Habilitationsschrift

eingereicht bei der

Fakultät für Mathematik und Informatik

der

RUPRECHT–KARLS–UNIVERSITÄT
HEIDELBERG

Vorgelegt von

Dr. rer. nat. Sebastian Sager

aus Westerstede in Niedersachsen

2011

On the Integration of Optimization Approaches for Mixed-Integer Nonlinear Optimal Control

Sebastian Sager
Interdisciplinary Center for Scientific Computing
University of Heidelberg

Zusammenfassung

In dieser Arbeit untersuchen wir gemischt-ganzzahlige nichtlineare Optimalsteuerungsprobleme (MIOCPs). Darunter verstehen wir Optimierungsprobleme mit unterliegenden Differentialgleichungen und Steuerfunktionen, die Ganzzahligkeitsbedingungen erfüllen müssen. Typische Beispiele sind die Gangwahl von Fahrzeugen, logische Entscheidungen wie “messe ich zu einem gegebenen Zeitpunkt oder nicht?” oder verfahrenstechnische Prozesse mit ein-aus Pumpen. In praktischen Anwendungen kommen häufig zusätzliche Charakteristika hinzu. Beispiele sind kombinatorische Bedingungen, Unsicherheiten verschiedener Art, Verzögerungseffekte oder die Existenz multipler Optimierungskriterien.

In dieser Arbeit erläutern wir algorithmische Ansätze für MIOCPs. Wir stellen neue Algorithmen vor, die in der Lage sind, auch praktische Optimierungsprobleme schnell und zuverlässig zu lösen. Sie basieren auf neuen theoretischen Ergebnissen. Auf der unteren Ebene werden relaxierte, kontinuierliche Optimalsteuerungsprobleme gelöst. Hierzu nutzen wir direkte simultane Methoden, insbesondere Bocks direkte Mehrzielmethode. Diese grundsätzliche Vorgehensweise wurde schon in früheren Arbeiten vorgeschlagen. Die Beiträge dieser Arbeit gehen dahingehend weiter, als der internationale Forschungsstand für MIOCPs erweitert wird durch

- einen Algorithmus für eine generische Problemklasse von MIOCPs mit bewiesener Terminierung und Konvergenz zu einer ε-zertifizierten Lösung,
- Theoreme, die die beste untere Schranke für MIOCPs garantieren,
- ein Korollar, das die Abschätzung des Hausdorff-Abstandes zwischen den Erreichbarkeitsmengen einer disjunkten Steuerungsmenge und ihrer konvexen Hülle verbessert und das Resultat auf den nichtlinearen Fall verallgemeinert,
- struktur-ausnutzende lineare Algebra für die Äußere Konvexifizierung,
- ein Theorem, das die Dekomposition eines MIOCP in ein OCP und ein gemischt-ganzzahliges lineares Programm (MILP) rechtfertigt,
- einen struktur-ausnutzenden Branch and Bound Algorithmus, der die MILP Lösungszeit um Größenordnungen gegenüber kommerziellen Lösern verbessert,
- Algorithmen, um Unsicherheiten und Verzögerungseffekte behandeln zu können,
- eine benchmark-Bibliothek von MIOCPs mit einem neuen Klassifizierungsschema,
- eine detaillierte Untersuchung eines zeit-diskreten MIOCPs, das das psychologische Anwendungsfeld “Komplexes Problemlösen” Methoden der Optimierung zugänglich macht,
- notwendige Optimalitätsbedingungen für Versuchsplanungsprobleme (OEDPs),
- eine neue Formulierung von OEDPs, die zu besserer Konditionierung und Konvergenz führt,
- und Lösungen für mehr als ein Dutzend MIOCPs.

Abstract

The central topic of this thesis are mixed-integer nonlinear optimal control problems (MIOCPs). These are optimization problems that involve differential equations and control functions constrained by integrality requirements. Typical examples are the choice of gears in transport, logical decisions like “do I measure at a given point in time or not?”, or processes in chemical engineering involving on-off pumps. In practical applications often further characteristics evolve for this class of problems. Examples are combinatorial and logical constraints, uncertainties, delays, and multiple objectives.

In this thesis we present, compare, and discuss possible approaches to treat MIOCPs, and present novel algorithms that are reliable, fast, and can cope with many generalizations that are necessary from a practical point of view. These algorithms are backed up by novel insight from a theoretical perspective. Our algorithms are based on state-of-the-art direct methods on the lower level to solve relaxed problems, in particular all-at-once approaches like direct multiple shooting and direct collocation. This has already been proposed in previous work. However, the contributions in this thesis go further. The international state-of-the-art in mixed-integer optimal control has particularly been advanced by

- an algorithm for a generic problem class of MIOCPs, for which termination and convergence to an ε-certificated solution has been proven,
- Theorems that yield the best possible lower bound for MIOCPs,
- a Corollary that sharpens the estimate of the Hausdorff distance between reachability sets of a disjoint control set and its convex hull from the previously known $\sqrt{\Delta t}$ to linear grid size Δt, and generalizes to the nonlinear case,
- a structure-exploiting linear algebra approach that drastically reduces the computational extra effort due to the outer convexification,
- a theorem that justifies a practically important decomposition of MIOCPs into continuous nonlinear control problems and mixed-integer linear programs (MILPs), and hence the inclusion of combinatorial constraints into the proposed solution framework,
- a structure-exploiting Branch and Bound algorithm that reduces the MILP solution time by orders of magnitude compared to state-of-the-art commercial solvers,
- a framework to treat uncertainties and control delays,
- a benchmark library of MIOCPs with a detailed and novel classification scheme,
- the necessary conditions of optimality for Optimum Experimental Design (OED) problems,
- a novel formulation of OED problems that helps to overcome the intrinsic ill-conditioning,
- and more than a dozen MIOCPs that have been solved, most of them for the first time.

Danksagung

Für meine wissenschaftliche Sozialisation an dem “besonderen Ort IWR” danke ich meinen Lehrern und Mentoren Georg Bock, Johannes Schlöder, Gerd Reinelt, Willi Jäger und Rolf Rannacher, von denen ich so viel wichtiges lernen durfte.
Die Heidelberger Graduiertenschule für Mathematische und Computergestützte Methoden für die Wissenschaften am IWR der Universität Heidelberg hat mir die Arbeit der vergangenen Jahre in einem einmaligen wissenschaftlichen und menschlichen Umfeld ermöglicht. An diese Zeit, die wissenschaftliche Aufbruchstimmung und alle Freunde und Kollegen am IWR werde ich mich zeitlebens mit großer Freude (und Wehmut?) erinnern!
Insbesondere danke ich allen Mitgliedern der Arbeitsgruppen Bock, Körkel und Mombaur, den zentralen Mitarbeitern des IWR; und natürlich den Doktoranden meiner Nachwuchsgruppe: Holger Diedam, Michael Engelhart, Janick Frasch, Tony Huschto, Michael Jung, Florian Kehrle und Jonas Rauch.
Hervorheben möchte ich Andreas Potschka und Christian Kirches, mit denen ich immer wissenschaftliche Exzellenz, stimulierende Diskussionen, gemeinsame Konferenz-, Kontinent- und Kneipenbesuche, Quizfragen und menschliche Größe verbinden werde.

Wesentlich beigetragen zur Fertigstellung dieses Werkes haben natürlich die Coautoren der Veröffentlichungen, die in diese Arbeit eingeflossen sind. Dies betrifft Georg Bock, Moritz Diehl, Christian Kirches, Johannes Schlöder, Michael Jung, Tony Huschto, Gustav Feichtinger, Peter Kort, Richard Hartl, Andrea Seidl, Carola Barth, Holger Diedam, Michael Engelhart und Joachim Funke. Danke, es hat Spaß gemacht!

Ohne die Unterstützung von Freunden und Familie geht gar nichts, auch nicht wissenschaftlich. Ein herzliches *danke schön* daher an Carola, Daniel, Jens, Lars, Olaf, Uli, Rolf und Walter dafür, dass ich von Euch immer Verständnis, wahlweise aufmunternde oder kritische Worte (“wird das etwa wieder so eine unendliche Geschichte?”), Ablenkung und Liebe bekommen habe – und dass Ihr immer erkannt habt, was davon gerade gebraucht wurde!

Heidelberg, im August 2011 *Sebastian Sager*

„Zu mancher richtigen Entscheidung kam es nur,
weil der Weg zur falschen gerade nicht frei war.“
HANS KRAILSHEIMER

Contents

1 Introduction

The central topic of our work are so-called mixed-integer nonlinear optimal control problems (MIOCPs). These are optimization problems that involve differential equations and control functions that need to obey integrality requirements. Typical examples are the choice of gears in transport, logical decisions like “do I measure at a given point in time or not?”, or processes in chemical engineering involving on-off valves. In practical applications often further characteristics evolve for this class of problems. Examples are combinatorial and logical constraints, uncertainties, delays, and multiple objectives.

From an algorithmical point of view the integer requirement makes this problem class extremely challenging. Most approaches to standard optimal control problems use gradient information and hence assume a connected feasible set. Thus, there is a strong demand for efficient and stable algorithms for MIOCPs that are in particular able to cope with the mentioned extended problem formulations.

1.1 Goals of this thesis

Mathematical optimization is a discipline of high importance for science, industry, and economics, with much progress over the last decades. Unfortunately, this field has also seen a separation into at least three major subdisciplines: continuous, discrete, and stochastic optimization. In addition, communities focussing on optimal control, design problems, multiple objective functions, nonsmooth optimization, or global optimization have evolved. However, only very few mathematical optimization and control problems fall precisely into only one of these subdisciplines. This habilitation aims towards an integration of deterministic, gradient-based methods from these subdisciplines in the context of MIOCPs, and hence to an extension of numerically solvable optimal control problems.

Although the first MIOCPs were already solved in the early 1980s, the so-called *indirect methods* that were used at the time do not seem appropriate for generic large-scale optimal control problems with underlying nonlinear differential algebraic equation systems. It is also difficult to extend the algorithms to be able to cope with the mentioned complications. Instead *direct methods*, in particular simultaneous approaches, have become the methods of choice for most practical (non-integer) problems.

By direct method we refer to methods that *discretize first, then optimize* and work directly on the optimality conditions of the discretized control problem. In our terminology *indirect methods* for optimal control are methods that *optimize first, then discretize* by applying necessary conditions of optimality in function space, and then solving the control problem indirectly by solving the resulting boundary value problem numerically. This is not to be confused with the notion of direct and indirect methods in PDE constrained optimization. In that context direct methods sometimes refer to descent-based algorithms that directly use derivative information, while indirect methods refer to solving necessary first order conditions of optimality.

In our context, the direct methods discretize infinite-dimensional control functions with basis functions and corresponding finite-dimensional variables that enter into the optimization problem. We further distinguish between *sequential* and *simultaneous* direct methods, depending on whether they solve an outer optimization loop with sequential simulations, or whether they solve the simulation and optimization task simultaneously. The drawback of direct methods with integer control functions obviously is that they may lead to high-dimensional vectors of integer variables. For many practical applications a fine control discretization is required, however. Therefore, in general, techniques from mixed-integer nonlinear programming like Branch&Bound or Outer Approximation will work only on coarse grids, because of the exponentially growing complexity of the problem.

In this thesis we present, compare, and discuss possible approaches to treat MIOCPs, and present novel algorithms that are reliable, fast, and can cope with many generalizations that are necessary from a practical point of view. The algorithms are based on new theoretical results.

1.2 Outline of this thesis

Our goal to shed light on mixed-integer nonlinear optimal control problems from different points of view is challenging. Some of the results of this thesis could only be obtained in (interdisciplinary) cooperations. The thesis is structured in chapters, which are based on submitted or published papers. On the one hand, the chapters can hence be read independent from the rest of the thesis. On the other hand, they highlight different aspects of mixed-integer optimal control and contribute to an overall understanding.

Chapter 2 has a special role as the backbone of this thesis. It gives an overview and refers to details in the other chapters. This is also the chapter which differs most from the original publication, as it includes additional material. All chapters were rewritten to link to one another, to reduce redundancy, and to jointly yield a comprehensive work on a challenging optimization problem class. They start with a comprehensive summary of content and a specific, but non-redundant introduction. They vary from the mentioned publications, in particular with respect to bibliography, a unified format for algorithms, tables, and plots, and links to other chapters of this work.

