\
&\qquad\qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \\
&\qquad\qquad \boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \leq \boldsymbol{0} \\
&\qquad\qquad \boldsymbol{\rho}(\boldsymbol{u}(t), t) \leq \boldsymbol{0}
\end{aligned}
\tag{3.1}
$$

On one or more intervals:

Step 1: Parametrise the inputs using a finite number of decision variables, i.e. piece-wise using any standard collocation method (e.g. orthogonal collocation with Legendre roots) or piece-wise constant

$$\boldsymbol{u}_K(t) = \sum_{i=1}^{k} = \boldsymbol{u}_i \chi_i(t) \quad \text{where} \quad \chi_i(t) = \prod_{k=1,j}^{K} \frac{(t - t_k)}{(t_i - t_k)}, \quad \boldsymbol{u}_K(t_i) = \boldsymbol{u}_i \tag{3.2}$$

Step 2: Substitute the parametrised inputs from step 1 in process model (3.1)

$$\begin{aligned} &\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}_K, t) \\ &i = 1, \ldots, K \\ &\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \end{aligned} \tag{3.3}$$

Step 3: Substitute the modified dynamic model given by (3.3) into the problem described by (3.1). The updated problem statement of the sequential method reads:

$$\begin{aligned} &\min_{\boldsymbol{u}_K} \mathcal{J}(\boldsymbol{x}, \boldsymbol{u}_K) \\ \text{s.t.} \quad &\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}_K, t), \quad t_0 \leq t \leq t_1 \\ &\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \\ &\boldsymbol{u}_K = \sum_{i=1}^{k} = \boldsymbol{u}_i \chi_i \quad \text{where} \quad \chi_i = \prod_{k=1,j}^{K} \frac{(t - t_k)}{(t_i - t_k)}, \quad \boldsymbol{u}_K(t_i) = \boldsymbol{u}_i \\ &\boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \leq \boldsymbol{0} \\ &\boldsymbol{\rho}(\boldsymbol{u}_K(t_i), t_i) \leq \boldsymbol{0} \end{aligned} \tag{3.4}$$

Over time horizon:

Step 4: Choose t_i using orthogonal collocation method and evaluate $\boldsymbol{u}_K$ given by equation (3.2) at times t_i.

Step 5: Choose an initial guess for decision variables $\boldsymbol{u}_K$ on one or more time intervals and solve IVP, given by (3.3) for the inputs obtained from (3.2) on each interval, by employing any ODE solver (Gill, Murray, et al., 1981) (e.g. Runge-Kutta or Newton-Raphson Methods).

Step 6: Evaluate the objective function (3.4) using state and control profiles obtained in Step 5. Then update the values of decision variables $\boldsymbol{u}_K$ via any standard optimisation routine, e.g. a steepest descent or Quasi-Newton methods (Gill, Murray, et al., 1981). Repeat Step 4 through Step 6 until the convergence is satisfied. Update the decision variables according to gradients.

While the sequential approach is straightforward to implement, it tends to have slow convergence rate due the fact that the feasible path methods require to solve repeatedly and extensively a set of the differential equations. The most computationally intensive part of this approach is the step where ODEs are solved with higher accuracy, even though the decision variables are far from the optimal solution. Furthermore, the quality of the solution strongly depends on quality of parametrisation of the controls. If a feasible solution is provided as a starting guess, the solution converges quickly. However, it is a non-trivial problem to find such feasible initial guess.

3.1.2 Simultaneous Approach

Although sequential methods guarantee an optimal solution by following a feasible path, in opposite, they can be prohibitively expensive because they tend to converge slowly and require solution of IVP at each iteration. In the simultaneous approach, the state profiles are approximated in addition to the control profiles, thus the dynamic optimisation problem becomes a pure NLP problem expressed by a set of algebraic equations. This NLP problem then simultaneously converges to the optimum even from infeasible starting guess. The idea of orthogonal collocation coupled with Quasi-Newton method, introduced by Hertzberg and Asbjornsen in (Hertzberg and Asbjornsen, 1977), was used to perform simultaneous parameter estimation and integration of a non-linear system dynamics. Sensitivity equations for the dependent variables with respect to the parameters along system dynamics were also replaced by an approximated set of algebraic equations. So, the optimisation was performed in the subspace of the parameters. A low order polynomial approximation was found to give good accuracy while keeping the dimension of the NLP low. This method proved to be superior in computational efficiency to the other existing parameter estimation algorithms.

A similar approach is discussed by Biegler (Biegler, 1984), in which orthogonal collocation is applied to the system of differential equations, too. The control and state profiles are transformed into a set of algebraic equations. Then, the optimisation strategy solves the transformed problem. The main improvement of the Biegler's simultaneous algorithm over Hertzberg and Asbjornsen algorithm is in different approximation of the time varying independent variables. Biegler in (Biegler, 1984) approximated them by the Lagrange polynomials instead of the constant independent variables used by Hertzberg and Asbjornsen in (Hertzberg and Asbjornsen, 1977).

A generalisation of these two collocation methods is presented in (Renfro, Morshedi, et al., 1987). The major difference in this approach is the application of collocation procedure to convert the ODEs into an approximating set of algebraic equations. Global spline collocation (Villadsen and Michaelsen, 1978) is chosen over ordinary global collocation because of its ability to solve a wider spectrum of problems, e.g. stiff systems and boundary value problems. This method is also labelled as collocation on finite elements (Carey and B.A. 1975; Finlayson, 1980). The continuous independent variables are specified as piecewise constant functions. Algorithm can specify the number and location of spline points. This makes (Renfro, Morshedi, et al., 1987) algorithm slightly more complex and dimensionally larger then Biegler's method.

The simultaneous algorithms introduces an approximation of dynamic system equations in order to explicitly avoid of integration process. Hence, the optimisation is carried out in the full space of approximated inputs and states. In general, ODEs are satisfied only at the solution of optimisation problem (Vassiliadis, Sargent, et al., 1994a). So, this method is called the infeasible path approach. The approach can be found in several batch applications (Cuthrell and Biegler, 1989; Eaton and Rawlings, 1990; Logsdon and Biegler, 1989; Ruppen, Benthack, et al., 1995).

Very general problems, such as boundary value problems or state constrained problems can be handled by simultaneous methodology. This versatility combined with the super linear convergence is the strength of this methodology. The weaknesses are the large problem size resulting from approximation on finite elements, and the need for efficient non-linear large scale optimisation algorithms. With the development of Successive Quadratic Programming (SQP), reduced space SQP, interior-point approach and the conjugate gradients methods, the NLPs resulting from the simultaneous method can be solved efficiently (Biegler, 1984; Biegler, Cervantes, et al., 2002; Cervantes and Biegler, 1998; Renfro, Morshedi, et al., 1987; Srinivasan, Bonvin, et al., 2003a).

This numerical approach has been implemented and developed as part of this work. We have improved problem reformulations and convergence. Graphical user interface has been developed in order to simplify the problem definition (Podmajerský, Čižniar, et al., 2007). As the NLPs usually exhibit more local optima, the global optimisation frameworks, i.e. multistart and αBB approach have been implemented to improve the performance and the feasibility. The results were published in (Čižniar, Podmajerský, et al., 2009). In addition, NMPC framework based on global optimisation that employs simultaneous approach has been implemented and benchmarked in (Čižniar, Hirmajer, et al., 2008).

The basic procedure of simultaneous methods is as follows (Cuthrell and Biegler, 1989; Srinivasan, Bonvin, et al., 2003a):

Problem:

$$\begin{aligned} &\min_{\boldsymbol{u}} \mathcal{J}(\boldsymbol{x},\boldsymbol{u}) \\ &\qquad \text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x},\boldsymbol{u}), \quad t_0 \le t \le t_1 \\ &\qquad\qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \\ &\qquad\qquad \boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \le \boldsymbol{0} \\ &\qquad\qquad \boldsymbol{\rho}(\boldsymbol{u}, t) \le \boldsymbol{0} \end{aligned} \tag{3.5}$$

On one or more intervals:

Step 1: Approximate the process states and inputs using any standard collocation method (e.g. orthogonal collocation with n_{cp} collocation points on finite number of elements/intervals n_i). On an interval i with times $t \in [t_i, t_{i+1}]$, the states and the controls are approximated accordingly:

$$\begin{aligned} &\boldsymbol{x}_{K_x}(t) = \sum_{j=0}^{K_x} \boldsymbol{x}_{ij}\xi_j(t) \quad \text{where} \quad \xi_j(t) = \prod_{k=0,j}^{K_x} \frac{(t-t_{ik})}{(t_{ij}-t_{ik})}, \quad \boldsymbol{x}_{K_x}(t_{ij}) = \boldsymbol{x}_{ij} \\ &\boldsymbol{u}_{K_u}(t) = \sum_{j=1}^{K_u} \boldsymbol{u}_{ij}\chi_j(t) \quad \text{where} \quad \chi_j(t) = \prod_{k=1,j}^{K_u} \frac{(t-t_{ik})}{(t_{ij}-t_{ik})}, \quad \boldsymbol{u}_{K_u}(t_{ij}) = \boldsymbol{u}_{ij} \\ &\text{for} \quad i = 1, \dots n_i. \end{aligned} \tag{3.6}$$

where $\boldsymbol{x}_{K_x}(t)$ represents the polynomial of degree (K_x+1), and $\boldsymbol{u}_{K_u}(t)$ represents the polynomial of degree K_u.

Step 2: Substitute approximated states and inputs into process equations and obtain the algebraic expression for residuals.

$$\begin{aligned}
&\boldsymbol{R}(t_{ik}) = \sum_{j=0}^{K_x} \boldsymbol{x}_{ij}\dot{\xi}_j(t_i) - \boldsymbol{F}(\boldsymbol{x}_{K_x}(t_{ik}), \boldsymbol{u}_K(t_{ik}), t_{ik}) \\
&j = k = 1, \dots, K_x, \quad i = 1, \dots n_i \\
&\text{with} \quad \boldsymbol{x}_{K_x}(t_0) = \boldsymbol{x}_0
\end{aligned} \tag{3.7}$$

Step 3: Substitute the approximated dynamic model into the problem.

$$\begin{aligned}
\min_{\boldsymbol{x}_{K_x}(t_{ik}), \boldsymbol{u}_{K_u}(t_{ik})} \quad & \mathcal{J}(\boldsymbol{x}(t_1), \boldsymbol{u}(t_1)) \\
\text{s.t.} \quad & \boldsymbol{R}(t_{ik}) = \sum_{j=0}^{K_x} \boldsymbol{x}_{ij}\dot{\xi}_j(t_i) - \boldsymbol{F}(\boldsymbol{x}_{K_x}(t_{ik}), \boldsymbol{u}_K(t_{ik}), t_{ik}) \\
& \boldsymbol{x}_{K_x}(t_0) = \boldsymbol{x}_0 \\
& \boldsymbol{x}_{K_x}(t_i) = \boldsymbol{x}_{K_x}(t_i) \\
& \boldsymbol{\psi}(\boldsymbol{x}_{K_x}(t_1), t_1) \leq \mathbf{0} \\
& \boldsymbol{\rho}(\boldsymbol{u}_{K_u}(t_i), t_i) \leq \mathbf{0} \\
& j = k = 1, \dots, K_x, \quad i = 1, \dots n_i.
\end{aligned} \tag{3.8}$$

Over time horizon:

Step 4: Choose collocation points t_{ij} in each interval i using orthogonal collocation method (e.g. Legendre collocation, Radau collocation) and evaluate $\boldsymbol{u}_{K_u}$ and $\boldsymbol{x}_{K_x}$ given by equation (3.6) at times t_{ik}.

Step 5: Solve problem given by (3.8) in Step 3 at the t_{ik} chosen in Step 4 using any non-linear programming solver available (e.g. Successive Quadratic Programming).

The main advantage of the simultaneous method is that optimal solution can be found even by starting from a very poor initial guess. On the other hand, the main disadvantage of the method is that the resulting non-linear problem becomes more complicated with rising number of discretisation points, and it may have many local minima. Furthermore, the use of simultaneous methods requires an awareness of the trade-off between approximation and optimisation (Srinivasan, Myszkorowski, et al., 1995). It can turn out that a less accurate approximation of the problem results in better cost. On the other hand, such solution can correspond to an inadequate problem approximation. Then, the approximation can be improved either by introducing accuracy as an additional constraint or by increasing the number of collocation points. Especially, stiff problems require a fine grid of approximation points what results in a larger number of decision variables (Terwiesch, Agarwal, et al., 1994; Villadsen and Michaelsen, 1978).

3.1.3 Other Direct Approaches

In addition to sequential and simultaneous methods alone, several other direct methods are available in literature (Vemuri, 2004). The direct shooting method (Bock and Platt, 1984) is a hybrid methodology between the sequential and simultaneous methods. In this methodology, the total time is divided into several stages. Except for the first stage, the initial conditions of the stages are considered as decision variables. Then is required that the initial states of each stage should match the final ones of the proceeding stage. This procedure belongs to an infeasible path methods as simultaneous methods while the integration is accurate as in sequential methods (Srinivasan, Bonvin, et al., 2003a). Extensions of the direct multiple shooting methods to differential-algebraic systems are described in (Schulz, Bock, et al., 1998).

3.2 Indirect Optimisation Methods

This class of optimisation methods is based on the optimal control theory. Three basic strategies are classified under optimal control theory: (i) calculus of variations developed by Euler in 1744, (ii) minimum (maximum) principle developed by Pontryagin in 1962, and (iii) dynamic programming developed by Bellman in 1957, are very closely related. Under certain differentiability assumptions, one can be deduced from the other (Solheim, 1976). Pontryagin and his group were the first who give a rigorous proof of the most general version of the optimal control theory (Pesch and Bulirsch, 1994).

3.2.1 Pontryagin's Formulation

According to PMP, the problem of minimising the scalar objective function $\mathcal{J}$ in (1.3c) subject to dynamic constraints (1.2) and static constraints (1.6) can be rephrased as a minimisation of Hamiltonian function $\mathcal{H}(t)$ as follows:

$$
\begin{aligned}
\min_{\boldsymbol{u}} \mathcal{H} =& L(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{\lambda}^T \boldsymbol{F}(\boldsymbol{x},\boldsymbol{u},t) + \boldsymbol{\mu}^T \boldsymbol{\rho}(\boldsymbol{u},t) \\
\text{s.t.} \quad & \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x},\boldsymbol{u},t), \qquad \boldsymbol{x}(0) = \boldsymbol{x}_0 \\
& \dot{\boldsymbol{\lambda}} = -\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{x}}, \qquad \boldsymbol{\lambda}(t_1) = \left.\frac{\partial \phi}{\partial \boldsymbol{x}}\right|_{t_1} + \boldsymbol{\nu}^T \left.\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\right|_{t_1} \\
& \boldsymbol{\mu}^T \boldsymbol{\rho} = \boldsymbol{0}, \qquad \boldsymbol{\nu}^T \boldsymbol{\psi} = \boldsymbol{0} \\
& \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{F}}{\boldsymbol{u}} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\rho}}{\boldsymbol{u}} = \boldsymbol{0}
\end{aligned} \tag{3.9}
$$

where $\boldsymbol{\lambda} \neq 0$ is the n-dimensional vector of adjoint variables (Lagrange multipliers for system variables), and $\boldsymbol{\mu}(t) \geq 0$ the p-dimensional vector of Lagrange multipliers for the path constraints, $\boldsymbol{\nu} \geq 0$ the q-dimensional vector of Lagrange multipliers for terminal constraints. The Lagrange multipliers $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ are non-zero when the corresponding constraints are active and zero otherwise so the following equations are always satisfied:

$$
\boldsymbol{\rho}(\boldsymbol{u},t) \leq 0, \qquad \boldsymbol{\psi}(\boldsymbol{x}(t_1),t_1) = 0 \tag{3.10}
$$

PMP formulation represented by (3.9) provides an indirect method for optimal solution. One can see that to find optimal inputs for redefined problem, the state equations ($\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}, t)$) have to be solved in forward direction with initial conditions ($\boldsymbol{x}(0) = \boldsymbol{x}_0$), and subsequently the adjoint equations ($\dot{\boldsymbol{\lambda}} = -\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{x}}$) in backward direction with terminal conditions ($\boldsymbol{\lambda}(t_1) = \frac{\partial \phi}{\partial \boldsymbol{x}}|_{t_1} + \boldsymbol{\nu}^T \frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}|_{t_1}$). Thus, a two-point boundary value problems must be solved numerically in order to obtain the optimal inputs $\boldsymbol{u}$ satisfying following NCO:

$$\frac{\partial \mathcal{H}}{\partial \boldsymbol{u}} = \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{F}}{\boldsymbol{u}} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{u}} = \boldsymbol{0} \tag{3.11}$$

3.2.2 Hamilton-Jacobi-Bellman Formulation

The Hamilton-Jacobi-Bellman formulation transforms the problem of direct formulation (1.14) into the partial differential equations by utilising the following principle of optimality (Bryson and Ho, 1975; Kirk, 1970):

$$\begin{aligned} &\frac{\partial \boldsymbol{V}(\boldsymbol{x}, t)}{\partial t} + \min_{\boldsymbol{u}, \boldsymbol{\mu}, \boldsymbol{\nu}} \left(\frac{\partial \boldsymbol{V}(\boldsymbol{x}, t)}{\partial \boldsymbol{x}} \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\mu}^T \boldsymbol{\rho}(\boldsymbol{u}, t) \right) = \boldsymbol{0} \\ &\boldsymbol{V}(\boldsymbol{x}(t_1), t_1) = \boldsymbol{\phi}(\boldsymbol{x}(t_1)) + \boldsymbol{\nu}^T \boldsymbol{\psi}(\boldsymbol{x}(t_1)) \end{aligned} \tag{3.12}$$

where $\boldsymbol{V}(\boldsymbol{x}, t)$ is the return function or, equivalently, the minimum cost if the system has the state $\boldsymbol{x}$ at time $t \leq t_1$. The link between PMP and HJB formulations is that the adjoints are the sensitivities of the cost (return function) with respect to the states:

$$\boldsymbol{\lambda} = \frac{\partial \boldsymbol{V}}{\partial \boldsymbol{x}} \tag{3.13}$$

The term to be minimised in (3.12) is the Hamiltonian $\mathcal{H}$. Thus, the partial differential equation (3.12) represents the time evolution of the adjoints:

$$\dot{\boldsymbol{\lambda}} = \frac{d}{dt} \frac{\partial \boldsymbol{V}}{\partial \boldsymbol{x}} = \frac{\partial}{\partial \boldsymbol{x}} \frac{\partial \boldsymbol{V}}{\partial t} = -\frac{\partial \mathcal{H}_{\min}}{\partial \boldsymbol{x}} \tag{3.14}$$

where $\mathcal{H}_{\min}$ is the minimum value of the Hamiltonian.

3.2.3 Boundary Condition Iteration – Single-Shooting

The basic algorithm to solve two-point boundary value problem (TPBVP) is very similar for most of the methods. The algorithm for the intuitive solution strategy, single-shooting method, is shown below:

Problem:

$$\begin{aligned} &\min_{\boldsymbol{u}} \mathcal{J}(\boldsymbol{x}, \boldsymbol{u}) \\ \text{s.t.} \quad &\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}, t), \quad t_0 \leq t \leq t_1 \\ &\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \\ &\boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \leq 0 \qquad \qquad \boldsymbol{\rho}(\boldsymbol{u}, t) \leq \boldsymbol{0} \end{aligned} \tag{3.15}$$

Step 1: Apply PMP formulation to a given problem and transform it to given form (3.9).

Step 2: Parametrise the Lagrange multipliers $\boldsymbol{\mu}^T$ using any standard collocation method, e.g. orthogonal collocation and choose t_i:

$$\boldsymbol{\mu}_K(t) = \sum_{i=1}^{k} = \boldsymbol{\mu}_i \varphi_i(t) \quad \text{where} \quad \varphi_i(t) = \prod_{k=1,j}^{K} \frac{(t - t_k)}{(t_i - t_k)}, \quad \boldsymbol{\mu}_K(t_i) = \boldsymbol{\mu}_i, \quad i = 1, \dots, K \tag{3.16}$$

Step 3: Substitute the parametrised Lagrange multipliers $\boldsymbol{\mu}_K$ into the problem. The modified optimisation problem is then given as:

$$\begin{aligned} \min_u \mathcal{H} =& \boldsymbol{\lambda}^T \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}, t) + \boldsymbol{\mu}_K^T \boldsymbol{\rho}(\boldsymbol{u}, t) \\ \text{s.t.} \quad & \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}, t) \\ & \boldsymbol{x}(0) = \boldsymbol{x}_0 \\ & \dot{\boldsymbol{\lambda}} = -\frac{\partial \boldsymbol{H}}{\partial \boldsymbol{x}} \\ & \boldsymbol{\lambda}(t_1) = \left.\frac{\partial \phi}{\partial \boldsymbol{x}}\right|_{t_1} + \boldsymbol{\nu}^T \left.\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\right|_{t_1} \\ & \boldsymbol{\mu}_K^T \boldsymbol{\rho} = \boldsymbol{0} \\ & \boldsymbol{\nu}^T \boldsymbol{\psi} = \boldsymbol{0} \\ & \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{F}}{\boldsymbol{u}} + \boldsymbol{\mu}_K^T \frac{\partial \boldsymbol{\rho}}{\boldsymbol{u}} = \boldsymbol{0} \end{aligned} \tag{3.17}$$

Step 4: Choose an initial guess for the vector of decision variables: $\boldsymbol{\lambda}(t_0)$ and $\boldsymbol{\mu}_K(t_0)$ and integrate the dynamics forward in time using $\boldsymbol{x}(0)$, $\boldsymbol{\lambda}(t_0)$, and $\boldsymbol{\mu}_K(t_0)$. Compute $\boldsymbol{\lambda}(t_1)$.

Step 5: Check whether equations

$$\boldsymbol{\rho}(\boldsymbol{u}, t) = 0, \qquad \boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) = 0 \tag{3.18}$$

and

$$\frac{\partial \mathcal{H}}{\partial \boldsymbol{u}} = \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{F}}{\boldsymbol{u}} + \boldsymbol{\mu}^T \frac{\partial \boldsymbol{\rho}}{\partial \boldsymbol{u}} = \boldsymbol{0} \tag{3.19}$$

hold. The values of $\boldsymbol{\lambda}$ at final time obtained by integration in Step 3 should match those specified by equation:

$$\boldsymbol{\lambda}(t_1) = \left.\frac{\partial \phi}{\partial \boldsymbol{x}}\right|_{t_1} + \boldsymbol{\nu}^T \left.\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{x}}\right|_{t_1} \tag{3.20}$$

Step 6: Update the decision variables by optimisation method and repeat Step 3 through Step 5 until the solution is found.

In the single-shooting method, also called Boundary Condition Iteration (BCI), missing interval values are guessed and updated according to numerical strategy as long as the final value from IVP does not fit the final conditions. The optimal inputs are expressed analytically as functions of the states and adjoints. They results from NCO (3.19). The decision variables include the initial conditions $\boldsymbol{\lambda}(t_0)$ that are chosen in order to satisfy $\boldsymbol{\lambda}(t_1)$.

There are several difficulties associated with the shooting method (Murthy, Gangiah, et al., 1980). First, it can exhibit stability problems during forward integration of adjoints. Furthermore, unless a good initial guess for the adjoints is available, it is computationally expensive to find the optimum. The good initial guess is rarely the case hence the adjoints represent sensitivities. Next, the method does not work when there are discontinuities in the adjoint variables.

3.2.4 Multiple Shooting

In the multiple shooting method, the $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ values are guessed not only at the initial time, but also at several time points in-between. The IVP is then restarted at guessed time points. The difference between guessed and obtained values at time points are then reduced by new guesses updated by optimisation routine. The shorter intervals in multiple shooting lead to less non-linearity and less sensitivity to the guessed values than in single-shooting methodology.

3.2.5 Quasi-Linearisation or discretisation

Quasi-Linearisation is Newton-Raphson's method applied to a TPBVP, and is based on a first-order Taylor expansion of differential equations. In this methodology, successive linearisation (Bryson and Ho, 1975; Kirk, 1970; Lee and Markus, 1968) is applied to problem described by (3.9). In the same way, a discretisation can be assess: the states and adjoints are by time approximated functions. The basic procedure is as follows:

Step 1: Parametrise $\boldsymbol{x}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$ using finite number of decision variables.

Step 2: Choose an initial vector of guesses for the decision variables including $\boldsymbol{\nu}$.

Step 3: Linearise the process ODEs in (3.9) around the current guess or discretise them on finite elements. In case of linearisation, the original dynamic problem described by non-linear equations is transformed into a set of linear differential-algebraic equations. In case of discretisation, the original dynamic problem described is transformed into a set of non-linear differential-algebraic equations.

Step 4: Iteratively solve the optimal set of decision variables using appropriate optimisation method.

These methods works well if the solution is smooth and the unknown boundary conditions are not particular sensitive to initialisation errors. Furthermore, a good initial guess for the set of decision variables is needed as in shooting methods.

3.2.6 Control Vector Iteration

Control Vector Iteration (CVI) is the gradient method in which the necessary condition for optimality (3.11) provides the gradient along which are decision variables updated. The basic procedure is as follows:

Step 1: Parametrise $\boldsymbol{u}$, and $\boldsymbol{\mu}$ using finite number of decision variables.

Step 2: Choose an initial vector of guesses for the decision variables including $\boldsymbol{\nu}$.

Step 3: Integrate the state equations in (3.9) forward in time from t_0 to t_1

Step 4: Integrate the adjoint equations in (3.9) backward in time from t_1 to t_0 and compute the gradient $\partial\mathcal{H}/\partial\boldsymbol{u}$ expressed by (3.11).

Step 5: Use an optimisation algorithm to update the values of the decision variables. Repeat Step 3 through Step 5 until $\mathcal{H}$ is minimised.

The main advantage of gradient method lies in the fact that a good initial guess for the decision variables is beneficial but not crucial for convergence.

Despite the availability of all these solution methods, TPBVPs are demanding problems, with limited robustness. In addition, the adjoint variables are not physically meaningful quantities, which makes the first guess difficult. Another major obstacle for using the above approaches, is the symbolic manipulation needed in deriving the TPBVP.

Chapter 4

Nominal Solution

The previous chapters discuss the definition of the optimisation problem, the necessary conditions that indicate the optimality and single out the candidates of the nominal solution. The last chapter discussed various numerical methods that calculate the nominal solution. Note that direct optimisation methods deliver piece-wise continuous solution, either for controls, or for both controls and states. In opposite, the indirect optimisation methods delivers smooth continuous nominal solution as these methods are based on theory of optimal control. The major drawback of indirect methods is the need of good initial values for optimised variables, especially for complementary variables such as $\boldsymbol{\lambda}, \boldsymbol{\mu}, \boldsymbol{\nu}$. In order to overcome this, one may initially compute the nominal optimisation problem by the one of the direct methods and then, to pass these results as the starting values for optimised variables in indirect methods. A different approach is to compute the nominal solution directly from conditions of optimality. Basic idea of numerical procedure that computes such a solution follows next. This approach is also considered as the best option for further computations.

Next, let us consider the general form of the nominal optimisation problem:

$$\begin{aligned}
&\min_{\boldsymbol{u}} \mathcal{J}(\boldsymbol{x}, \boldsymbol{u}) \\
\text{s.t.} \quad &\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}, t), \quad t_0 \le t \le t_1 \\
&\boldsymbol{x}(t_0) = \boldsymbol{x}_0 \\
&\boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \le 0 \\
&\boldsymbol{u}^L \le \boldsymbol{u}(t) \le \boldsymbol{u}^U.
\end{aligned} \tag{4.1}$$

If $\boldsymbol{u}^*$ provides an optimal solution for the problem (4.1), there must also exist sextuple $(\boldsymbol{u}^*, \boldsymbol{x}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}_L^*, \boldsymbol{\mu}_U^*, \boldsymbol{\nu}^*)$ that satisfies the following Euler-Lagrange equations:

$$\dot{\boldsymbol{x}}^* = \mathcal{H}_{\boldsymbol{\lambda}}(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}_L^*, \boldsymbol{\mu}_U^*, t); \qquad \text{with} \quad \boldsymbol{x}^*(t_0) = \boldsymbol{x}_0 \tag{4.2}$$

$$\dot{\boldsymbol{\lambda}}^* = -\mathcal{H}_{\boldsymbol{x}}(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*, t); \qquad \text{with} \quad \boldsymbol{\lambda}^*(t_1) = \boldsymbol{\Phi}_{\boldsymbol{x}}(\boldsymbol{x}^*(t_1), t_1) \tag{4.3}$$

$$\boldsymbol{0} = \mathcal{H}_{\boldsymbol{u}}(\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*, \boldsymbol{\mu}_L^*, \boldsymbol{\mu}_U^*, t) \tag{4.4}$$

$$\tag{4.5}$$

along with complementary conditions

$$0 = \mu_j \rho_j, \quad j = 1, \ldots, n_\rho \tag{4.6}$$

$$0 = \nu_k \psi_k, \quad k = 1, \ldots, n_\psi \tag{4.7}$$

and where

$$\boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \leq 0 \tag{4.8}$$

$$\Phi = \phi + \boldsymbol{\nu}^{*T} \boldsymbol{\psi} \tag{4.9}$$

$$\mathcal{H} = L + \boldsymbol{\lambda}^{*^T} \boldsymbol{F} + \boldsymbol{\mu}_L^{*^T} (\boldsymbol{u}^L - \boldsymbol{u}) + \boldsymbol{\mu}_U^{*^T} (\boldsymbol{u} - \boldsymbol{u}^U). \tag{4.10}$$

Note that it is non-trivial to obtain the precise nominal solution for previous problem (4.1) that also satisfies every necessary condition of optimality represented by (4.2)–(4.4). The main difficulty here is the instability of states in backward direction and of the adjoints in forward direction. In addition, the Euler-Lagrange equations (4.2)–(4.4) represent the system of a differential algebraic equations that need to be solved collectively. Direct and indirect methods overcome this shortcoming by parametrising one or more of the variables, i.e. some variables are transformed from infinite domain into finite domain. The accuracy of such solution might be compromised as the parametrised solution may not absolutely copy the original one. Partial differential equations represented by HJB equations (3.12) are difficult to solve effectively. Furthermore, the nominal solution $\boldsymbol{u}^*$ cannot be directly exploited from a system of Euler-Lagrange equations (4.2)–(4.4), because of its constraints. Without a prior knowledge of a character of the nominal solution, it is not precisely know if the nominal solutions is constrained at the moment or if it is not. Similarly, the exact values of Lagrange multipliers $\boldsymbol{\nu}, \boldsymbol{\mu}$ needs to be determined.

Normally, the nominal solution $\boldsymbol{u}^*$ can be characterised as a non-singular, singular, constrained or unconstrained. When the nominal solution $\boldsymbol{u}^*$ can be explicitly given from the NCO (4.4), the problem is non-singular. The opposite situation is, when the $\boldsymbol{u}^*$ cannot be directly expressed from the NCO (4.4). In the this case, subsequent time derivatives of the NCO (4.4) are required. The condition is differentiated until the nominal solution $\boldsymbol{u}^*$ can be explicitly determine. Each of these solutions can be further function of (i) the states and adjoints, (ii) the states alone, (iii) the adjoints alone, or (iv) neither of them in case of constant control. Also, regard that this nominal solution holds only along an interior arc. Along the boundary arc, the nominal solution is then given by the path constraints. A continuity of all trajectories in nominal solution is required in the joints between the arcs.

Usually, the sequence of these arcs is unknown and needs to be determined by different way. In this case, numerical methods can provide the sequence of an interior or a boundary arcs, as well as, the values of unknown Lagrange multipliers. The switching times between the arcs become additional undetermined variables. One may estimate the initial values of the switching times by visual investigation of the piece-wise continuous solution or from the Lagrange multipliers $\boldsymbol{\mu}$ (for path constraints). The latter is provided by the conditions (4.7). Observe that when the one of path or terminal constraints is active (satisfied) the equivalent Lagrange multiplier becomes non-zero. Otherwise, the appropriate multiplier is equal to zero, thus the concerned path or terminal constraint is inactive (violated). Afterwards, the active path constraints (for controls) determine the structure of nominal solution and the analogous NCO. It is further assumed that such sequence is unchangeable by any perturbation.