In the following we give details and explain the contributions of the author of this thesis. Out of a total of eight papers that form the basis of this thesis, S. Sager authored three as single author and three as first, corresponding author. For the two remaining publications, PhD students from the junior research group headed by S. Sager were first authors.

Chapter 2 is based on the publication

[204] S. Sager. Reformulations and Algorithms for the Optimization of Switching Decisions in Nonlinear Optimal Control. *Journal of Process Control*, 2009, Vol. 19:1238–1247.

This chapter gives an overview of the field of mixed-integer nonlinear optimal control and refers to later chapters. S. Sager coauthored more papers on MIOC that do not enter as chapters in this thesis. References to and short summaries of this additional work, in particular [170, 157, 146, 147, 213, 141], are provided. Also a generic introduction to MIOC and to competitive approaches are given.
The chapter can be seen as the backbone of this thesis and should be read first.

Chapter 3 is based on the publication

[208] S. Sager, H.G. Bock, M. Diehl. The Integer Approximation Error in Mixed-Integer Optimal Control. *Mathematical Programming A*, 2011, DOI 10.1007/s10107-010-0405-3.

In this chapter the most important theoretical results are achieved. We show that an outer convexification of the integer control functions has very beneficial properties. We present a strategy that calculates integer controls in linear time, but still allows to guarantee upper bounds with respect to objective function and constraints that depend on the control discretization grid. The Sum Up Rounding strategy is a constructive part of the proof of these guarantees.
S. Sager wrote the publication and worked out the details of the mathematical proofs. The coauthors contributed in several discussions and reviewed the paper before submission.

Chapter 4 is based on the publication

[145] C. Kirches, H.G. Bock, J.P. Schlöder, S. Sager. Block Structured Quadratic Programming for the Direct Multiple Shooting Method for Optimal Control. *Optimization Methods and Software*, 2010, Vol. 26(2):239–257.

To overcome the drawback of additional control functions due to the outer convexification, we propose a tailored structure-exploitation for the solution of the underlying Karush-Kuhn-Tucker systems.
S. Sager as last author initiated this work with the basic idea to use an alternative to condensing for MIOCPs. C. Kirches worked out the details in his PhD thesis that S. Sager mentored within the *Heidelberg PhD student mentoring system*. Especially the numerical implementation has been done completely by C. Kirches. H.G. Bock and J.P. Schlöder worked on a predecessor of the method in a different context and supervised C. Kirches during his PhD thesis. The paper has been written jointly by C. Kirches and S. Sager.

Chapter 5 is based on the publication

[212] S. Sager, M. Jung, C. Kirches. Combinatorial Integral Approximation. *Mathematical Methods for Operations Research*, 2011, Vol. 73(3):363–380.

The Sum Up Rounding strategy is not able to take combinatorial constraints into account. We discuss a decomposition approach that couples the solution of a mixed-integer linear program to the approximation results that guarantee a-priori bounds on the loss of optimality. The decomposition of a mixed-integer nonlinear optimal control problem into one continuous nonlinear control problem and one mixed-integer linear program usually has dramatic advantages in terms of computational complexity.
S. Sager is the first author of this publication. He developed the main mathematical ideas and proofs. The coauthoring PhD students M. Jung and C. Kirches contributed with an efficient implementation of the Branch and Bound algorithm.

Chapter 6 is based on the publication

[128] T. Huschto, G. Feichtinger, P. Kort, R.F. Hartl, S. Sager, A. Seidl. Numerical Solution of a Conspicuous Consumption Model with Constant Control Delay. *Automatica*, 2011, DOI 10.1016/j.automatica.2011.06.004.

This chapter is special, because the control problem under consideration does not include integer controls. However, it could be easily extended by requiring that the prices need to be from a finite set, as is often the case for airlines, hotels, and so on. Hence, the control problem can be seen as the relaxed control problem, for which an integer solution can be determined in a second step. The optimal control problem is interesting, as it includes an uncertain scenario in which an expected value is optimized subject to worst case constraints. Additionally, it contains delays in the control functions.
S. Sager initiated the mathematical approach to this problem and supervised the progress of the paper, mainly due to the first author T. Huschto, who is a PhD student in the junior research group headed by S. Sager. The other coauthors contributed from the economical application point of view and by providing the challenging and interesting test problem. The mathematical parts have been written jointly by T. Huschto and S. Sager.

Chapter 7 is based on the publication

[205] S. Sager. A benchmark library of mixed-integer optimal control problems. *Proceedings MINLP09 IMA Minneapolis*, accepted.

We present a number of different MIOCPs that are also included in an open online benchmark library the author maintains. The control problems are classified according to different criteria and either solutions or references to solutions are given.

Chapter 8 is based on the publication

[207] S. Sager, C.M. Barth, H. Diedam, M. Engelhart, J. Funke. Optimization as an Analysis Tool for Human Complex Problem Solving. *SIAM Journal on Optimization*, accepted.

We discuss a particularly interesting example of a time-discrete optimal control problem with integer-valued decisions that stems from the analysis of human complex problem solving. Optimization is applied as an analysis tool and as a means to come up with an objective indicator function to quantitatively assess the performance of subjects in complex problem solving.
S. Sager initiated and wrote the paper as first author. H. Diedam and M. Engelhart worked as "Wissenschaftliche Hilfskräfte" in this project and provided a graphical user interface and automatic model generation tools. C.M. Barth and J. Funke are cooperation partners from the Institute of Psychology in Heidelberg and contributed expertise from the field of complex problem solving.

Chapter 9 is based on the publication

[206] S. Sager. Sampling Decisions in Optimum Experimental Design in the Light of Pontryagins Maximum Principle. *SIAM Journal on Control and Optimization*, submitted.

The optimal design of experiments can be seen as an important subclass of MIOC. We analyze sampling decisions via necessary conditions in function space to exemplarily show how problem classes can be further investigated by means of optimization theory.

1.3 Contributions to the international state-of-the-art

In this thesis we present, compare, and discuss possible approaches to treat MIOCPs, and present novel algorithms that are reliable, fast, and can cope with many generalizations that are necessary from a practical point of view. These algorithms are backed up by novel insight from a theoretical perspective.
Our favored algorithms use state-of-the-art direct methods on the lower level to solve relaxed problems, in particular all-at-once approaches like direct multiple shooting and direct collocation. This has already been proposed in previous work [203, 214]. However, the contributions in this thesis go further. The international state-of-the-art in mixed-integer optimal control has particularly been advanced by

- Algorithm 2.1 on page 25 for a generic problem class of MIOCPs, for which termination and convergence to an ε-certificated solution has been proven,
- Theorem 3.2.2 on page 36 that gives an upper bound on the distance between differential states as unique solutions of a system of ordinary differential equations with different control functions,
- Theorem 3.3.1 on page 38 that bounds the sliding mode norm between a relaxed and a Sum Up Rounding based integer control function,

- Theorem 3.4.2 on page 40 that extends this to the nonlinear case,
- Corollary 3.5.3 on page 44 that yields the best possible lower bound for MIOC problems,
- Corollary 3.5.7 on page 45 that sharpens the estimate of the Hausdorff distance between reachability sets of a disjoint control set and its convex hull from the previously known $\sqrt{\Delta t}$ to linear grid size Δt, and generalizes to the nonlinear case,
- a structure-exploiting linear algebra approach in Chapter 4 that drastically reduces the computational extra effort due to the outer convexification,
- Theorem 5.3.1 on page 78 that justifies a practically important decomposition of MIOCPs into continuous nonlinear control problems and mixed-integer linear programs (MILPs), and hence the inclusion of combinatorial constraints into the proposed solution framework,
- the structure-exploiting Branch and Bound Algorithm 5.1 on page 82 that reduces the MILP solution time by orders of magnitude compared to state-of-the-art commercial solvers,
- a framework to treat uncertainties and control delays in Chapter 6,
- a benchmark library of MIOCPs with a detailed and novel classification scheme,
- the extensive study of a time-discrete MIOCP which not merely describes an application, but opens up the whole new *application area* of complex problem solving for optimization,
- Corollary 9.5.1 on page 198 that states the necessary conditions of optimality for Optimum Experimental Design (OED) problems,
- the concept of the *Local and Global Information Gain* functions in OED and Lemmata 9.5.5, 9.5.6, 9.5.7, 9.5.8 that allow an a posteriori analysis of sampling decisions in OED,
- a novel formulation of OED problems that helps to overcome the intrinsic ill-conditioning and improves convergence properties,
- and more than a dozen of challenging MIOC applications that have been solved, most of them for the first time.

A short outlook to future work is given in Chapter 10.

2 Overview: Mixed–Integer Optimal Control

The contents of this chapter are based on the paper

[204] S. Sager. Reformulations and Algorithms for the Optimization of Switching Decisions in Nonlinear Optimal Control. *Journal of Process Control*, 2009, Vol. 19:1238–1247.

Chapter Summary. In model-based nonlinear optimal control switching decisions that can be optimized often play an important role. Prominent examples of such hybrid systems are gear switches for transport vehicles or on/off valves in chemical engineering. Optimization algorithms need to take the discrete nature of the variables that model these switching decisions into account. Unnecessarily, for many applications still an equidistant time discretization and either rounding or standard mixed–integer solvers are used. In this chapter we survey recent progress in theoretical bounds, reformulations, and algorithms for this problem class and show how process control can benefit from them. We propose a comprehensive algorithm based on the solution of a sequence of purely continuous problems and simulations.

2.1 Introduction

We give an introduction to the problem class we are interested in and review the literature. Note that more specific, complementary literature surveys are provided in the introductory sections of the next chapters.

Problem class. We are interested in model-based nonlinear optimal control including switching decisions that are to be optimized together with continuous controls. For the sake of readability, we proceed as follows. We focus on a specific case of a mixed–integer nonlinear optimal control problem (MIOCP) in ordinary differential equations (ODE) of the following form. Later on, in Section 2.8, we discuss extensions to include different objective functionals, multi-point constraints, algebraic variables, more general hybrid systems, and the like. For now, we want to minimize a Mayer term

$$\min_{x,u,v} \Phi(x(t_\mathrm{f})) \tag{2.1a}$$

over the differential states $x(\cdot)$ and the control functions $(u,v)(\cdot)$ subject to the n_x-dimensional ODE system

$$\dot{x}(t) \quad = \quad f(x(t),u(t),v(t)), \quad t \in [0,t_\mathrm{f}], \tag{2.1b}$$

with fixed initial values

$$x(0) \quad = \quad x_0, \tag{2.1c}$$

a feasible domain for the measurable controls

$$u(t) \quad \in \quad \mathscr{U}, \quad t \in [0, t_{\mathrm{f}}], \tag{2.1d}$$

and integrality of the control function $v(\cdot)$

$$v(t) \quad \in \quad \Omega := \{v^1, v^2, \ldots, v^{n_\omega}\}, \quad t \in [0, t_{\mathrm{f}}]. \tag{2.1e}$$

Additionally, nonlinear path and control constraints of the form

$$0 \quad \leq \quad c(x(t), u(t), v(t)), \quad t \in [0, t_{\mathrm{f}}] \tag{2.1f}$$

may need to be considered. Our main focus lies on the control function $v(\cdot)$ that needs to take a value v^i from a finite set $\Omega \subset \mathbb{R}^{n_{\mathrm{v}}}$ at all times. In the following all functions are assumed to be sufficiently often continuously differentiable, and $\|\cdot\|$ denotes the maximum norm $\|\cdot\|_\infty$.
We use the term *integer control* for (2.1e), while *binary control* refers to the special case

$$\omega(t) \quad \in \quad \{0, 1\}^{n_\omega}. \tag{2.2a}$$

We use the expression *relaxed*, whenever a restriction $v(\cdot) \in \Omega$ is relaxed to a larger subset of Ω, in particular to its convex hull. Of interest is a recently proposed outer convex relaxation [203] that we define as follows. For every element v^i of Ω a binary control function $\omega_i(\cdot)$ is introduced. The ODE (2.1b) can then be written as

$$\dot{x}(t) \quad = \quad \sum_{i=1}^{n_\omega} f(x(t), u(t), v^i) \; \omega_i(t), \quad t \in [0, t_{\mathrm{f}}]. \tag{2.2b}$$

If we impose the special ordered set type one condition

$$\sum_{i=1}^{n_\omega} \omega_i(t) \quad = \quad 1, \quad t \in [0, t_{\mathrm{f}}], \tag{2.2c}$$

there is obviously a bijection between every feasible integer function $v(\cdot) \in \Omega$ and an appropriately chosen binary function $\omega(\cdot) \in \{0,1\}^{n_\omega}$, compare Section 2.6.5. The relaxation of $\omega(t) \in \{0,1\}^{n_\omega}$ is given by $\omega(t) \in [0,1]^{n_\omega}$.
We use the expression *outer convexification* or *partial outer convexification* for the formulation (2.2b, 2.2c). Note that the resulting problem is only convex if $f(\cdot)$ is convex also in the arguments $x(\cdot)$ and $u(\cdot)$. Hence, the expression *convexification* only addresses the integer component.
Note that an equivalent formulation that is sometimes used, especially in the hybrid systems

community, is to write (2.1b, 2.1e) as

$$\dot{x}(t) \quad = \quad \tilde{f}_i(x(t),u(t)), \quad t \in [0,t_\mathrm{f}],\ 1 \le i \le n_\omega \tag{2.3}$$

as the choice of a model *i* to use.