In detail, the procedure to obtain a continuous nominal solution is as follows:

Step 1: Solve the optimisation problem (4.1) by any numerical approach and obtain optimal values for controls $\boldsymbol{u}_0^*$, for states $\boldsymbol{x}_0^*$, adjoints $\boldsymbol{\lambda}_0^*$, and Lagrange multipliers $\boldsymbol{\nu}_0^*$ and $\boldsymbol{\mu}_0^*$, respectively.

Step 2: Analyse the nominal solution $\boldsymbol{u}_0^*$ by visual inspection. Determine the sequence of arcs: (i) visually, or (ii) from Lagrange multipliers $\boldsymbol{\mu}^*$.

Step 3: Determine the initial values for switching times $\boldsymbol{\tau}_0^*$, i.e. times where the one arc become the different arc.

Step 4: Construct control sequence $\boldsymbol{u}$ of constrained and unconstrained arcs with initial switching times $\boldsymbol{\tau}_0^*$. Regard that the constrained arcs follows directly from box condition $\boldsymbol{u}^L \leq \boldsymbol{u}(t) \leq \boldsymbol{u}^U$ and the unconstrained arcs are given from necessary condition for controls $\mathcal{H}_{\boldsymbol{u}} = 0$.

Step 5: If $\mathcal{H}_{\boldsymbol{u}}$ is only a function of the states $\boldsymbol{x}$:

- Solve IVP problem of (4.1) with $\boldsymbol{u}$ in forward direction. Obtain state variables $\boldsymbol{x}$.
- Then, obtain adjoints $\boldsymbol{\lambda}$ by solving (4.3) in backward direction.

If $\mathcal{H}_{\boldsymbol{u}}$ is function of the both states $\boldsymbol{x}$ and adjoints $\boldsymbol{\lambda}$:

- Solve the problem given by differential algebraic equations (DAE) (4.2)–(4.4) with $\boldsymbol{u}$. Obtain state variables $\boldsymbol{x}$ and adjoints $\boldsymbol{\lambda}$.
- Note that the task to find solution of DAE becomes non-trivial when either one of states or adjoints becomes unstable during integration in reverse direction. One way to overcome this problem is to use adjoints from Step 1, and to solve DAE problem with less precise adjoints that have been obtained by numerical approach. The adjoints will be updated and improved during the iterations.

Step 6: Check whether $\boldsymbol{x}(t_1)$ matches $\boldsymbol{x}_0^*(t_1)$, and whether $\boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \leq 0$ holds.

Step 7: Use an optimisation algorithm to update the values of switching times $\boldsymbol{\tau}$ and $\boldsymbol{\nu}$, respectively. Iterate Step 5 through Step 7 until (4.1) is optimised.

It is need to be clarified that starting values of switching times $\boldsymbol{\tau}_0$, multiplier $\boldsymbol{\nu}_0$, or adjoint profiles $\boldsymbol{\lambda}_0$ must be as best as possible whereas only the sensitivities computed by finite-differences are available for the optimiser routine.

4.1 Batch Reactor Control

4.1.1 Unconstrained Case

Consider a problem of a batch reactor example according to Crescitelli and Nicoletti, 1973. The series of reactions $\mathsf{R} \xrightarrow{k_1} \mathsf{P} \xrightarrow{k_2} \mathsf{Q}$ takes place in the reactor. The goal is to maximise the production of P, whereas Q is an undesired by-product. The reactor is initially loaded with the reactant R and some product P. The final time is set to $t_1 = 10$. The manipulated variable $u(t)$ is related to the reactor temperature $T(t)$ by

$$T = -\frac{E_1}{R\, ln \frac{u}{k_{1,0}}}. \tag{4.11}$$

No bound constraints are imposed on the control variable in this case. The mathematical model is derived based on material-balance considerations and together with objective function, the dynamic optimisation problem reads:

$$\begin{aligned} &\max_{T}\; J = c_{\mathsf{P}}(t_1) \\ \text{s.t.}\quad &\dot{c}_{\mathsf{R}} = -u\, c_{\mathsf{R}}; \quad c_{\mathsf{R}}(t_0) = \beta_{\mathsf{R}} \\ &\dot{c}_{\mathsf{P}} = u\, c_{\mathsf{R}} - C\, c_{\mathsf{P}}\, u^{\alpha}; \quad c_{\mathsf{P}}(t_0) = \beta_{\mathsf{P}}. \end{aligned} \tag{4.12}$$

The model parameters and initial conditions are given in Table 4.1, below.

Table 4.1: Model parameters and initial conditions

$k_{1,0} = 0.535e11$	
$k_{2,0} = 0.461e18$	
$E_1 = 18$	$[\mathrm{kcal\,mol^{-1}}]$
$E_2 = 30$	$[\mathrm{kcal\,mol^{-1}}]$
$R = 2$	$[\mathrm{cal\,mol^{-1}\,K^{-1}}]$
$\alpha = E_1/E_2$	
$C = (k_{2,0}/k_{1,0})^{\alpha}$	
$t_1 = 10$	$[\mathrm{min}]$
$\beta_{\mathsf{R}} = 0.53$	$[\mathrm{mol\,L^{-1}}]$
$\beta_{\mathsf{P}} = 0.43$	$[\mathrm{mol\,L^{-1}}]$
$c_{\mathsf{R}}^{\min} = 0.2$	$[\mathrm{mol\,L^{-1}}]$
$u^{\min} = 0.09$	
$u^{\max} = 0.11$	

For this problem, the necessary conditions of optimality can then be expressed together as:

$$\begin{aligned} \begin{bmatrix} \dot{c}_{\mathsf{R}} \\ \dot{c}_{\mathsf{P}} \end{bmatrix} &= \begin{bmatrix} -u\, c_{\mathsf{R}} \\ u\, c_{\mathsf{R}} - C\, c_{\mathsf{P}}\, u^{\alpha} \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} c_{\mathsf{R}} \\ c_{\mathsf{P}} \end{bmatrix}(t_0) = \begin{bmatrix} \beta_{\mathsf{R}} \\ \beta_{\mathsf{P}} \end{bmatrix} \\ \begin{bmatrix} \dot{\lambda}_{\mathsf{R}} \\ \dot{\lambda}_{\mathsf{P}} \end{bmatrix} &= \begin{bmatrix} \lambda_{\mathsf{R}}\, u - \lambda_{\mathsf{P}}\, u \\ C\, \lambda_{\mathsf{P}}\, u^{\alpha} \end{bmatrix} \quad \text{with} \quad \begin{bmatrix} \lambda_{\mathsf{R}} \\ \lambda_{\mathsf{P}} \end{bmatrix}(t_1) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ 0 &= \lambda_{\mathsf{P}} \left(c_{\mathsf{R}} - \alpha\, C\, u^{\alpha-1}\, c_{\mathsf{P}} \right) - \lambda_{R}\, c_{\mathsf{R}}, \end{aligned} \tag{4.13}$$

where λ_{R} and λ_{P} are adjoint variables.

The nominal solution for the problem (4.12) can then be obtained from the differential algebraic equations (4.13). Note that the nominal solution consists of one unconstrained arc only. In order to obtain initial guess for the state a control profiles, one at first can solve the problem by direct numerical approach (CVP approach in this situation). The solution obtained by CVP approach is depicted in Figure 4.1 by dotted red line. As only one unconstrained arc is presented here, the continuous nominal solution can be directly acquired from the NCO (4.13). So, DAE (4.13) are solved in backward direction. As the algebraic equation $\mathcal{H}_u = 0$ may produce more then one solution, the starting point for DAE solver can then be the final value of the sequential control. In order to overcome the instability of states produced by backward integration, it is advised to use the state profile from CVP approach to approximate the states at the integration points. Such nominal solution is then shown in Figure 4.1 via solid blue line. The comparison of the both solutions is presented in Figure 4.1. Also, see that these solutions are in good agreement.

4.1.2 Constrained case

The constrained problem is an extension of the unconstrained problem. Here, the optimisation objective is to maximise the concentration of product P at final time, $c_{\mathsf{P}}(t_1)$, while keeping the residual concentration of reactant R at final time below the maximum threshold $c_{\mathsf{R}}^{\min}$. Also, box constraints are imposed on control variable. Essentially, the constrained optimisation problem reads:

$$\begin{aligned}
&\max_{T}\ J = c_{\mathsf{P}}(t_1)\\
\text{s.t.}\quad &\dot c_{\mathsf{R}} = -u\,c_{\mathsf{R}};\quad c_{\mathsf{R}}(t_0) = \beta_{\mathsf{R}}\\
&\dot c_{\mathsf{P}} = u c_{\mathsf{R}} - C\,c_{\mathsf{P}}\,u^{\alpha};\quad c_{\mathsf{P}}(t_0) = \beta_{\mathsf{P}}\\
&c_{\mathsf{R}}(t_1) \le c_{\mathsf{R}}^{\min}\\
&u^{\min} \le u \le u^{\max}
\end{aligned}\tag{4.14}$$

The correspondent NCO are moreover expressed as follows:

$$\begin{aligned}
\begin{bmatrix}\dot c_{\mathsf{R}}\\ \dot c_{\mathsf{P}}\end{bmatrix} &= \begin{bmatrix}-u\,c_{\mathsf{R}}\\ u\,c_{\mathsf{R}} - C\,c_{\mathsf{P}}\,u^{\alpha}\end{bmatrix}\quad\text{with}\quad \begin{bmatrix}c_{\mathsf{R}}\\ c_{\mathsf{P}}\end{bmatrix}(t_0) = \begin{bmatrix}\beta_{\mathsf{R}}\\ \beta_{\mathsf{P}}\end{bmatrix}\\
\begin{bmatrix}\dot\lambda_{\mathsf{R}}\\ \dot\lambda_{\mathsf{P}}\end{bmatrix} &= \begin{bmatrix}\lambda_{\mathsf{R}}\,u - \lambda_{\mathsf{P}}\,u\\ C\,\lambda_{\mathsf{P}}\,u^{\alpha}\end{bmatrix}\quad\text{with}\quad \begin{bmatrix}\lambda_{\mathsf{R}}\\ \lambda_{\mathsf{P}}\end{bmatrix}(t_1) = \begin{bmatrix}\nu\\ -1\end{bmatrix}\\
0 &= \lambda_{\mathsf{P}}\left(c_{\mathsf{R}} - \alpha\,C\,u^{\alpha-1}\,c_{\mathsf{P}}\right) - \lambda_{\mathsf{R}}\,c_{\mathsf{R}}\\
0 &= \nu\left(c_{\mathsf{R}}(t_1) - c_{\mathsf{R}}^{\min}\right)\quad\text{with}\quad 0\le\nu,
\end{aligned}\tag{4.15}$$

Again, the control structure needs to be determined at first by visual inspection of the solution that is obtained by direct numerical approach (CVP approach in this situation, too). The sequential solution is depicted in Figure 4.2 by dotted red line and comprises three arcs: an upper bound arc, an interior arc, and a lower bound arc. In addition, this solution provides the value of Lagrange multiplier ν, the switching times τ_i of the arcs, and the initials for next computations. Note that the terminal inequality constraint is active at the optimum. Thus the multiplier $\nu \neq 0$. The continuous nominal solution is then given by sequence of an upper bound arc, the interior arc provided by DAE (4.15), and a lower bound arc. As in unconstrained case, the solution is obtained by backward integration of DAE (4.15) with Lagrange multiplier ν taken from previous computation. Note that during the backward integration the control u is initially set to its lower bound until time reaches the switching point τ_2, then it switches to interior arc that is computed from the algebraic equation $\mathcal{H}_u = 0$ till the switching time τ_1. Finally, the control remains on its upper bound until it enters the initial time t_0. Similar to previous case, the algebraic equation $\mathcal{H}_u = 0$ produces more than one solution, so it is important to start from the point provided by CVP approach. Further, the stability issues of states during backward integration are addressed by approximating the pre-computed states (by CVP approach) at the integration points. The nominal solution is represented by solid blue line in Figure 4.2. The comparison of the both solutions is demonstrated in Figure 4.2. Indeed, these solutions are compatible.

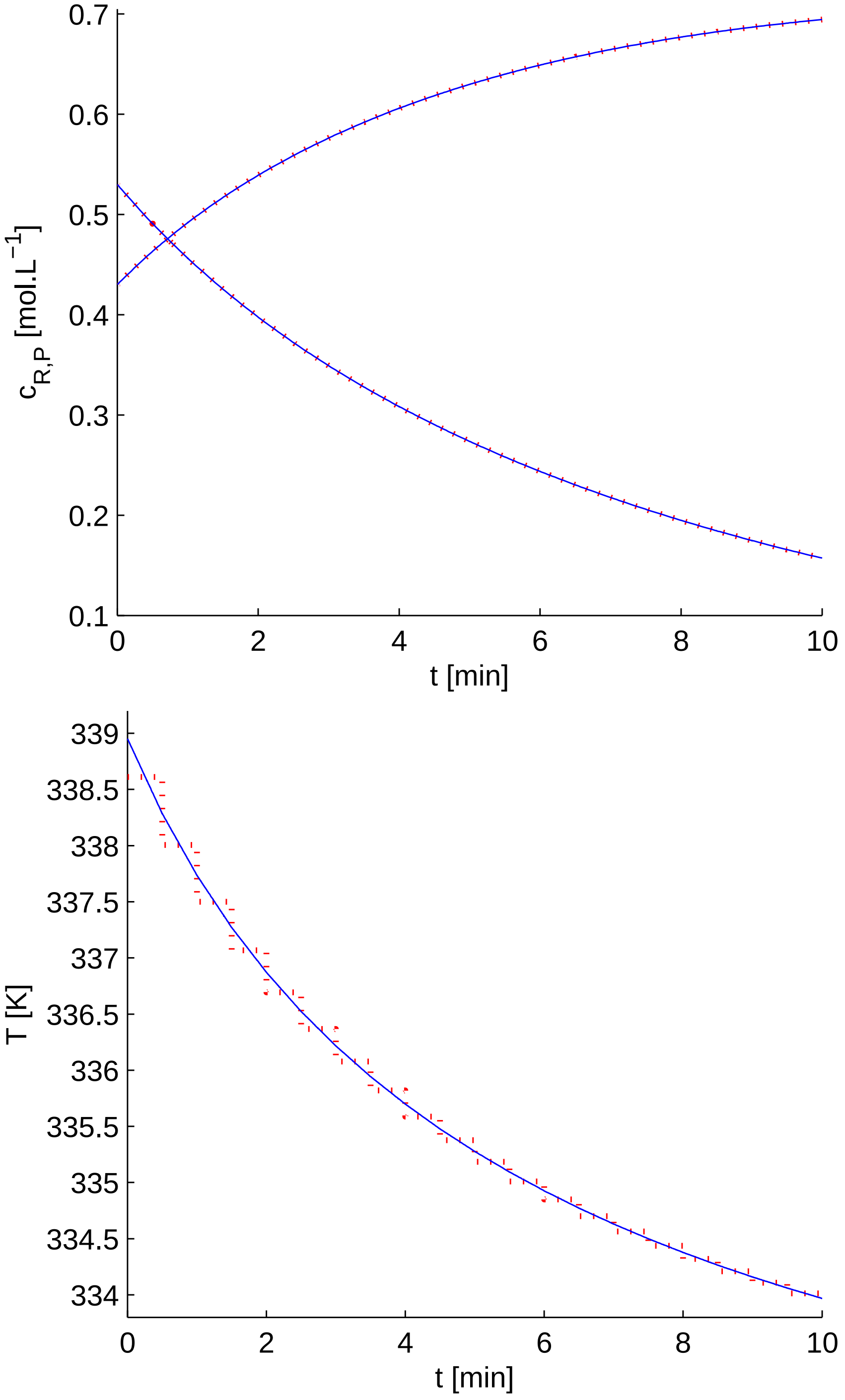

Figure 4.1: **Unconstrained case:** Nominal solution vs. sequential solution of state (top figure) and control (bottom figure) variables. **Dotted line:** Solution obtained via sequential approach. **Solid line:** Nominal solution.

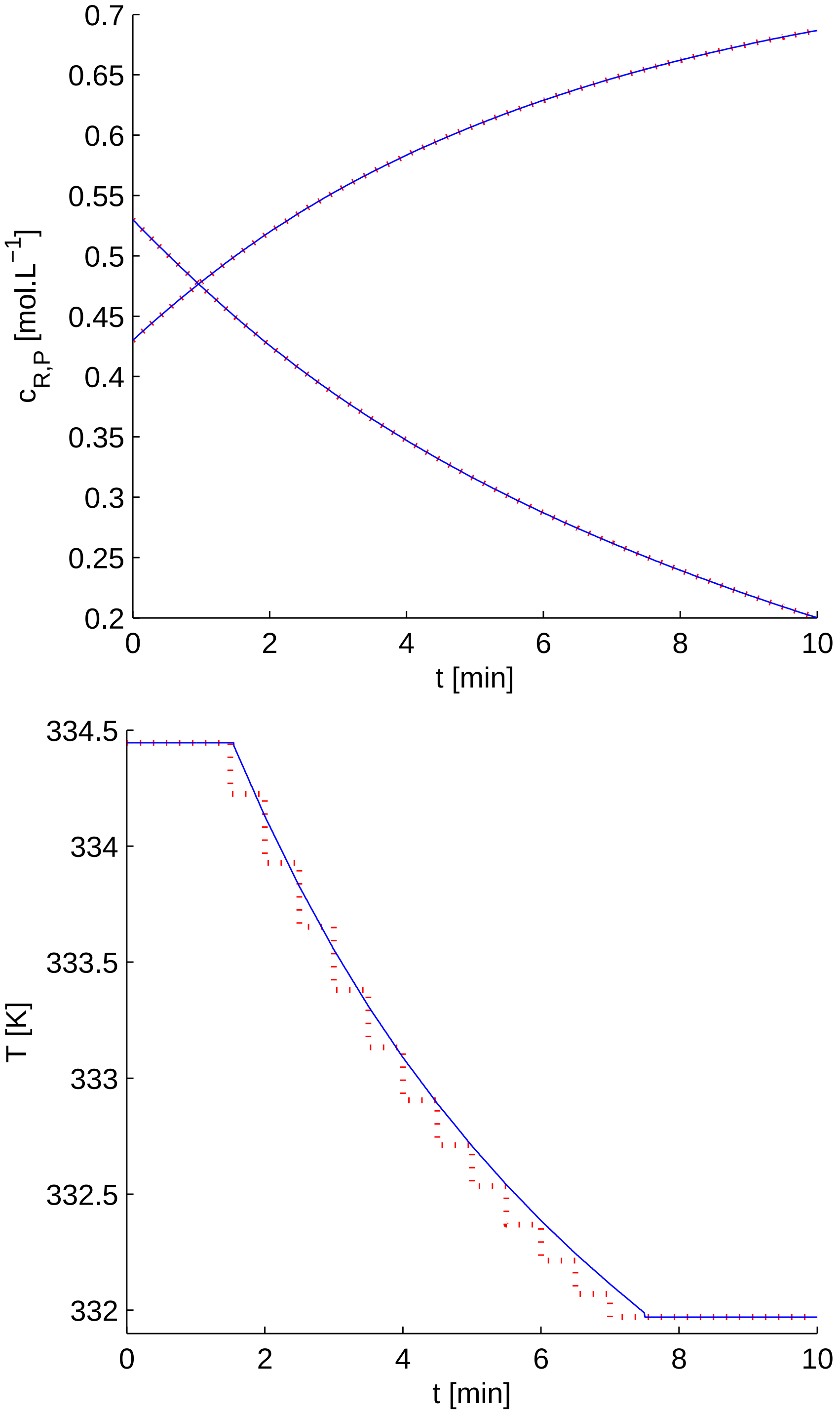

Figure 4.2: **Constrained case:** Nominal solution vs. sequential solution of state (top figure) and control (bottom figure) variables. **Dotted line:** Solution obtained via sequential approach. **Solid line:** Nominal solution.

Part II

Dynamic Optimisation under Uncertain Conditions

Chapter 5

Tracking of Necessary Conditions for Optimality

Previous part discussed a unified framework called nominal optimisation (also, dynamic optimisation under ideal conditions) that improves control policy of a particular process. In real-world applications, this case is very uncommon, as the process is usually subject to disturbances, model mismatch and another forms of an uncertainty. Such changes in process behaviour naturally affect the operational conditions that are no longer optimal when controlled by nominal solution. In addition, the NCO as well are not satisfied. The obvious limitation of standalone nominal optimisation comes from the fact that it is an open-loop control scheme that leads to an optimality loss and violations of NCO in the presence of an uncertainty.

A natural approach to control a perturbed process is to employ the measurements that provides the feedback information. To satisfy the NCO also in perturbed process, one way is to re-optimise the problem by taking the perturbations into account. Usually, it is difficult to express an uncertainty mathematically. In such a case, some robust or closed-loop control approach is clearly needed. The other way around is to enforce NCO directly without re-optimisation. Since the control structure (given by successive sequence of the arcs) is given by the nominal solution and if the sequence of arcs is not affected by an uncertainty, the NCO can be easily checked at certain points and then they can be pushed towards them. This approach is known as NCO-tracking and has been introduced by (Srinivasan and Bonvin, 2004b; Srinivasan, Bonvin, et al., 2003a,b; Welz, Srinivasan, et al., 2006).

In essence, the enforcement of the NCO in DOP can be described as the use of an arbitrary control approach to meet four sets of necessities (Srinivasan and Bonvin, 2004b). Subsequently, the measurements provide the fundamental information about the process behaviour and the constraints. The problem of tracking NCO can be divided into two main methodologies: (i) the NCO-tracking associated with tracking of active constraints, and (ii) tracking of NCO associated with sensitivities. The relation between standalone NCO parts and various control schemes found in literature is presented in Table 5.1.

We can see that path objectives in NCO can be handled by constrained control presented in (Srinivasan and Bonvin, 2004b; Srinivasan, Primus, et al., 2001) or by constraint adaptation introduced in (Chachuat, Marchetti, et al., 2008; Marchetti, Chachuat, et al., 2007). A general idea here is to make the path constraints active and to optimise the length of their intervals. The active terminal constraints can be enforced by pure run-to-run constraint control introduced in (Srinivasan and Bonvin, 2004b; Srinivasan, Primus, et al., 2001), by constraint adaptation technique (CA) explained in (Chachuat, Marchetti, et al., 2008; Marchetti, Chachuat, et al., 2007), or via integrated two-time-scale control described in (Podmajerský, Chachuat, et al., 2011a; Podmajerský and Fikar, 2011). The first approach optimises the terminal objectives by I controller linked to an appropriate decision variable. The second approach updates the constraints in a model in run-to-run fashion and in such way that the real-world constraints are satisfied. Last approach implements CA approach in the outer loop and NE control in the inner loop. The sensitivities along unconstrained arcs are handled by NE control (Bryson and Ho, 1975; Gros, 2007; Pesch and Bulirsch, 1994; Pesh, 1990; Ruppen, Benthack, et al., 1995) that uses actual measurements to minimise the variations around nominal solution. Different approach is to use NE control inside NMPC structure as proposed (Würth, Hannemann, et al., 2009b, 2011). Terminal sensitivities can be addressed using static optimisation technique like extremum-seeking control (Kristic and Wang, 2000) or via NE control proposed by (Bryson and Ho, 1975; Pesch and Bulirsch, 1994; Pesh, 1990) which updates path sensitivities to correct inputs toward terminal objectives.

Table 5.1: Relation between the four NCO parts and various control schemes

	Path objectives	**Terminal objectives**
Constraints	Constraint control [1] Constraint adaptation [3,4]	Run-to-run constraint control [1,2] Integrated two-time-scale control [5,6]
Sensitivities	NE control [7,8] NE control for NMPC [10,11]	Extremum-seeking control [9] NE control [12,13,14]

[1] B. Srinivasan and D. Bonvin (2004b). “Dynamic Optimization under Uncertainty via NCO Tracking: A Solution Model Approach”. In: *BatchPro Symposium*. Plenary talk, pp. 17–35

[2] B Srinivasan, C.J. Primus, et al. (2001). “Run-to-run Optimization of Batch Processes via Generalized Constraint Control”. In: *Control Eng. Practice* **9**, pp. 911–919

[3] A. Marchetti, B. Chachuat, et al. (2007). “Batch process optimization via run-to-run constraints adaptation”. In: *European Control Conference*. Kos, Greece

[4] B. Chachuat, A. Marchetti, et al. (2008). “Process Optimization via Constraints Adaptation”. In: *Journal of Process Control* **18**(3-4), pp. 244–257

[5] M. Podmajerský, B. Chachuat, et al. (2011a). “Integrated Two-Time-Scale Scheme for Real-time Optimisation of Batch Processes”. In: *Proc. of the 18th IFAC World Congress*. Milano, Italy

[6] M. Podmajerský and M. Fikar (2011). “Real-time Dynamic Optimisation by Integrated Two-Time-Scale Scheme”. In: *Proceedings of the 18th International Conference on Process Control 11*. Slovak University of Technology in Bratislava. Štrbské Pleso, Slovakia

[7] S. Gros (2007). “Neighboring Extremals in Optimization and Control”. PhD thesis. École Polytechnique Fédérale de Lausanne, Lausanne

[8] D. Ruppen, C. Benthack, et al. (1995). “Optimization of Batch Reactor Operation under Parametric Uncertainty – Computational Aspects”. In: *Journal of Process Control* **5**(4), pp. 235–240

[9] M Kristic and H-H. Wang (2000). “Stability of Extremum Seeking Feedback for General Non-linear Dynamic Systems”. In: *Automatica* **36**, pp. 595–601

[10] L. Würth, R. Hannemann, et al. (2009b). “Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization”. In: *Journal of Process Control* **19**(8), pp. 1277–1288

[11] L. Würth, R. Hannemann, et al. (2011). “A two-layer architecture for economically optimal process control and operation”. In: *Journal of Process Control* **21**(3), pp. 311–321

[12] A. E. Bryson and Yu-Chi Ho (1975). *Applied Optimal Control – Optimization, Estimation and Control*. Hemisphere publishing corporation

[13] H. J. Pesh (1990). “The Accessory Minimum Problem and Its Importance for the Numerical Computation of Closed-Loop Controls”. In: *Conference on Decision and Control*. Honolulu, HI, pp. 952–953

[14] H.J. Pesch and R. Bulirsch (1994). “The Maximum Principle, Bellman’s Equation, and Caratheodory’s Work”. In: *J. Optimization Theory and Applications* **80**(2), pp. 199–225

Chapter 6

NCO-tracking Associated with Active Constraints

The key idea behind NCO-tracking is that once the nominal solution is obtained, the sequence of arcs can be linked to the correspondent NCO and then indirectly optimised to satisfy NCO. Several tasks are involved in order to implement the methodology.

6.1 Switching Structure

The nominal solution, introduced in Chapter 4, embodies a sequence of constrained and unconstrained arcs. This can be further converted into a control scheme, i.e. a switching structure that comprises the sequence of arcs, their switching times, the active constraints and the appropriate NCO linked to the switching times and to the arcs. The optimisation problem then becomes a control problem. Note that sequence of arcs provided by the nominal solution is a minimal parametrisation of a switching structure. Indeed, the nominal solution can be parametrised by different switching structure. More on this topic can be found in (Srinivasan, Bonvin, et al., 2003a).

6.2 Fixed and Free Input Elements

Previous section presented a control structure that consists of a set of elements. These include time-functions (arcs) and time-invariant parameters (switching times, and possibly final time). Some of these input elements are affected by perturbations (free variables) and some are inherent (fixed variables). For example, constrained input arcs remain at its minimal or maximal values and they do not change with changing perturbations. Thus, these elements are considered as fixed. In opposite, the switching times of the input arcs may vary. Such elements are therefore considered free to adjust.

6.3 Linking the Variables to the NCO

The fixed input elements are known and they can be implemented directly without any feedback. In contrast, the free variables requires to be further adjusted, according to the amount of uncertainty, within the control framework. The active path and terminal constraints are determined by a set of certain arcs and parameters. Some of the free or additional parameters can be assigned to handle NCO. In particular, the objectives related to the NCO are the following:

- The *path constraint variables* are adjusted by making the corresponding path constraint active.
- The *terminal constraint variables* are adjusted by making the corresponding terminal constraint active.
- The *path sensitivity variables* need path sensitivity measurements or estimation for adaptation.
- The *terminal sensitivity variables* need terminal sensitivity measurements or estimation for adaptation.

Note that these control variables are part of the switching structure. They also ensure a certain pairing between the control variables and the NCO. This assignment requires a judgement of designer. We note that a different pairing policy implies different adaptation strategy, thus different control structure. An important assumption for this assignment to become effective is that the nominal solution is precise and the set of active constraints does not change with uncertainty.

6.4 Two-stage Batch Reactor Control

The aim of NCO-tracking is to avoid a re-optimisation process and to drive the variations to zero. This section reviews the design of controller that tracks NCO associated with constraints straightforwardly using standard control tools. In the presence of uncertainty, the resulting variations can be estimated explicitly from the measurements. The case study with step-by-step control design follows.

6.4.1 Problem Definition

A connected two-stage batch reactors (Hirmajer and Fikar, 2006; Vassiliadis, Sargent, et al., 1994a,b), non-linear in nature are considered. The first one is filled with diluted solution of compound A with initial concentration $c_{\mathsf{A}}(t_0)$ up to volume V_1 and some portion of catalyst. The heating coil is a control variable during the first stage. In the first reactor, the chain reaction

$$\mathsf{A} \overset{k_1}{\rightarrow} \mathsf{B} \overset{k_2}{\rightarrow} \mathsf{C} \tag{6.1}$$

takes place till an undetermined time t_p. At this instant, the dynamic of the process will change. The second batch reactor is filled with the products from the first reaction and an amount S of diluted solution of compound B with the concentration c_{B}^{s} is added. Three parallel reactions at isothermal conditions take place in the reactor

$$\mathsf{B} \rightarrow \mathsf{D} \tag{6.2}$$

$$\mathsf{B} \rightarrow \mathsf{E} \tag{6.3}$$

$$2\mathsf{B} \rightarrow \mathsf{F}. \tag{6.4}$$

The model of above-mentioned process is based under assumption of perfect mixing and ideal liquid mixture behaviour. For detailed process information, please refer to (Srinivasan and Bonvin, 2007).

6.4.2 Nominal Solution

The objective is to maximise an amount of compound D equal to $V_2 c_{\mathsf{D}}$ subject to a minimal desired concentration c_{D}^{w} of compound D at final time t_1 and subject to process equations. The decision variables are the reactor temperature T at the first reaction stage, the switching time t_p between two stages, final time t_1 and the amount of S. In overall, the optimisation problem is given as follows:
objective function:

$$\min_{S, t_p, T[0,t_1]} V_2 c_{\mathsf{D}}(t_1) \tag{6.5a}$$

constraints:
path

$$\begin{bmatrix} \dot{c}_{\mathsf{A}} \\ \dot{c}_{\mathsf{B}} \\ \dot{c}_{\mathsf{C}} \\ \dot{c}_{\mathsf{D}} \\ \dot{c}_{\mathsf{E}} \\ \dot{c}_{\mathsf{F}} \end{bmatrix} = \begin{bmatrix} -2\,k_1(T)\,c_{\mathsf{A}}^2 \\ k_1(T)\,c_{\mathsf{A}}^2 - k_2(T)\,c_{\mathsf{B}} \\ k_2(T)\,c_{\mathsf{B}} \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad t \in [0, t_p] \tag{6.5b}$$

$$\begin{bmatrix} \dot{c}_{\mathsf{A}} \\ \dot{c}_{\mathsf{B}} \\ \dot{c}_{\mathsf{C}} \\ \dot{c}_{\mathsf{D}} \\ \dot{c}_{\mathsf{E}} \\ \dot{c}_{\mathsf{F}} \end{bmatrix} = \begin{bmatrix} 0 \\ -0.02\,c_{\mathsf{B}} - 0.05\,c_{\mathsf{B}} - 0.00008\,c_{\mathsf{B}}^2 \\ 0 \\ 0.02\,c_{\mathsf{B}} \\ 0.05\,c_{\mathsf{B}} \\ 0.00004\,c_{\mathsf{B}}^2 \end{bmatrix} \qquad t \in [t_p, t_1] \tag{6.5c}$$

terminal

$$c_{\mathsf{D}}(t_1) - c_{\mathsf{D}}^{w} \geq 0 \tag{6.5d}$$

with kinetic rate constants defined as

$$k_1(T) = 0.0444\,e^{\frac{-2500}{T}} \tag{6.6a}$$

$$k_2(T) = 6889\,e^{\frac{-5000}{T}} \tag{6.6b}$$

and mixing operations at the switching time t_p are

$$V_2\, c_{\mathsf{A}}(t_p^+) = V_1\, c_{\mathsf{A}}(t_p^-) \tag{6.7a}$$
$$V_2\, c_{\mathsf{B}}(t_p^+) = V_1\, c_{\mathsf{B}}(t_p^-) + S\, c_{\mathsf{B}}^s \tag{6.7b}$$
$$V_2\, c_{\mathsf{C}}(t_p^+) = V_1\, c_{\mathsf{C}}(t_p^-) \tag{6.7c}$$

where $V_2 = V_1 + S$; $c_{\mathsf{A}}(t_p^-), c_{\mathsf{B}}(t_p^-), c_{\mathsf{C}}(t_p^-)$ are output concentrations of compounds A, B, and C in the first stage; $c_{\mathsf{A}}(t_p^+), c_{\mathsf{B}}(t_p^+), c_{\mathsf{C}}(t_p^+)$ are initial input concentrations of compounds A, B, and C in the second stage; S stands for an amount of addition of compound B at switching time t_p with fixed concentration c_{B}^s. The values of process parameters are the following: $V_1 = 0.1\,\mathrm{m}^3$, $c_{\mathsf{A}}(t_0) = 2000\,\mathrm{mol\,m}^{-3}$, $c_{\mathsf{B-F}}(t_0) = 0\,\mathrm{mol\,m}^{-3}$, $c_{\mathsf{B}}^s = 600\,\mathrm{mol\,m}^{-3}$, $c_{\mathsf{D}}^w = 150\,\mathrm{mol\,m}^{-3}$. The final time is constrained by $t_1 \le 180\,\mathrm{min}$.