As already done in [203], we refer to (2.1) as a *mixed-integer nonlinear optimal control problem*. Whereas the term ”optimal control” is commonly agreed to denote the optimization of processes that can be described by an underlying system of (partial) differential and algebraic equations with so–called control functions, there are several names for optimal control problems containing binary or integer variables in the literature. Sometimes it is referred to as *mixed–integer dynamic optimization* or *mixed-logic dynamic optimization* (MIDO or MLDO, see, e.g., [185]), sometimes as *hybrid optimal control* (e.g., [12], [232] or [56]), sometimes as a special case of *mixed–integer nonlinear program* (MINLP) optimization. As controls that take only values at their boundaries are known as *bang–bang controls* in the optimal control community, very often expressions containing bang–bang are used, too (e.g., [177]). Although there may be good reasons for each of these names, we use the expressions *mixed–integer (nonlinear) optimal control* (MIOC) and *mixed–integer (nonlinear) optimal control problem* (MIOCP). The reason is that the expression *mixed–integer* describes very well the nature of the variables involved and is well–established in the optimization community, while *optimal control* is used for the optimization of control functions and parameters in dynamic systems, whereas the term dynamic optimization might also refer to *parameter estimation*.

Typical examples for the problem class (2.1) are the choice of gears in transport, [237, 106, 213, 214, 147, 141], or processes involving on/off valves, [137, 54, 174]. Also Optimum Experimental Design can be interpreted as a special non–standard subclass of (2.1), compare Chapter 9. An open online benchmark library of MIOCPs is available, [202], see also Chapter 7.

MIOC approaches in the literature. MIOCPs include features related to different mathematical disciplines. Hence, it is not surprising that very different approaches have been proposed to analyze and solve them, ranging from theoretical discussions based on variations of the maximum principle to mixed-integer linear programming on piecewise linearly approximated discretizations of the control problem.

There are three generic approaches to solve model-based optimal control problems, compare [37, 36]. With its explicit approach, *Dynamic Programming* seems to be suited for a treatment of integer variables. Proofs of concept can be found, e.g., for applications in automatic truck control with gear switching, [125, 52]. However, the approach suffers in general from the so-called *curse of dimensionality*, an exponential increase in runtime when the state dimension increases. It is therefore not the method of choice for generic large–scale optimal control problems with underlying nonlinear differential (algebraic) equation systems.

The same holds true for *indirect methods*, also known as the *first optimize, then discretize* approach. A global minimum principle for disjoint control sets and (noncontinuous) ordinary differential equations (ODEs) has been formulated and solved numerically via the newly developed method of *Competing Hamiltonians* in the work of Bock and Longman in the early 1980s, [42, 43, 161]. To our knowledge this was the first time that a global minimum principle has been applied to solve a MIOCP. Theoretical results on hybrid systems have been determined, e.g., in [232, 223, 234]. Based on *hybrid maximum principles* or extensions of *Bellman's equation* approaches to treat switched systems have been proposed, e.g., in [224, 17, 5]. Unfortunately, indirect methods are not appropriate for generic large-scale optimal control problems with underlying nonlinear differential algebraic equation systems, and have problems to deal with path-constrained arcs. It is important to stress, however, that functional analysis yields important insight into solution structures, as exemplified in Chapter 9 for the special case of Optimum Experimental Design.

The third generic approach, *direct methods* and in particular *all–at–once approaches*, have become the methods of choice for most practical problems, see [37]. Even in the case of direct methods, there are multiple alternatives to proceed. Various approaches have been proposed to discretize the differential equations by means of shooting methods or collocation, e.g., [44, 34], to use global optimization methods by under- and overestimators, e.g., [85, 188, 63], to consider a static optimization problem instead of the transient behavior, e.g., [121], to approximate nonlinearities by piecewise-linear functions, e.g., [174], or by approximating the combinatorial decisions by continuous formulations, as in [55] for drinking water networks. Also problem (re)formulations play an important role, e.g., outer convexification of nonlinear MIOCPs [214], the modeling of MPECs and MPCCs [27, 26], or mixed-logic problem formulations leading to disjunctive programming, [195, 120, 184]. The approach to optimize the time-points for a given switching structure has been proposed by several authors, e.g., [139, 164, 199, 140, 106, 214]. It is well known that such a formulation introduces nonconvexities (see, e.g., an example in [203]). Hence, this approach should be combined with a proper initialization of the switching points and the calculation of an accurate lower bound, as pointed out in [214]. Another interesting technique is the method of Monotone Structural Evolution proposed in [233]. This method uses knowledge from the maximum principle to obtain criteria for an adaptive refinement of discretization structures, unfortunately at the price of having to solve the adjoint equations.

Interesting recent developments include problem-specific reformulations and decompositions, as in [55] for drinking water networks. The authors reformulate the MIOCP as a large-scale, structured nonlinear program (NLP) and solve a small scale integer program on a second level to approximate the calculated continuous aggregated output of all pumps in a water works.

Powerful commercial MILP solvers and advances in MINLP solvers, [1, 45], make the usage of general purpose MILP/MINLP solvers more and more attractive. The MIOCP may be discretized by a direct method and results in MILP, e.g., [174], or a MINLP, e.g., [105], with a finite number of mixed-integer variables. However, due to the high complexity of MINLPs and the increase in the number of integer variables, whenever the discretization grid is refined, this only works for small problems with limited time horizons, see [238] for a discussion.

Outline of the chapter. We start by giving an overview of Dynamic Programming, the indirect, and the direct approach to mixed-integer optimal control in Sections 2.2, 2.3 and 2.4, respectively. Tackling generic problems of the form (2.1) is difficult because of the combined nonlinear and discrete nature. Several algorithms are capable of producing a sub-optimal solution with the strong property of integer feasibility. For these approaches a bound on the performance loss is of utmost importance. This is addressed in Section 2.5. We first consider the case of linearly entering binary controls. In Sections 2.6.3 and 2.6.5 we obtain an equivalent formulation of this control-affine structure for (2.1). Sections 2.6 and 2.7 list different approaches to overcome the intrinsic problem of direct approaches with integer variables, cumulating into a comprehensive algorithm in Section 2.7.4. In Section 2.8 generalizations of the simple control problem (2.1) are discussed. We conclude with a summary in Section 2.9.

2.2 Dynamic programming

The methodology *dynamic programming* (DP) was developed in the fifties and sixties of the 20th century, most prominently by Richard Bellman [28]. It can be applied to discrete time and discrete control and state spaces, as well as to the continuous case, which leads to the so-called Hamilton-Jacobi-Bellman equation.

Dynamic programming is based on a backward recursion in time, starting at the end of the time horizon t_f with a *cost-to-go function* that is identical to the Mayer term contribution. Based on Bellman's *principle of optimality*, the optimization task is partitioned into N smaller time horizon optimization problems on intervals $[t_i, t_{i+1}]$.

When DP is applied to systems with continuous state spaces, such as (2.1), some approximations have to be made, usually by discretization. Generally, this discretization leads to an exponential growth of computational cost with respect to the dimension n_x of the state space, what Bellman called the "curse of dimensionality". It is the only, but unfortunately decisive drawback of DP and limits its practical applicability to systems with small n_x. However, DP can easily deal with all kinds of hybrid systems or non-differentiable dynamics, and it allows to treat stochastic optimal control problems and min-max games without much additional effort.

In the context of the MIOCP (2.1) the treatment of the integer requirements (2.1e) is easily achieved by means of an enumeration of all possible choices on each small time horizon problem. The partition of the optimization problem leads to a linear dependence of the runtime on the number of discretization points N, which makes DP the method of choice for small n_x and long time horizons.

Applications of dynamic programming for MIOC can be found, e.g., in [125, 52]. Both discuss the energy–optimal control of heavy duty trucks, based on GPS data and only one or two differential states, but possibly long prediction time horizons. An excellent textbook on discrete time optimal control and dynamic programming is [32].

2.3 Indirect approach to optimal control

The basic idea of indirect approaches is *optimize, then discretize*. In other words, first necessary conditions for optimality are applied to the optimization problem in function space, and in a second step the resulting boundary value problem is solved by an adequate discretization, such as multiple shooting. The necessary conditions for optimality are given by the famous Pontryagin's maximum principle. Assume we want to solve the following optimal control problem.

$$\begin{aligned} &\min_{x,w} \quad \Phi(x(t_\mathrm{f})) \\ &\text{subject to} \\ &\qquad \begin{aligned} \dot{x}(t) &= f(x(t),w(t)), & t \in [0,t_\mathrm{f}], \\ w(t) &\in \mathscr{W}, & t \in [0,t_\mathrm{f}], \\ x(0) &= x_0, \end{aligned} \end{aligned} \tag{2.4}$$

with an arbitrary, essentially bounded feasible set $\mathscr{W}$ for the control $w(\cdot)$. To state the maximum principle, we need the concept of the Hamiltonian.

Definition 2.3.1. (Hamiltonian, adjoint states)
The Hamiltonian of the corresponding optimal control problem (2.4) is given by

$$\mathscr{H}(x(t),w(t),\lambda(t)) \quad := \quad \lambda(t)^T\, f(x(t),w(t))$$

with variables $\lambda : [t_0,t_f] \to \mathbb{R}^{n_x}$ called adjoint variables. The end–point Lagrangian function ψ is defined as $\psi(x(t_f)) := \Phi(x(t_f))$.

The *maximum principle* in its basic form, also sometimes referred to as *minimum principle*, goes back to the early fifties and the works of Hestenes, Boltyanskii, Gamkrelidze, and of course Pontryagin. Precursors of the maximum principle as well as of the Bellman equation can already be found in Carathéodory's book of 1935, compare [189] for details.
The maximum principle states the existence of adjoint variables $\lambda^*(\cdot)$ that satisfy adjoint differential equations and transversality conditions. The optimal control $w^*(\cdot)$ is characterized as an implicit function of the states and the adjoint variables — a minimizer $w^*(\cdot)$ of problem (2.4) also minimizes the Hamiltonian subject to additional constraints.