Numerical solution of the optimisation problem is obtained by simultaneous approach, namely by orthogonal collocation method. Both control and state profiles are parametrised and transformed from infinite time domain into finite time domain, i.e. continuous problem becomes NLP. In the first stage, 4 intervals with 10 discretisation points are used for state variable, 3 intervals with 4,5,3 collocation points are used for control variables, respectively. In the second stage, single interval with 10 collocation points is used for state variables. No control variable is presented here as the process operates at isothermal conditions. The initial conditions are summarised as follows: interval lengths are fixed to $\Delta t_i = 15\,\mathrm{min}$, controls are initialised as $T_i = 350\,\mathrm{K}$. The initial addition is considered as $S = 0.1\,\mathrm{m}^3$ and bounds defined as $T_i \in [298, 398]$, for $t \in [0, t_p]$, $S \in [0, 0.1]$, $\Delta t_i \in [1, 100]$. Absolute and relative tolerances of NLP solver are set to 10^{-6}.

The optimal value of performance index is $\mathcal{J} = 25.56\,\mathrm{mol}$ which is in good agreement with (Hirmajer and Fikar, 2006). Also, value of the addition is reported $S = 0.0702\,\mathrm{m}^3$ which matches the amount reported in (Hirmajer and Fikar, 2006). Both constraints are satisfied and active: the final concentration c_{D} of compound D is equal to desired $150\,\mathrm{mol\,m}^{-3}$ and the final time t_1 coincides with the maximum $180\,\mathrm{min}$. The resulting optimal switching time takes value $t_p = 105.8\,\mathrm{min}$. The nominal solutions for control and state variables are shown in Figure 6.1.

6.4.3 Proposed Control Structures

The nominal solution can be parametrised by the following function $\wp(\eta[t_s, t_s'], \pi)$ where $\eta[t_s, t_s']$ represents an input arc defined on particular interval $[t_s, t_s']$. Next, the parameter vector π represents switching times. For our problem of the two-stage batch reactor, there exist two possible parametrisations of nominal solution that track the necessary condition of optimality. In detail, the input structures **Model 1** and **Model 2** are as follows:

Model 1 The input profile begins with the sensitivity arc 1 $\frac{\partial \mathcal{H}}{\partial \eta_1}(0, \pi_1)$ that terminates when concentration of compound B reaches an optimal value $c_{\mathsf{B}}(\pi_1) \ge c_{\mathsf{B}}^*$. This point denotes the switching time in the input structure π_1 as well as switching time between the batches. At the time π_1, the input profile becomes a constant arc 2 $\eta_2\,(\pi_1, \pi_2) = u_{t_p}$ and terminates at the time π_2. Time π_2 is reached when the concentration of compound D is greater or equal to desired value $c_{\mathsf{D}}(\pi_2) \ge c_{\mathsf{D}}^w$. The corresponding

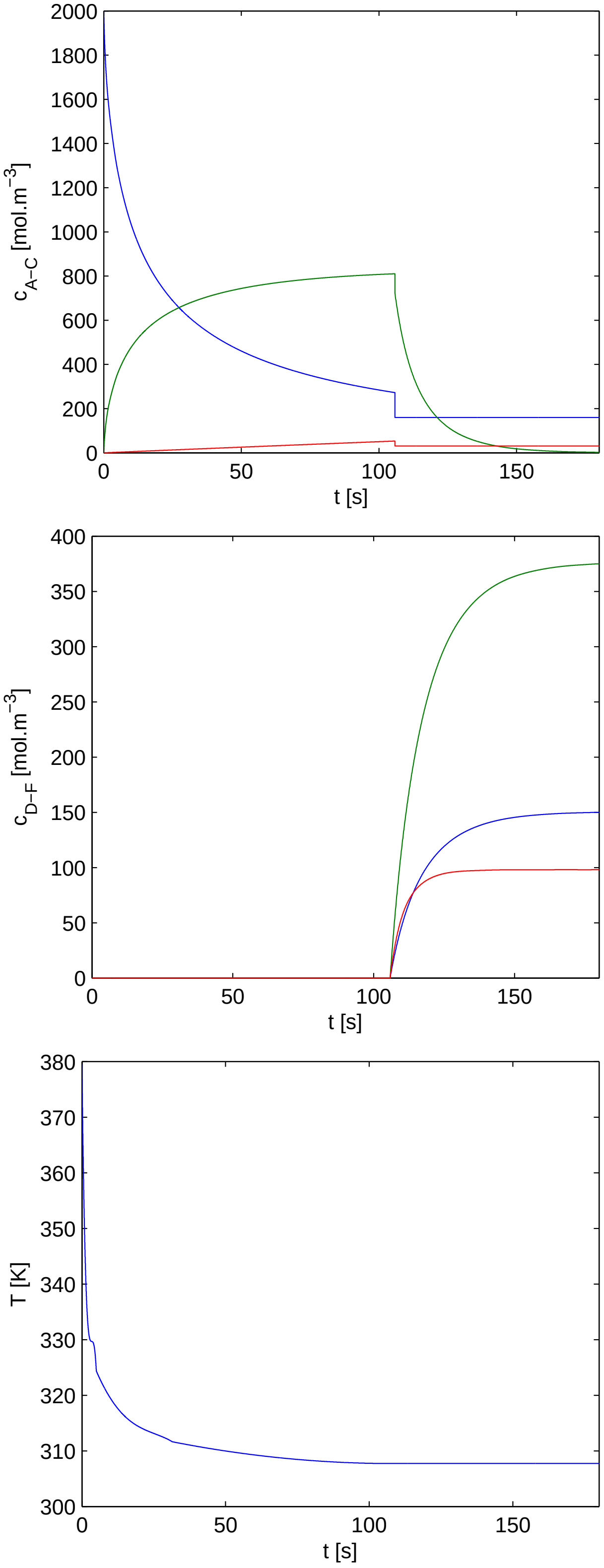

Figure 6.1: Nominal solution for two-stage reactor. **Top plot:** optimal concentration profiles of compounds A – C. **Middle plot:** optimal concentration profiles of compounds E – F. **Bottom plot:** nominal control profile.

Table 6.1: NCO corresponding to input Model 1

	path objectives	point-wise objectives
constraints	**Arc 2:** -	$c_{\mathsf{B}}(\pi_1) \geq c_{\mathsf{B}}^*$ $c_{\mathsf{D}}(\pi_2) \geq c_{\mathsf{D}}^w$
sensitivities	**Arc 1:** $\frac{\partial \mathcal{H}}{\partial \eta_1}(0, \pi_1)$	-

Table 6.2: NCO corresponding to input Model 2

	path objectives	point-wise objectives
constraints	**Arc 2:** -	$c_{\mathsf{D}}(\pi_1) \geq c_{\mathsf{D}}^w$
sensitivities	**Arc 1:** $\frac{\partial \mathcal{H}}{\partial \eta_1}(0, t_p)$	-

NCO are summarised in Table 6.1. For Model 1, the switching structure involves two continuous variables and two scalar variables. The schematic of this structure is displayed in Figure 6.2.

$$u(t) = \mathcal{P}(\eta_1, \eta_2, \pi_1, \pi_2) = \begin{cases} \eta_1(t) & (\text{for } 0 \leq t \leq \pi_1) \\ \eta_2(t) & (\text{for } \pi_1 \leq t \leq \pi_2) \end{cases} \tag{6.8}$$

Model 2 The input profile begins with the sensitivity arc 1 $\frac{\partial \mathcal{H}}{\partial \eta_1}(0, t_p)$ and it ends at switching time t_p. Then, the control profile changes to a constant arc 2 $\eta_2(t_p, \pi_1) = u_{t_p}$ and it finishes at the time π_1 which is determined by desired concentration level of compound D. It is required to achieve at least the desired value $c_{\mathsf{D}}(\pi_1) \geq c_{\mathsf{D}}^w$. The corresponding NCO are presented in detail in Table 6.2. The switching structure then involves two continuous variables and one scalar variable. The schematic of this structure is displayed in Figure 6.2.

$$u(t) = \mathcal{P}(\eta_1, \eta_2, \pi_1) = \begin{cases} \eta_1(t) & (\text{for } 0 \leq t \leq t_p) \\ \eta_2(t) & (\text{for } t_p \leq t \leq \pi_1) \end{cases} \tag{6.9}$$

Control design for NCO-tracking scheme continues with following tasks:

Formulation of control problem – Next, the manipulated and controlled variables need to be determined. For Model 1 formulated by equations (6.10), the initial concentration $c_{\mathsf{B}0}$ of compound B is chosen as a manipulated variable. The switching time t_p, and amount of addition S are chosen as a manipulated variables for Model 2 formulated by equations (6.11). Similarly, the final concentration $c_{\mathsf{D}}(t_1)$ of compound D and the concentration of compound B at the switching time $c_{\mathsf{B}}(t_p)$ are chosen as a controlled variables for Model 1, whereas only a final concentration $c_{\mathsf{D}}(t_1)$ of compound D is a controlled variable for Model 2.

$$u(t) = \mathcal{P}(\eta_1, \eta_2, \pi_1, \pi_2) = \left[\begin{array}{ll} \eta_1(t) = P_c(c_{\mathsf{B}}^{nom} - c_{\mathsf{B}}(t)) & (\text{for } 0 \leq t \leq \pi_1) \\ \eta_2(t) = u_{t_p} & (\text{for } \pi_1 \leq t \leq \pi_2) \end{array} \right. \tag{6.10}$$

$$u(t) = \mathcal{P}(\eta_1, \eta_2, \pi_1) = \left[\begin{array}{ll} \eta_1(t) = P_c(c_{\mathsf{B}}^{nom} - c_{\mathsf{B}}(t)) & (\text{for } 0 \leq t \leq t_p) \\ \eta_2(t) = u_{t_p} & (\text{for } t_p \leq t \leq \pi_1) \end{array} \right. \tag{6.11}$$

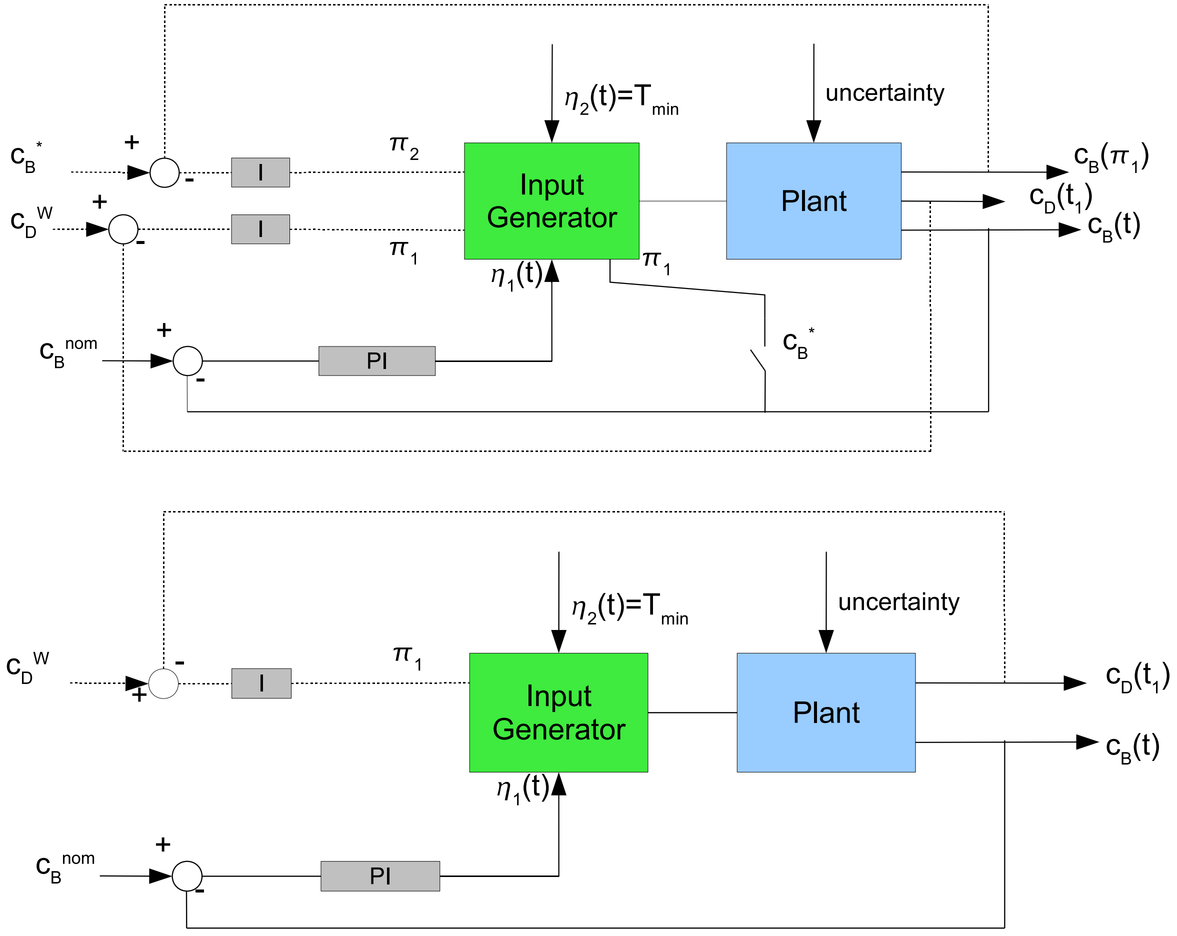

Figure 6.2: NCO control schemes. **Top plot:** Model 1. **Bottom plot:** Model 2.

Controller design – When the input (manipulated) and output (controlled) variables are chosen, decentralisation and pairing process follows. Decentralisation is helpful, because all manipulated variables are not adapted simultaneously. Pairing process links manipulated variables to control variables. Specifically, $c_{\mathsf{B}0}$ is linked to c^*_{B}, t_p to $c_{\mathsf{D}}(t_1)$ in Model 1, S is linked to $c_{\mathsf{D}}(t_1)$ in Model 2. In detail, the control schemes are the following:

- on-line

  Model 1 – On-line measurement of concentration c_{B} is used to push NCO variations to zero by PI controller in the arc 1. NCO are tracked alongside the arc 1 until c_{B} achieves optimal value of concentration c^*_{B}. This point determines the switching time t_p.

  Model 2 – Concentration c_{B} is measured on-line in this control scheme, too. Measurements are used to diminish NCO variations by PI controller, in the first arc 1. Alongside this arc PI controller tracks NCO till the switching time t_p, where reactors changes dynamics.

- off-line (run-to-run)

  Model 1 – Difficult on-line measurable variables can be updated off-line, in run-to-run manner. The switching time t_p is adapted by discrete I controller using actual and previous measurement of final concentration $c_{\mathsf{D}}(t_1)$. In the same way, measurements of compound at switching time $c_{\mathsf{B}}(t_p)$ are used to adapt initial concentration $c_{\mathsf{B}0}$.

  Model 2 – Here, measurements of final concentration $c_{\mathsf{D}}(t_1)$ are used to update the amount of addition S at switching time t_p.

The performance of these controllers is closely investigated in the next section. Additionally, the obtained solutions for both models are discussed.

6.4.4 Results

Optimisation results for the problem (6.5) are discussed in this section. It is assumed that uncertainty is presented in the form of time-invariant kinetic rates constants, activation energies, initial conditions, an addition or a concentration of addition. Their influence on process behaviour are studied separately as well as together. All uncertainties are positively or negatively adjusted by 10% for simulation purposes and then compared with the nominal solution for perturbed problem (further referred to as perturbed solution).

The contribution of particular perturbation on observed variables is discussed next:

activation energies Process model of the first dynamic contains two activation energies that can be subject to perturbations. If both of them are changed positively then the initial concentration $c_{\mathsf{B}0}$ of compound B rises, final t_1 and switching t_p times decrease compared to the nominal solution. If both of them are changed negatively then only final time t_1 decreases and other variables stay identical with nominal solution. In the case where one activation energy is bigger than the other one, behaviour is similar to previous scenarios depending on size of influence of each activation energy. For the both switching structures the behaviour is the same in the presence of perturbations.

kinetic rate constants Kinetic rate constants are presented in both dynamics, so they can be perturbed together or separately. It is assumed that they all are perturbed. Negative perturbation results in a shorter final time t_1, in the both control schemes. Positive perturbations result in an infeasibility in the both control schemes. It is caused by slow dynamic that cannot meet the active constraints at the perturbed value.

initial conditions Perturbations of initial conditions can only be positive except concentration of compound A which can be also negative. All tested possibilities are feasible and result in shorter final times for both control schemes in direct comparison to the nominal solution. In the case of Model 1 an initial concentration $c_{\mathsf{B}0}$ of compound B may decrease the switching time t_p as it is chosen as manipulated variable.

addition If an error in an amount of addition is positive, the final time t_1 decreases. If an error in an amount of addition is negative, optimisation becomes infeasible. Note that in this particular case, the performance index varies according to an error in an amount of addition. The second control scheme cannot be used in this scenario.

concentration of addition Positive perturbations in a concentration of an addition result in final time t_1 decrease for both control schemes. Negative perturbations in a concentration of the addition result in infeasibility for Model 1. They also result in a decrease of the final time t_1 and a smaller amount of addition S required in the process opposite to the amount in the nominal solution.

Combinations of perturbations in all variables are assumed in order to verify a performance of the control schemes. The amount of perturbation is randomly generated from 0 to 10% of the original values and with a random sign. Figure 6.3 and Figure 6.4 present the resulting performance of Model 1 and Model 2. It is obvious that both control schemes are feasible and that they manage to eliminate the influence of perturbations.

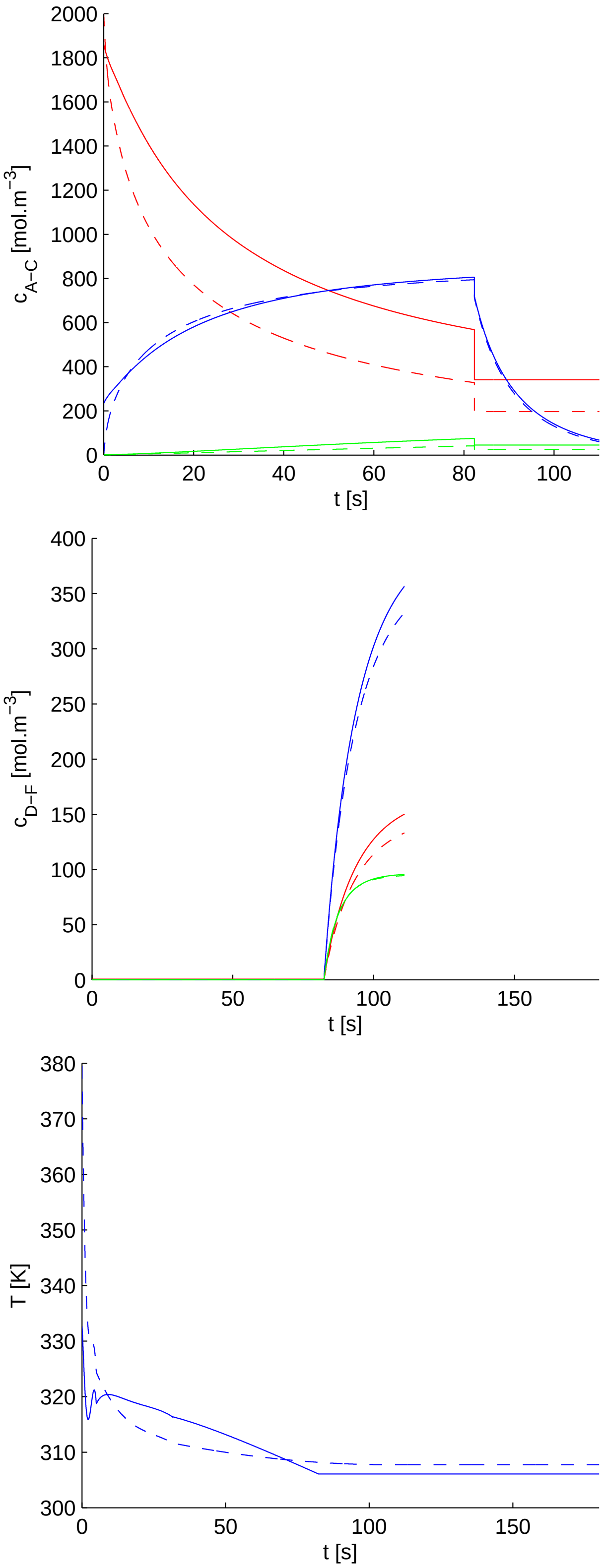

Figure 6.3: Perturbed solution (dashed line) and solution after NCO-tracking (solid line) for the switching structure of Model 1.

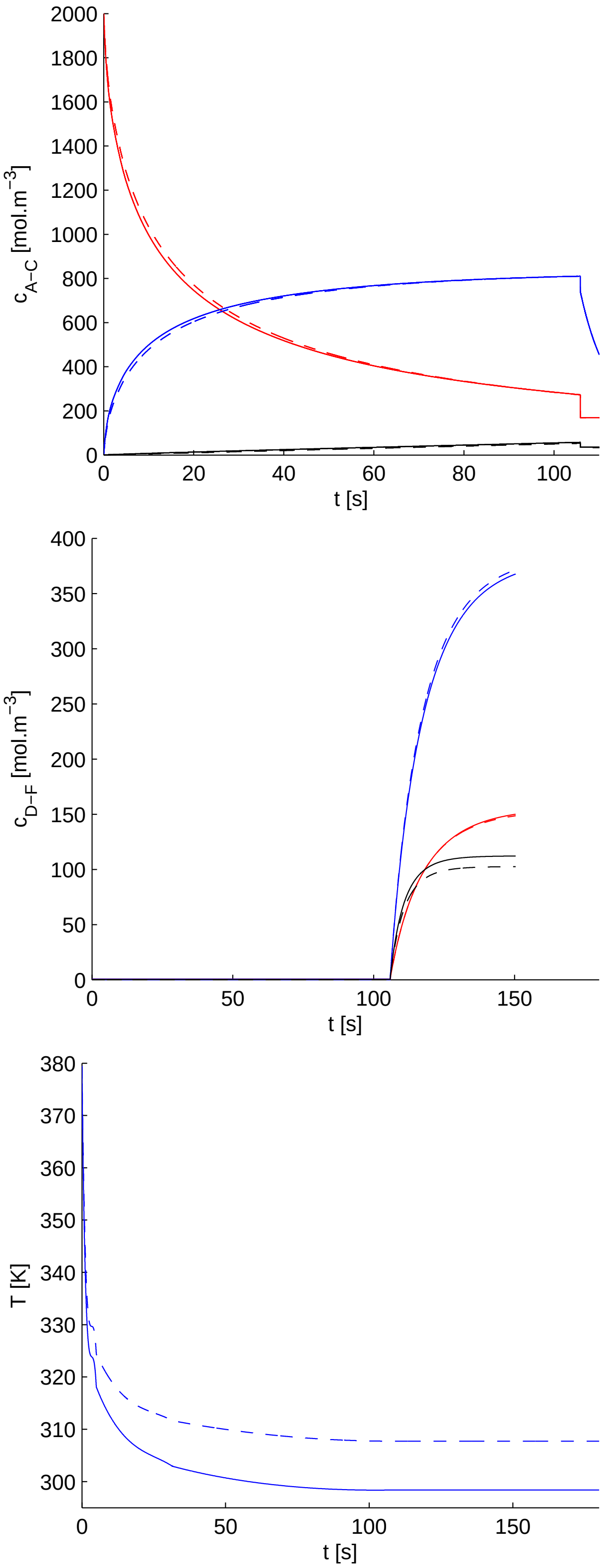

Figure 6.4: Perturbed solution (dashed line) and solution after NCO-tracking (solid line) for the switching structure of Model 2.

Chapter 7

NCO Associated with Sensitivities

The tracking of the sensitivity part of NCO is less straightforward in opposite to the tracking of NCO associated with constraints. In addition, unlike the constraints, the sensitivity terms cannot be estimated explicitly from the state measurements. An estimation of their values at a given time and for a given state of the system requires the known values of adjoint variables. These variables are part of the nominal solution.

This chapter introduces an approach of neighbouring-extremal control that corrects the nominal solution in a way that the variations between nominal and perturbed system are minimal. In addition, no further re-optimisation is required. This methodology has been subject of our research and it has been published in (Podmajerský, Chachuat, et al., 2011a; Podmajerský and Fikar, 2009b, 2011).

7.1 Neighbouring-extremal Control for Non-singular Problems without Terminal Constraints

7.1.1 Problem Formulation

Let us again consider the dynamic optimisation problem with simple input bound constraints:

$$\begin{aligned}\min_{\boldsymbol{u}} \mathcal{J} = \phi(\boldsymbol{x}(t_1)) + \int_{t_0}^{t_1} & L(\boldsymbol{x},\boldsymbol{u})\mathrm{d}t \\ \text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x},\boldsymbol{u}); & \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \\ t_0 \leq t \leq t_1 & \\ \boldsymbol{u}^L \leq \boldsymbol{u} \leq \boldsymbol{u}^U & \end{aligned} \tag{7.1}$$

7.1.2 Neighbouring-extremal Control

Let us now consider a small disturbance in the model parameters. This will result in the changes in the nominal solution $\boldsymbol{u}^*(t)$, $t_0 \leq t \leq t_1$. We will consider the first-order approximation for augmented optimal trajectory of a perturbed control

$$\boldsymbol{u}(t;\zeta) = \boldsymbol{u}^*(t) + \zeta\delta\boldsymbol{u}(t) + \boldsymbol{o}(\zeta), \tag{7.2}$$

and use the theory of neighbouring-extremal (Bryson and Ho, 1975) for computing the correction $\delta\boldsymbol{u}$ in such a way that the first-order variation of necessary conditions for optimality vanishes along the augmented control $\boldsymbol{u}^*(t) + \zeta\delta\boldsymbol{u}(t)$. The correction of $\delta\boldsymbol{u}$ is computed as the solution of the variational LQ minimum problem (Breakwell, Speyer, et al., 1963; Kelley, Kopp, et al., 1963)

$$\min \delta^2 J(\delta\boldsymbol{u}) = \frac{1}{2}\delta\boldsymbol{x}(t_1)^T\Phi^*_{\boldsymbol{xx}}\delta\boldsymbol{x}(t_1) + \frac{1}{2}\int_{t_0}^{t_1}\begin{pmatrix}\delta\boldsymbol{x}\\ \delta\boldsymbol{u}\end{pmatrix}^T\begin{pmatrix}\mathcal{H}^*_{\boldsymbol{xx}} & \mathcal{H}^*_{\boldsymbol{xu}}\\ \mathcal{H}^*_{\boldsymbol{ux}} & \mathcal{H}^*_{\boldsymbol{uu}}\end{pmatrix}\begin{pmatrix}\delta\boldsymbol{x}\\ \delta\boldsymbol{u}\end{pmatrix}\mathrm{d}t$$

s.t.

$$\delta\dot{\boldsymbol{x}} = \boldsymbol{F}^*_{\boldsymbol{x}}\delta\boldsymbol{x} + \boldsymbol{F}^*_{\boldsymbol{u}}\delta\boldsymbol{u}; \tag{7.3}$$

$$\delta\boldsymbol{x}(t_0) = \delta\boldsymbol{x}_0 \tag{7.4}$$

$$\boldsymbol{u}^L - \boldsymbol{u}^*(t) \leq \delta\boldsymbol{u}(t) \leq \boldsymbol{u}^U - \boldsymbol{u}^*(t) \tag{7.5}$$

that corresponds to minimisation of the second-order variation of the cost functional subject to the linearised dynamics. A superscript * (e.g. $\mathcal{H}^*_{\boldsymbol{uu}}$) means that the variable is evaluated upon nominal trajectories $\boldsymbol{u}^*(t), \boldsymbol{x}^*(t), \boldsymbol{\lambda}^*(t)$, for $t_0 \leq t \leq t_1$. A perturbed optimal control $\boldsymbol{u}(t;\zeta)$ exists in a neighbourhood of $\zeta = 0$, provided that the LQ problem (7.3)–(7.5) itself has an optimal solution (Pesh, 1990). The control variation $\delta\boldsymbol{u}$ satisfying the strengthened Legendre-Clebsch condition of positive definiteness $\mathcal{H}^*_{\boldsymbol{uu}} > \boldsymbol{0}$ and for unconstrained problems $\mu_i^L = 0$, $\mu_i^U = 0$, $i \in \{1, ..., n_u\}$ is then given by:

$$\delta\boldsymbol{u}(t) = -(\mathcal{H}^*_{\boldsymbol{uu}})^{-1}(\mathcal{H}^*_{\boldsymbol{ux}}\delta\boldsymbol{x}(t) + \boldsymbol{F}^{*T}_{\boldsymbol{u}}\delta\boldsymbol{\lambda}(t)) \tag{7.6}$$

where $\delta\boldsymbol{x}(t)$ and $\delta\boldsymbol{\lambda}(t)$ satisfy the following two-point boundary-value problem (TPBVP)

$$\begin{pmatrix}\delta\dot{\boldsymbol{x}}(t)\\ \delta\dot{\boldsymbol{\lambda}}(t)\end{pmatrix} = \begin{pmatrix}\boldsymbol{T}_{\boldsymbol{x}} & \boldsymbol{T}_{\boldsymbol{\lambda}}\end{pmatrix}\begin{pmatrix}\delta\boldsymbol{x}(t)\\ \delta\boldsymbol{\lambda}(t)\end{pmatrix} \tag{7.7}$$

$$\delta\boldsymbol{x}(t_0) = \delta\boldsymbol{x}_0, \quad \delta\boldsymbol{\lambda}(t_1) = \boldsymbol{\Phi}^*_{\boldsymbol{xx}}\delta\boldsymbol{x}(t_1) \tag{7.8}$$

where $\boldsymbol{T}_{\boldsymbol{x}}$ and $\boldsymbol{T}_{\boldsymbol{\lambda}}$ along unconstrained arcs are given by:

$$\boldsymbol{T}_{\boldsymbol{x}} = \begin{pmatrix}\boldsymbol{F}^*_{\boldsymbol{x}} - \boldsymbol{F}^*_{\boldsymbol{u}}(\mathcal{H}^*_{\boldsymbol{uu}})^{-1}\mathcal{H}^*_{\boldsymbol{ux}}\\ -\mathcal{H}^*_{\boldsymbol{xx}} + \mathcal{H}^*_{\boldsymbol{xu}}(\mathcal{H}^*_{\boldsymbol{uu}})^{-1}\mathcal{H}^*_{\boldsymbol{ux}}\end{pmatrix} \tag{7.9}$$

$$\boldsymbol{T}_{\boldsymbol{\lambda}} = \begin{pmatrix}-\boldsymbol{F}^*_{\boldsymbol{u}}(\mathcal{H}^*_{\boldsymbol{uu}})^{-1}\boldsymbol{F}^{*T}_{\boldsymbol{u}}\\ -(\boldsymbol{F}^*_{\boldsymbol{x}} - \boldsymbol{F}^*_{\boldsymbol{u}}(\mathcal{H}^*_{\boldsymbol{uu}})^{-1}\mathcal{H}^*_{\boldsymbol{ux}})^T\end{pmatrix} \tag{7.10}$$

and along constrained arc by:

$$\boldsymbol{T}_{\boldsymbol{x}} = \begin{pmatrix}\boldsymbol{F}^*_{\boldsymbol{x}}\\ -\mathcal{H}^*_{\boldsymbol{xx}}\end{pmatrix} \tag{7.11}$$

$$\boldsymbol{T}_{\boldsymbol{\lambda}} = \begin{pmatrix}\boldsymbol{0}\\ -\boldsymbol{F}^{*T}_{\boldsymbol{x}}\end{pmatrix} \tag{7.12}$$

7.1.3 Numerical Computation

The linear TPBVP (7.7)–(7.8) can be used to calculate the NE control correction $\delta \boldsymbol{u}(t)$, $t_0 \leq t \leq t_1$, in one of two situations:

i. The variations $\delta \boldsymbol{x}_0$ are available continuously in time, in which case the backward sweep method (Bryson and Ho, 1975) is used to derive an explicit feedback control law.

ii. The initial state variations $\delta \boldsymbol{x}_0$ are available at discrete time instants, in which case the discrete feedback control is obtained by re-solving of the TPBVP.