Theorem 2.3.2. (Maximum principle)
Let problem (2.4) have a feasible optimal solution $w^*(\cdot)$ with a system response $x^*(\cdot)$. Then there exist adjoint variables $\lambda^*(\cdot)$ such that for $t \in [0,t_f]$ it holds almost everywhere

$$\dot{x}^*(t) \quad = \quad \mathscr{H}_\lambda(x^*(t),w^*(t),\lambda^*(t)) = f(x^*(t),w^*(t)), \tag{2.5a}$$

$$\dot{\lambda}^{*T}(t) \quad = \quad -\mathscr{H}_x(x^*(t),w^*(t),\lambda^*(t)), \tag{2.5b}$$

$$x^*(t_0) \quad = \quad x_0, \tag{2.5c}$$

$$\lambda^{*T}(t_f) = -\psi_x(x^*(t_f)), \tag{2.5d}$$

$$w^*(t) = \arg\min_{w\in\mathscr{W}} \mathscr{H}(x^*(t), w(t), \lambda^*(t)). \tag{2.5e}$$

For a proof of the maximum principle and further references see, e.g., [51, 192]. The interesting part about the *global* maximum principle is that the constraint $w(t) \in \mathscr{W}$ has been transferred towards the inner minimization problem (2.5e). This is done on purpose, so no assumptions need to be made on the feasible control domain $\mathscr{W}$. The global maximum principle also applies to nonconvex and disjoint sets $\mathscr{W}$. Hence, if we write $w(\cdot) = (u,v)(\cdot)$ and $\mathscr{W} = \mathscr{U} \times \Omega$, the maximum principle also covers problem (2.1) and the inner minimization problem (2.5e) reads

$$(u^*, v^*)(t) = \arg\min_{u\in\mathscr{U}, v\in\Omega} \mathscr{H}(x^*(t), u(t), v(t), \lambda^*(t)). \tag{2.6}$$

This is of course only possible if a *global* maximum principle is applied, as derived in [234, 108]. For a disjoint set Ω of moderate size, the pointwise minimization of (2.6) can be performed by enumeration between the n_ω different choices, implemented as switching functions that determine changes in the minimum. This approach, the *Competing Hamiltonians* approach, has been developed based on a global maximum principle and applied to the optimization of operation of subway trains with discrete acceleration stages in New York by Bock and Longman [43]. Their approach was even able to cope with non–continuous right hand sides.

Additional work has been done on the formulation of more general results, in particular the *hybrid maximum principle*, [232], and a *hybrid necessary principle*, [103]. Furthermore, the proofs were simplified by making a direct connection to the classical maximum principle, [76]. Based on hybrid maximum principles or extensions of Bellman's equation approaches to treat switched systems have been proposed that extend indirect methods or dynamic programming, e.g., in [224, 17]. While the maximum principle and knowledge about solution behavior keeps being important for analytical reasons, direct methods have become the methods of choice for larger control problems of practical relevance in ordinary differential equations. It is interesting to observe, however, that this might be different in the case of partial differential equations (PDE). In the PDE constraint optimization community the two approaches *first optimize, then discretize* and *first discretize, then optimize* are still competing. One reasons for this is probably the fact that adjoints need to be determined for an efficient calculation of derivatives, which involves a second discretization grid for the backward solve. In higher dimensions the question which grid to choose becomes more important and favors an indirect approach. Also there is a tendency to treat spatial phenomena like shock waves rather in function space than by discretization, e.g., [59].

We revise the maximum principle in Section 9.2 to use it to analyze sampling functions in the Optimal Design of Experiments. New results from [107] for a global maximum principle also for the DAE case have been applied to a benchmark problem in Section 7.3.3 for illustration.

2.4 Direct approach to optimal control

The main idea of direct approaches is *first discretize, then optimize*. The control problem in a function space is discretized by means of parametric functions with local support, and then the resulting nonlinear program (NLP) in finitely many optimization variables is solved. There are basically three different approaches: *single shooting*, *Bock's direct multiple shooting*, and *collocation*. Details on these methods and how they relate to one another can be found, e.g., in [37, 33, 36]. Details on the direct multiple shooting method are given in Section 4.2.

There are important differences between the approaches, mainly in the parameterization of the underlying differential equations and the respective connections to the optimization algorithm by means of derivative information. There are also good reasons why collocation and multiple shooting, both dating back to the early eighties, [40, 44, 34], are most often superior to the single shooting approach. However, all further algorithms and reformulations yet to be presented can be equally applied to any one of the three.

We restrict ourselves to a short presentation of the discretization of the respective functions in time, common to all three methods. Generally, any appropriate set of basis functions will do, e.g., splines or piecewise linear functions, if they can be described by means of finitely many values that become the optimization variables. For the following it is sufficient to assume a piecewise constant discretization of the form

$$\hat{u}(t,q_i^{\mathrm{u}}) \ := \ q_i^{\mathrm{u}}, \ \ \hat{v}(t,q_i^{\mathrm{v}}) := q_i^{\mathrm{v}}, \ \ t \in [t_i,t_{i+1}] \tag{2.7}$$

on an appropriate time grid $0 = t_0 < t_1 < \ldots < t_m = t_{\mathrm{f}}$ and with control values $q_i^{\mathrm{u}} \in \mathbb{R}^{n_{\mathrm{u}}}$ and $q_i^{\mathrm{v}} \in \mathbb{R}^{n_{\mathrm{v}}}$. The control space is hence reduced to functions that can be written as in (2.7), depending on finitely many controls $(q^{\mathrm{u}}, q^{\mathrm{v}})$.

If present, also the path constraints $c(\cdot) \geq 0 \, \forall \, t \in [0,t_{\mathrm{f}}]$, compare Section 2.8, are discretized on an appropriately chosen grid. From this discretization and the (algorithm specific) parameterization of the differential states results a highly structured NLP that is usually solved by either an interior point or an active set based algorithm. For details on an efficient implementation and further references see, e.g., [169, 35, 36, 143].

If switching decisions or disjoint feasible sets are present as in (2.1e), the discretization (2.7) leads to control variables that inherit this integrality condition. For a piecewise constant discretization $q_i^{\mathrm{v}} \in \Omega$ needs to hold for all $0 \leq i < m$. Formally a mixed–integer nonlinear program is obtained.

2.5 Theory for control-affine systems

Most of the algorithms that have been applied to solve problem (2.1) cannot provide a rigorous lower bound on the optimal solution value. Even if global MINLP methods are applied, one does not know how good the solution really is, as the underlying control discretization grid might be too coarse in some regions or simply not hit the optimal switching points. Only recently the connection between rigorous bounds on the optimal integer solution value and results of relaxed, continuous control problems has been made, [203, 214]. Let us, for now, consider a *binary-control-affine problem* of the form

$$\begin{array}{llll} \min\limits_{x,u,\omega} & \Phi(x(t_\mathrm{f})) & & \\ \text{subject to} & & & \\ \quad \dot{x}(t) & = & \tilde{f}(x(t),u(t)) \cdot \omega(t), & t \in [0,t_\mathrm{f}], \\ \quad u(t) & \in & \mathscr{U}, & t \in [0,t_\mathrm{f}], \\ \quad \omega(t) & \in & \{0,1\}^{n_\omega}, & t \in [0,t_\mathrm{f}], \\ \quad C(\omega(t)) & = & 0, & t \in [0,t_\mathrm{f}], \\ \quad x(0) & = & x_0, & \end{array} \tag{2.8}$$

with $\tilde{f} : \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}^{n_\mathrm{u}} \to \mathbb{R}^{n_\mathrm{x} \times n_\omega}$ and $C : \mathbb{R}^{n_\omega} \to \mathbb{R}^{n_C}$ an arbitrary constraint on the binary control. We see later how the special case (2.8) relates to the more general problem (2.1) that we are really interested in. One of the observations in [203, 214] was that the optimal solution of the relaxation of control problem (2.8) yields the exact lower bound for (2.1), i.e., the value that can either be reached or be approximated arbitrarily close by an integer control. However, the proof used arguments from functional analysis and hence this result does not apply to a finite number of switches.

Now we extend this statement: for any $\delta > 0$ it holds that if the control discretization grid is chosen fine enough, then there exists a binary solution with a *finite number* of switches that yields an objective value closer than δ to the one of the relaxed problem. The basis for this is

Theorem 2.5.1. *Let $x(\cdot)$ and $y(\cdot)$ be solutions of the initial value problems*

$$\dot{x}(t) = A(t,x(t)) \cdot \alpha(t), \quad x(0) = x_0, \tag{2.9a}$$

$$\dot{y}(t) = A(t,y(t)) \cdot \omega(t), \quad y(0) = y_0, \tag{2.9b}$$

with $t \in [0,t_f]$, for given measurable functions $\alpha, \omega : [0,t_f] \to [0,1]^{n_\omega}$ and a differentiable $A : \mathbb{R}^{n_x+1} \mapsto \mathbb{R}^{n_x \times n_\omega}$. If positive numbers $C, L \in \mathbb{R}^+$ exist such that for $t \in [0,t_f]$ almost everywhere it holds that

$$\left\| \frac{\mathrm{d}}{\mathrm{d}t} A(t,x(t)) \right\| \leq C, \tag{2.9c}$$

$$\| A(t,y(t)) - A(t,x(t)) \| \leq L \| y(t) - x(t) \|, \tag{2.9d}$$

and $A(\cdot,x(\cdot))$ is essentially bounded by $M \in \mathbb{R}^+$ on $[0,t_f]$, and it exists $\varepsilon \in \mathbb{R}^+$ such that for all

$t \in [0, t_f]$

$$\left\| \int_0^t \alpha(\tau) - \omega(\tau) \, \mathrm{d}\tau \right\| \quad \leq \quad \varepsilon \tag{2.9e}$$

then it also holds

$$\| \, y(t) - x(t) \, \| \leq (\| \, x_0 - y_0 \, \| + (M + Ct)\varepsilon) \, e^{Lt} \tag{2.9f}$$

for all $t \in [0, t_f]$.

A proof for this theorem is provided in Chapter 3, together with a formal connection to the problem (2.8). The basic idea is to assume that we have found the feasible and optimal trajectory (x^*, u^*, α^*) of the relaxation of problem (2.8). We fix the continuous control functions $u^*(\cdot)$ and write the right hand side $f(x(\cdot), u(\cdot))\omega(\cdot)$ as a function $A(y(\cdot); u^*(\cdot))\omega(\cdot)$ of $y(\cdot)$ and $\omega(\cdot)$ only. The ODE in (2.8) is then in the form of (2.9b). We see in Section 2.7.3 a constructive way to determine a binary control $\omega(\cdot)$ from $\alpha^*(\cdot)$ in a way that ε is a mere multiple of the control discretization grid size, and can hence be made arbitrarily small. This is done for two cases of interest: the one when there are no constraints, $C(\omega) = 0$, and when a special ordered set property $C(\omega) = \sum \omega_i - 1$ has to hold that stems from an equivalent reformulation of the general nonlinear case.

Theorem 2.5.1 now helps to estimate the performance loss between the optimal relaxed control $\alpha^*(\cdot)$ and the binary control $\omega(\cdot)$. The difference between the differential states is determined from (2.9f), if (2.9e) holds. The Mayer function $\Phi(\cdot)$ is assumed to be differentiable, hence continuous. Therefore the difference between the objective function values of the original, binary control problem (2.8) and of its relaxation are bounded by a constant times ε.

The most interesting assumption of Theorem 2.5.1 is (2.9e). At first sight the condition is somewhat unusual, as one might expect an L^∞ norm,

$$\int_0^{t_\mathrm{f}} \| \, \omega(\tau) - \alpha(\tau) \, \| \, \mathrm{d}\tau \quad \leq \quad \varepsilon. \tag{2.10}$$

This condition is far too strong, however. While one direction is obvious,

$$\left\| \int_0^t \omega(\tau) - \alpha(\tau) \, \mathrm{d}\tau \right\| \quad \leq \quad \int_0^t \| \, \omega(\tau) - \alpha(\tau) \, \| \, \mathrm{d}\tau \leq \int_0^{t_\mathrm{f}} \| \, \omega(\tau) - \alpha(\tau) \, \| \, \mathrm{d}\tau$$

one can construct an example for which the gap between the two expressions (2.9e) and (2.10) becomes as large as it can get. Assume an equidistant time grid $0 = t_0 < t_1 < \cdots < t_m = t_\mathrm{f}$, with $t_{i+1} - t_i = \frac{t_\mathrm{f}}{m}$. Define

$$\alpha(\tau) \quad := \quad \frac{1}{2}, \quad \omega(\tau) := \begin{cases} 1 & \tau \in [t_i, t_{i+1}], i \text{ even} \\ 0 & \tau \in [t_i, t_{i+1}], i \text{ odd} \end{cases}$$

We obtain

$$\int_0^{t_\mathrm{f}} \| \, \omega(\tau) - \alpha(\tau) \, \| \, \mathrm{d}\tau \;=\; \frac{t_\mathrm{f}}{2} \quad \text{and} \quad \left\| \int_0^{t} \omega(\tau) - \alpha(\tau) \, \mathrm{d}\tau \right\| \leq \frac{t_\mathrm{f}}{m},$$

where the second term vanishes for $m \to \infty$ for all $t \in [0, t_\mathrm{f}]$. Again, the details are given in Chapter 3.