Backward Sweep Method

This methodology assumes a linear relation between the state and adjoint variables:

$$\delta \boldsymbol{\lambda}(t) = \boldsymbol{S}(t)\delta \boldsymbol{x}(t) \tag{7.13}$$

where $\boldsymbol{S}(t) \in \mathcal{R}^{n_x \times n_x}$ is a matrix time function with following terminal condition

$$\boldsymbol{S}(t_1) = \boldsymbol{\Phi}^*_{\boldsymbol{xx}}. \tag{7.14}$$

Further differentiation of (7.13) with respect to time gives

$$\delta \dot{\boldsymbol{\lambda}}(t) = \dot{\boldsymbol{S}}(t)\delta \boldsymbol{x}(t) + \boldsymbol{S}(t)\delta \dot{\boldsymbol{x}}(t). \tag{7.15}$$

Equating the terms in (7.15) to those in (7.7)–(7.8) and in (7.13) results in the following Riccati equation:

$$\dot{\boldsymbol{S}}(t) = -\boldsymbol{S}\boldsymbol{F}^*_{\boldsymbol{x}} - \boldsymbol{F}^{T*}_{\boldsymbol{x}}\boldsymbol{S} + \boldsymbol{S}\mathcal{H}^*_{\boldsymbol{xx}}\boldsymbol{S}. \tag{7.16}$$

Next, the combination of (7.6) and (7.13) produces:

$$\delta \boldsymbol{u}(t) = -\boldsymbol{K}(t)\delta \boldsymbol{x}(t). \tag{7.17}$$

Putting (7.17) into (7.16), the system of Riccati equations becomes:

$$\begin{aligned} \delta \boldsymbol{u} &= -\boldsymbol{K}\delta \boldsymbol{x} && (7.18)\\ \boldsymbol{K} &= (\mathcal{H}^*_{\boldsymbol{uu}})^{-1}(\mathcal{H}^*_{\boldsymbol{ux}} + \boldsymbol{F}^{*T}_{\boldsymbol{u}}\boldsymbol{S}) && (7.19)\\ \dot{\boldsymbol{S}} &= -\mathcal{H}^*_{\boldsymbol{xx}} - \boldsymbol{S}\boldsymbol{F}^*_{\boldsymbol{x}} - \boldsymbol{F}^{*T}_{\boldsymbol{x}}\boldsymbol{S} + (-\mathcal{H}^*_{\boldsymbol{xu}} + \boldsymbol{S}\boldsymbol{F}^*_{\boldsymbol{u}})\boldsymbol{K} && (7.20)\\ \boldsymbol{S}(t_1) &= \boldsymbol{\Phi}^*_{\boldsymbol{xx}} && (7.21) \end{aligned}$$

Shooting Method

The linear TPBVP (7.7)–(7.8) can be also rewritten as

$$\begin{pmatrix} \delta \dot{\boldsymbol{x}}(t) \\ \delta \dot{\boldsymbol{\lambda}}(t) \end{pmatrix} = \underbrace{\begin{pmatrix} \boldsymbol{\alpha}(t) & -\boldsymbol{\beta}(t) \\ -\boldsymbol{\gamma}(t) & -\boldsymbol{\alpha}(t) \end{pmatrix}}_{=:\, \boldsymbol{\Delta}(t)} \begin{pmatrix} \delta \boldsymbol{x}(t) \\ \delta \boldsymbol{\lambda}(t) \end{pmatrix}, \tag{7.22}$$

with the boundary conditions:

$$\begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \delta\boldsymbol{x}(t_0) \\ \delta\boldsymbol{\lambda}(t_0) \end{pmatrix} + \begin{pmatrix} \boldsymbol{0} & \boldsymbol{0} \\ -\left[\boldsymbol{\phi}^*_{\boldsymbol{xx}}\right]_{t_1} & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \delta\boldsymbol{x}(t_1) \\ \delta\boldsymbol{\lambda}(t_1) \end{pmatrix} = \begin{pmatrix} \delta\boldsymbol{x}_0 \\ \boldsymbol{0} \end{pmatrix}. \tag{7.23}$$

The shooting approach proceeds by guessing the missing initial (or terminal) conditions in (7.23), and adjusting them in such a way that the corresponding terminal (or initial) conditions are satisfied (see, e.g., Pesch, 1989a; Pesch, 1989b). Given the guess $\delta\boldsymbol{\lambda}(0) = \delta\boldsymbol{\lambda}_0$ for the adjoints variations at initial time $t = t_0$, the (unique) solution to the linear ODE system (7.22) is of the form:

$$\begin{pmatrix} \delta\dot{\boldsymbol{x}}(t;\delta\boldsymbol{\lambda}_0) \\ \delta\dot{\boldsymbol{\lambda}}(t;\delta\boldsymbol{\lambda}_0) \end{pmatrix} = \underbrace{\begin{pmatrix} \boldsymbol{\Upsilon}_1(t;t_0) & \boldsymbol{\Upsilon}_2(t;t_0) \\ \boldsymbol{\Upsilon}_3(t;t_0) & \boldsymbol{\Upsilon}_4(t;t_0) \end{pmatrix}}_{=:\ \boldsymbol{\Upsilon}(t;t_0)} \begin{pmatrix} \delta\boldsymbol{x}_0 \\ \delta\boldsymbol{\lambda}_0 \end{pmatrix}, \tag{7.24}$$

where the transition matrix $\boldsymbol{\Upsilon}(t;t_0)$ is obtained as the solution to the initial value problem

$$\frac{\partial}{\partial t}\boldsymbol{\Upsilon}(t;t_0) = \boldsymbol{\Delta}(t)\boldsymbol{\Upsilon}(t;t_0),\ t_0 \le t \le t_1; \quad \boldsymbol{\Upsilon}(t_0;t_0) = \boldsymbol{I}. \tag{7.25}$$

Substituting (7.24) into (7.23) leads to the following linear system in the variable $\delta\boldsymbol{\lambda}(0)$:

$$\left[\boldsymbol{\phi}^*_{\boldsymbol{xx}}\right]_{t_1}\boldsymbol{\Upsilon}_2(t_1;t_0) - \boldsymbol{\Upsilon}_4(t_1;t_0)\delta\boldsymbol{\lambda}(t_0) = -\left[\boldsymbol{\phi}^*_{\boldsymbol{xx}}\right]_{t_1}\boldsymbol{\Upsilon}_1(t_1;t_0) - \boldsymbol{\Upsilon}_3(t_1;t_0)\delta\boldsymbol{x}_0. \tag{7.26}$$

For given initial state $\delta\boldsymbol{x}_0$, the solution to the linear system (7.26) provides the corresponding initial adjoint $\delta\boldsymbol{\lambda}(t_0)$. Finally, the NE control variation can be calculated from (7.6) as

$$\delta\boldsymbol{u}(t) = -(\mathcal{H}^*_{\boldsymbol{uu}})^{-1}\begin{pmatrix} \mathcal{H}^*_{\boldsymbol{ux}} & \boldsymbol{F}^{*T}_{\boldsymbol{u}} \end{pmatrix} \boldsymbol{\Upsilon}(t;t_0) \begin{pmatrix} \delta\boldsymbol{x}_0 \\ \delta\boldsymbol{\lambda}(t_0) \end{pmatrix}. \tag{7.27}$$

Implementation Issues

It is implicitly assumed for constrained control sequence that the uncertainty is sufficiently small for the perturbed optimal control to have the same sequence of constrained and unconstrained arcs as the nominal solution. Clearly, at each switching point t_k between an unconstrained and a constrained arcs, a continuity of control, state and adjoint variables must be preserved:

$$\boldsymbol{x}^*(t_k^+) = \boldsymbol{x}^*(t_k^-), \quad \boldsymbol{\lambda}^*(t_k^+) = \boldsymbol{\lambda}^*(t_k^-), \quad \boldsymbol{u}^*(t_k^+) = \boldsymbol{u}^*(t_k^-) \tag{7.28}$$

For example, at a switching point between a lower bound and an interior arc, the value of control on lower bound matches the value of control in the interior arc $u^{\mathcal{H}} = u^L$. Here, $u^{\mathcal{H}}$ represents the control obtained from solving the condition $\mathcal{H}_{\boldsymbol{u}} = \boldsymbol{0}$. Taking the previous into account, NE control is obtained by solving either TPBVP or a set of Riccati equations with possible discontinuities at junction times t_k between constrained and unconstrained arcs. In practice, this assumption does not cause an apparent performance loss. Also note that control corrections along unconstrained arcs may take values out of their limits, hence such controls need to be saturated.

7.1.4 Van de Vusse Reaction

Plant Model

A chemical reactor of van de Vusse scheme (Vusse, 1964) is considered in this example. In the reactor, side and follow-up reactions take place where desired cyclopentenol (B) is produced from cyclopentadiene (A) by acid-catalysed electrophilic addition of water in dilute solution. In addition, cyclopentanediol (C) is consecutive product of cyclopentenol (B) and addition of another water molecule, and dicyclopentadiene (D) is a side product of strong Diels-Alder reaction between the educt and the product.

The plant model presented in Klatt and Engell, 1993 consists of material balances of the reactant (A) and the product (B) as well as energy balances of the plant and the cooling jacket as follows:

$$\dot{c_{\mathsf{A}}} = -k_1(T)c_{\mathsf{A}} - k_2(T)c_{\mathsf{A}}^2 + (c_{in} - c_{\mathsf{A}})u_1, \tag{7.29a}$$

$$\dot{c_{\mathsf{B}}} = k_1(T)(c_{\mathsf{A}} - c_{\mathsf{B}}) - c_{\mathsf{B}}u_1, \tag{7.29b}$$

$$\dot{T} = h_r(c_{\mathsf{A}}, c_{\mathsf{B}}, T) + \alpha(T_c - T) + (T_{in} - T)u_1, \tag{7.29c}$$

$$\dot{T_c} = \beta(T - T_c) + \gamma u_2 \tag{7.29d}$$

with reaction enthalpy given as

$$h_r(c_{\mathsf{A}}, c_{\mathsf{B}}, T) = -\sigma[k_1(T)(c_{\mathsf{A}}\Delta H_{\mathsf{AB}} + c_{\mathsf{B}}\Delta H_{\mathsf{BC}}) + k_2(T)c_{\mathsf{A}}^2\Delta H_{\mathsf{AD}}] \tag{7.30}$$

and kinetic rate constants are expressed as Arrhenius functions of temperature in °C.

$$k_i(T) = k_{i0}\mathrm{e}^{-\frac{\frac{E_i}{R}}{T}}, \quad i = 1, 2. \tag{7.31}$$

Table 7.1: Model parameters and the main set-points values

$\alpha = 30.8285$	$[\mathrm{h}^{-1}]$	$\beta = 86.688$	$[\mathrm{h}^{-1}]$
$\gamma = 0.1$	$[\mathrm{K\,kJ}^{-1}]$	$\sigma = 3.556 \times 10^{-4}$	$[\mathrm{m}^3\,\mathrm{K\,kJ}^{-1}]$
$k_{10} = 1.287 \pm 20\% \times 10^12$	$[\mathrm{h}^{-1}]$	$\frac{E_1}{R} = 9758.3$	
$k_{20} = 9.043 \pm 20\% \times 10^6$	$[\mathrm{m}^3\,\mathrm{mol}^{-1}\,\mathrm{h}^{-1}]$	$\frac{E_2}{R} = 8560$	
$\Delta H_{\mathsf{AB}} = 4.2$	$[\mathrm{kJ\,mol}^{-1}]$	$\Delta H_{\mathsf{BC}} = -11$	$[\mathrm{kJ\,mol}^{-1}]$
$\Delta H_{\mathsf{AD}} = -41.85$	$[\mathrm{kJ\,mol}^{-1}]$	$c_{in} = 5100 \pm 20\%$	$[\mathrm{mol\,m}^{-3}]$
$T_{in} = 104.9$	$[\mathrm{K}]$	$c_{\mathsf{A},sp_1} = 3517.5$	$[\mathrm{mol\,m}^{-3}]$
$c_{\mathsf{B},sp_1} = 740$	$[\mathrm{mol\,m}^{-3}]$	$T_{sp_1} = 87$	$[\mathrm{K}]$
$T_{c,sp_1} = 79.8$	$[\mathrm{K}]$	$u_{1,sp_1} = 8.256$	$[\mathrm{h}^{-1}]$
$u_{2,sp_1} = -6239$	$[\mathrm{kJ\,h}^{-1}]$	$c_{\mathsf{A},sp_2} = 2985$	$[\mathrm{mol\,m}^{-3}]$
$c_{\mathsf{B},sp_2} = 960$	$[\mathrm{mol\,m}^{-3}]$	$T_{sp_2} = 106$	$[\mathrm{K}]$
$T_{c,sp_2} = 100.7$	$[\mathrm{K}]$	$u_{1,sp_2} = 18.037$	$[\mathrm{h}^{-1}]$
$u_{2,sp_2} = -4556$	$[\mathrm{kJ\,h}^{-1}]$		

The model parameters are defined in Table 7.1, the state variables are $\boldsymbol{x} = [c_{\mathsf{A}}, c_{\mathsf{B}}, T, T_c]^T$. The controls are input flow rate q normalised by the volume of the plant V_R and cooling system capacity $\dot{Q}$. Both inputs are constrained as

$$u_1 = \frac{q}{V_R}, \quad 5\,\mathrm{h}^{-1} \leq u_1 \leq 35\,\mathrm{h}^{-1} \tag{7.32a}$$

$$u_2 = \dot{Q}, \quad -8500\,\mathrm{kJ.h}^{-1} \leq u_1 \leq 0\,\mathrm{kJ.h}^{-1} \tag{7.32b}$$

The product concentration and the plant temperature were chosen as controlled outputs

$$y_1 = c_{\mathsf{B}}, \qquad y_2 = T. \tag{7.33}$$

The aim of the optimisation problem is to drive reactor's operational conditions from the original steady-state to another operational point. Note that we study set-point transition only and we do not stabilise the process around final set-point. The particular numeric values of states and inputs at the operational points are summarised in Table 7.1. The transition is performed with several scenarios, whereby the desired final point is always reached without violating input constraints. Thus, the performance index is defined as finite time LQ integral functional where the normalised tracking error variations between original and new stationary point are driven to zero in a finite time $t_1 = 20\,\mathrm{min}$. The cost function then reads

$$\min_u \mathcal{J}_0 = \int_{t_0}^{t_f} (\hat{\boldsymbol{y}}^T \boldsymbol{Q}_I \hat{\boldsymbol{y}} + \hat{\boldsymbol{u}}^T \boldsymbol{R}_I \hat{\boldsymbol{u}})\mathrm{d}t \tag{7.34}$$

where

$$\hat{\boldsymbol{y}} = \begin{bmatrix} \frac{c_{\mathsf{B}} - c_{\mathsf{B},sp_2}}{c_{\mathsf{B},sp_2}} & \frac{T - T_{sp_2}}{T_{sp_2}} \end{bmatrix}^T \tag{7.35}$$

$$\hat{\boldsymbol{u}} = \begin{bmatrix} \frac{u_1 - u_{1,sp_2}}{u_{1,sp_2}} & \frac{u_2 - u_{2,sp_2}}{u_{2,sp_2}} \end{bmatrix}^T \tag{7.36}$$

and matrices $\boldsymbol{Q}_I$ and $\boldsymbol{R}_I$ are positive-definite and symmetric weights

$$\boldsymbol{Q}_I = \begin{bmatrix} q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & q_n \end{bmatrix}, \quad \boldsymbol{R}_I = \begin{bmatrix} r_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_n \end{bmatrix} \tag{7.37}$$

Open-loop Optimal Control

To find the optimal sequence of arcs, a numerical solution of dynamic optimisation problem (7.29)–(7.34) has been obtained. In-house dynamic optimisation package has been employed that is based on CVP approach and implemented in MATLAB environment: SUNDIALS toolbox (Hindmarsh, Brown, et al., 2005) for the forward and backward numerical integrations and MATLAB version of IPOPT solver (Wächter and Biegler, 2006) for NLP solutions. For this particular case study, the both control inputs has been piecewise parametrised on 40 stages of equidistant width (30 s).

The nominal open-loop control solution of more aggressive control scenario is shown in the bottom two plots of Figure 7.1 (dash-dotted line). The responses of perturbed process to the nominal control are displayed in top two plots in Figure 7.1. Observe that u_1 starts on upper bound and u_2 on lower bound and then they are followed by an unconstrained arc. Similarly, the nominal control solution of less aggressive control scenario is depicted in the bottom two plots of Figure 7.2. See that upper bound of u_1 and lower bound for u_2 are shorter in comparison to the more aggressive scenario. In both cases, the unconstrained arcs are given by the necessary conditions (see (2.17b)):

$$\mathcal{H}_{u_1} = 2\left(\frac{u_1 - u_{1,sp_2}}{u_{1,sp_2}^2}\right) r_1 + \lambda_{c_\mathsf{A}}(c_{in} - c_\mathsf{A}) - \lambda_{c_\mathsf{B}} c_\mathsf{B} + \lambda_T(T_{in} - T) = 0 \tag{7.38}$$

$$\mathcal{H}_{u_2} = 2\left(\frac{u_2 - u_{2,sp_2}}{u_{2,sp_2}^2}\right) r_2 + \lambda_{T_c}\gamma = 0 \tag{7.39}$$

One can express the nominal control solution from these equations as:

$$u_1^* = \frac{\frac{2}{u_{1,sp_2} r_1} - \lambda_{c_\mathsf{A}}(c_{in} - c_\mathsf{A})}{\frac{2r_1}{u_{1,sp_2}^2}} + \frac{\lambda_{c_\mathsf{B}} c_\mathsf{B} - \lambda_T(T_{in} - T)}{\frac{2r_1}{u_{1,sp_2}^2}} \tag{7.40}$$

$$u_2^* = \frac{\frac{2}{u_{2,sp_2} r_2} - \lambda_{T_c}\gamma}{\frac{2r_2}{u_{2,sp_2}^2}} \tag{7.41}$$

Note that nominal control trajectories u_1^* and u_2^* are computed iteratively as adjoints become unstable during forward integration. The numerical procedure can be the following: dynamic process (7.29)–(7.31) is solved in forward direction where the controls are explicitly given from (7.40)–(7.41). The unknown adjoint variables $\boldsymbol{\lambda}_0^*$ are taken from the CVP solution and then they are approximated during forward integration. Subsequently, in next iteration steps, the adjoint variables $\boldsymbol{\lambda}_k^*$ are corrected during backward integration. In the final iteration, the final value of the computed adjoints $\boldsymbol{\lambda}_n^*(t_1)$ must equal adjoints given by the term $\boldsymbol{\lambda}(t_1) = \boldsymbol{\phi}_{\boldsymbol{xx}}^*$ given by the optimal problem.

Neighbouring-extremal Feedback Control

The standard approach of the real-time optimisation consists of process model update using available measurements and followed by numerical re-optimisation that provides updated inputs to the plant. Instead of re-optimisation, the so called NCO-tracking approach is used here. The main idea is based on the fact that optimality requires to meet the necessary conditions for optimality. NCO-tracking provides almost optimal operation conditions via precomputed state-feedback without re-solving of DOP. This approach cancels out the residual variations around nominal solution caused by uncertainty. Note that the switching times between arcs in the nominal solution are considered fixed. They do not vary with uncertainty. Also note that the NE controller updates only interior arcs.

To test the performance of proposed NE controller two scenarios are studied: more and less aggressive control. Performance is demonstrated on perturbed process with various combination of parameter uncertainty in an inlet concentration c_{in}, in a vector of initial conditions $\boldsymbol{x}_0$, and in kinetic rate constants k_{10}, k_{20} that may vary in range of ±20%. The weight constants for controls are $r_1 = r_2 = 1$ and the state penalisation is $q_1 = q_2 = 500$ and $q_1 = q_2 = 200$ for more aggressive and for less aggressive scenario, respectively.

The corresponding control updates and their responses for several perturbation scenarios are plotted in Figure 7.1–Figure 7.3 for more aggressive, and in Figure 7.2–Figure 7.4 for less aggressive scenario, respectively. The open-loop implementation of the nominal solution is clearly unable to deal with the perturbations. In addition, the desired set-points are not reached in any perturbation case. In contrast, the proposed NE controller recovers an influence of the perturbations. The reactor is controlled in almost optimal way and it ends up in close proximity of the desired set-point. This can be observed in both control scenarios. NE controller roughly copies the control actions of the optimal inputs for perturbed problem. Similar performance of NE controller can be observed in the next scenarios with various perturbations:

C1 : $\Delta c_{in} = 20\%, \Delta k_{10} = -20\%, \Delta k_{20} = 10\%, \Delta x_0 = -20\%$

C2 : $\Delta c_{in} = -10\%, \Delta k_{10} = -20\%, \Delta k_{20} = 10\%$

C3 : $\Delta c_{in} = 10\%, \Delta k_{10} = -10\%, \Delta k_{20} = -20\%$

C4 : $\Delta c_{in} = -20\%, \Delta x_0 = 10\%$.

7.2 Neighbouring-extremal Control for Non-singular Problems with Terminal Constraints

7.2.1 Problem Formulation

Throughout this section, the following dynamic optimisation problem with terminal bound constraints is considered:

$$\min_{u} \ \mathcal{J} = \phi(\boldsymbol{x}(t_1)) + \int_{t_0}^{t_1} L(\boldsymbol{x}, \boldsymbol{u})\mathrm{d}t \tag{7.42}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}); \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0$$

$$t_0 \le t \le t_1$$

$$\boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \le \boldsymbol{0} \tag{7.43}$$

$$\boldsymbol{u}^L \le \boldsymbol{u} \le \boldsymbol{u}^U. \tag{7.44}$$

All the functions in (7.42)–(7.44) are assumed to be two times continuously differentiable with respect to all their arguments. Further, it is assumed that there exists an unique nominal solution $\boldsymbol{u}^*(t)$, i.e. the problem is feasible.

7.2.2 Necessary Conditions for Optimality

Following Bryson and Ho, 1975, the Hamiltonian function $\mathcal{H}$ is defined as follows:

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}, \boldsymbol{\mu}^L, \boldsymbol{\mu}^U) = L(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})^T\boldsymbol{\lambda} + \boldsymbol{\mu}^{L^T}(\boldsymbol{u}^L - \boldsymbol{u}) + \boldsymbol{\mu}^{U^T}(\boldsymbol{u} - \boldsymbol{u}^U), \tag{7.45}$$

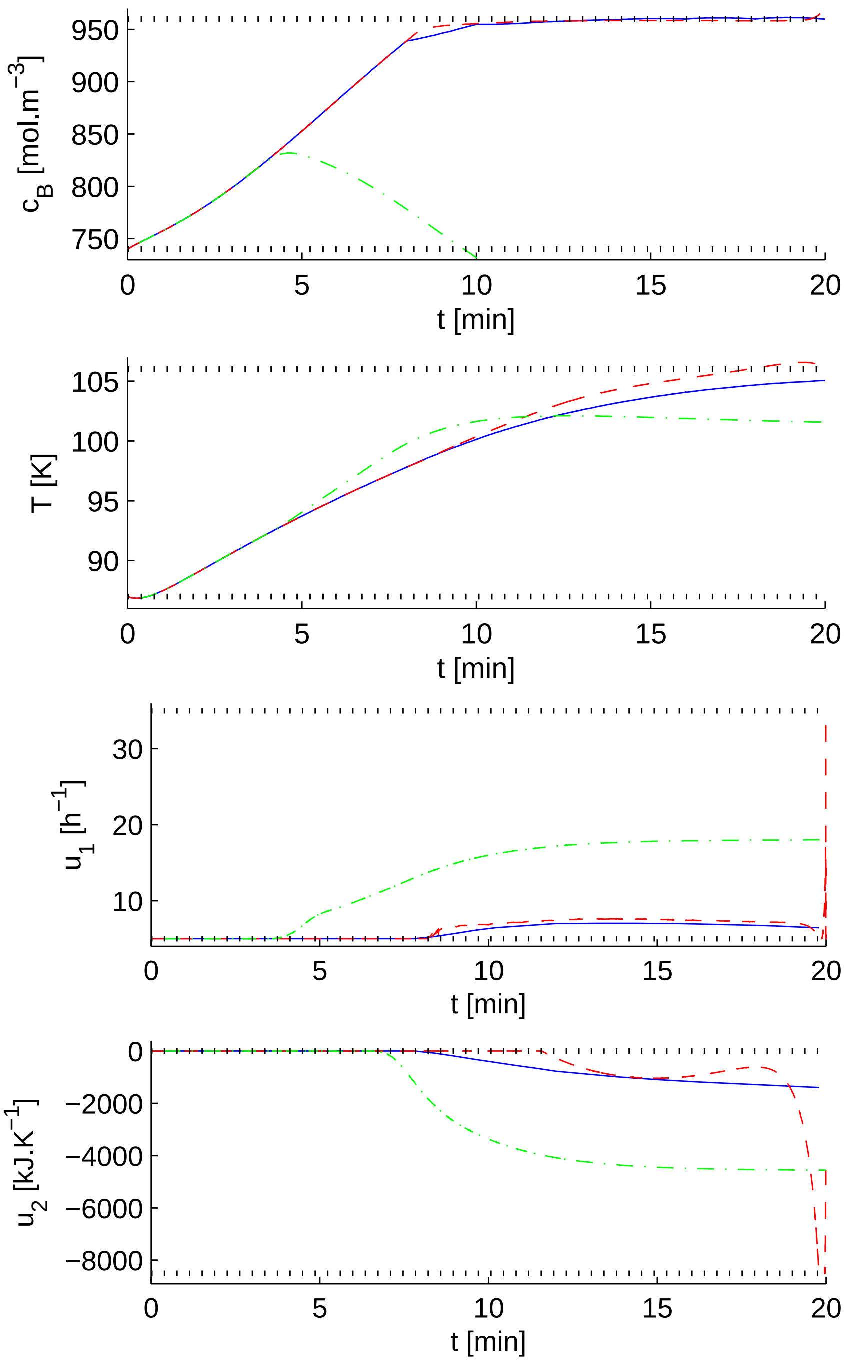

Figure 7.1: More aggressive control with the perturbation scenario: $\Delta c_{in} = -20\%, \Delta k_{10} = -20\%, \Delta k_{20} = -20\%$. **Dashed line:** NE controller; **Solid line:** optimal inputs to the perturbed problem; **Dash-dotted line:** open-loop.

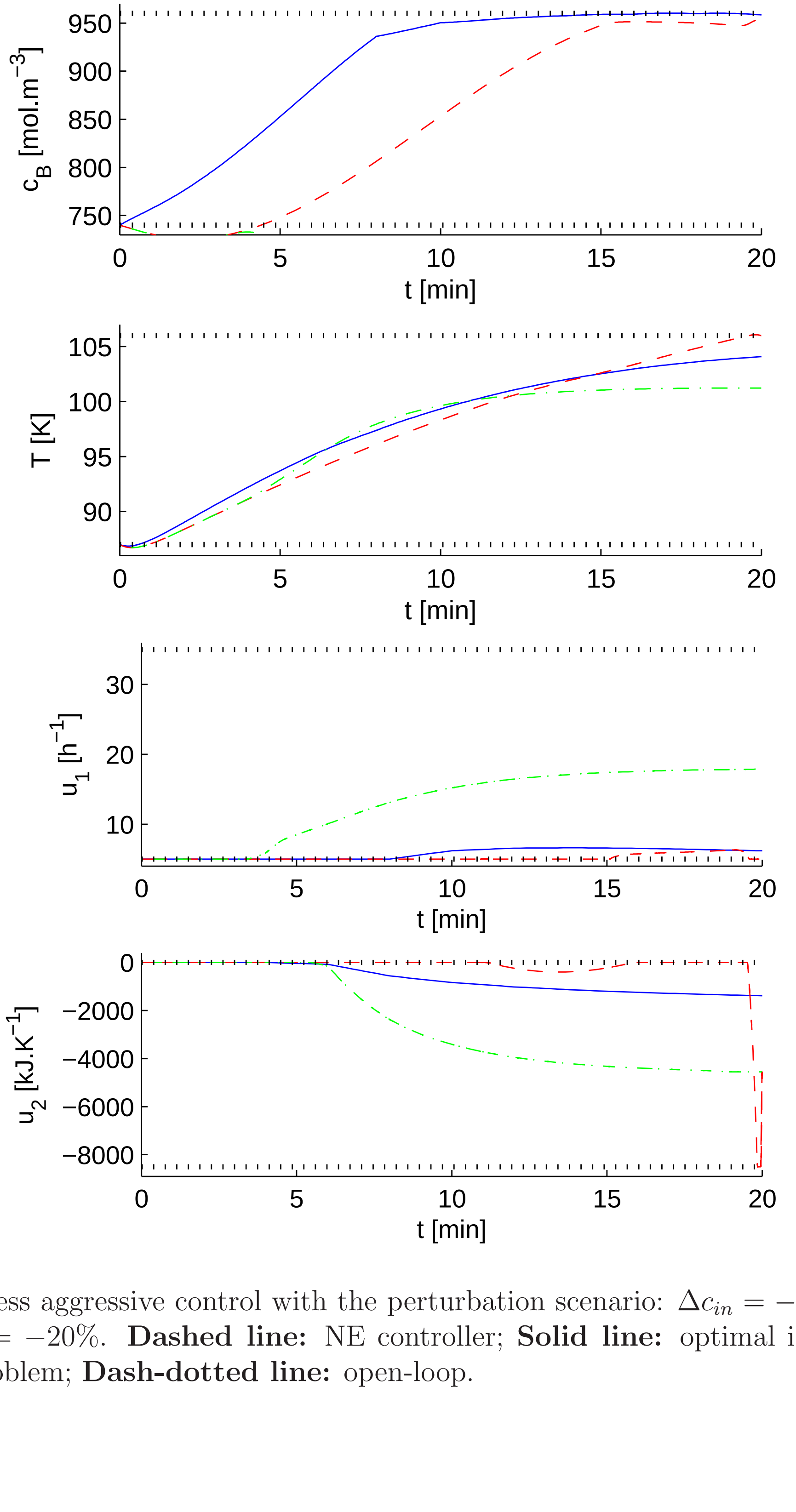

Figure 7.2: Less aggressive control with the perturbation scenario: $\Delta c_{in} = -20\%, \Delta k_{10} = -20\%, \Delta k_{20} = -20\%$. **Dashed line:** NE controller; **Solid line:** optimal inputs to the perturbed problem; **Dash-dotted line:** open-loop.