2.6 Reformulations

In the previous sections we developed both general methodology and theory that guarantees performance loss bounds for binary-control-affine systems. In this section we survey different reformulations. The first two, a switching time optimization approach in 2.6.1 and penalization strategies in 2.6.2, aim at producing sub-optimal integer feasible solutions. In subsections 2.6.3 and 2.6.5 the target is to reformulate the nonlinear problem equivalently to obtain a binary-control-affine system.

2.6.1 Switching time optimization

One possibility to solve problem (2.1) is motivated by the idea to optimize the switching times directly, and to take the values of the integer controls fixed on given intervals. This concept is old and well known from a) indirect approaches, where switching functions (derivatives of the Hamiltonian with respect to the controls) are used to determine switching times, from b) hybrid systems, where switching functions determine phase transitions, and from c) multi-stage processes, such as batch processes in chemical engineering, consisting of several phases with open duration, e.g., [167].

The main idea consists of a reformulation. The control $v(t)$ is fixed to a value $v^{i_j} \in \Omega$ on each interval $[t_j, t_j + 1]$, with an (a priori) fixed order of the v^{i_j}. The control problem to be solved reads

$$\begin{aligned}
&\min_{x,u,t_j} \quad \Phi(x(t_\mathrm{f})) \\
&\text{subject to} \\
&\qquad \begin{array}{rcll}
\dot{x}(t) &=& f(x(t), u(t), v^{i_j}), & t \in [t_j, t_{j+1}], \\
u(t) &\in& \mathscr{U}, & t \in [0, t_\mathrm{f}], \\
x(0) &=& x_0. &
\end{array}
\end{aligned} \tag{2.11}$$

In practice one does not optimize the switching points t_j directly, but the scaled vector h of model stage lengths $h_j := t_{j+1} - t_j$, see [167, 106]. This approach is visualized in Figure 2.1 for a one-dimensional binary control. Although the algorithm looks very promising at first sight, it has some severe disadvantages. First, a nonregular situation that may occur when stage lengths are reduced to zero. Assume the length of an intermediate stage, say h_2, has been reduced to

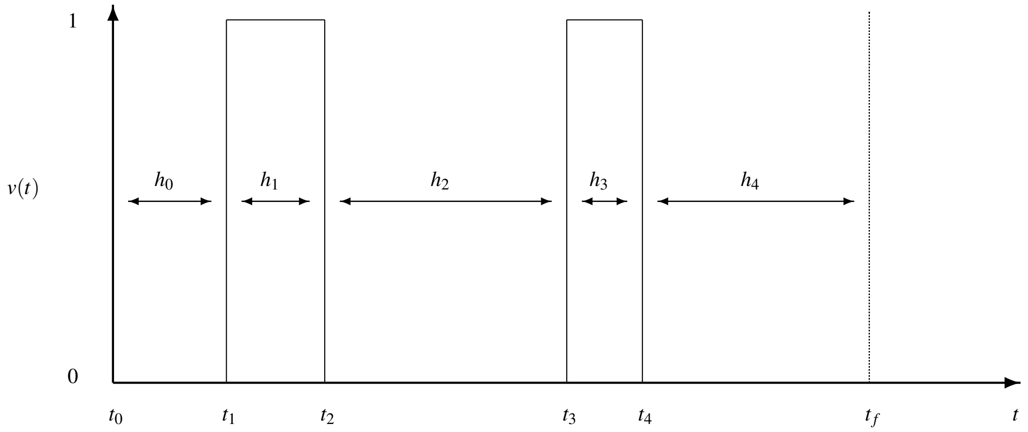


Figure 2.1: Switching time optimization, one–dimensional example.

zero by the optimizer. Then the sensitivity of the optimal control problem with respect to h_1 and h_3 is given by the value of their sum $h_1 + h_3$ only. Thus special care has to be taken to treat the case where stage lengths diminish during the optimization procedure. In [139], [140] and [176] an algorithm to eliminate such stages is proposed. This is possible, still the stage cannot be reinserted, as the time when to insert it is undetermined. Also, in the presence of explicit combinatorial constraints in the mixed path and control constraints (2.1f) a feasible initialization is not straightforward.

The second drawback is that the number of switches is typically not known, left alone the precise switching structure. Some authors propose to iterate on the maximum number of intervals until there is no further decrease in the objective function of the corresponding optimal solution, [139, 140, 176]. But it should be stressed that this can only be applied to more complex systems, if very good initial values for the location of all switching points are available. This is closely connected to the third and most important drawback of the switching time approach. The reformulation yields additional nonconvexities in the optimization space. Even if the optimization problem is convex in the optimization variables resulting from a constant discretization of the control function $v(\cdot)$, the reformulated problem may be nonconvex, compare[203].

The mentioned drawbacks of the switching time optimization approach can be overcome, though, if it is combined with a bunch of other concepts, compare [203, 106]. This includes good initial values, a strategy to deal with diminishing stage lengths and a direct all–at–once approach like direct multiple shooting that helps when dealing with nonconvexities as discussed in [203]. Also, making use of the theoretical results of Section 2.5, termination criteria for an iterative refinement of the switching structure need to be determined.

2.6.2 Reformulations to avoid integrality

The first idea to replace a binary variable $y \in \{0,1\}$ by a continuous variable $y \in [0,1]$ is to add the constraint $y\,(1-y)=0$ to the problem formulation. Unfortunately this equality constraint is nonconvex with a disjoint feasible set and optimization solvers perform badly on such equations, as the necessary constraint qualification is violated.

Penalization strategies have the same aim as switching time optimization: working with continuous variables only, but obtaining an integer feasible solution. To do so for, say, the binary case $\omega(t) \in \{0,1\}^{n_\omega}$, we first relax towards $\omega(t) \in [0,1]^{n_\omega}$ for all $t \in [0,t_\mathrm{f}]$. To enforce a binary solution, we have two possibilities. One is to add a concave penalty function, e.g.,

$$\min_{x,u,\omega} \Phi(x(t_\mathrm{f})) + \sum_{i=1}^{n_\omega} \varepsilon_i \int_{t_0}^{t_f} (1-\omega_i(t))\,\omega_i(t)\,\mathrm{d}t$$

for $\varepsilon_i \geq 0$. The other one would be to impose additional constraints,

$$(1-\omega_i(t))\,\omega_i(t) \leq \varepsilon_i \quad \forall\, t \in [0,t_\mathrm{f}].$$

An extension is to use a penalty term homotopy, by solving a series of continuous optimal control problems with relaxed $\omega(\cdot)$. One initializes problem P^{k+1} with the solution of P^k and raises ε_i^k until all $\omega_i(t)$ are 0 or 1.

Both approaches depend very much on the choice of ε and impose bad numerical behavior by either making the objective nonconvex, or splitting the feasible region into disjoint parts. Either approach may work well for special cases, but is generally not to be recommended. Details and a numerical case study can be found, e.g., in [203].

2.6.3 Reformulations to avoid nonlinearity

Another target for reformulations are the nonlinearities. We consider general linear approximations and products containing binary variables.

The basic idea to use underestimating and overestimating linear functions is best exemplified by replacing a bilinear term xy by a new variable z and additional constraints. This reformulation was proposed by [178]. For the new variable z we obtain the linear constraints

$$\begin{array}{rcccl} y^{\mathrm{lo}}x + x^{\mathrm{lo}}y - x^{\mathrm{lo}}y^{\mathrm{lo}} & \leq & z & \leq & y^{\mathrm{lo}}x + x^{\mathrm{up}}y - x^{\mathrm{up}}y^{\mathrm{lo}}, \\ y^{\mathrm{up}}x + x^{\mathrm{up}}y - x^{\mathrm{up}}y^{\mathrm{up}} & \leq & z & \leq & y^{\mathrm{up}}x + x^{\mathrm{lo}}y - x^{\mathrm{lo}}y^{\mathrm{up}}, \end{array} \tag{2.12}$$

for given bounds on x and y, i.e., $x \in [x^{\mathrm{lo}}, x^{\mathrm{up}}]$ and $y \in [y^{\mathrm{lo}}, y^{\mathrm{up}}]$. The inequalities follow from $(x-x^{\mathrm{lo}})(y-y^{\mathrm{lo}}) \geq 0$ and three similar equations. The snag is of course that very tight bounds are needed for a successful optimization, which is not the case in the presence of strong nonlinearities. See [235] or [92] for references on general under– and overestimation of functions.

When binary variables enter in a nonlinear way into the right hand side function $f(\cdot)$, often simplifications are possible. All higher exponents can be skipped, as it holds $\omega_i(t) \cdot \omega_i(t) = \omega_i(t)$ for $\omega_i(t) \in \{0,1\}$. Also for mixed products of binary variables a reduction of nonlinearity is possible. We introduce an additional variable, e.g., for $\omega_i(t) \cdot \omega_j(t)$:

$$\omega_{ij}(t) \quad := \quad \begin{cases} 1 & \text{if } \omega_i(t) = \omega_j(t) = 1 \\ 0 & \text{else} \end{cases}$$

The new binary variables can be incorporated into the optimization problem by adding the constraints

$$\omega_{ij}(t) \leq \omega_i(t), \quad \omega_{ij}(t) \leq \omega_j(t), \quad \omega_i(t) + \omega_j(t) \leq 1 + \omega_{ij}(t).$$

2.6.4 Reformulations to decouple integrality and nonlinearity

A clever problem-specific reformulation is proposed in [54, 55]. For the optimal operation of a water network the authors propose to *decompose the problem* in the sense that a pure NLP is solved for the overall network with a (continuous) aggregated output of the discrete-valued pumps in each waterworks. In a second step this optimal continuous output is approximated by solving a small-scale integer program for every waterworks in the system.

Based on the same principle of decoupling nonlinearity and integrality, a more generic approach is presented in Chapter 5. It allows to approximate the solution of the nonlinear MIOCP by means of the solution of a continuous OCP and a mixed–integer linear program, with possibly huge computational savings. Yet, it is possible to bound the gap between the exact and the approximative solution.

2.6.5 Outer convexification

We saw in Section 2.5 that for binary-control-affine models we get an estimate of the performance loss of any feasible binary solution by solving a relaxed problem. If nonlinearities with respect to the $v(\cdot)$ in (2.1) occur, they can sometimes be transformed as in Section 2.6.3. If this is not the case, a partial *outer convexification* with respect to the integer functions has been proposed in [203, 214]. Consider the following reformulation of problem (2.1),

$$\begin{aligned} &\min_{x,u,\omega} \quad \Phi(x(t_\mathrm{f})) \\ &\text{subject to} \\ &\qquad \begin{array}{lcll} \dot{x}(t) & = & \sum_{i=1}^{n_\omega} f(x(t),u(t),v^i) \cdot \omega_i(t), & t \in [0,t_\mathrm{f}], \\ u(t) & \in & \mathscr{U}, & t \in [0,t_\mathrm{f}], \\ \omega(t) & \in & \{0,1\}^{n_\omega}, & t \in [0,t_\mathrm{f}], \\ \sum_{i=1}^{n_\omega} \omega_i(t) & = & 1, & t \in [0,t_\mathrm{f}], \\ x(0) & = & x_0, & \end{array} \end{aligned} \tag{2.13}$$

with fixed $v^i \in \Omega, 1 \le i \le n_\omega$. Problem (2.13) has two important properties: first, there is a bijection between solutions of (2.13) and of (2.1), hence any optimal solution is also optimal for the other problem. Second, it fits into the context of Section 2.5, as the binary controls enter linearly. In fact, as all v^i are fixed, problem (2.13) can be written in the form (2.8) with a matrix $\tilde{f}(\cdot)$ that contains $f(x(t),u(t),v^i)$ as its n_ω columns. There is one important modification: the additional linear constraint $0 = C(\omega) = \sum \omega_i - 1$ to ensure the controls form a special ordered set at each instant in time. This constraint needs to be taken into account whenever a binary solution is constructed from a relaxed one, compare Section 2.7.3.

Problem (2.13) yields a tight relaxation of the original problem (2.1). This reformulation comes at the price of additional control functions, as $v(\cdot)$ is replaced by n_ω controls $\omega_i(\cdot)$ (one less, if the linear equality constraint is used to eliminate one of them).

Note that depending on $f(\cdot)$, integer controls may decouple, leading to a reduced number n_ω. Assume we have

$$\begin{array}{rcl} \dot{x}(t) & = & g(\cdot, v_1(t)) + h(\cdot, v_2(t)), \\ v_1(t) & \in & \Omega_1, \quad v_2(t) \in \Omega_2. \end{array}$$

Then an equivalent reformulation is given by

$$\begin{array}{rcl} \dot{x}(t) & = & \left(\sum_{i=1}^{n_{\omega_1}} g(\cdot, v_1^i)\ \omega_{1,i}(t) \right) + \left(\sum_{i=1}^{n_{\omega_2}} h(\cdot, v_2^i)\ \omega_{2,i}(t) \right), \\ \sum_{i=1}^{n_{\omega_1}} \omega_{1,i}(t) & = & 1, \quad t \in [t_0, t_f], \\ \sum_{i=1}^{n_{\omega_2}} \omega_{2,i}(t) & = & 1, \quad t \in [t_0, t_f], \\ \omega_1 & \in & \{0,1\}^{n_{\omega_1}}, \quad \omega_2 \in \{0,1\}^{n_{\omega_2}}, \end{array}$$

leading to $n_\omega = n_{\omega_1} + n_{\omega_2}$ controls instead of $n_\omega = n_{\omega_1} n_{\omega_2}$. The proof is straightforward. As in most practical applications the binary control functions enter linearly (such as valves that indicate whether a certain term is present or not), or n_ω increases linearly with the number of choices (e.g., the gears), or integer controls decouple, the drawback of an increased number n_ω of control functions is clearly out-weighted by the advantages.

2.7 Algorithms

We present algorithms to solve problems of the form (2.1) and (2.13).

2.7.1 Rounding

One idea to solve problem (2.13) is to solve its relaxation (hence, $\omega(t) \in [0,1]^{n_\omega}$) and to round all $\omega_i(t)$ (alternatively their finite-dimensional parameterization q_i^ω) to the nearest binary value. In general mixed–integer programming this approach is not a good idea, rounded solutions are often very poor solutions or even infeasible. However, for control problems the optimal solution in function space is often of bang-bang type, i.e., the optimal control only takes values at its bounds. For these cases rounding performs well, if it is combined with an adaptive control discretization grid.

As follows from the results of Sections 2.5 and 2.6.5, we do have the exact lower bound from the solution of the relaxation of problem (2.13) and can hence estimate the performance loss associated with rounding. This is an important difference and advantage compared to general integer programming.

2.7.2 MI(N)LP algorithms

In the last 20 years important contributions in the field of algorithms for mixed–integer nonlinear programs (MINLPs) have been achieved. Of course both the classical algorithms Branch&Bound, Outer Approximation, and Bender's decomposition as newer developments including cutting planes and treatment of nonconvexities can be applied to the MINLP that stems from a discretization with a direct approach of problem (2.13).

If switching decisions or disjoint feasible sets are present as in (2.1e), discretization (2.7) leads to control variables that inherit this integrality condition. For a piecewise constant discretization $q_i^\omega \in \{0,1\}$ needs to hold for all i. The drawback of direct methods with integer control functions is obviously that they lead to high–dimensional vectors of binary/integer variables.

For many practical applications a fine control discretization is required. Therefore MINLP techniques works only on limited and small time horizons because of the exponentially growing complexity of the problem, [238].

We recommend to use global MINLP algorithms only in two cases: first, when the control discretization grid is fixed and a global solution on this grid is of importance, and second, in an outer loop, when both integer control functions and non-time-dependent combinatorial decisions have to be made. In this case the problem can be decoupled, treating combinatorial decisions in an outer loop, and working with a relaxation of the integer control functions in the inner loop, compare Section 2.8.

2.7.3 Sum up rounding

A novel rounding strategy that is especially tailored to minimize expression (2.9e) on page 16 has first been proposed in the context of mixed-integer optimal control in [203]. We consider a

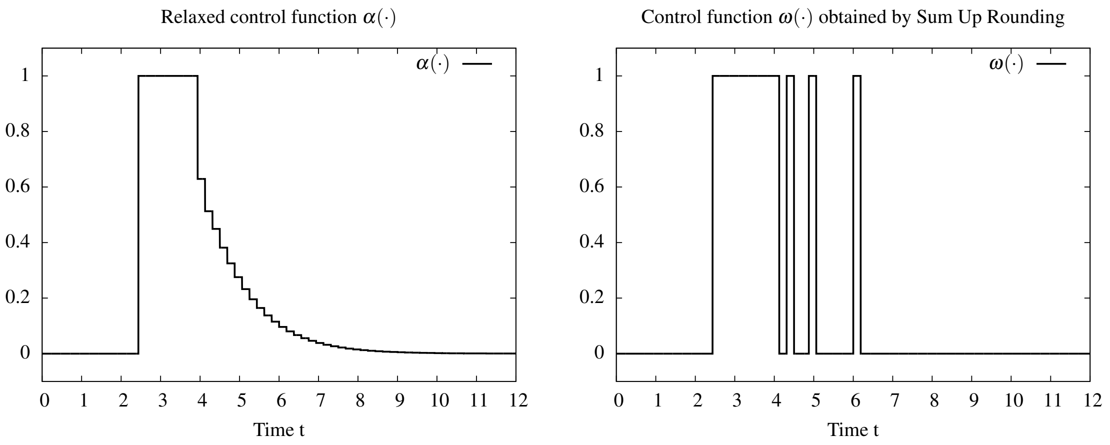


Figure 2.2: Relaxed and Sum Up Rounding binary controls for $m = 64$ time intervals.

piecewise constant control function

$$\alpha_j(t) = q^{\alpha}_{j,i} \in [0,1], \quad t \in [t_i, t_{i+1}] \tag{2.14}$$

with $j = 1 \ldots n_\omega$ and $i = 0 \ldots m-1$ on a fixed time grid $0 = t_0 < t_1 < \cdots < t_m = t_f$, as introduced in Section 2.4. Such a function could be the result of an optimization with a direct approach that discretizes the control functions by piecewise constant functions. We write $\Delta t_i := t_{i+1} - t_i$ and Δt for the maximum distance between two time points,

$$\Delta t := \max_{i=0\ldots m-1} \Delta t_i = \max_{i=0\ldots m-1} \{t_{i+1} - t_i\}. \tag{2.15}$$

Let then a function $\omega(\cdot) : [0, t_f] \mapsto \{0,1\}^{n_\omega}$ be defined by

$$\omega_j(t) = p_{j,i}, \quad t \in [t_i, t_{i+1}] \tag{2.16}$$

where the $p_{j,i}$ are binary values given by

$$p_{j,i} \quad = \quad \begin{cases} 1 & \text{if } \sum_{k=0}^{i} q^{\alpha}_{j,k} \Delta t_k - \sum_{k=0}^{i-1} p_{j,k} \Delta t_k \geq 0.5 \Delta t_i \\ 0 & \text{else} \end{cases} . \tag{2.17}$$

See Figure 2.2 for an example.

We have the following estimate on the integral over the difference between the control functions $\alpha(\cdot)$ and $\omega(\cdot)$.

Theorem 2.7.1. (Sum Up Rounding Integral Deviation)
Let the functions $\alpha : [0, t_f] \mapsto [0,1]^{n_\omega}$ and $\omega : [0, t_f] \mapsto \{0,1\}^{n_\omega}$ be given by (2.14) and (2.16,

2.17), respectively. Then it holds

$$\left\| \int_0^t \omega(\tau) - \alpha(\tau)\, \mathrm{d}\tau \right\| \leq 0.5\, \Delta t.$$

For a proof see Chapter 3. In combination with Theorem 2.5.1 this theorem allows us to relate the difference between differential states corresponding to any (relaxed) solution and a specific integer solution obtained by Sum Up Rounding to the size of the control discretization grid.
Note that the Sum Up Rounding strategy (2.17) does not work for problems with the additional special ordered set property $\sum \omega_i = 1$ as in (2.13), as can be seen by the easy example of two functions that have the constant value $\alpha_1(t) = \alpha_2(t) = 0.5$. If we define $p_{j,i}$ to be

$$\hat{p}_{j,i} = \sum_{k=0}^{i} q^{\alpha}_{j,k} \Delta t_k - \sum_{k=0}^{i-1} p_{j,k} \Delta t_k \tag{2.18a}$$

$$p_{j,i} = \begin{cases} 1 & \text{if } \hat{p}_{j,i} \geq \hat{p}_{k,i} \;\forall\; k \neq j \text{ and } j < k \;\forall\; k : \hat{p}_{j,i} = \hat{p}_{k,i} \\ 0 & \text{else} \end{cases} \tag{2.18b}$$

a similar result to Theorem 2.7.1 holds, compare Chapter 3.

2.7.4 MS MINTOC

We propose to use the following algorithm for the solution of problem (2.1). We denote the control discretization grid in iteration k with $\mathscr{G}^k$, and the optimal trajectory of (2.8) with $\mathscr{T}^k = (x^k(\cdot), u^k(\cdot), \alpha^k(\cdot))$. For the sake of notational simplicity we use $u^k(\cdot)$ and $\alpha^k(\cdot)$ and not the discretization variables $q^{\mathrm{u}}, q^{\alpha}$.
As for all algorithms we have to ask whether it is well-posed and terminates in a finite number of steps. The answer is given by the following

Theorem 2.7.2. (Well-posedness of MS MINTOC)
If the assumptions

1. *On all grids $\mathscr{G}^k$ an optimal solution to the relaxed problem* (2.8) *is found in a finite number of operations.*
2. *Bisection is used for the refinement of $\mathscr{G}^k$.*
3. *After a finite number k^{max} of refinements we freeze the optimal relaxed solution, $\mathscr{T}^k = \mathscr{T}^{k^{max}}$ and $\Phi^{REL}_{\mathscr{G}^k} = \Phi^{REL}_{\mathscr{G}^{k^{max}}} \;\forall\; k > k^{max}$.*

hold, then Algorithm 2.1 terminates in a finite number of steps with a feasible binary solution, for which $\Phi^* < \Phi^{REL}_{\mathscr{G}^k} + TOL$ holds.