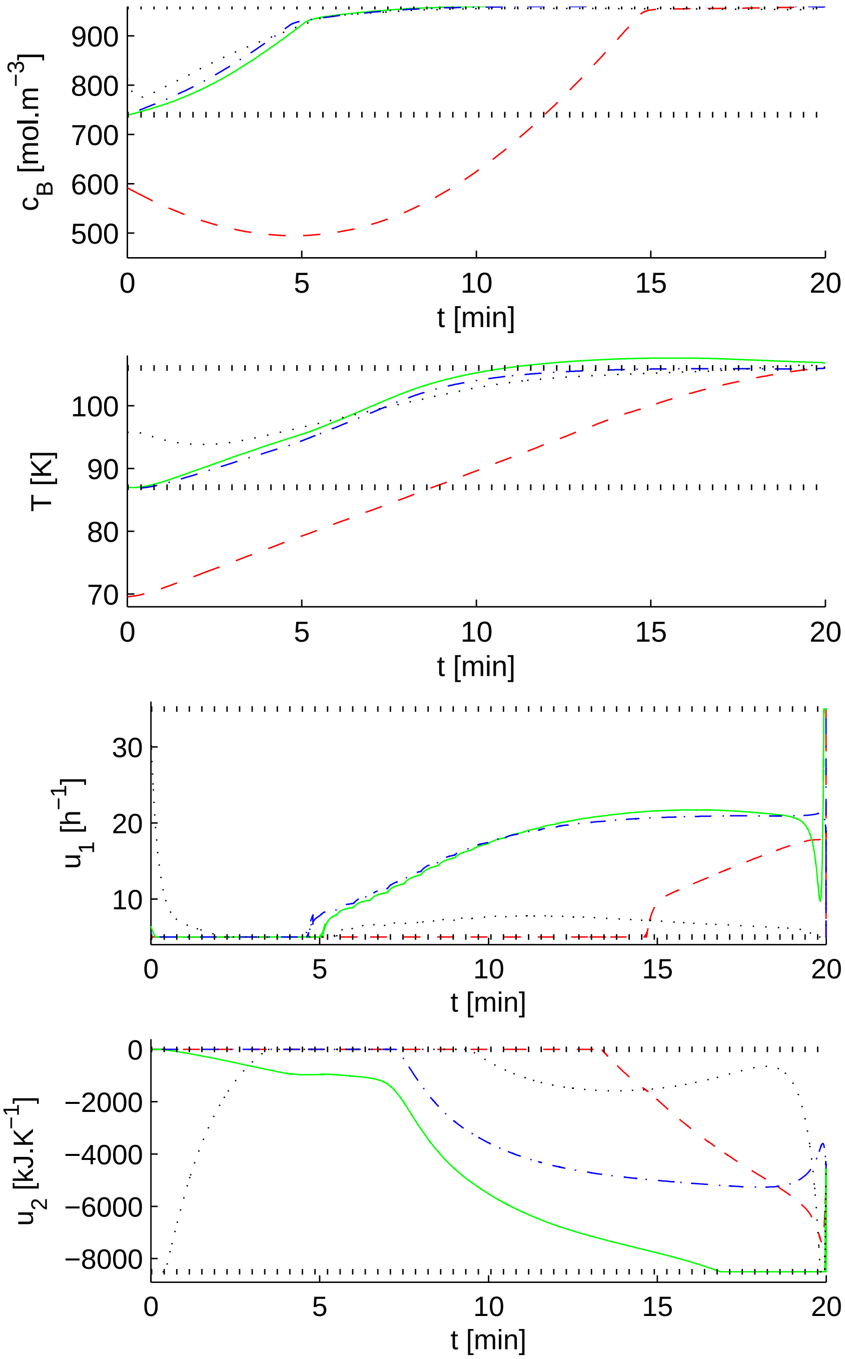

Figure 7.3: Performance of NCO-tracking with more aggressive control. **Dashed line:** C1, **solid line:** C2, **dash-dotted line:** C3, **dotted line:** C4.

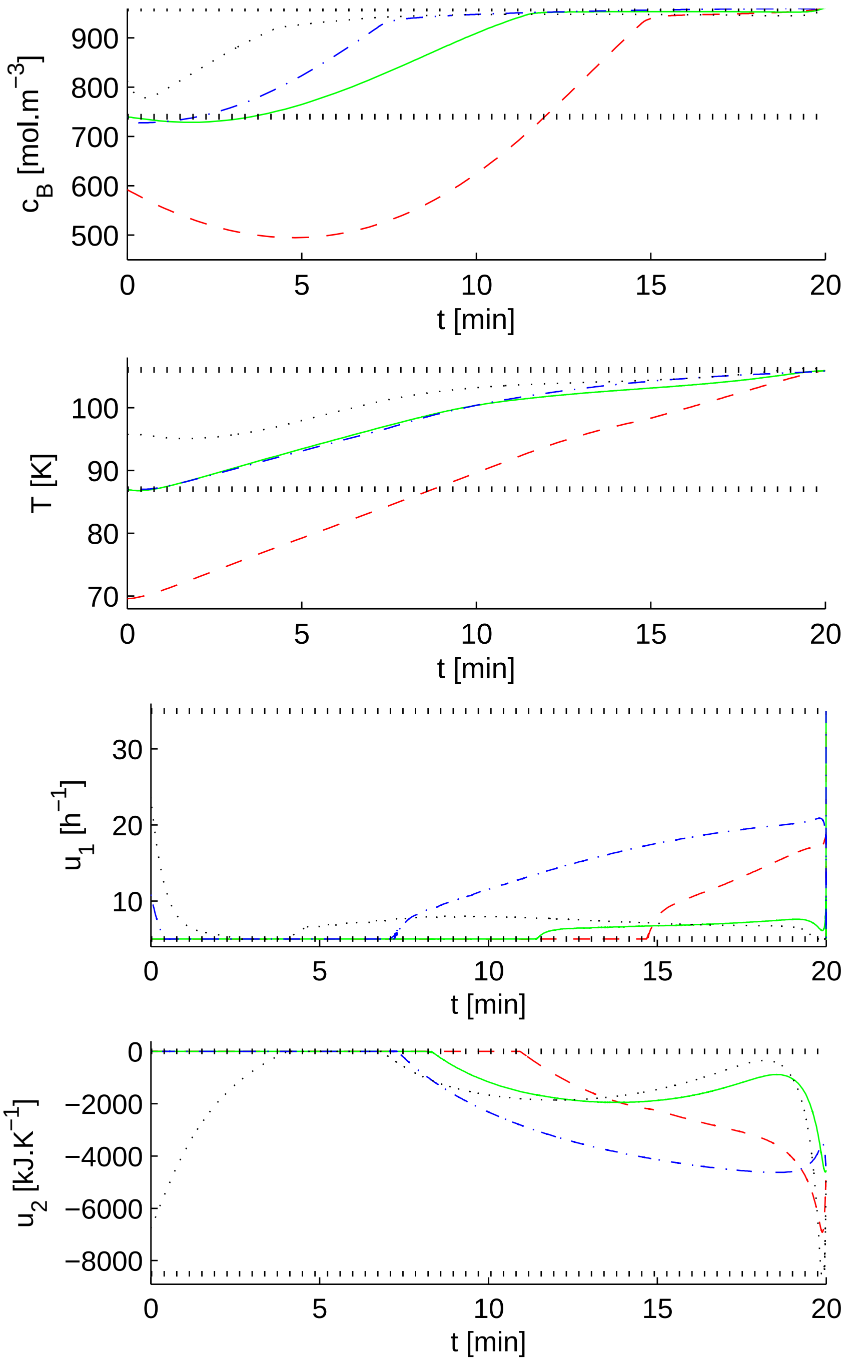

Figure 7.4: Performance of NCO-tracking with less aggressive control. **Dashed line:** C1, **solid line:** C2, **dash-dotted line:** C3, **dotted line:** C4.

$\boldsymbol{\lambda} \in \mathcal{R}^{n_x}$ denotes the adjoints that satisfy:

$$\dot{\boldsymbol{\lambda}} = -\mathcal{H}_{\boldsymbol{x}} = -\boldsymbol{F}_{\boldsymbol{x}}^T \boldsymbol{\lambda} - L_{\boldsymbol{x}}, \; t_0 \le t \le t_1, \tag{7.46}$$

with the terminal conditions given by:

$$\boldsymbol{\lambda}(t_1) = \left[\phi_{\boldsymbol{x}} + \boldsymbol{\nu}^T \boldsymbol{\psi}_{\boldsymbol{x}}\right]_{t=t_1}, \tag{7.47}$$

$\boldsymbol{\mu}^L(t), \boldsymbol{\mu}^U(t) \in \mathcal{R}^{n_u}$ are Lagrange multiplier vector functions satisfying the following:

$$\left.\begin{array}{ll} \mu_k^{L^T}(t)(u_k^L - u_k(t)) = 0; & \mu_k^L(t) \ge 0 \\ \mu_k^{U^T}(t)(u_k(t) - u_k^U) = 0; & \mu_k^U(t) \ge 0 \end{array}\right\} \begin{array}{c} \text{for each } k = 1, \ldots, n_u; \\ t_0 \le t \le t_1 \end{array} \tag{7.48}$$

and $\boldsymbol{\nu} \in \mathcal{R}^{n_\psi}$ are Lagrange multipliers for the terminal constraints such that

$$0 = \nu_k \psi_k, \quad \nu_k \ge 0, \quad \text{for each } k = 1, \ldots, n_\psi. \tag{7.49}$$

Provided that the optimal control problem is not abnormal, the first- and second-order necessary conditions for optimality (NCO) read:

$$\mathcal{H}_{\boldsymbol{u}} = L_{\boldsymbol{u}} + \boldsymbol{F}_{\boldsymbol{u}}^T \boldsymbol{\lambda} - \boldsymbol{\mu}^L - \boldsymbol{\mu}^U = \boldsymbol{0} \tag{7.50}$$

$$\mathcal{H}_{\boldsymbol{uu}} \succeq 0 \tag{7.51}$$

Equations in (7.46)–(7.51) determine the set of active terminal constraints at the optimum, which is denoted by the vector $\bar{\boldsymbol{\psi}}$ of dimension $n_{\bar{\psi}}$ and by complementary multiplier $\bar{\boldsymbol{\nu}}^*$. The constraints are inactive when the corresponding Lagrange multiplier is equal to zero.

7.2.3 Neighbouring-Extremal Control

Let us assume that $\boldsymbol{u}^*(t)$ is an optimal control for the optimisation problem (7.42)–(7.44), which consists of a finite sequence of boundary and interior arcs. The optimal solution then comprises $\boldsymbol{u}^*(t)$, $\boldsymbol{x}^*(t)$, $\boldsymbol{\lambda}^*(t)$, $\bar{\boldsymbol{\nu}}^*$, $\boldsymbol{\mu}^{*L}(t)$, $\boldsymbol{\mu}^{*U}(t)$, $t_0 \le t \le t_1$. For the control sequence, it is also assumed that the uncertainty is sufficiently small for the perturbed optimal control to have the same sequence of boundary and interior arcs as the nominal solution $\boldsymbol{u}^*(t)$. The constrained optimal control problem obtained with a small variation in the initial condition $\boldsymbol{x}(t_0) = \boldsymbol{x}_0 + \delta\boldsymbol{x}_0$ and in active terminal constraints $\bar{\boldsymbol{\psi}}(\boldsymbol{x}(t_1), t_1) = \delta\bar{\boldsymbol{\psi}}$ produces variations in the controls $\delta\boldsymbol{u}(t)$, states $\delta\boldsymbol{x}(t)$, adjoints $\delta\boldsymbol{\lambda}(t)$, Lagrange multipliers $\delta\bar{\boldsymbol{\nu}}$ (for the active terminal constraints $\bar{\boldsymbol{\psi}}$) and multiplier functions $\delta\boldsymbol{\mu}^{L^T}(t)$, $\delta\boldsymbol{\mu}^{U^T}(t)$.

These variations can be calculated from the linearisation of the first-order NCO (7.46)–(7.51) around the extremal path (Bryson and Ho, 1975). Along each arc composing $\boldsymbol{u}^*$, a control variable $u_i^*(t)$ may:

- belong to the interior of the control region $u_i^L < u_i^*(t) < u_i^U$, in which case a neighbouring-extremal solution is such that $\delta\mu_i^L(t) = \delta\mu_i^U(t) = 0$, and $\delta u_i(t)$ is obtained from the first variation of (7.50) as:

$$\delta\mathcal{H}_{u_i} = \mathcal{H}_{u_i\boldsymbol{x}}^* \delta\boldsymbol{x} + \boldsymbol{F}_{u_i}^{*T} \delta\boldsymbol{\lambda} + \mathcal{H}_{\boldsymbol{u}u_i}^* = 0; \tag{7.52}$$

- be at one of its boundaries u_i^L or u_i^U, in which case a NE control is simply given by $\delta u_i(t) = 0$.

These n_u conditions can be written collectively in the form :

$$\boldsymbol{A}_0 \delta \boldsymbol{\lambda} + \boldsymbol{B}_0 \delta \boldsymbol{x} + \boldsymbol{C}_0 \delta \boldsymbol{u} = \boldsymbol{0}, \tag{7.53}$$

where $\boldsymbol{A}_0(t), \boldsymbol{B}_0(t) \in \mathcal{R}^{n_u \times n_x}$, $\boldsymbol{C}_0(t) \in \mathcal{R}^{n_u \times n_u}$, and

$$\delta \dot{\boldsymbol{x}} = \boldsymbol{F}_{\boldsymbol{x}}^* \delta \boldsymbol{x} + \boldsymbol{F}_{\boldsymbol{u}}^* \delta \boldsymbol{u} \tag{7.54}$$

$$\delta \dot{\boldsymbol{\lambda}} = -\boldsymbol{F}_{\boldsymbol{x}}^{*T} \delta \boldsymbol{\lambda} - \mathcal{H}_{\boldsymbol{xx}}^* \delta \boldsymbol{x} - \mathcal{H}_{\boldsymbol{xu}}^* \delta \boldsymbol{u} \tag{7.55}$$

with additional conditions:

$$\delta \boldsymbol{x}(t_0) = \delta \boldsymbol{x}_0 \tag{7.56}$$

$$\delta \boldsymbol{\lambda}(t_1) = \left[\left(\boldsymbol{\phi}_{\boldsymbol{xx}}^* + \bar{\boldsymbol{\nu}}^{*T} \bar{\boldsymbol{\psi}}_{\boldsymbol{xx}}^* \right) \delta \boldsymbol{x} + \bar{\boldsymbol{\psi}}_{\boldsymbol{x}}^{*T} \delta \bar{\boldsymbol{\nu}} \right]_{t=t_1} \tag{7.57}$$

$$\delta \bar{\boldsymbol{\psi}} = \left[\bar{\boldsymbol{\psi}}_{\boldsymbol{x}}^* \delta \boldsymbol{x} \right]_{t=t_1}. \tag{7.58}$$

A superscript * indicates that the corresponding quantity is evaluated along the extremal path $\boldsymbol{u}^*(t)$, $t_0 \leq t \leq t_1$, and corresponding states, adjoints and Lagrange multipliers. In the case that the control problem is regular, $\boldsymbol{C}_0$ is invertible along $t_0 \leq t \leq t_1$, and a NE control law can be readily obtained as:

$$\delta \boldsymbol{u}(t) = -\boldsymbol{C}_0^{-1} \left[\boldsymbol{A}_0 \delta \boldsymbol{\lambda}(t) + \boldsymbol{B}_0 \delta \boldsymbol{x}(t) \right] \tag{7.59}$$

Note that this control law corresponds to the standard NE control law in the case where no input constraint is active; that is:

$$\boldsymbol{A}_0 = \boldsymbol{F}_{\boldsymbol{u}}^{*T}, \quad \boldsymbol{B}_0 = \mathcal{H}_{\boldsymbol{ux}}^*, \quad \boldsymbol{C}_0 = \mathcal{H}_{\boldsymbol{uu}}^* \tag{7.60}$$

Overall, $\delta \boldsymbol{x}(t)$ and $\delta \boldsymbol{\lambda}(t)$ satisfy the following two-point boundary value problem (TPBVP):

$$\begin{aligned} &\begin{pmatrix} \delta \dot{\boldsymbol{x}}(t) \\ \delta \dot{\boldsymbol{\lambda}}(t) \end{pmatrix} = \boldsymbol{\Delta}(t) \begin{pmatrix} \delta \boldsymbol{x}(t) \\ \delta \boldsymbol{\lambda}(t) \end{pmatrix}, \\ &\delta \boldsymbol{x}(0) = \delta \boldsymbol{x}_0, \quad \delta \bar{\boldsymbol{\psi}} = \left[\bar{\boldsymbol{\psi}}_{\boldsymbol{x}}^* \delta \boldsymbol{x} \right]_{t=t_1}, \\ &\delta \boldsymbol{\lambda}(t_1) = \left[\left(\boldsymbol{\phi}_{\boldsymbol{xx}}^* + \bar{\boldsymbol{\nu}}^{*T} \bar{\boldsymbol{\psi}}_{\boldsymbol{xx}}^* \right) \delta \boldsymbol{x} + \bar{\boldsymbol{\psi}}_{\boldsymbol{x}}^{*T} \delta \bar{\boldsymbol{\nu}} \right]_{t=t_1} \end{aligned} \tag{7.61}$$

where:

$$\boldsymbol{\Delta}(t) = \begin{pmatrix} \boldsymbol{\alpha}(t) & -\boldsymbol{\beta}(t) \\ -\boldsymbol{\gamma}(t) & -\boldsymbol{\omega}(t) \end{pmatrix} \tag{7.62}$$

with:

$$\boldsymbol{\alpha}(t) := \boldsymbol{F}_{\boldsymbol{x}}^* - \boldsymbol{F}_u^* \boldsymbol{C}_0^{-1} \boldsymbol{B}_0 \tag{7.63}$$

$$\boldsymbol{\beta}(t) := \boldsymbol{F}_u^* \boldsymbol{C}_0^{-1} \boldsymbol{A}_0 \tag{7.64}$$

$$\boldsymbol{\gamma}(t) := \mathcal{H}_{\boldsymbol{xx}}^* - \mathcal{H}_{\boldsymbol{xu}}^* \boldsymbol{C}_0^{-1} \boldsymbol{B}_0 \tag{7.65}$$

$$\boldsymbol{\omega}(t) := \boldsymbol{F}_{\boldsymbol{x}}^{*T} - \mathcal{H}_{\boldsymbol{xu}}^* \boldsymbol{C}_0^{-1} \boldsymbol{A}_0 \tag{7.66}$$

At a switching point between a lower bound and an interior arc, the value of control on lower bound matches the value of control in the interior arc $u^{\mathcal{H}} = u^{L}$, where $u^{\mathcal{H}}$ represents the control obtained from solving the condition $\mathcal{H}_{\boldsymbol{u}} = \mathbf{0}$. In addition, state, adjoint, and control trajectories are continuous at this point, too:

$$\boldsymbol{x}^*(t_k^+) = \boldsymbol{x}^*(t_k^-), \quad \boldsymbol{\lambda}^*(t_k^+) = \boldsymbol{\lambda}^*(t_k^-), \boldsymbol{u}^*(t_k^+) = \boldsymbol{u}^*(t_k^-). \tag{7.67}$$

In general, the switching junction times between the various arcs that constitute the optimal solution $\boldsymbol{u}^*(t)$, $t_0 \leq t \leq t_1$, vary when the initial conditions or terminal condition vary. However, such switching time variations are difficult to determine and complicate the calculation of the NE control. To make this implementable, we consider the fixed switching points at their nominal values, and the control values are then updated between these fixed times.

7.2.4 Numerical Computation of Neighbouring Feedback Control

The linear TPBVP (7.61) can be used to calculate the NE control corrections $\delta\boldsymbol{u}(t)$, $0 \leq t \leq t_f$, in either one of two situations:

i. The initial state and (active) terminal constraint variations $\delta\boldsymbol{x}_0$ and $\delta\bar{\boldsymbol{\psi}}$ are available at discrete time instants, in which case the discrete feedback control can be obtained by directly re-solving the TPBVP. This can be done via a shooting method as described in Pesch, 1989a; Pesch, 1989b;

ii. The variations $\delta\boldsymbol{x}_0$ and $\delta\bar{\boldsymbol{\psi}}$ are available continuously in time, in which case the backward sweep method can be used to derive an explicit feedback control law. This approach is closely explained in Bryson and Ho, 1975.

7.2.5 Shooting Method

The boundary value problem (7.61) can be rewritten in the form

$$\begin{pmatrix} \delta\dot{\boldsymbol{x}}(t) \\ \delta\dot{\boldsymbol{\lambda}}(t) \end{pmatrix} = \boldsymbol{\Delta}(t) \begin{pmatrix} \delta\boldsymbol{x}(t) \\ \delta\boldsymbol{\lambda}(t) \end{pmatrix}, \tag{7.68}$$

with the boundary conditions

$$\begin{pmatrix} \boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \delta\boldsymbol{x}(t_0) \\ \delta\boldsymbol{\lambda}(t_0) \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{B}_1 & \boldsymbol{I} \end{pmatrix} \begin{pmatrix} \delta\boldsymbol{x}(t_1) \\ \delta\boldsymbol{\lambda}(t_1) \end{pmatrix} = \begin{pmatrix} \delta\boldsymbol{x}_0 \\ \boldsymbol{B}_2 \end{pmatrix}, \tag{7.69}$$

where

$$\boldsymbol{B}_1 = -\left[\boldsymbol{\phi}^*_{\boldsymbol{xx}} + \bar{\boldsymbol{\nu}}^{*T}\bar{\boldsymbol{\psi}}^*_{\boldsymbol{xx}}\right]_{t_1}$$
$$\boldsymbol{B}_2 = \left[\bar{\boldsymbol{\psi}}^{*T}_{\boldsymbol{x}}\right]_{t_1}\delta\bar{\boldsymbol{\nu}}.$$

The shooting approach proceeds by guessing the missing initial (or terminal) conditions in (7.69), and adjusting them in such a way that the corresponding terminal (or initial) conditions are satisfied (see, e.g., Pesch, 1989a; Pesch, 1989b). Given the guess $\delta\boldsymbol{\lambda}(t_0) = \delta\boldsymbol{\lambda}_0$ for the adjoint variations at initial time $t = t_0$, the (unique) solution to the linear ODE system (7.68) is of the form:

$$\begin{pmatrix} \delta\boldsymbol{x}(t) \\ \delta\boldsymbol{\lambda}(t) \end{pmatrix} = \underbrace{\begin{pmatrix} \boldsymbol{\Upsilon}_1(t;t_0) & \boldsymbol{\Upsilon}_2(t;t_0) \\ \boldsymbol{\Upsilon}_3(t;t_0) & \boldsymbol{\Upsilon}_4(t;t_0) \end{pmatrix}}_{=:\,\boldsymbol{\Upsilon}(t;t_0)} \begin{pmatrix} \delta\boldsymbol{x}_0 \\ \delta\boldsymbol{\lambda}_0 \end{pmatrix}, \tag{7.70}$$

where the transition matrix $\boldsymbol{\Upsilon}(t;t_0)$ is obtained as the solution to the initial value problem

$$\frac{\partial}{\partial t}\boldsymbol{\Upsilon}(t;t_0) = \boldsymbol{\Delta}(t)\boldsymbol{\Upsilon}(t;t_0),\ t_0 \le t \le t_1; \quad \boldsymbol{\Upsilon}(t_0;t_0) = \boldsymbol{I}. \tag{7.71}$$

Substituting (7.70) into (7.69) and (7.58) leads to the following linear system in the variables $\delta\boldsymbol{\lambda}_0, \delta\bar{\boldsymbol{\nu}}$:

$$\begin{pmatrix} \boldsymbol{Z}_1 & \boldsymbol{Z}_2 \\ \boldsymbol{Z}_3 & \boldsymbol{0} \end{pmatrix} \begin{pmatrix} \delta\boldsymbol{\lambda}_0 \\ \delta\bar{\boldsymbol{\nu}} \end{pmatrix} = \begin{pmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{pmatrix} \delta\bar{\boldsymbol{\psi}} - \begin{pmatrix} \boldsymbol{Z}_4 \\ \boldsymbol{Z}_5 \end{pmatrix} \delta\boldsymbol{x}_0, \tag{7.72}$$

where

$$\begin{aligned}
\boldsymbol{Z}_1 &= \left[\boldsymbol{\phi}^*_{\boldsymbol{xx}} + \bar{\boldsymbol{\nu}}^{*T}\bar{\boldsymbol{\psi}}^*_{\boldsymbol{xx}}\right]_{t_1} \boldsymbol{\Upsilon}_2(t_1;t_0) - \boldsymbol{\Upsilon}_4(t_1;t_0) \\
\boldsymbol{Z}_2 &= \left[\bar{\boldsymbol{\psi}}^{*T}_{\boldsymbol{x}}\right]_{t_1} \\
\boldsymbol{Z}_3 &= \left[\bar{\boldsymbol{\psi}}^*_{\boldsymbol{x}}\right]_{t_1} \boldsymbol{\Upsilon}_2(t_1;t_0) \\
\boldsymbol{Z}_4 &= \left[\boldsymbol{\phi}^*_{\boldsymbol{xx}} + \bar{\boldsymbol{\nu}}^{*T}\bar{\boldsymbol{\psi}}^*_{\boldsymbol{xx}}\right]_{t_1} \boldsymbol{\Upsilon}_1(t_1;t_0) - \boldsymbol{\Upsilon}_3(t_1;t_0) \\
\boldsymbol{Z}_5 &= \left[\bar{\boldsymbol{\psi}}^*_{\boldsymbol{x}}\right]_{t_1} \boldsymbol{\Upsilon}_1(t_1;t_0).
\end{aligned}$$

For given initial state and active terminal constraint variations $\delta\boldsymbol{x}_0$ and $\delta\bar{\boldsymbol{\psi}}$, the solution to the linear system (7.72) provides the corresponding initial adjoint and Lagrange multiplier variations $\delta\boldsymbol{\lambda}_0$ and $\delta\bar{\boldsymbol{\nu}}$. Finally, the NE control variation can be calculated from (7.59) as

$$\delta\boldsymbol{u}(t) = -\boldsymbol{C}_0(t)^{-1}\begin{pmatrix} \boldsymbol{B}_0(t) & \boldsymbol{A}_0(t) \end{pmatrix} \boldsymbol{\Upsilon}(t;t_0) \begin{pmatrix} \delta\boldsymbol{x}_0 \\ \delta\boldsymbol{\lambda}_0 \end{pmatrix}. \tag{7.73}$$

Backward Sweep Method

The idea behind the sweep method consists in postulating a linear relationship between the state, adjoint and Lagrange multiplier variations as follows:

$$\delta\boldsymbol{\lambda}(t) = \boldsymbol{\Sigma}(t)\delta\boldsymbol{x}(t) + \boldsymbol{\Pi}(t)\delta\bar{\boldsymbol{\nu}} \tag{7.74}$$

$$\delta\bar{\boldsymbol{\psi}} = \boldsymbol{\Pi}^T(t)\delta\boldsymbol{x}(t) + \boldsymbol{\Omega}(t)\delta\bar{\boldsymbol{\nu}} \tag{7.75}$$

where $\boldsymbol{\Sigma}(t) \in \mathcal{R}^{n_x \times n_x}$, $\boldsymbol{\Pi}(t) \in \mathcal{R}^{n_x \times n_{\bar{\psi}}}$, $\boldsymbol{\Omega}(t) \in \mathcal{R}^{n_{\bar{\psi}} \times n_{\bar{\psi}}}$ are matrix functions of t. From the terminal conditions (7.57) and (7.58), it is easily seen that

$$\boldsymbol{\Sigma}(t_1) = \left[\bar{\boldsymbol{\phi}}^*_{\boldsymbol{xx}} + \bar{\boldsymbol{\nu}}^{*T}\bar{\boldsymbol{\psi}}^*_{\boldsymbol{xx}}\right]_{t=t_1} \tag{7.76}$$

$$\boldsymbol{\Pi}(t_1) = \left[\bar{\boldsymbol{\psi}}^{*T}_{\boldsymbol{x}}\right]_{t=t_1} \tag{7.77}$$

$$\boldsymbol{\Omega}(t_1) = \boldsymbol{0}. \tag{7.78}$$

A system of Riccati matrix differential equations can be obtained by differentiating (7.74) and (7.75) with respect to time:

$$\delta\dot{\boldsymbol{\lambda}}(t) = \dot{\boldsymbol{\Sigma}}(t)\delta\boldsymbol{x}(t) + \boldsymbol{\Sigma}(t)\delta\dot{\boldsymbol{x}}(t) + \dot{\boldsymbol{\Pi}}(t)\delta\bar{\boldsymbol{\nu}} \tag{7.79}$$

$$\mathbf{0} = \dot{\boldsymbol{\Pi}}^T(t)\delta\boldsymbol{x}(t) + \boldsymbol{\Pi}^T(t)\delta\dot{\boldsymbol{x}}(t) + \dot{\boldsymbol{\Omega}}(t)\delta\bar{\boldsymbol{\nu}}, \tag{7.80}$$

and then equating the various terms in (7.79),(7.80) to those in (7.61):

$$\dot{\boldsymbol{\Sigma}}(t) = -\boldsymbol{\Sigma}(t)\boldsymbol{\alpha}(t) - \boldsymbol{\rho}(t)\boldsymbol{\Sigma}(t) + \boldsymbol{\Sigma}(t)\boldsymbol{\beta}(t)\boldsymbol{\Sigma}(t) - \boldsymbol{\gamma}(t) \tag{7.81}$$

$$\dot{\boldsymbol{\Pi}}(t) = -\left[\boldsymbol{\rho}(t) - \boldsymbol{\Sigma}(t)\boldsymbol{\beta}(t)\right]\boldsymbol{\Pi}(t) \tag{7.82}$$

$$\dot{\boldsymbol{\Omega}}(t) = \boldsymbol{\Pi}(t)^T\boldsymbol{\beta}(t)\boldsymbol{\Pi}(t), \tag{7.83}$$

with $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\rho}$ given by (7.63)–(7.66).

At this point, values for $\boldsymbol{\Sigma}(t_0)$, $\boldsymbol{\Pi}(t_0)$, $\boldsymbol{\Omega}(t_0)$ can be calculated by integrating (7.81)–(7.83) backward in time to $t = t_0$, from the terminal conditions (7.76)–(7.78); in other words, the terminal conditions (7.76)–(7.78) are 'swept back' to the initial time.

Having solved (7.81)–(7.83), the variation $\delta\bar{\boldsymbol{\nu}}$ in the Lagrange multipliers that are needed to counteract the variation $\delta\bar{\boldsymbol{\psi}}$ in the active terminal constraints can be obtained from (7.75) as

$$\delta\bar{\boldsymbol{\nu}} = \boldsymbol{\Omega}^{-1}(t)\left[\delta\bar{\boldsymbol{\psi}} - \boldsymbol{\Pi}^T(t)\delta\boldsymbol{x}(t)\right]. \tag{7.84}$$

Note that the latter can be solved at any time t, provided that the square matrix $\boldsymbol{\Omega}(t)$ in non-singular. Clearly, (7.78) precludes the calculation of $\delta\bar{\boldsymbol{\nu}}$ at final time t_1. In turn, the adjoint variation $\delta\boldsymbol{\lambda}(t)$ can be calculated, at any $t_0 \leq t \leq t_1$ from the following equation:

$$\delta\boldsymbol{\lambda}(t) = \left[\boldsymbol{\Sigma}(t) - \boldsymbol{\Pi}(t)\boldsymbol{\Omega}(t)^{-1}\boldsymbol{\Pi}(t)^T\right]\delta\boldsymbol{x}(t) + \boldsymbol{\Pi}(t)\boldsymbol{\Omega}(t)^{-1}\delta\bar{\boldsymbol{\psi}}. \tag{7.85}$$

Finally, substituting (7.74) and (7.85) into (7.59) yields the following explicit expression of the control variation $\delta\boldsymbol{u}(t)$ as a function of the state and terminal constraint variations $\delta\boldsymbol{x}(t)$ and $\delta\bar{\boldsymbol{\psi}}$:

$$\delta\boldsymbol{u}(t) = -\boldsymbol{K}_{\boldsymbol{x}}(t)\delta\boldsymbol{x}(t) - \boldsymbol{K}_{\bar{\psi}}(t)\delta\bar{\boldsymbol{\psi}} \tag{7.86}$$

with:

$$\boldsymbol{K}_{\boldsymbol{x}} := (\mathcal{H}^*_{\boldsymbol{uu}})^{-1}\left[\mathcal{H}^*_{\boldsymbol{ux}} + \boldsymbol{F}^{*T}_{\boldsymbol{u}}\left(\boldsymbol{\Sigma} - \boldsymbol{\Pi}\boldsymbol{\Omega}^{-1}\boldsymbol{\Pi}^T\right)\right], \qquad \boldsymbol{K}_{\bar{\psi}} := (\mathcal{H}^*_{\boldsymbol{uu}})^{-1}\boldsymbol{F}^{*T}_{\boldsymbol{u}}\boldsymbol{\Pi}\boldsymbol{\Omega}^{-1}$$

By construction, the continuous-time feedback law (7.86) enforces satisfaction of the first variation of the NCO (7.46)–(7.51) in the presence of state and constraint variations $\delta\boldsymbol{x}(t)$ and $\delta\boldsymbol{\psi}$.