Proof. By Assumption 1 all optimal control problems are solved in finite time, and so is the simulation in 3.(c). If the algorithm stops in 3.(b), a binary solution with $\Phi^* = \Phi^{\mathrm{REL}}_{\mathscr{G}^k}$ has been

Algorithm 2.1: MS MINTOC

```
input  : initial control discretization grid $\mathscr{G}^0$, tolerance $TOL \in \mathbb{R}^+$, $k = 0$.
output : $\varepsilon$–optimal solution
begin
    If necessary, reformulate and convexify (Sections 2.6.3, 2.6.5) problem (2.1).
    Obtain problem of type (2.8). Relax this problem to $\alpha(\cdot) \in [0,1]^{n_\omega}$.
    repeat
        Solve relaxed problem on $\mathscr{G}^k$. Obtain $\mathscr{T}^k = (x^k(\cdot), u^k(\cdot), \alpha^k(\cdot))$ and the
        grid–dependent optimal value $\Phi^{\text{REL}}_{\mathscr{G}^k}$.
        If $\mathscr{T}^k$ on $\mathscr{G}^k$ fulfills $\omega^k(\cdot) := \alpha^k(\cdot) \in \{0,1\}^{n_\omega}$ then STOP.
        Apply Sum Up Rounding (Section 2.7.3) to $\alpha^k(\cdot)$. Fix $u^k(\cdot)$.
        Obtain $y^k(\cdot)$ and upper bound $\Phi^{\text{BIN}}_{\mathscr{G}^k}$ by simulation.
        If $\Phi^{\text{BIN}}_{\mathscr{G}^k} < \Phi^{\text{REL}}_{\mathscr{G}^k} + TOL$ then STOP.
        Refine the control grid $\mathscr{G}^k$.
        $k = k + 1$.
    until
    Bijection to obtain solution for problem (2.1) with objective $\Phi^* = \Phi^{\text{BIN}}_{\mathscr{G}^k}$.
end
```

found. It is left to show that the algorithm does not loop infinitely often. Let $\omega^k(\cdot)$ be the control that we obtain from applying Sum Up Rounding on grid $\mathscr{G}^k$ to $\alpha^k(\cdot)$, and $y^k(\cdot)$ the vector of corresponding differential states. From Theorem 2.7.1 we have

$$\left\| \int_0^{t_\text{f}} \omega^k(\tau) - \alpha^k(\tau)\, \mathrm{d}\tau \right\| \leq 0.5\, \Delta t,$$

hence with Theorem 2.5.1 on page 15

$$\| \, y^k(t_\text{f}) - x^k(t_\text{f}) \, \| \leq M \Delta t e^{L t_\text{f}}.$$

Due to Assumption 3, $x^k(\cdot)$ stays constant for $k \geq k^{\max}$. Reducing Δt by bisection causes a strictly monotonic decrease of this expression, and this holds also for $\Phi(y^k(t_\text{f})) - \Phi(x^k(t_\text{f}))$, as $\Phi(\cdot)$ is a continuous function. □

Note that Algorithm 2.1 is modified in practice for efficiency. Of particular interest are solutions with a small number of switches, but good performance. Therefore we recommend to include an intermediate switching time optimization (Section 2.6.1), initialized with the $\omega^k(\cdot)$ in 3.(c) to improve $\Phi^{\text{BIN}}_{\mathscr{G}^k}$. It may also be advantageous to leave $u^k(\cdot)$ open for optimization, to compensate for the coarser grid. Also, adaptive refinements of the grid $\mathscr{G}^k$, based on control values $\alpha^k(\cdot)$ are preferable to bisection, see [203, 217].

2.8 More general problem classes

Problem (2.1) does not include all features that a mathematical model of a control process might show. In this section we discuss some straightforward extensions of the aforementioned approach, plus two features at the end, where special attention and further work is necessary.

Bolza type functionals. Every Lagrange term $\int L(x(t),u(t),v(t))\mathrm{d}t$ can be transformed equivalently into a Mayer term, hence the objective can also be of the more general Bolza type.
Multi-point constraints. Whenever multi-point constraints of the form

$$\begin{array}{rcl} 0 & \leq & r^{\text{ieq}}(x(t_0),x(t_1),\ldots,x(t_\text{f})), \\ 0 & = & r^{\text{eq}}(x(t_0),x(t_1),\ldots,x(t_\text{f})) \end{array}$$

have to be fulfilled, the same argument as for the objective function can be used: All differential states corresponding to a relaxed solution can be approximated arbitrarily close by the ones corresponding to an integer solution, and $r^{\text{ieq}}(\cdot), r^{\text{eq}}(\cdot)$ are assumed to be at least continuous functions. Algorithm 2.1 needs to be extended in the sense that for all constraints an additional tolerance has to be checked in step 3.(d).
Path constraints. Path constraints $c(x(t),u(t)) \geq 0 \;\forall\; t \in [0,t_\text{f}]$ are discretized in direct approaches, see Section 2.4, hence with a fixed $u^*(\cdot)$ the same argument as for multi-point constraints applies.
Time-independent continuous and combinatorial variables. For many processes also time-independent control values enter the problem formulation, say of continuous type, $p^{\min} \leq p \leq p^{\max}$, and of integer type, such as $\rho \in \{\rho^1,\rho^2,\ldots,\rho^{n_\rho}\}$. These control values are optimized together with the continuous controls $u^*(\cdot)$ and the relaxed binary controls $\alpha(\cdot)$. Once determined, $(u^*(\cdot),p^*,\rho^*)$ are fixed. In a second stage, the REPEAT loop of Algorithm 2.1, feasible binary controls are determined. Especially integer control values ρ^* are typically hard to compute. Our procedure allows thus for a decoupling of the determination of optimal integer control values and optimal binary control functions, resulting in a huge reduction of complexity.
Multi-stage processes. Often complex practical processes, such as batch processes in chemical engineering or robot control, consist of several successive phases with different models and transition phases that may even change the number of differential states, see, e.g., [167]. The main additional effect of multiple stages that plays a role in Theorem 2.5.1 are the initial values of the differential states on each model stage determined by a continuous transition function. The expression $\| y_0^i - x_0^i \|$ for model stage i is nothing else than a function of the difference of the differential states on model stage $i-1$. Hence, also $\| y_0^i - x_0^i \|$ depends on the control discretization grid size Δt.
Global optimization. Algorithm 2.1 works for both global as local optimization. If a global method is applied in step 3.(a), the integer solution approximates arbitrarily close the global optimum. If a local approach is chosen, the result is an approximation of this local optimum.

Multi-objective optimization. There is an important implication in the context of multi-objective optimization: whenever the Pareto front is to be calculated, it suffices to solve the relaxed convexified problem. The Pareto front of optimal control problems involving integer functions can hence be calculated without actually solving a single integer problem. This has been shown exemplarily in [170], where a combined multi-objective mixed-integer optimal control algorithm is presented.

State-dependent switches. In hybrid systems a second type of discrete events may occur, namely state-dependent switches. Prominent examples are overflows in chemical engineering or ground contact in robotics, both dependent on a differential state (volume, vertical position) and triggering a model change. Mathematically these systems can be modeled by means of continuous switching functions. For all possible orderings of such events Theorems 2.5.1 and 2.7.1 can be adapted.

Algebraic variables and conditions. Theory and algorithms have to be extended for the case that algebraic equations involving the binary control functions are present, e.g., in an explicit system of index 1,

$$\begin{aligned} \dot{x}(t) &= f(x(t), z(t), u(t), \omega(t)), \quad t \in [0, t_\mathrm{f}], \\ 0 &= g(x(t), z(t), u(t), \omega(t)), \quad t \in [0, t_\mathrm{f}]. \end{aligned}$$

Formally, index 1 DAE systems can be transformed into an ODE, making it possible to treat them within the proposed methodology. However, for many systems special DAE solvers have been developed, as the additional derivation of the system is not beneficial from a numerical point of view. Further analysis is needed on how to exploit occurring structures.

(Mixed Path-) Control constraints. For generic constraints of the type

$$c(x(t), u(t), v(t)) \geq 0 \, \forall \, t \in [0, t_\mathrm{f}]$$

no termination criterium for Algorithm 2.1 can be guaranteed (think about a constraint that simply cuts off all binary solutions). However, in most practical applications the constraints are usually of one of the following types.

- **Special Ordered Set type 1 constraints**
  After a partial outer convexification, compare Section 2.6.5, the resulting MIOCP contains coupling constraints on the values of the binary control functions $\omega(\cdot)$. The constraints (2.2c)

  $$\sum_{i=1}^{n_\omega} \omega_i(t) \quad = \quad 1, \quad t \in [0, t_\mathrm{f}]$$

  however can be addressed with the specialized Sum Up Rounding strategy presented in Section 2.7.3.
- **Combinatorial linear constraints**
  Many explicit constraints on the integer control functions are linear, or can be equivalently

reformulated as linear constraints. For these constraints we propose to use a decoupling of the nonlinearity and the integrality requirements. This is discussed in detail in Chapter 5.

- **Logical implications**
  One general idea is to reformulate the constraints (2.1f) for $t \in [0, t_f]$ to either

  $$0 \leq \sum_{j=1}^{n_\omega} c(x(t), u(t), v^j) \ \omega_j(t) \tag{2.19}$$

  or for $j = 1, \dots, n_\omega$ to

  $$0 \leq c(x(t), u(t), v^j) \ \omega_j(t) \tag{2.20}$$

  which are equivalent for $\omega_j(t) \in \{0, 1\}$. The first one, unfortunately, leads to compensation effects once $\omega(\cdot)$ is relaxed. This formulation is similar to the convex hull formulation in a disjunctive programming approach, as discussed in [195, 120, 184].
  The second formulation can also be interpreted as "only when choice j is active, the constraint $0 \leq c(x(t), u(t), v^j)$ needs to hold". Note that by constraint (2.20) only positive relaxed solutions are feasible, for which also the corresponding binary vector is feasible. This makes it more unlikely (although not impossible) that the index j corresponding to a value $q^\alpha_{j,i} = 0$ is chosen as the maximum in (2.18), whenever $c(x(t), u(t), v^j) < 0$ on $[t_i, t_{i+1}]$. Furthermore this constraint should be included in the rounding decision to avoid infeasibilities.
  Constraints of type (2.20) are called *vanishing constraints*. Note that every optimization problem with vanishing constraints can be transformed into an optimization problem with equilibrium constraints [3, 133]. Unfortunately, they may violate constraint qualifications and hence lead to severe numerical and theoretical problems. One possible approach to overcome this is to treat the vanishing constraints on the QP level with a tailored active set strategy, compare [143].

MIOC under Uncertainties.

When optimal control is applied in practice, uncertainties need to be taken into account. Typical sources of uncertainties are model mismatch (wrong or approximative model), external disturbances (wind, rain, economic behavior, ...), or strategic uncertainties (e.g., uncertain future use of designed plant).

There are different algorithmic approaches to uncertainties, e.g., worst-case (robust) optimization, optimization of expectation value, optimization with feedback, optimization of Value-at-Risk, optimization of Conditional Value-at-Risk, multi-stage stochastic programming. An excellent survey is given by [201]. All theoretical results and proposed methods carry over to our methodology for MIOC, due to the direct first discretize, then optimize approach and the theoretically established connection between relaxed and integer feasible trajectories.

One example are worst case scenarios. Here we want to make sure safety critical constraints are satisfied for all possible parameters p. In an abstract setting the optimization problem

$$\begin{aligned}&\min_{x,u} \quad \Phi[x,u,p]\\&\text{subject to}\\&\qquad 0 \;=\; F[x,u,p],\\&\qquad 0 \;\leq\; C[x,u,p].\end{aligned} \tag{2.21}$$

for particular values of the parameters p is replaced by the robust counterpart

$$\begin{aligned}&\min_{x,u} \quad \Phi[x,u,p]\\&\text{subject to}\\&\qquad 0 \;=\; F[x,u,p],\\&\qquad 0 \;\leq\; \min_{\|p-\bar{p}\|_{2,\Sigma^{-1}}\leq\gamma} C[x,u,p].\end{aligned} \tag{2.22}$$

Problem (2.22) is a semi-infinite optimization problem with an infinite number of constraints, which is very difficult to tackle. One approach to approximate the solution has been proposed by [31, 73]. It is easy to show that up to first order

$$\min_{\|p-\bar{p}\|_{2,\Sigma^{-1}}\leq\gamma} C[x,u,p] \approx C[x,u,\bar{p}] + \gamma \left\| \frac{\mathrm{d}}{\mathrm{d}p} C[x,u,\bar{p}] \right\|_{2,\Sigma} \tag{2.23}$$

So we can approximate the solution of (2.22) by

$$\begin{aligned}&\min_{x,u} \quad \Phi[x,u,\bar{p}]\\&\text{subject to}\\&\qquad 0 \;=\; F[x,u,\bar{p}],\\&\qquad 0 \;\leq\; C[x,u,\bar{p}] + \gamma \left\| \tfrac{\mathrm{d}}{\mathrm{d}p} C[x,u,\bar{p}] \right\|_{2,\Sigma}.\end{aligned} \tag{2.24}$$

For robust MIOC we calculate the relaxed solution to the robust counterpart OCP, which we approximate arbitrarily close as described above. How to include statistical information on the uncertainty is exemplarily discussed in Chapter 6.