7.3 Run-to-run Constraint Adaptation

Different approach that deals with terminal constraints in presence of uncertainty is Constraint Adaptation (CA). The principle behind run-to-run optimisation is similar to MPC.

But instead of adapting the initial conditions and moving the control horizon as is done in MPC, the adaptation is performed on the optimisation model (e.g., model parameters or constraint biases) before re-running the optimiser. In run-to-run constraint adaptation (Chachuat, Marchetti, et al., 2008; Marchetti, Chachuat, et al., 2007), more specifically, it is the terminal constraints (7.43) in the optimisation model which are adapted after each run as:

$$\boldsymbol{\psi}(\boldsymbol{x}(t_1), t_1) \leq \delta\boldsymbol{\psi}, \tag{7.87}$$

where $\delta\boldsymbol{\psi}$ stands for the terminal constraint bias. Such a bias can be directly updated as the difference between the available terminal constraint measurements, $\boldsymbol{\psi}^{\text{meas}}$, at the end of each run and the predicted constraint values. This simple strategy can however lead to excessive correction when operating far away from the optimum, and it may also exacerbate the sensitivity of the adaptation scheme to measurement noise. A better strategy consists of filtering the bias, e.g., with a first-order exponential filter:

$$\delta\boldsymbol{\psi}_{k+1} = [\boldsymbol{I} - \boldsymbol{W}]\,\delta\boldsymbol{\psi}_k + \boldsymbol{W}\,[\boldsymbol{\psi}_k^{\text{meas}} - \boldsymbol{\psi}(\boldsymbol{x}_k(t_1), t_1)]\,, \tag{7.88}$$

with k the run index, and $\boldsymbol{W}$ a gain matrix—typically, a diagonal matrix with entries w_i such that $0 < w_i \leq 1, i = 1, ..., n_\psi$.

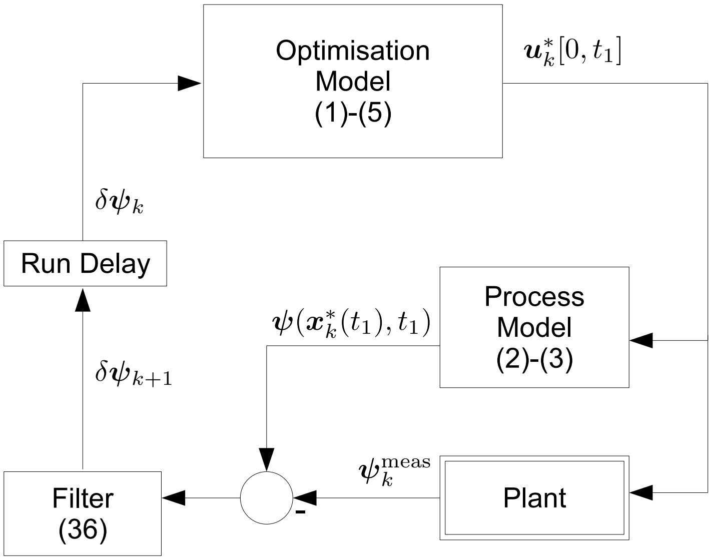
Optimisation Model (1)-(5)
$\boldsymbol{u}_k^*[0, t_1]$
$\delta\boldsymbol{\psi}_k$
Run Delay
Process Model (2)-(3)
$\boldsymbol{\psi}(\boldsymbol{x}_k^*(t_1), t_1)$
$\delta\boldsymbol{\psi}_{k+1}$
$\boldsymbol{\psi}_k^{\text{meas}}$
Filter (36)
-
Plant

Figure 7.5: Run-to-run constraint adaptation scheme.

The run-to-run constraint-adaptation scheme is shown in Figure 7.5. The constrained dynamic optimisation problem uses the available nominal process model. It is solved between each run, using a numerical procedure such as the sequential (Edgar and Himmelblau, 1988; Guntern, Keller, et al., 1998) or the simultaneous approach (Biegler, 1984; Renfro, Morshedi, et al., 1987) of dynamic optimisation. The optimal control trajectory $\boldsymbol{u}_k^*(t)$, $t_0 \leq t \leq t_1$, is computed and applied to the plant during the k-th run. The predicted optimal response is denoted by $\boldsymbol{x}_k^*(t)$. The discrepancy between the measured terminal constraint values $\boldsymbol{\psi}_k^{\text{meas}}$ and the optimiser predictions $\boldsymbol{\psi}(\boldsymbol{x}_k^*(t_1), t_1)$ is then used to adjust the constraint bias as described earlier, before re-running the optimiser and so on.

Chapter 8

NCO-tracking by Two-time-scale Control

Run-to-run constraint adaptation was shown to be a promising approach in Chachuat, Marchetti, et al., 2008; Marchetti, Chachuat, et al., 2007. It provides a natural framework for handling changes in active constraints in dynamic process systems and it is quite robust towards model mismatch and process disturbances. Moreover, its practical implementation is rather simple. Inherent limitations of this scheme, however, are that (i) it does not perform any control corrections during the runs, and (ii) it typically leads to sub-optimal performance since only the terminal constraints are adapted.

On the other hand, NE control is able to correct small deviations around the nominal extremal path in order to deliver similar performance as with re-optimisation. Since no costly on-line re-optimisation is needed, this approach is well suited for processes with fast dynamics that require frequent updates. However, the performance of NE control typically decreases dramatically in the presence of large model mismatch and process disturbances, and it requires a full-state measurement. This leads to sub-optimality or, worse, infeasibility when constraints are present or limited measurements are available.

The proposal is to combine the advantages of these two approaches: run-to-run constraint adaptation is applied at the slow time scale (outer loop) to handle large model mismatch and changes in active constraints, based on run-end measurements only. Furthermore, NE control is applied at the fast time scale (inner loop) and uses measurement information available within each run, in order to enhance convergence speed and mitigate sub-optimality. Another advantage is that the switching times between control arcs are adapted in the outer loop, thereby reducing the fact that they are not adapted by the NE controller itself (see Section 7.2).

A current limitation of the proposed scheme is that full-state measurement is required by NE controller. In practice, this is rarely the case as the relation between measured outputs and states is never known perfectly.

The results in this Chapter have been also published in (Podmajerský, Chachuat, et al., 2011a,b,c; Podmajerský and Fikar, 2011).

The proposed integrated two-time-scale optimisation scheme is depicted in Figure 8.1. The implementation procedure is as follows:

Initialisation:

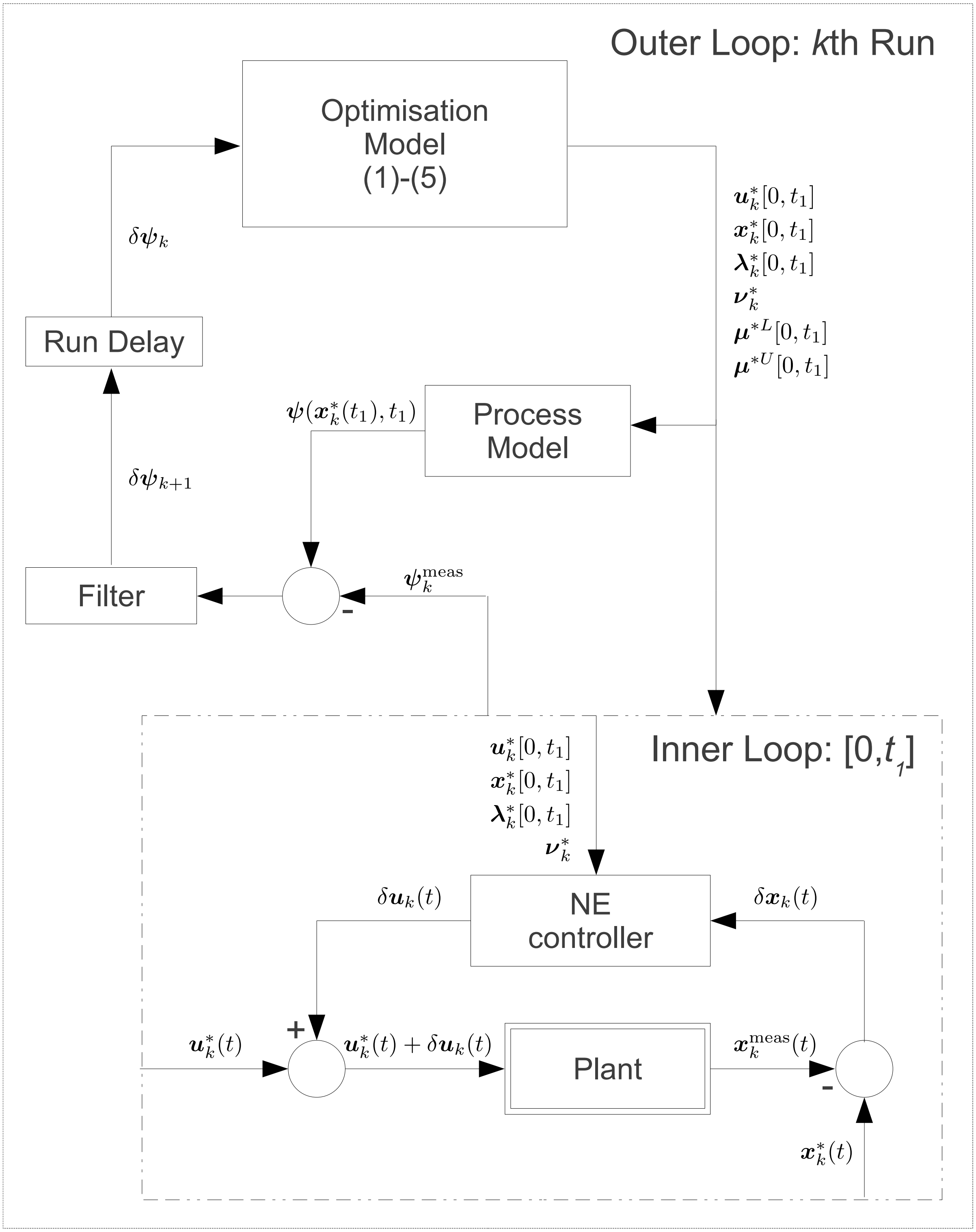

Figure 8.1: Two-times-scale optimisation scheme employing NE control in the inner loop and run-to-run constraint adaptation in the outer loop.

0. Initialise the constraint bias $\delta\boldsymbol{\psi} = \mathbf{0}$, select a gain matrix $\boldsymbol{W}$ and set the run index to $k = 1$.

Outer Loop:

1. Determine $\boldsymbol{u}_k^*$ by solving the optimal control problem (7.42)–(7.44), then obtain the corresponding states $\boldsymbol{x}_k^*$ and adjoints $\boldsymbol{\lambda}_k^*$, with the active terminal constraints $\bar{\boldsymbol{\psi}}$ and corresponding Lagrange multipliers $\bar{\boldsymbol{\nu}}_k^*$, and the active input constraints and corresponding multiplier functions $\boldsymbol{\mu}^{*L}$ and $\boldsymbol{\mu}^{*U}$.
2. Design a NE controller around the extremal path $\boldsymbol{u}_k^*$, either by using the backward sweep approach (continuous measurements), or by applying the shooting method (discrete measurements).
3. **Inner Loop:**
   Implement the NE controller during the k-th run in order to calculate the corrections $\delta\boldsymbol{u}_k(t)$ to $\boldsymbol{u}_k^*(t)$ based on the available (continuous or discrete) process measurements.
4. Update the constraint bias $\delta\boldsymbol{\psi}_{k+1}$ as the filtered difference between the measured values of the terminal constraints and their predicted counterparts.
5. Increment the run index $k \leftarrow k + 1$, and return to Step 1.

8.1 Batch Reactor Control

Consider again the batch reactor example from Section 4.1 to illustrate the proposed integrated two-times-scale approach as well as pure NE control with terminal constraints and pure run-to-run constraint adaptation.

8.1.1 Results

In order to illustrate the benefits of the proposed integrated scheme, two case studies are considered where the initial conditions are changed by −40% with respect to the nominal scenario,

$$\delta\beta_{\mathsf{R}} = -0.212, \qquad \delta\beta_{\mathsf{P}} = -0.172.$$

No measurement noise is presented in the ideal case and a measurement noise is presented in the second case. Also note that NE controller can be designed and applied in two different ways which are presented in Section 7.2.5. In this study, the NE controller in the integrated run-to-run constraint-adaptation scheme is designed by backward sweep method that requires presence of continuous state measurements. Within each case study, two adaptation strategies are compared to reject these perturbations:

- *Strategy 1:* The original run-to-run constraint-adaptation scheme (run-end state measurements are required) is applied;

- *Strategy 2:* The integrated run-to-run constraint-adaptation and NE control scheme (run-time continuous state measurements are required) is used.

Note that for the sake of comparison, the run-to-run adaptation is initialized with a constraint bias of $\delta\psi = 0$ and considers a filter gain of $W = 0.5$ in both strategies. This filter parameter was chosen so as to avoid oscillations during the adaptation process.

Case 1: No measurement noise

Figure 8.2 shows the evolution of the temperature profile during the runs 1, 2 and 20, for both strategies. Observe that control profiles in Strategy 2 are almost identical during the runs 1, 2 and 20. NE controller corrects all the perturbations during the runs such that the constraint adaptation is minimal. Also note that run 20 is meant to represent the adapted plant operation after convergence. Similarly, the evolution of the terminal constraint ψ is depicted in Figure 8.3 for both strategies. It is clearly seen from these plots that proposed integrated scheme exhibits a much faster convergence than the original run-to-run constraint-adaptation scheme. In particular, the integrated scheme is found to reach the same operation after just one run whereas the original scheme after seven runs. The converged solution after 20 runs versus the perturbed optimal control is shown in Figure 8.4, for both strategies. It is found that both approaches match the terminal constraint and the objective with perturbed optimal control.

Case 2: Presence of measurement noise

Figure 8.5 depicts the evolution of temperature profile during the runs 1, 2 and 20, for both strategies in presence of noisy measurements. Observe that the control profiles in Strategy 2 reflect the correction of perturbations brought by continuous measurements and by perturbed initial conditions. Only 2 runs are needed for Strategy 2 to get in neighbourhood of terminal objectives, whereas Strategy 1 needed 12 runs. It is found in Figure 8.6 that the terminal constraint in Strategy 1 approaches asymptotically the reference and oscillates closely to the reference in Strategy 1. The converged solution after 20 runs against the perturbed optimal control is shown in Figure 8.7, for both strategies in the presence of measurement noise. Note that the terminal constraints are met as well as the performance index.

8.2 Semi-batch Reactor Control

A semi-batch reactor example considered in Chen and Hwang, 1990 illustrates the proposed integrated two-times-scale approach and pure NE control as well as pure run-to-run constraint adaptation. The goal is to maximise the yield of ethanol using the feed rate $u(t)$ as the control variable, while keeping the liquid volume below some maximum threshold. Simple bound constraints are imposed on the feed rate.

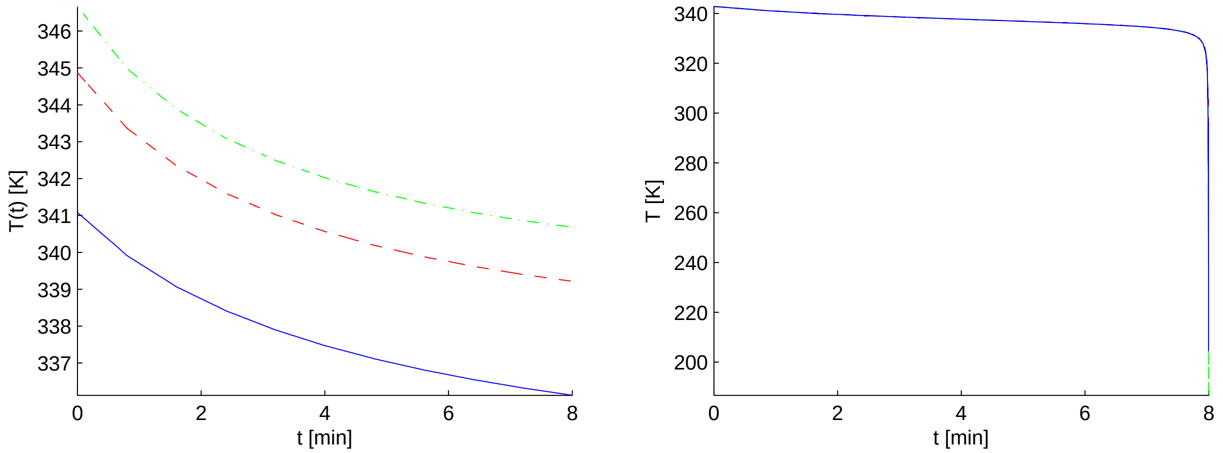

Figure 8.2: Evolution of the temperature profile in presence of no measurement noise. *Left plot:* Strategy 1; *Right plot:* Strategy 2. **Solid line:** Control input in the last run. **Dash-dotted line:** Control input in the first run. **Dashed line:** Control input in the second run.

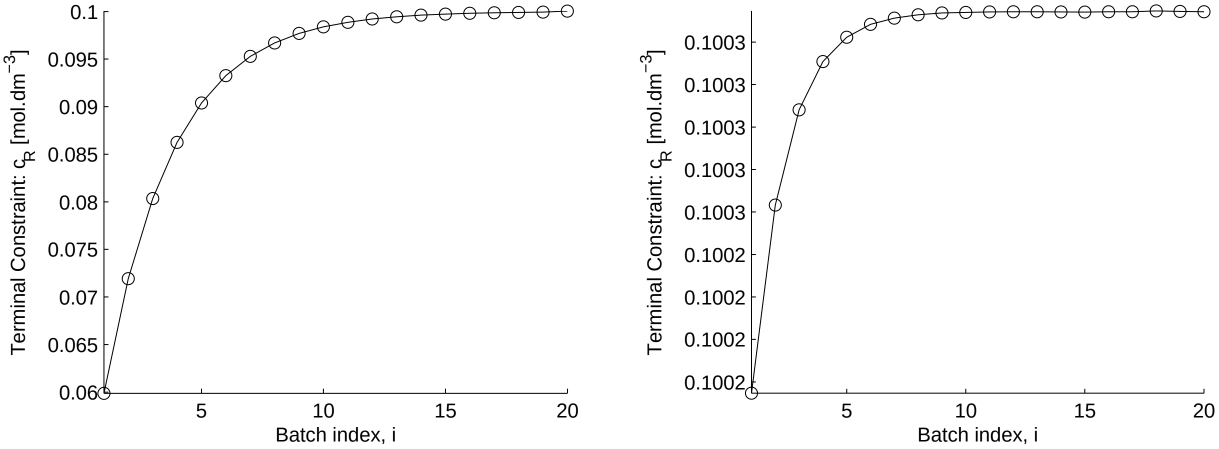

Figure 8.3: Evolution of the terminal constraint ψ in presence of no measurement noise. *Left plot:* Strategy 1; *Right plot:* Strategy 2.

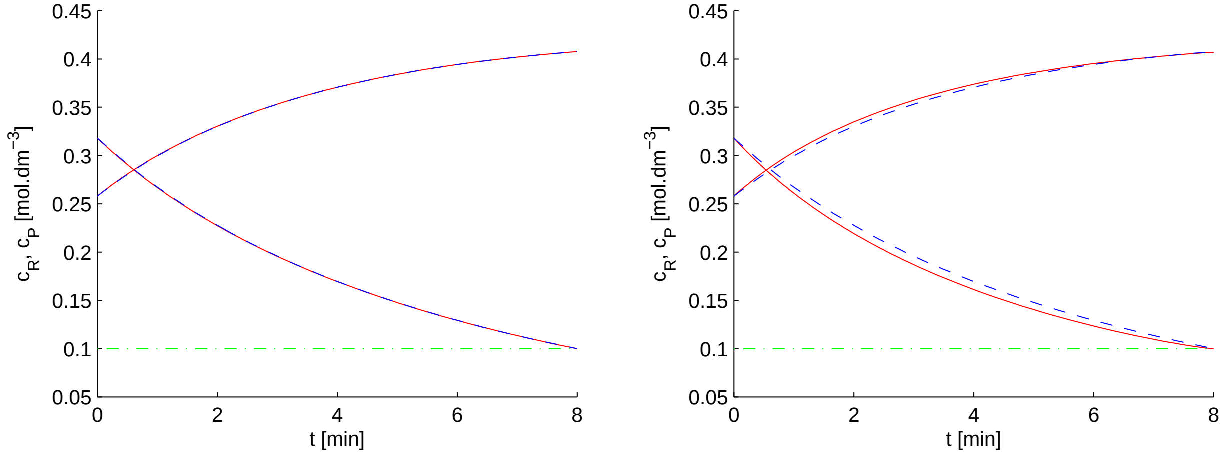

Figure 8.4: Converged solution after 20 runs compared to the perturbed optimal control in presence of no measurement noise. *Left plot:* Strategy 1; *Right plot:* Strategy 2. **Solid line:** System response of the proposed solution. **Dashed line:** System response of the re-optimised solution.

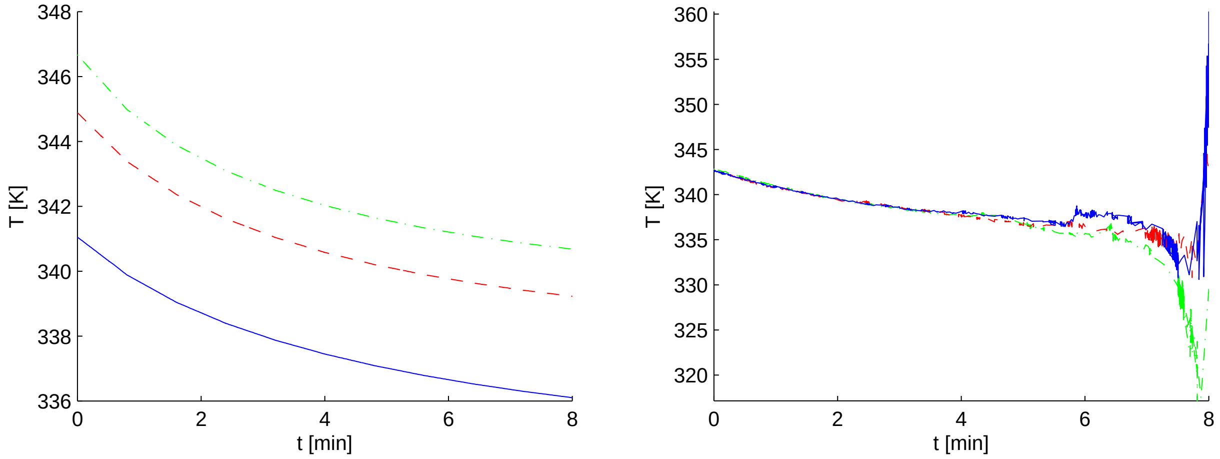

Figure 8.5: Evolution of the temperature profile in presence of measurement noise. *Left plot:* Strategy 1; *Right plot:* Strategy 2. **Solid line:** Control input in the last run. **Dash-dotted line:** Control input in the first run. **Dashed line:** Control input in the second run.

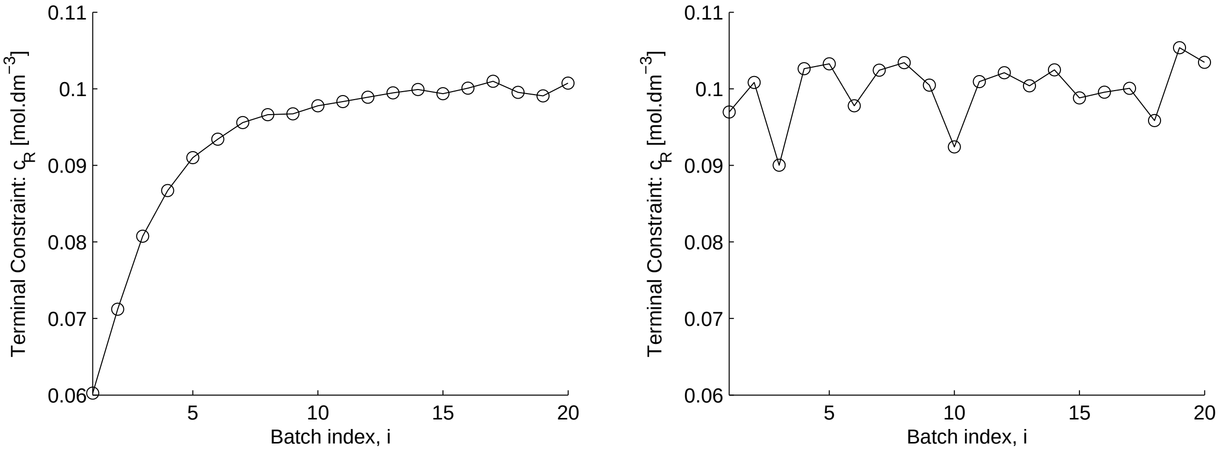

Figure 8.6: Evolution of the terminal constraint ψ in presence of measurement noise. *Left plot:* Strategy 1; *Right plot:* Strategy 2.

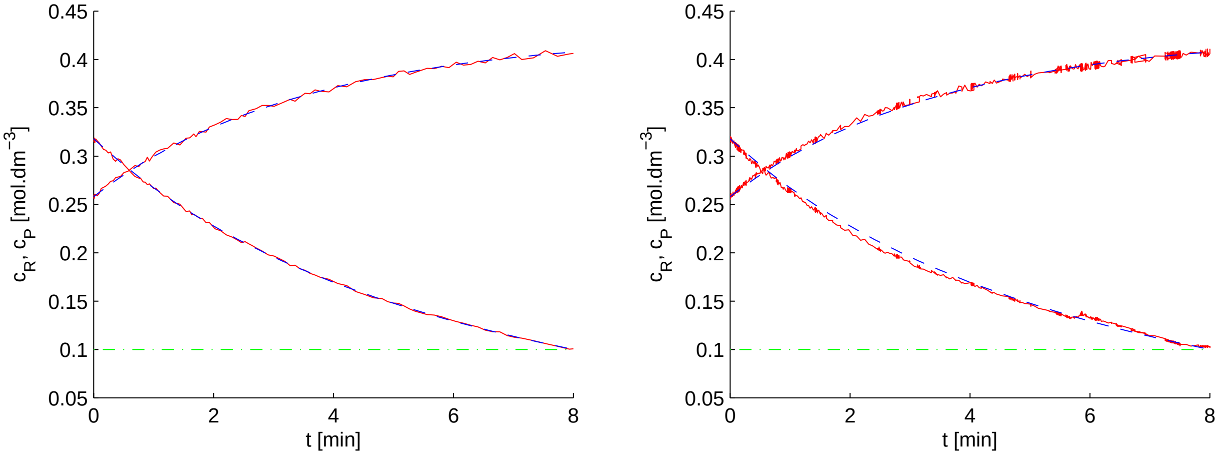

Figure 8.7: Converged solution after 20 runs compared to the perturbed optimal control in presence of measurement noise. *Left plot:* Strategy 1; *Right plot:* Strategy 2. **Solid line:** System response of the proposed solution. **Dashed line:** System response of the re-optimised solution.

8.2.1 Problem Formulation

The mathematical formulation of this problem is to maximise:

$$\mathcal{J} = c_{\mathsf{E}}(t_1)V(t_1) - 0.1\int_{t_0}^{t_1} [u(t)]^2 \mathrm{d}t. \tag{8.1}$$

The process model is given by:

$$\dot{c}_{\mathsf{MS}}(t) = p_1(t)c_{\mathsf{MS}}(t) - u(t)\frac{c_{\mathsf{MS}}(t)}{V(t)} \tag{8.2}$$

$$\dot{c}_{\mathsf{S}}(t) = -10p_1(t)c_{\mathsf{MS}}(t) + u(t)\frac{150 - c_{\mathsf{S}}(t)}{V(t)} \tag{8.3}$$

$$\dot{c}_{\mathsf{E}}(t) = p_2(t)c_{\mathsf{MS}}(t) - u(t)\frac{c_{\mathsf{E}}(t)}{V(t)} \tag{8.4}$$

$$\dot{V}(t) = u(t) \tag{8.5}$$

where:

$$p_1(t) = \left(\frac{0.408}{1 + \frac{c_{\mathsf{E}}}{16}}\right)\left(\frac{c_{\mathsf{S}}}{0.22 + c_{\mathsf{S}}}\right) \tag{8.6}$$

$$p_2(t) = \left(\frac{1}{1 + \frac{c_{\mathsf{E}}}{71.5}}\right)\left(\frac{c_{\mathsf{S}}}{0.44 + c_{\mathsf{S}}}\right). \tag{8.7}$$

The state values $c_{\mathsf{MS}}(t)$, $c_{\mathsf{S}}(t)$, $c_{\mathsf{E}}(t)$ and $V(t)$ are the cell biomass, substrate, and ethanol concentrations [g/L], and the volume [L]. The final time is set to $t_1 = 60\,\mathrm{h}$. The reactor container is initially loaded with $V(t_0) = 10\,\mathrm{L}$ of reaction mixture with biomass and substrate concentrations $c_{\mathsf{MS}}(t_0) = 1\,\mathrm{g/L}$ and $c_{\mathsf{S}}(t_0) = 150\,\mathrm{g/L}$. No ethanol is initially present in the reaction mixture. The feed rate [L/h] is bounded as:

$$0 \le u \le 12\,[\mathrm{L/h}]. \tag{8.8}$$

The liquid volume is limited by $V^{\max} = 200\,\mathrm{L}$, so the terminal condition reads:

$$V(t_1) \le V^{\max}\,[\mathrm{L}]. \tag{8.9}$$

Note that the integral term $\int_{t_0}^{t_1}[u(t)]^2\mathrm{d}t$ augments the original objective function in order to make the control problem regular. This way $\mathcal{H}_{\boldsymbol{u}}$ depends on the control variable and is invertible.

8.2.2 Open-Loop Optimal Control

The optimisation problem (8.1)–(8.9) was solved using the sequential method (Edgar and Himmelblau, 1988; Guntern, Keller, et al., 1998) with piecewise constant control. It is seen in Figure 8.8 that the optimal control profile consists of three arcs: a lower bound; an interior arc; and another short lower bound. Since the problem is regular, the control action along interior arc can be explicitly determined from the conditions (7.46)–(7.51). Note that

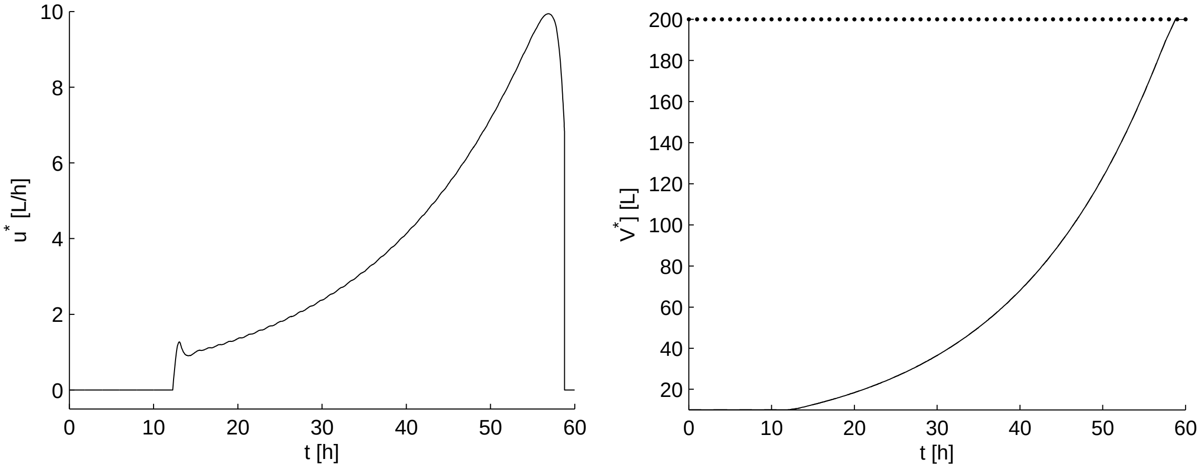

Figure 8.8: Optimal state trajectory of liquid volume (solid line) and maximum value (dotted line) – the nominal solution

along boundary arcs, the control action is equal to the lower bound. The switching times τ_1 and τ_2 between these arcs are not explicitly known and they need to be estimated too. Good initial guess for these switching times can be estimated using e.g. a piecewise constant control approximation. Overall, the optimal control solution is stated as:

$$\dot{\boldsymbol{x}} = \text{(8.2)–(8.7)}; \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{8.10}$$

$$\dot{\boldsymbol{\lambda}} = -\mathcal{H}_{\boldsymbol{x}}; \quad \boldsymbol{\lambda}(t_1) = \left[\phi_{\boldsymbol{x}} + \boldsymbol{\nu}^T \boldsymbol{\psi}_{\boldsymbol{x}}\right]_{t=t_1} \tag{8.11}$$

$$0 = \begin{cases} u^L(t), & t_0 \le t \le \tau_1 \\ \mathcal{H}_u(t), & \tau_1 \le t \le \tau_2 \\ u^L(t), & \tau_2 \le t \le t_1 \end{cases} . \tag{8.12}$$

(8.10)–(8.12) is a non-linear multi-point boundary value problem. The optimal control profile is obtained by computing the switching times τ_1 and τ_2, the initial conditions for adjoints $\boldsymbol{\lambda}_0$, and the Lagrange multiplier for terminal constraint $\boldsymbol{\nu}$, according to the indirect shooting method (Bryson and Ho, 1975). It is found that the performance index $\mathcal{J}^* = 20689$ matches the objective value obtained with the sequential method $\mathcal{J} = 20699$. Figure 8.8 also shows that the terminal constraint (8.9) is active.

8.2.3 Closed-Loop Optimal Control

Full-state measurements and an addition of white noise with variance of 1% to the measured states are considered. Moreover, the following variations in initial conditions $\delta\boldsymbol{x}_0 = \begin{bmatrix}0.17 & -6 & 0.9 & 0.8\end{bmatrix}$ is considered, too. These variations are chosen to cause a performance loss and terminal constraint violation, when applying the open-loop control profile. The measured outputs are taken with frequency of 0.1 s. Run-to-run constraint adaptation is initialised with a constraint bias of $\delta\psi = 0$ and considers a filter gain of $W = 0.2$. This gain is chosen so as to meet the terminal constraint as quickly as possible while avoiding oscillations during the adaptation.

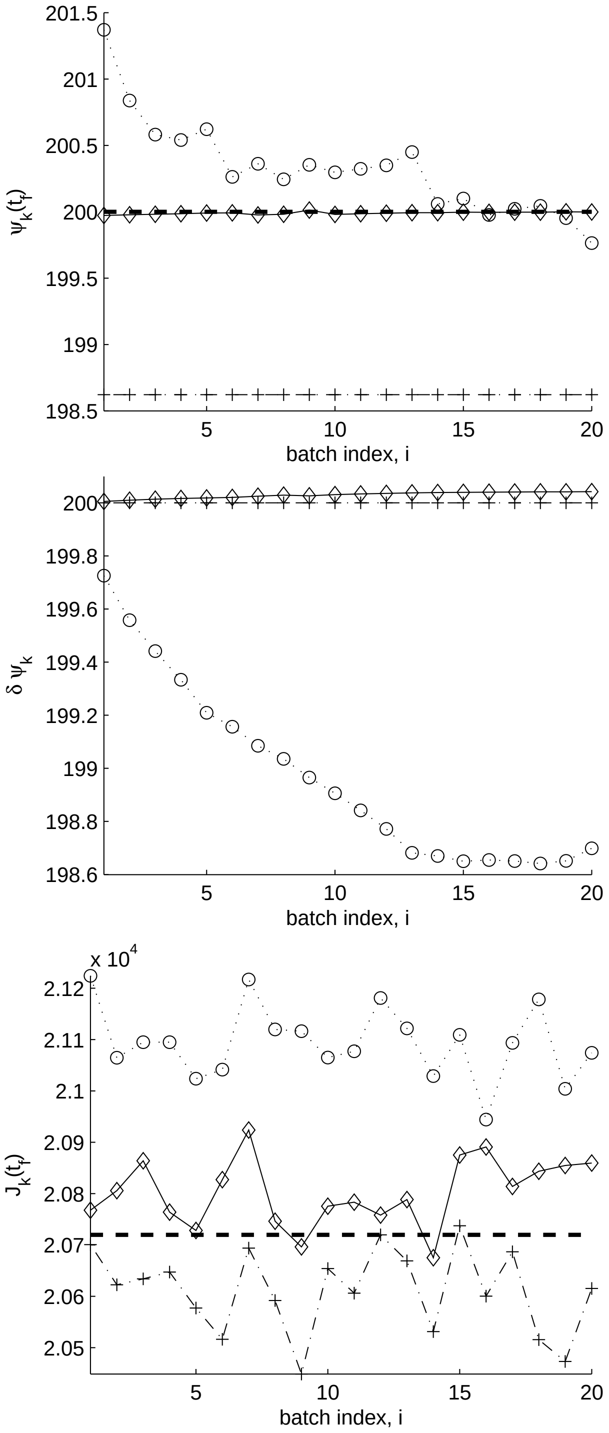

Figure 8.9: **Dotted lines with circles**: constraint adaptation alone, **dash-dotted lines with crosses**: neighbouring extremal control alone, **dashed lines**: optimal solution for perturbed system, **solid lines with diamonds**: integrated two-time-scale scheme control. **Left plot**: evolution of the terminal constraint; **Middle plot**: evolution of the terminal constraint bias; **Right plot**: evolution of the performance index.

Figure 8.9 compares the evolution of the performance during the first 20 batches. The evolution of the terminal constraint is presented in the upper left plot. Observe that pure NE control satisfies the terminal constraint, but is rather conservative. In contrast, pure constraint adaptation violates the terminal constraint in most of the batches. In the last 5 batches, the method almost reaches the terminal constraint. Note that this approach also appears to be more sensitive to measurement noise than other two. The integrated scheme remains in the neighbourhood of the terminal constraint for all batches. Due to the fact that control corrections are applied during each batch as well, this approach is able to correct the control profile with lower sensitivity to measurement noise. The upper right plot of Figure 8.9 shows that the terminal constraint bias varies slightly for the integrated scheme, because the NE controller in the inner loop is able to recover a large portion of the potential constraint violation. In contrast, constraint adaptation requires much larger bias adaptation since no correction is made within the batches. The lower plot of Figure 8.9 shows the evolution of the performance index. The worst average case is for pure NE control. In contrast, pure constraint adaptation exhibits the highest performance, but this is a direct consequence terminal constraint violation. The cost function of the proposed integrated approach is very close to the optimal performance of the perturbed system.

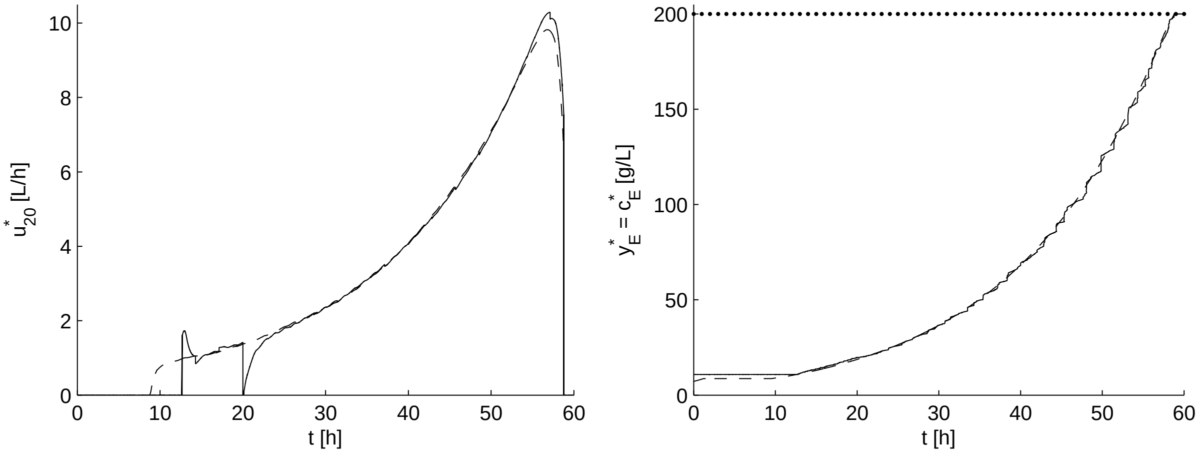

Figure 8.10: Performance with perturbed initial conditions after 20 runs of adaptation. **Left plot**: control trajectory; **Right plot**: state trajectory of liquid volume; **Solid line**: perturbed system with two-time-scale integrated scheme; **Dashed line**: optimal solution for perturbed system.

The resulting control profile after 20 batches is shown in left plot of Figure 8.10. The control still consists of the tree same arcs, but the switching times are now different from the nominal solution displayed in Figure 8.8, as a result of the constraint adaptation. The corresponding measured output of liquid volume is presented in the right plot of Figure 8.10. It can be seen that the measured output of the perturbed process is in very good agreement with the re-optimised solution.

Part III

Application

Chapter 9

Two-tanks Connected in Series

The level control of two connected tanks with liquid interaction is considered to illustrate the integrated two-times-scale approach against a constraint adaptation control scheme and a neighbouring-extremal controller alone. The various control methods are first tested using numerical simulations and then validated experimentally. The experiments are carried out on an Amira DTS200 device, which is shown in Figure 9.1. It consists of 3 connected tube-shaped tanks connected through their bases and comprises six valves to regulate the outflows and leakages. The levels are measured by pressure sensors situated at the bottom of each tank. Also, two inlet flows are available, with pumps feeding the liquid to the first and third tanks.

Only the first two tanks are used in this case study, as illustrated in Figure 9.2. The objective is to control the transition from an initial level to specified level in the second tank, by manipulating the inlet flow $u(t)$ pumped into the first tank. Both the levels h_1 and h_2 are measured. The measurements are provided by pressure sensors. The outflow from the second tank is regulated by the half-opened valve k_{22} and the two tanks interact through the valve k_{11} at their bases.

9.1 Problem Formulation

The process model assumes a constant liquid density and vertical walls for the two tanks. From material balances and Bernoulli's equation, the resulting mathematical model is given by:

$$\dot{h}_1(t) = \frac{u}{F_1} - \frac{k_{11}}{F_1}\sqrt{h_1 - h_2} \tag{9.1}$$

$$\dot{h}_2(t) = \frac{k_{11}}{F_2}\sqrt{h_1 - h_2} - \frac{k_{22}}{F_2}\sqrt{h_2}, \tag{9.2}$$

where the state variables $h_1(t)$ and $h_2(t)$ are levels [cm] in the first tank and in the second tank, respectively; the constants F_1 and F_2 define cross-sectional area of tank bases [cm^2]; k_{11} and k_{22} are valve constants [cm$^{2.5}$ s^{-1}]; and the variable u represents the inlet flow in the first tank [ml s^{-1}].

The initial level values corresponds to a constant inlet flow of $u = 25\,\text{ml}\,\text{s}^{-1}$ which are $h_1(t) = 16\,\text{cm}$ and $h_2(t) = 8\,\text{cm}$. The numerical values of the constants are $F_1 = F_2 =$

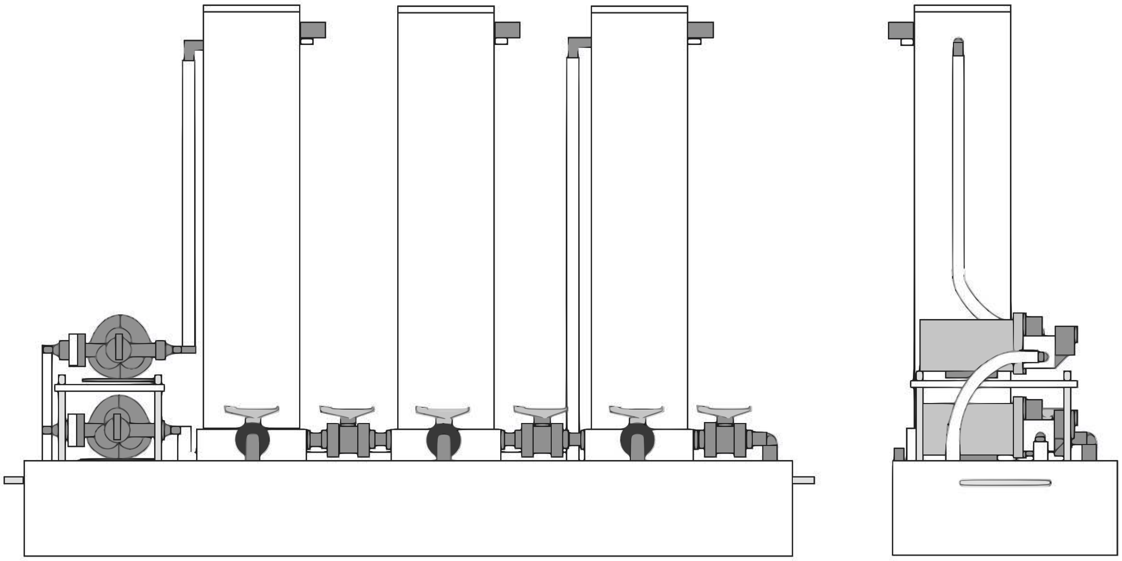

Figure 9.1: **Amira DTS200** – Process for level control of tanks.

$154\,\text{cm}^2$, $k_{11} = 10.68\,\text{cm}^{2.5}\,\text{s}^{-1}$, and $k_{22} = 7.5\,\text{cm}^{2.5}\,\text{s}^{-1}$. The final time is set to $t_1 = 500\,\text{s}$ and the inlet flow u is bounded as:

$$0 \le u \le 100\,[\text{ml}\,\text{s}^{-1}]. \tag{9.3}$$

The terminal constraint reads:

$$h_2(t_1) = h_2^{\text{ref}}. \tag{9.4}$$

with the desired level in the second tank set to $h_2^{\text{ref}} = 25\,\text{cm}$. The objective function is defined so as to minimise the LQ cost:

$$\min_u \mathcal{J} = \int_{t_0}^{t_1} q[h_2(t) - h_2^{\text{ref}}]^2 + r[u(t)]^2 \text{d}t \tag{9.5}$$

where weighting constants are $r = 0.001$ and $q = 1000$. Note $r > 0$ is needed in order to make the control problem non-singular. This way $\mathcal{H}_u$ depends on the control variable and the Hamiltonian $\mathcal{H}$ is regular.

9.2 Open-loop Optimal Control

The analysis of the solution of the optimisation problem (9.1)–(9.5) in Figure 9.3 (left plot) indicates that the optimal control consists of an upper bound, a lower bound, and an interior arc. Good initial guess for the switching times are obtained by application of sequential approach with piecewise constant control parametrisation. The optimal control profile is then obtained by determining the switching times τ_1 and τ_2, the initial conditions for adjoints $\boldsymbol{\lambda}_0$, and the Lagrange multipliers for terminal constraints $\boldsymbol{\nu}$, using the indirect shooting approach (Bryson and Ho, 1975).

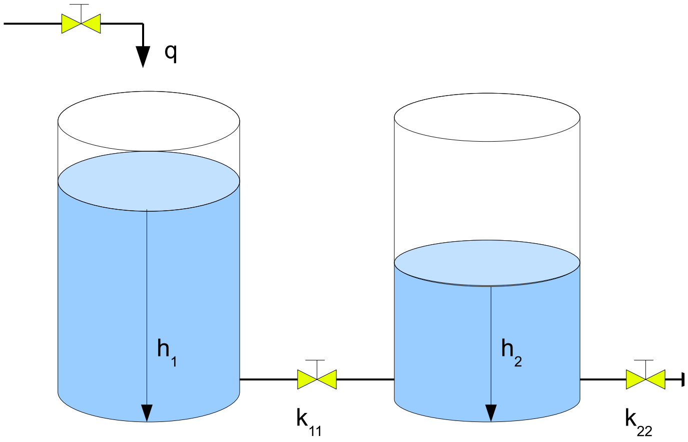

Figure 9.2: Configuration of two tanks connected in series.

9.3 Closed-loop Optimal Control

In order to simulate the real behaviour of the process, the valve constants are perturbed to the following values: $k_{11} = 10.08\,\mathrm{cm}^{2.5}\,\mathrm{s}^{-1}$ and $k_{22} = 8.82\,\mathrm{cm}^{2.5}\,\mathrm{s}^{-1}$. The initial conditions remain unchanged. The measured outputs are two states h_1 and h_2 with addition of white noise of variance 1%, and the measurement frequency is every second. While the NE controller is designed using the nominal mathematical model, the simulations are performed for measured outputs from the perturbed model. The differences between the nominal and perturbed models result in a performance loss and terminal constraint violation, when applying the open-loop control profile from Figure 9.3. The run-to-run constraint adaptation is initialised with a constraint bias of $\delta\psi = 0$ and a filter gain of $W = 0.6$. In the proposed integrated scheme a filter gain of $W = 0.4$ is considered.

Figure 9.4 compares the performance of various approaches during the first 15 runs. The evolution of the terminal constraint is presented in the left plot. Observe that in the first run pure constraint adaptation starts far from the desired value compared to NE control. In the subsequent runs, this constraint remains violated while it converges to the desired value h_2^{ref} after about 5 runs. The integrated scheme starts quite close to the terminal constraint, and converges to the desired value. Due to the fact that control corrections are applied during each run as well, this approach is able to correct the control profile with lower sensitivity to measurement noise than pure constraint adaptation. The right plot of Figure 9.4 shows the evolution of the original terminal constraint plus constraint bias. This value only varies slightly for the integrated scheme because the NE controller in the inner loop is able to recover a large portion of the constraint violation. In contrast, constraint adaptation approach requires larger adaptation since no correction is made during the runs.

The resulting control profile after 15 runs is shown in left plot of Figure 9.5. The optimal control profile still consists of the same tree arcs, but the switching times have now changed compared to nominal solution in Figure 9.3, as a result of the terminal constraint adaptation. Figure 9.6 shows that the level in second tank meets the desired level with the proposed integrated two-time-scale control approach.

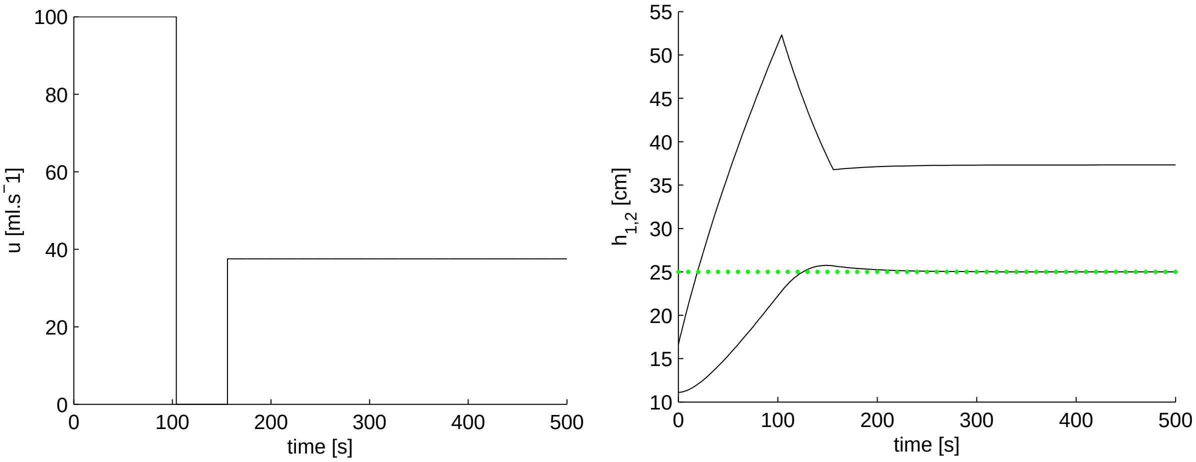

Figure 9.3: **Left:** Nominal control trajectory; **Right:** Simulated response for open-loop implementation of nominal control with the nominal model, and **bold dotted line:** desired level in the second tank.

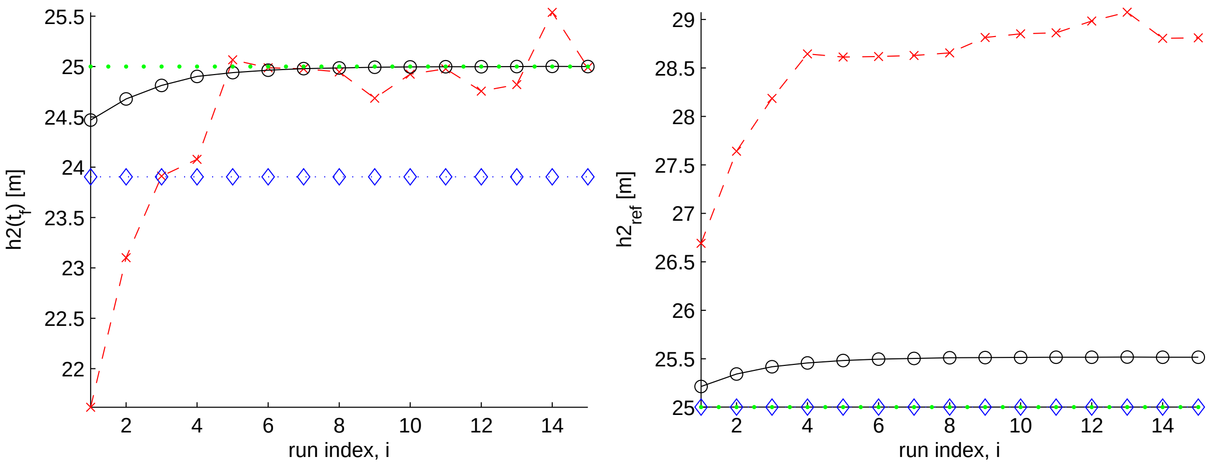

Figure 9.4: Evolution of the terminal constraint during 15 runs. **Dashed line with crosses**: pure constraint adaptation, **dotted line with diamonds**: pure neighbouring extremal control, **solid line with circles**: integrated two-time-scale scheme control, **bold dotted line:** desired level. **Left plot**: Evolution of the measured terminal constraint; **Right plot**: Evolution of the modelled terminal constraint.

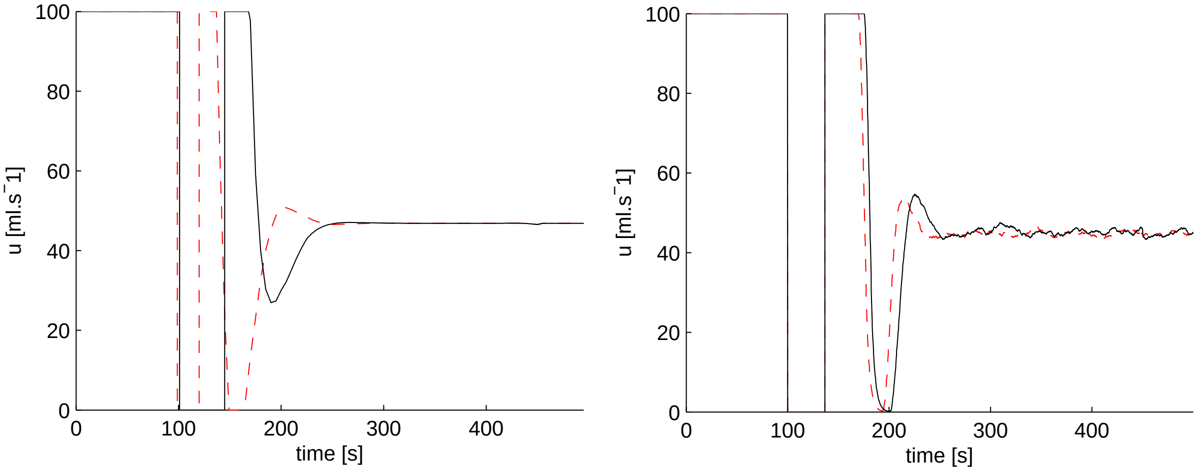

Figure 9.5: Control trajectory of proposed scheme with perturbed valve constants in the 1st and 15th runs. **Left plot**: Simulated results; **Right plot**: Experimental results; **Dashed line**: 1st run; **Solid line**: 15th run.

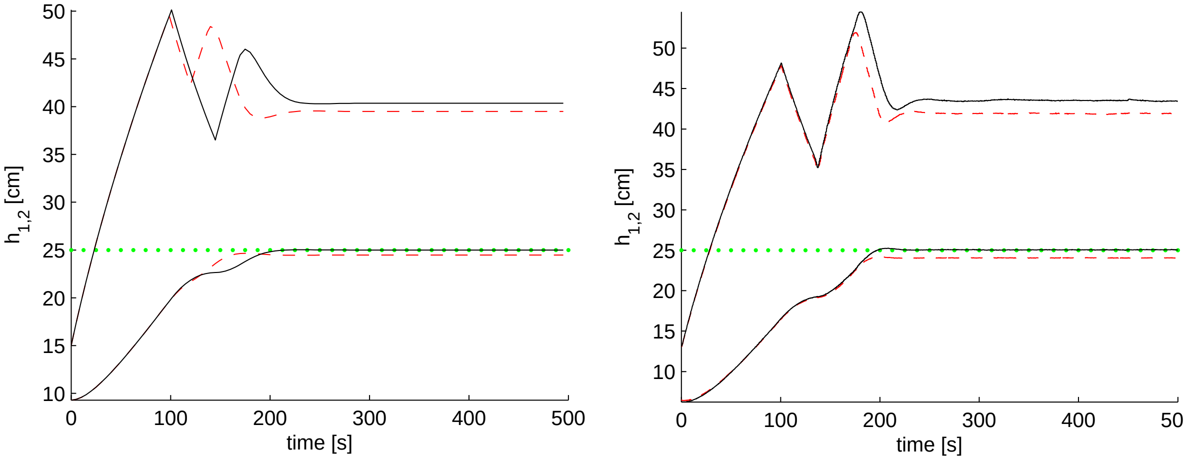

Figure 9.6: Plant response of proposed control scheme with perturbed valve constants in 1st and 15th runs. **Left plot**: Simulated results; **Right plot**: Experimental results; **Dashed line**: 1st run; **Solid line**: 15th run; and **bold dotted line:** desired level in the second tank.

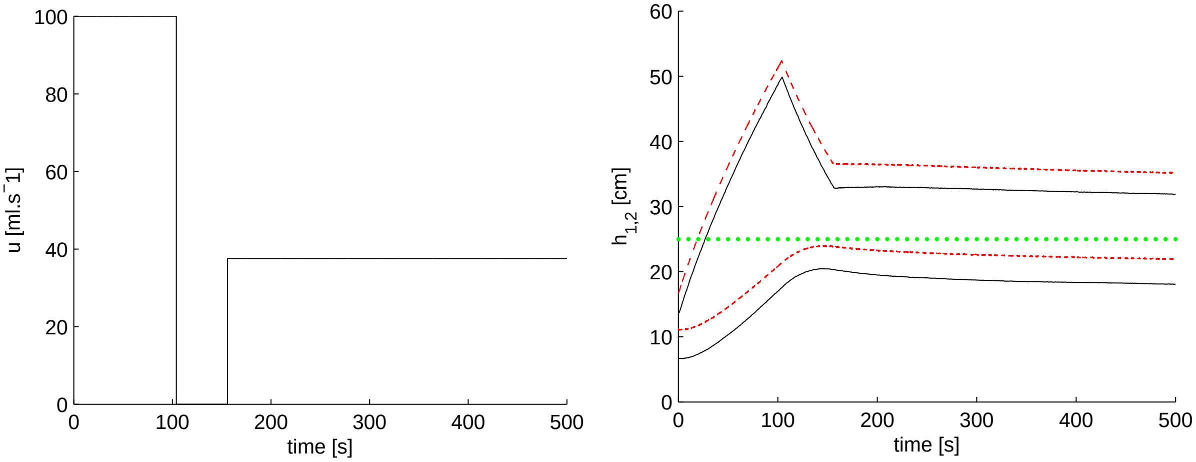

Figure 9.7: **Left:** Nominal control trajectory; **Right:** Plant response for open-loop implementation of nominal control trajectory, **solid line:** obtained for nominal model, **dashed line:** obtained for perturbed model, and **bold dotted line:** desired level in the second tank.

9.4 Experimental Results

The nominal solution is first obtained for certain positions of the outflow valves (leakages). In order to test the performance of the control approaches, the outflow is then increased. This change also causes minor variations in the initial conditions. Both levels can be measured on-line, with a sampling time of 1 s. The conversion between measured outputs (in volts) and states (in centimetres) is considered as another perturbation. The mismatch between the model and the process results in performance loss and terminal constraint violation (see Figure 9.7).

The performance evolution of various control approaches over 15 runs is compared in Figure 9.8. The left plot shows the evolution of the terminal constraint. A similar behaviour as in numerical simulations can be observed. In the first run, NE control and the proposed two-time-scale approach start closer to the desired value than constraint adaptation. NE controller is able to recover some of the constraint violation but not all of it. In the subsequent runs, the two-time-scale controller slowly increases the terminal constraint to reach the desired level after about 6 runs. In contrast, pure run-to-run constraint adaptation also takes 6 runs to come close to optimum, although with a higher value of the filter gain. Moreover, the proposed two-time-scale approach exhibits a lower sensitivity to measurement noise, due to the fact that control corrections are applied during each run as well.

The evolution of the terminal constraint plus constraint bias is depicted in the right plot of Figure 9.8. As previously, pure constraint adaptation approach needs heavier adaptation.

The right plot in Figure 9.5 displays the adapted control profile and Figure 9.6 shows adapted plant response on such a control, after 1 run and 15 runs. Note that the unconstrained arc is no longer constant due to the NE control corrections. Moreover, large uncertainty may cause the NE controller to saturate, as in the case here in Figure 9.5.

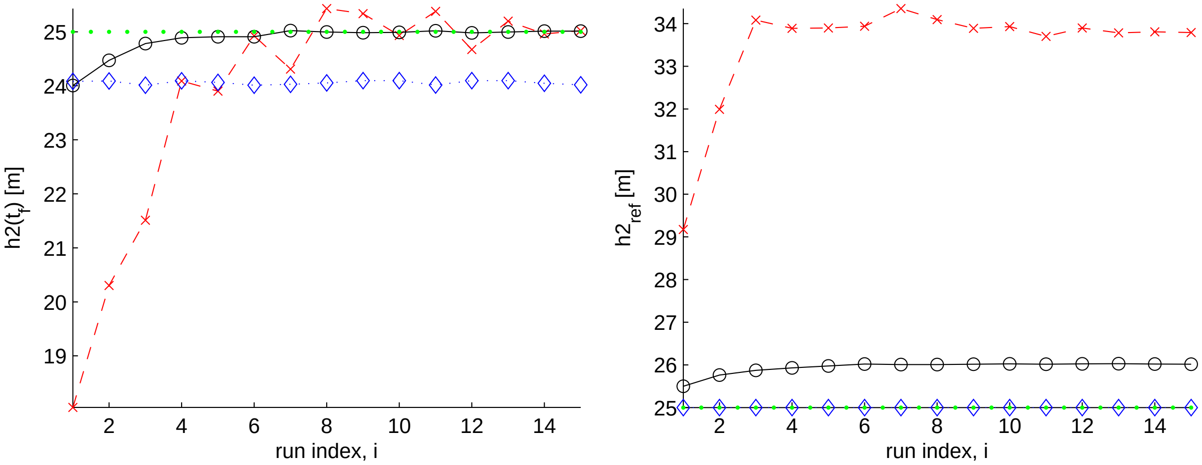

Figure 9.8: Evolution of the terminal constraint during 15 runs (experiment). **Dashed line with crosses**: pure constraint adaptation, **dotted line with diamonds**: pure NE control, **solid line with circles**: integrated two-time-scale control scheme, **bold dotted line:** desired level. **Left plot**: Evolution of the measured terminal constraint; **Right plot**: Evolution of the modelled terminal constraint.

Chapter 10

Conclusions

The thesis deals with the optimisation of the batch processes in the presence of uncertainties. Theoretical as well as practical issues are addressed. The first part is devoted to process optimisation under ideal circumstances. The optimisation problem, the necessary conditions for optimal solution, and the numerical procedures that find the optimal solution are presented here. Finally, the computational procedure to obtain the continuous nominal solution that satisfies the necessary condition of optimality is provided. The second part is devoted to real-time optimisation of the batch processes in the presence of uncertainties. The closed-loop control framework is explained that incorporates the full-state measurements and that tracks the NCO of the perturbed process. Design and the implementation aspects of the NCO-tracking is discussed in detail. Finally, an integrated two-times-scale control scheme for batch processes is proposed. The standalone contributions of the thesis are summarised in the sequel.

The first contribution to this thesis is the overview of the numerical methods for problems of dynamic optimisation. It allows to choose the suitable method for the particular problem. In particular, the orthogonal collocation (OC) approach alongside control vector parametrisation (CVP) approach have widely been used in practice. CVP method is preferred for non-stiff problems with small number of constraints. In opposite, OC is more suitable for stiff problems or for problems with higher number of constraints.

The second result is the implementation of the procedure that combines the direct method with the indirect method to obtain continuous solution that satisfies the NCO and the required constraints. This approach uses the direct method (mostly CVP or OC) to find a good initial guess for the indirect method which is an extension of the optimal control theory. Obviously, such a solution guarantees an optimal control policy of the process when implemented open-loop.

The third result is the implementation of the NCO-tracking scheme for batch processes in chemical industries. The approach proposes systematic algorithms for the design and the implementation phase which have been verified in simulations. This control scheme recovers most of the optimality loss caused by uncertainty and perturbation in the real process. We use simple controllers to meet a set of control points where NCO must hold. Standard PID controllers track both the active constraints and the sensitivities where the cheap solution but less accurate is required for the real-time implementation. For more accurate solution, run-to-run re-optimisation approach (CA) or the time-variant optimal feedback controller

(NE) are used to track the active constraints or the sensitivities, respectively. Another contribution here, is the numerical implementation of the NE controller with terminal constraints. Traditional approach is to pre-compute a system of Ricatti equations and to find gain matrix of the state-feedback controller. Our proposal is to resolve linear TPBVP in every time instant and to apply small predicted horizon of the control variations and the variations in the terminal constraints.

The main contribution, which has been experimentally tested, is the NCO-tracking by two-time-scale control where two decoupled control approaches handle different NCO. The combination of the run-to-run constraint adaptation and NE control with terminal constraints allows to complement the benefits of each other, while mitigating some of their deficiencies. With this control approach, sensitivities along unconstrained arcs are handled by NE controller that corrects the control trajectory in-batch according to the latest output measurements and by taking the terminal conditions into account. The set of active constraints is then adapted via run-to-run re-optimisation that updates the terminal constraints in the model according to the final measurements. The advantages of the integrated scheme are demonstrated in the last part by the case study of a connected two-tank system. The implementation aspects are discussed here, as well. As part of future work, an extension of the current NE control to singular control problems is currently under investigation (Gros, Srinivasan, et al., 2009a,b), as well as the ability to handle problems with state path constraints (Pesch, 1989a; Pesch, 1989b).

References

Abel, O. and W. Marquardt (1998). “A Model Predictive Control Scheme for Safe and Optimal Operation of Exothermic Semi-batch Reactors”. In: *IFAC DYCOPS-5*. Corfu, Greece, pp. 761–766.

Agrawal, S.K. and B.C. Fabien (1999). *Optimization of Dynamic Systems*. Kluwer Academic Publishers, Netherlands.

Allgöwer, F. and A. Zheng (2000). *Nonlinear Model Predictive Control*. Birkhäuser Verlag.

Äström, K.J. and B. Wittenmark (1983). “Theory and Application of Adaptive Control – A Survey”. In: *Automatica* **19**(5), pp. 471–486.

Äström, K.J. and B. Wittenmark (1989). *Adaptive Control*. Addison-Wesley, Massachusetts.

Bemporad, A., M. Morari, V. Dua, and E. N. Pistikopoulos (2002). “The explicit linear quadratic regulator for constrained systems”. In: *Automatica* **38**(1), pp. 3–20.

Biegler, L. T. and R.R. Hughes (1983). “Process Optimization: A comparative Case Study”. In: *Comp. Chem. Eng.* **7**(5), pp. 645–661.

Biegler, L.T. (1984). “Solution of Dynamic Optimization Problems by Successive Quadratic Programming and Orthogonal Collocation”. In: *Comp. Chem. Eng.* **8**(3-4), pp. 243–248.

Biegler, L.T., A.M. Cervantes, and A. Wächter (2002). “Advances in Simultaneous Strategies for Dynamic Process Optimization”. In: *Chemical Engineering Science* **57**(4), pp. 575–593.

Blackman, P.F. (1962). “Extremum-seeking Regulators”. In: *An exposition of adaptive control*. New York: The Macmillan Company.

Bock, H.G. and K.J. Platt (1984). “A Multiple Shooting Algorithm for Direct Solution of Optimal Control Problems”. In: *In the 9th IFAC world congress*. Budapest, Hungary, pp. 242–247.

Bonvin, D., B. Srinivasan, and D. Hunkeler (2006). “Control and Optimization of Batch Processes: Improvement of Process Operation in the Production of Specialty Chemicals”. In: *IEEE Control Systems Magazine* **26**(6), pp. 34–45.

Breakwell, J.V., J. Speyer, and A.E. Bryson (1963). “Optimization and Control of Nonlinear Systems using the Second Variation”. In: *SIAM J. Control* **1**(2), pp. 193–223.

Bryson, A. E. and Yu-Chi Ho (1975). *Applied Optimal Control – Optimization, Estimation and Control*. Hemisphere publishing corporation.

Cannon, M., V. Deshmukh, and B. Kouvaritakis (2003). “Nonlinear Model Predictive Control with Polytopic Invariant Sets”. In: *Automatica* **39**(8), pp. 1487–1494.

Cannon, M., V. Deshmukh, and B. Kouvaritakis (2004). “Enlargement of Polytopic Terminal Region in NMPC by Interpolation and Partial Invariance”. In: *Automatica* **40**(2), pp. 311–317.

Carey, G.F. and Finlayson. B.A. (1975). “Orthogonal Collocation on Finite Elements”. In: *Chemical Engineering Science* **30**, pp. 587–1596.

Cervantes, A. and L.T. Biegler (1998). “Large-Scale DAE Optimization Using a Simultaneous NLP Formulation”. In: *AIChE Journal* **44**(5), pp. 1038–1050.

Chachuat, B. (2007). *Nonlinear and Dynamic Optimization: From Theory to Practice*. URL: http://infoscience.epfl.ch/getfile.py?recid=111939&mode=best.

Chachuat, B., A. Marchetti, and D. Bonvin (2008). “Process Optimization via Constraints Adaptation”. In: *Journal of Process Control* **18**(3-4), pp. 244–257.

Chen, C.-T. and C. Hwang (1990). “Optimal control computation for differential-algebraic process systems with general constraints”. In: *Chemical Engineering Communications* **1**(97), pp. 9–26.

Chen, W.-H., D. J. Ballance, and J. O’Reilly (2001). “Optimization of Attraction Domains of Nonlinear MPC via LMI Methods”. In: *Proceedings of the American control conference*. Arlington, USA, pp. 3067–3072.

Čižniar, M., T. Hirmajer, M. Podmajerský, M. Fikar, and M. A. Latifi (2008). “NMPC based on global optimisation”. In: *Proceedings of the 8th International Scientific - Technical Conference Process Control 2008*. Kouty nad Desnou, Czech Republic: University of Pardubice, c025b–1–c025b–6.

Čižniar, M., M. Podmajerský, T. Hirmajer, M. Fikar, and M. A. Latifi (2009). “Global optimization for parameter estimation of differential-algebraic systems”. In: *Chemical Papers* **63**(3), pp. 274–283.

Crescitelli, S. and B. Nicoletti (1973). “Near Optimal Control of Batch Reactors”. In: *Chemical Engineering Science* **28**, pp. 463–471.

Cuthrell, J. E. and L. T. Biegler (1989). “Simultaneous-optimization and Solution Methods for Batch Reactor Control Profiles”. In: *AIChE J.* **13**, pp. 49–62.

Diehl, M., H. G. Bock, J. P. Schlöder, Findeisen R., Nagy Z., and Allgöwer F. (2002). “Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations”. In: *Journal of Process Control* **12**(4), pp. 577 –585.

Diehl, M., J. Gerhard, W. Marquardt, and M. Mönnigmann (2008). “Numerical solution approaches for robust nonlinear optimal control problems”. In: *Computers & Chemical Engineering* **32**, pp. 1279–1292.

Dominguez, L. F. and E. N. Pistikopoulos (2010). “Multiparametric programming based algorithms for pure integer and mixed-integer bilevel programming problems”. In: *Computers & Chemical Engineering* **34**(12), pp. 2097 –2106.

Dovi, V.G. and A.P. Reverberi (1993). “Optimal Solution of Processes Described by System of Differential Algebraic Equations”. In: *Chemical Engineering Science* **48**(14), pp. 2609–2614.

Doyle, J. C. and G. Stein (1981). “Multivariable Feedback Design: Concepts for a Classical/Modern Synthesis”. In: *IEEE Transactions on Automatic Control* **23**, pp. 607–611.

Eaton, J. W. and J. B. Rawlings (1990). “Feedback-control of Chemical Processes using Online Optimization Techniques”. In: *Comp. Chem. Eng.* **14**, pp. 469–479.

Edgar, T. F. and D. M. Himmelblau (1988). *Optimization of Chemical Processes.* McGraw-Hill, New York.

Edgar, T. F., T.A. Soderstrom, and D. M. Himmelblau (2001). “A Mixed Integer Optimization Approach for Simultaneous Data Reconciliation and Identification of Measurement Bias”. In: *Control Engineering Practice* **9**(8), pp. 869–876.

Fikar, M., Ľ. Čirka, M. Herceg, and M. Podmajerský (2008). “E-learning in Course Operating Systems”. In: *Proceedings of 9th International Conference Virtual University 2008.* FEI STU in Bratislava, E-academia Slovaca. Bratislava: E-academia Slovaca, fid000091.pdf.

Finlayson, B.A. (1980). *Nonlinear Analysis in Chemical Engineering.* McGraw-Hill, New York.

François, G., B. Srinivasan, and D. Bonvin (2007). “Use of Measurements for Enforcing the Necessary Conditions of Optimality in the Presence of Constraints and Uncertainty”. In: *Journal of Process Control* **15**, pp. 701–712.

Garcia, C. E., D. M. Prett, and M. Morari (1989). “Model Predictive Control: Theory and Practice – A Survey”. In: *Automatica* **25**(3), pp. 335–348.

Gill, P.E., W. Murray, and M.H. Wright (1981). *Practical Optimization.* Academic Press, London.

Gros, S. (2007). “Neighboring Extremals in Optimization and Control”. PhD thesis. École Polytechnique Fédérale de Lausanne.

Gros, S., B. Srinivasan, B. Chachuat, and D. Bonvin (2009a). “Neighboring-Extremal Control for Singular Dynamic Optimization Problems. I-Single-Input Systems”. In: *International Journal of Control* **82**(6), pp. 1099–1112.

Gros, S., B. Srinivasan, B. Chachuat, and D. Bonvin (2009b). “Neighboring-Extremal Control for Singular Dynamic Optimization Problems. II-Multiple-Input Systems”. In: *International Journal of Control* **82**(7), pp. 1193–1211.

Guntern, C., A.H. Keller, and K. Hungerbuhler (1998). “Economic Optimization of an Industrial Semi-batch Reactor Applying Dynamic Programming”. In: *Industrial and Engineering Chemistry Research* **37**(10), pp. 4017–4022.

Hertzberg, T. and O.A. Asbjornsen (1977). *Parameter Estimation in Nonlinear Differential Equations: Computer Applications in the Analysis of Data and Plants.* Science Press, Princeton.

Hindmarsh, A. C. et al. (2005). “SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers”. In: *ACM Transactions on Mathematical Software* **31**(3), pp. 363–396.

Hirmajer, T. and M. Fikar (2006). “Optimal Control of a Two-Stage Reactor System”. In: *Chemical Papers* **60**(5), pp. 381–387.

Ishikawa, T., Y. Natori, L. Liberis, and C.C. Pantelides (1997). “Modeling and Optimization of Industrial Batch Process for the Production of Dioctyl Phthalte”. In: *Comp. Chem. Eng.* **21**, pp. 1239–1244.

Jacoby, S.L.S., J.S. Kowalik, and J.T. Pizzo (1972). *Iterative Methods for Nonlinear Optimization Methods.* Prentice-Hall Inc., Englewood Cliffs, New Jersy.

Kadam, J. V. and W. Marquardt (2007). "Integration of Economical Optimization and Control for Intentionally Transient Process Operation". In: *Lecture Notes in Control and Information Sciences* **358**, pp. 419–434.

Kelley, H.J., R.E. Kopp, and G. Moyer (1963). "A Trajectory Optimization Technique based upon the Theory of the Second Variation". In: *AIAA Astrodynamics Conference*, pp. 193–223.

Kirk, D.E. (1970). *Optimal Control Theory: an Introduction.* Prentice-Hall, London.

Klatt, K.U. and S. Engell (1993). "Gain Scheduling Control of a non-minimum-phase CSTR". In: *European Control Conference.* Groning, Netherlands, pp. 2323–2328.

Kristic, M and H-H. Wang (2000). "Stability of Extremum Seeking Feedback for General Non-linear Dynamic Systems". In: *Automatica* **36**, pp. 595–601.

Kvasnica, M. (2009). *Real-Time Model Predictive Control via Multi-Parametric Programming: Theory and Tools.* Saarbruecken: VDM Verlag.

Lau, K.H. and D.L. Ulrichson (1992). "Effects of Local Constraints on the Convergence Behavior of Sequential Modular Simulators". In: *Comp. Chem. Eng.* **15**(9), pp. 887–892.

Lee, E. B. and L. Markus (1968). *Foundations of Optimal Control Theory.* John Wiley, New York.

Logsdon, J. S. and L. T. Biegler (1989). "Accurate Solution of Differential Algebraic Optimization Problems". In: *Ind. Eng. Chem. Res.* **18**(11), pp. 1628–1639.

Marchetti, A., B. Chachuat, and D. Bonvin (2007). "Batch process optimization via run-to-run constraints adaptation". In: *European Control Conference.* Kos, Greece.

McFarlane, D. C. and K. Glover (1989). *Robust Controller Design Using Normalized Coprime Factor Plant Descriptions (LNCIS).* Springer, New York.

Morosanov, I. S. (1957). "Method of Extremum Control". In: *Automatic & Remote Control* **18**, pp. 1077–1092.

Mujtaba, I. M. and S. Macchietto (1997). "Efficient Optimization of Batch Distillation with Chemical Reaction using Polynomial Curve Fitting Techniques". In: *Ind. Eng. Chem. Res.* **36**(6), pp. 2287–2295.

Murthy, B.S.N., K. Gangiah, and A. Husain (1980). "Performance of Various Methods in Computing Optimal Policies". In: *The Chemical Engineering Journal* **19**, pp. 201–208.

Ostrovskii, I. I. (1957). "Extremum Regulation". In: *Automatic & Remote Control* **18**, pp. 900–907.

Pervozvanskii, A. A. (1960). "Continuous Extremum Control System in the Presence of Random Noise". In: *Automatic & Remote Control* **21**, pp. 673–677.

Pesch, H. J. (1989a). "Real-Time Computation of Feedback Controls for Constrained Optimal Control Problems. Part II: A Correction Method based on Multiple Shooting". In: *Optimal Control Applications & Methods* **10**, pp. 147–171.

Pesch, H.J. (1989b). "Real-time computation of feedback controls for constrained optimal control problems. Part I: Neighbouring extremals". In: *Optimal Control Applications & Methods* **10**, pp. 129–145.

Pesch, H.J. and R. Bulirsch (1994). "The Maximum Principle, Bellman's Equation, and Caratheodory's Work". In: *J. Optimization Theory and Applications* **80**(2), pp. 199–225.

Pesh, H. J. (1990). “The Accessory Minimum Problem and Its Importance for the Numerical Computation of Closed-Loop Controls”. In: *Conference on Decision and Control*. Honolulu, HI, pp. 952–953.

Pistikopoulos, E., M. Georgiadis, and V. Dua (2007a). *Multi-Parametric Model-Based Control, volume 2 of Process Systems Engineering*. WILEY-VCH Verlag.

Pistikopoulos, E., M. Georgiadis, and V. Dua (2007b). *Multi-Parametric Programming, volume 1 of Process Systems Engineering*. WILEY-VCH Verlag.

Podmajerský, M. (2005). “Návrh PI regulátora cez Internet”. MA thesis. Bratislava: KIRP FCHPT STU.

Podmajerský, M. (2007). “Parameter Estimation in Processes from Experimental Data”. MA thesis. FCHPT STU.

Podmajerský, M., B. Chachuat, and M. Fikar (2010a). “Dynamic Optimisation of Batch Processes by Integrated Two-Time-Scale Scheme”. In: *Acta Chimica Slovaca* **3**(2), pp. 38–56.

Podmajerský, M., B. Chachuat, and M. Fikar (2010b). “Integrated Two-Time-Scale Scheme for Dynamic Real-Time Optimisation”. In: *Proceedings of the 9th International Scientific - Technical Conference Process Control 2010*. Kouty nad Desnou, Czech Republic: University of Pardubice, C088a–1–C088a–14.

Podmajerský, M., B. Chachuat, and M. Fikar (2010c). “Measurement-based Optimisation of Batch Process by Integrated Two-Time-Scale Scheme”. In: *VOCAL 2010, Program and Abstracts*, pp. 93–93.

Podmajerský, M., B. Chachuat, and M. Fikar (2011a). “Integrated Two-Time-Scale Scheme for Real-time Optimisation of Batch Processes”. In: *Proc. of the 18th IFAC World Congress*. Milano, Italy.

Podmajerský, M., B. Chachuat, and M. Fikar (2011b). “Measurement-based Optimisation of Batch Processes using an Integrated Two-Time-Scale Scheme”. In: *Optimization and Engineering*. (Submitted).

Podmajerský, M., B. Chachuat, and M. Fikar (2011c). “Run-to-run Optimisation of Batch Processes with In-batch Controller”. In: *Selected Topics on Constrained and Nonlinear Control. Preprints*. STU Bratislava - NTNU Trondheim, pp. 337–342.

Podmajerský, M., M. Čižniar, T. Hirmajer, M. Fikar, and M. A. Latifi (2007). “Recent Developments in Dynopt Package”. In: *Proceedings of the 16th International Conference Process Control ’07*. Slovak University of Technology in Bratislava, 032f.pdf.

Podmajerský, M. and M. Fikar (2008). “Measurement-based Run-to-run Optimisation of Hybrid Two-stage Reactor System”. In: *Proceedings of the 8th International Scientific - Technical Conference Process Control 2008*. Kouty nad Desnou, Czech Republic: University of Pardubice, C025a–1–C025a–11.

Podmajerský, M. and M. Fikar (2009a). “On-Line Neighbouring-Extremal Controller Design for Setpoint-Transition in Presence of Uncertainty”. In: *AT&P Journal Plus*(2), pp. 77–83.

Podmajerský, M. and M. Fikar (2009b). “On-Line Neighbouring-Extremal Controller Design for Setpoint-Transition in Presence of Uncertainty”. In: *Proceedings of the 17th International Conference on Process Control 09*. Štrbské Pleso, Slovakia: Slovak University of Technology in Bratislava, pp. 107–113.

Podmajerský, M. and M. Fikar (2010). “Measurement-based Run-to-run Optimisation of Hybrid Two-stage Reactor System”. In: *Selected Topics in Modelling and Control.* **6**. Slovak University of Technology Press, pp. 44–51.

Podmajerský, M. and M. Fikar (2011). “Real-time Dynamic Optimisation by Integrated Two-Time-Scale Scheme”. In: *Proceedings of the 18th International Conference on Process Control 11.* Slovak University of Technology in Bratislava. Štrbské Pleso, Slovakia.

Ray, W.H. (1981). *Advanced Process Control.* Mc Graw Hill, New York.

Ray, W.H. and J. Szekely (1973). *Process Optimization.* John Wiley, New York.

Reklaitis, G.V., A. Ravindran, and K.M. Ragsdell (1983). *Engineering Optimization: Methods and Applications.* John Wiley and Sons, New York.

Renfro, J.G., A.M. Morshedi, and O.A. Asbjornsen (1987). “Simultaneous Optimization and Solution of Systems Described by Differential Algebraic Equations”. In: *Comp. Chem. Eng.* **11**(5), pp. 503–517.

Ruppen, D., C. Benthack, and D. Bonvin (1995). “Optimization of Batch Reactor Operation under Parametric Uncertainty – Computational Aspects”. In: *Journal of Process Control* **5**(4), pp. 235–240.

Schulz, V.H., H.G. Bock, and M.C. Steinbach (1998). “Exploiting Invariants in the Numerical Solution of Multipoint Boundary Value Problems for DAEs”. In: *SIAM Journal of Scientific Computing* **19**, pp. 440–467.

Solheim, O.A. (1976). *Optimal Regulering.* Tapir, Norway.

Sorensen, E., S. Macchietto, G. Stuart, and S. Skogestad (1996). “Optimal Control and Online Operation of Reactive Batch Distillation”. In: *Comp. Chem. Eng.* **20**(12), pp. 1491–1498.

Srinivasan, B. and D. Bonvin (2004a). “Dynamic Optimization under Uncertainty via NCO Tracking: A Solution Model Approach”. In: Technical report. Switzerland: Laboratoire d’Automatique, École Polytechnique Fédérale de Lausanne.

Srinivasan, B. and D. Bonvin (2004b). “Dynamic Optimization under Uncertainty via NCO Tracking: A Solution Model Approach”. In: *BatchPro Symposium.* Plenary talk, pp. 17–35.

Srinivasan, B. and D. Bonvin (2007). “Real-Time Optimization of Batch Processes by tracking the Necessary Conditions of Optimality”. In: *Indus. Eng. Chem. Res.* **46**(2), pp. 492–504.

Srinivasan, B., D. Bonvin, E. Visser, and S. Palanki (2003a). “Dynamic Optimization of Batch Processes: I. Characterization of the Nominal Solution”. In: *Computing and Chemical Engineering* **27**, pp. 1–26.

Srinivasan, B., D. Bonvin, E. Visser, and S. Palanki (2003b). “Dynamic Optimization of Batch Processes: II. Role of Measurements in Handling Uncertainty”. In: *Computing and Chemical Engineering* **27**, pp. 27–44.

Srinivasan, B., P. Myszkorowski, and D. Bonvin (1995). “A Multi-Criteria Approach to Dynamic Optimization”. In: *ACC’95*, pp. 1767–1771.

Srinivasan, B, C.J. Primus, D. Bonvin, and N.L. Ricker (2001). “Run-to-run Optimization of Batch Processes via Generalized Constraint Control”. In: *Control Eng. Practice* **9**, pp. 911–919.

Srinivasan, B., E. Visser, and D. Bonvin (1997). "Optimization-Based Control with Imposed Feedback Structures". In: *IFAC Symposium ADCHEM*. Banff, Canada, pp. 635–640.

Storen, S. and T. Hertzberg (1995). "The Sequential Quadratic Programming Algorithm for Solving Dynamic Optimization Problems – A Review". In: *Comp. Chem. Eng.* **19**, pp. 495–500.

Terwiesch, P., M. Agarwal, and D.W.T. Rippin (1994). "Batch Unit Operation with Imperfect Modeling: A Survey". In: *Journal of Process Control* **4**, pp. 238–258.

Tikhomirov, V.M. (1986). *Stories about Maxima and Minima*. American Mathematical Society.

Troutman, J. L. (1995). *Variational Calculus and Optimal Control: Optimization with Elementary Convexity*. New York: Springer.

Vassiliadis, V. S., R. W. H. Sargent, and C. C. Pantelides (1994a). "Solution of a Class of Multistage Dynamic Optimization Problems. 1. Problems without Path Constraints". In: *Eng. Chem. Res.* **33**(9), pp. 2111–2122.

Vassiliadis, V. S., R. W. H. Sargent, and C. C. Pantelides (1994b). "Solution of a Class of Multistage Dynamic Optimization Problems. 2. Problems with Path Constraints". In: *Eng. Chem. Res.* **33**(9), pp. 2123–2133.

Vemuri, J.Y. (2004). "Real-time Optimization of Semi-batch Reactors". PhD thesis. The Florida State University, College of Engineering.

Villadsen, J. and M.L. Michaelsen (1978). *Solution of Differential Equation Models by Polynomial Approximation*. Prentice-Hall, Englewood Cliffs, New York.

Visser, E., B. Srinivasan, S. Palanki, and D. Bonvin (1999). "On-Line Optimization of Batch Processes by Tracking States and Co-states". In: *IFAC World Congress*. Beijing, China, pp. 31–36.

Víteček, A. and M. Vítečková (2002). *Optimal Control Systems*. 1st ed. (in Czech). Ostrava: VŠB Technická Univerzita Ostrava.

Vusse, J. G. van de (1964). "Plug-flow Type Reactor Versus Tank Reactor". In: *Chem. Eng. Sci*(19), pp. 994–998.

Wächter, A. and L. T. Biegler (2006). "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical Programming* **106** (1), pp. 25–57.

Welz, C., B. Srinivasan, A. Marchetti, D. Bonvin, and N.L. Ricker (2006). "Evaluation of Input Parameterization for Batch Process Optimization". In: *AIChE Journal* **52**(9), pp. 3155–3163.

Würth, L., R. Hannemann, and W. Marquardt (2009a). "Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization". In: *Journal of Process Control* **19**(8), pp. 1277–1288.

Würth, L., R. Hannemann, and W. Marquardt (2009b). "Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization". In: *Journal of Process Control* **19**(8), pp. 1277–1288.

Würth, L., R. Hannemann, and W. Marquardt (2011). "A two-layer architecture for economically optimal process control and operation". In: *Journal of Process Control* **21**(3), pp. 311–321.

Zavala, V. M. and Biegler. L. T. (2009). “The advanced-step NMPC controller: Optimality, stability and robustness”. In: *Automatica* **45**(1), pp. 86–93.

Zhou, K., J. C. Doyle, and K. Glover (1995). *Robust and Optimal Control*. Prentice Hall, Englewood Cliffs, New Jersey.

Publication List

Article in CC Journal

Čižniar, M., M. Podmajerský, T. Hirmajer, M. Fikar, and M. A. Latifi (2009). “Global optimization for parameter estimation of differential-algebraic systems”. In: *Chemical Papers* **63**(3), pp. 274–283.

Podmajerský, M., B. Chachuat, and M. Fikar (2011b). “Measurement-based Optimisation of Batch Processes using an Integrated Two-Time-Scale Scheme”. In: *Optimization and Engineering*. (Submitted).

Article in Journal

Podmajerský, M., B. Chachuat, and M. Fikar (2010a). “Dynamic Optimisation of Batch Processes by Integrated Two-Time-Scale Scheme”. In: *Acta Chimica Slovaca* **3**(2), pp. 38–56.

Podmajerský, M. and M. Fikar (2009a). “On-Line Neighbouring-Extremal Controller Design for Setpoint-Transition in Presence of Uncertainty”. In: *AT&P Journal Plus*(2), pp. 77–83.

Chapter or Pages in Book

Podmajerský, M. and M. Fikar (2010). “Measurement-based Run-to-run Optimisation of Hybrid Two-stage Reactor System”. In: *Selected Topics in Modelling and Control*. **6**. Slovak University of Technology Press, pp. 44–51.

Article in Conference Proceedings

Čižniar, M., T. Hirmajer, M. Podmajerský, M. Fikar, and M. A. Latifi (2008). “NMPC based on global optimisation”. In: *Proceedings of the 8th International Scientific - Technical Conference Process Control 2008*. Kouty nad Desnou, Czech Republic: University of Pardubice, c025b–1–c025b–6.

Fikar, M., Ľ. Čirka, M. Herceg, and M. Podmajerský (2008). “E-learning in Course Operating Systems”. In: *Proceedings of 9th International Conference Virtual University 2008*. FEI STU in Bratislava, E-academia Slovaca. Bratislava: E-academia Slovaca, fid000091.pdf.

Podmajerský, M., B. Chachuat, and M. Fikar (2010b). “Integrated Two-Time-Scale Scheme for Dynamic Real-Time Optimisation”. In: *Proceedings of the 9th International Scientific - Technical Conference Process Control 2010*. Kouty nad Desnou, Czech Republic: University of Pardubice, C088a–1–C088a–14.

Podmajerský, M., B. Chachuat, and M. Fikar (2010c). “Measurement-based Optimisation of Batch Process by Integrated Two-Time-Scale Scheme”. In: *VOCAL 2010, Program and Abstracts*, pp. 93–93.

Podmajerský, M., B. Chachuat, and M. Fikar (2011a). “Integrated Two-Time-Scale Scheme for Real-time Optimisation of Batch Processes”. In: *Proc. of the 18th IFAC World Congress*. Milano, Italy.

Podmajerský, M., B. Chachuat, and M. Fikar (2011c). “Run-to-run Optimisation of Batch Processes with In-batch Controller”. In: *Selected Topics on Constrained and Nonlinear Control. Preprints*. STU Bratislava - NTNU Trondheim, pp. 337–342.

Podmajerský, M., M. Čižniar, T. Hirmajer, M. Fikar, and M. A. Latifi (2007). “Recent Developments in Dynopt Package”. In: *Proceedings of the 16th International Conference Process Control ’07*. Slovak University of Technology in Bratislava, 032f.pdf.

Podmajerský, M. and M. Fikar (2008). “Measurement-based Run-to-run Optimisation of Hybrid Two-stage Reactor System”. In: *Proceedings of the 8th International Scientific - Technical Conference Process Control 2008*. Kouty nad Desnou, Czech Republic: University of Pardubice, C025a–1–C025a–11.

Podmajerský, M. and M. Fikar (2009b). “On-Line Neighbouring-Extremal Controller Design for Setpoint-Transition in Presence of Uncertainty”. In: *Proceedings of the 17th International Conference on Process Control 09*. Štrbské Pleso, Slovakia: Slovak University of Technology in Bratislava, pp. 107–113.

Podmajerský, M. and M. Fikar (2011). “Real-time Dynamic Optimisation by Integrated Two-Time-Scale Scheme”. In: *Proceedings of the 18th International Conference on Process Control 11*. Slovak University of Technology in Bratislava. Štrbské Pleso, Slovakia.

Diploma Thesis

Podmajerský, M. (2007). “Parameter Estimation in Processes from Experimental Data”. MA thesis. FCHPT STU.

Bachelor’s Thesis

Podmajerský, M. (2005). “Návrh PI regulátora cez Internet”. MA thesis. Bratislava: KIRP FCHPT STU.

Curriculum Vitae

Marián Podmajerský

born in 10th of January 1983 in Malacky, Slovakia.

09/2007 — 09/2011	Doctorate study (PhD study), Slovak University of Technology in Bratislava, Slovakia
09/2009 — 03/2010	Visiting researcher, McMaster University Hamilton, Canada
05/2008 — 08/2008	Visiting researcher, EPFL Lausanne, Switzerland
11/2006 — 05/2007	Visiting student, LSGC Nancy, France
09/2005 — 06/2007	Graduate study (MSc. study), Slovak University of Technology in Bratislava, Slovakia
09/2001 — 06/2005	Undergraduate study (BSc. study), Slovak University of Technology in Bratislava, Slovakia