Nonlinear model predictive control (NMPC). In practical control applications often *feedback information* is available in form of measurements. Nonlinear model predictive control strives to include these measurements in a closed loop, in which calculation of new controls that are being applied to the process and measurements are iterated. Each sampling time, one solves for a given system state x_0 that may be partially or fully determined from measurements, a MIOCP. Figure 2.3 visualizes the basic concept.

Clearly, the solution of the MIOC needs to be obtained fast to be able to apply it in real-time. An

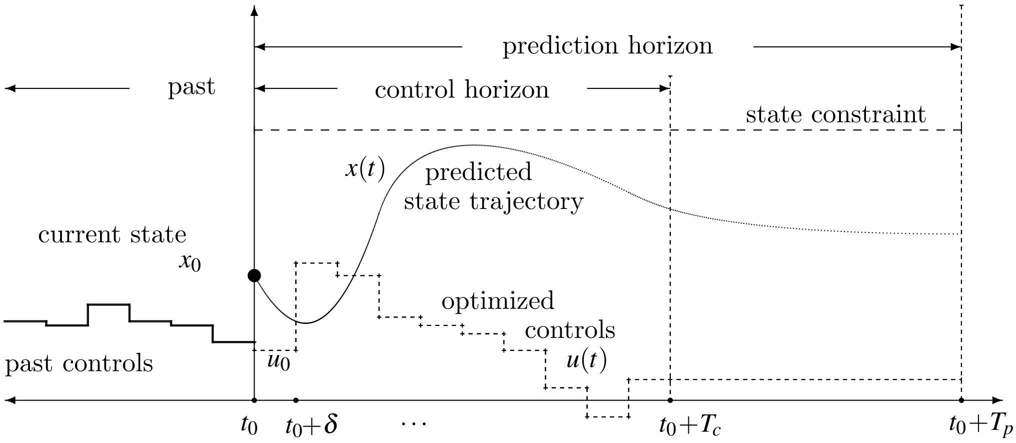


Figure 2.3: Sketch of NMPC: Using the current state x_0, a control problem is solved on a prediction horizon. The first control u_0 on the time interval $[t_0, t_0+\delta]$ is given “back to the real-world”, i.e., applied to the process. Then the horizon is shifted by one sampling interval to the right. This results in a feedback law $u_0(x_0)$ that allows to react to disturbances and modeling errors.

important ingredient is the speed up achieved by Algorithm 2.1 compared to traditional MINLP algorithms, compare Section 7.9.2 for an example with a speed up of four orders of magnitude. In addition to this, further speedup can and needs to be obtained.

Apparently, the value of the integer control functions needs to be determined only on the first time interval $[t_0, t_0+\delta]$, whereas the relaxed values can be taken for the future time intervals. Of more importance, however, is the fast solution of the relaxed control problems.

We apply ideas of Diehl, Bock and coworkers. In particular, the solution of the previous horizon problem is used to initialize the variables such that they are already close to the solution. This initialization is done using an *initial value embedding* [74]. While the conventional approach initializes with the new initial value x_0 and integrates with the old u_0, initial value embedding initializes with the old trajectory and adds a constraint $s_0^x - x_0 = 0$ that the optimization takes into account. The first iteration is a tangential predictor for the exact solution (for exact Hessian SQP) and is also valid for active set changes. This concept is combined with real-time iterations that optimize while the problem is changing. It exploits that derivatives can be computed *before* x_0 is known and allows to get the first iteration nearly without delay. As a result, one needs only one iteration despite the overall nonlinear approach. Also it is possible to show the nominal stability of the combined system-optimizer dynamics. For details see [72, 74].

Building on the SQP-based iterative scheme of [74], a new real-time multi-level optimization algorithm has been proposed. The four-level scheme proposed in [6] operates with ultra-fast feedback on the lowest level, where small QPs are solved very quickly using an online active set strategy, [89]. On the second level, which provides updated data down to the first level, the nonlinear constraints are evaluated to improve feasibility, on the third level the Lagrange gradient is evaluated by means of adjoint sensitivities to improve nonlinear optimality, [41, 250], and on the topmost level the complete derivative information if generated. Additional potential for speedup lies in an efficient parallel implementation and on the concept of an automatic code export [175, 127].

A final possibility for speedup is on the linear algebra level. In Chapter 4 we present a tailored solution approach that takes the special properties of MIOCPs for which an Outer Convexification has been applied into account.

Many numerical issues in NMPC are discussed in [148]. For further details on mixed–integer nonlinear model predictive control we refer to the PhD thesis [143]. Here also a proof for the nominal stability for rounding–based schemes can be found.

2.9 Summary

We presented a broad overview on recent mathematical developments in the efficient algorithmic treatment of switching decisions in nonlinear optimal control. Theoretical foundations for error estimates are given alongside a discussion of possible solution approaches. A comprehensive algorithm is presented. Well-posedness of the algorithm is discussed, as well as extensions to treat more general optimal control problems.

3 The Integer Approximation Error in Mixed-Integer Optimal Control

The contents of this chapter are based on the paper

[208] S. Sager, H.G. Bock, M. Diehl. The Integer Approximation Error in Mixed-Integer Optimal Control. *Mathematical Programming A*, 2011, DOI 10.1007/s10107-010-0405-3.

Chapter Summary. We extend recent work on nonlinear optimal control problems with integer restrictions on some of the control functions (mixed-integer optimal control problems, MIOCP). We improve a theorem [214] that states that the solution of a relaxed and convexified problem can be approximated with arbitrary precision by a solution fulfilling the integer requirements. Unlike in previous publications the new proof avoids the usage of the Krein-Milman theorem, which is undesirable as it only states the existence of a solution that may switch infinitely often. We present a constructive way to obtain an integer solution with a guaranteed bound on the performance loss in polynomial time. We prove that this bound depends linearly on the control discretization grid. A numerical benchmark example illustrates the procedure.
As a byproduct, we obtain an estimate of the Hausdorff distance between reachable sets. We improve the approximation order to linear grid size h instead of the previously known result with order $\sqrt{h}$ [123]. We are able to include a Special Ordered Set condition which allows a transfer of the results to a more general, multi-dimensional and nonlinear case compared to the theorems in [190].

3.1 Introduction

Our main motivation are mixed-integer optimal control problems (MIOCPs) in ordinary differential equations (ODE) that are of the form (2.1), compare page 8. See Chapter 2 for a generic introduction to MIOC and the relation to other approaches in MIOC.
Relation to own work. In [214] a new approach to MIOC was proposed. Based on insight from functional analysis, the exact lower bound for the nonlinear integer control problem is determined by solving a relaxed, continuous control problem. Integer solutions are obtained by a combination of grid adaptivity and the Sum Up Rounding Strategy described later on in this chapter.
We extend this work in two ways. First, a theorem stating that the solution of a relaxed and convexified problem can be approximated arbitrarily close by a solution fulfilling the integer

requirements is improved. Unlike before, a new short and self-contained proof avoids the usage of the Krein-Milman theorem, which is undesirable as it only states the existence of a solution that may switch infinitely often.

Second, the Sum Up Rounding strategy to obtain integer controls from continuous, relaxed ones, is analyzed. Previously, it has been described as a heuristic, similar to rounding methods in integer programming. However, it is used in the above proof. It yields a constructive way to obtain an integer solution with a guaranteed bound on the performance loss in polynomial time. We prove that this tolerance depends on the control discretization grid. The rounded solution is arbitrarily close to the relaxed one, if only the underlying grid is chosen fine enough.

The complete algorithm to solve MIOCPs has been described in Chapter 2. In there, the most important part of the proof for the algorithm's termination in a finite number of steps is missing, however. To fill this gap is the main contribution of this chapter.

Related work in error estimation for switched systems. In his PhD thesis [123] Gerhard Häckl estimated the Hausdorff distance between the reachable sets $cl(R^+(x_0))$ of a continuous time system and $cl(R^+(h,x_0))$ of a discrete time system with piecewise constant controls and grid size h. Parts of this dissertation entered in the book [68], the convergence result and approximation order are discussed in Section C.1. In comparison our results show that the approximation order is of order h instead of a constant multiple of $\sqrt{h}$ as claimed in [123, Corollary 2.4.8]. Also our estimation does hold for all values of h, and not only as $h \to 0$. The reason seems to be that Häckl and coworkers do not make use of the Sum Up Rounding strategy which is needed for the better approximation order. Also the extension from control-affine systems to nonlinear ones is not discussed.

A related result on error bounds has recently been obtained independently of this work by [190], building on work of [77, 117, 242, 241]. The authors give an upper bound of order h on the Hausdorff distance between the reachable set of relaxed controls and controls that are restricted to the space of piecewise constant functions that may only take the values 0 and 1 on a finite time grid. The mathematical approach is based on differential inclusions and Lie brackets. They use the Sum Up Rounding [203] strategy as well within their proof. Their study is restricted to the one-dimensional linear case, while we consider integer controls in arbitrary dimension and allow for nonlinearities.

To our knowledge, the approximation order h was first postulated in [241], for a locally Lipschitz continuous right-hand side. Veliov writes: "However, the author was able so far to prove this only in some special cases and the problem is still open." We refer to this as "Veliov's conjecture" in the following.

More remotely related is the question of the maximum number of switches for equivalent reachable sets. For a special case of a switched system it is shown in [225] that 4 switches are enough. A counterexample based on Fuller's phenomenon is given in [172].

Outline. We first consider the case where $v(\cdot) = \omega(\cdot)$ enters linearly in the optimization problem. This is the case for which theoretical results can be obtained, and we see later on that the nonlinearity with respect to the integer control function vanishes by a partial outer convexification using the reformulation (2.2b). We show that for any feasible relaxed solution we obtain a binary solution by the presented rounding strategy that is feasible and reaches the objective function value, both up to a given tolerance that depends on the control discretization grid size.

For this we deduce theoretical results concerning the difference between differential states that are obtained by integration with different control functions in Section 3.2. In Section 3.3 we present the rounding strategy and give an upper bound on the difference between the integral over the relaxed and the rounded control. In Section 3.4 we extend the results to the case in which the integer function $v(\cdot)$ enters in a nonlinear way. The partial outer convexification leads to additional Special Ordered Set constraints on the resulting binary control functions $\omega(\cdot)$ that we take into account in an extended Sum Up Rounding Strategy. In Section 3.5 we bring together the results and connect them to the optimization problem. In Section 3.6 we investigate a benchmark example to illustrate the procedure. We sum up the results in Section 3.7.

3.2 Approximating differential states

We want to show how the difference of the integrals of two differential states depends on the difference of the integrals of their corresponding control functions. Before we come to the main theorem of this section, we need the following lemma that can also be found, e.g., in [108, Lemma 1.3, page 4].

Lemma 3.2.1 (A variant of the Gronwall Lemma). *Let $[t_0,t_f]$ be an interval and $w,z : [t_0,t_f] \mapsto \mathbb{R}$ real-valued integrable functions. If for constant $L \geq 0$ it holds for $t \in [t_0,t_f]$ almost everywhere that*

$$w(t) \leq z(t) + L \int_{t_0}^{t} w(\tau)\, \mathrm{d}\tau$$

then also

$$w(t) \leq z(t) + L \int_{t_0}^{t} e^{L(t-\tau)} z(\tau)\, \mathrm{d}\tau$$

for $t \in [t_0,t_f]$ almost everywhere. If $z(\cdot)$ in addition belongs to $L^\infty([t_0,t_f],\mathbb{R})$ then it holds

$$w(t) \leq \| z(\cdot) \|_\infty e^{L(t-t_0)}$$

for $t \in [t_0,t_f]$ almost everywhere.

Proof. According to the assumption we may write

$$w(t) = a(t) + z(t) + \delta(t) \tag{3.1}$$

with the absolutely continuous function

$$a(t) := L \int_{t_0}^{t} w(\tau)\, \mathrm{d}\tau \tag{3.2}$$

and a non-positive function $\delta(\cdot) \in L^1([t_0,t_\mathrm{f}],\mathbb{R})$. Using (3.1) in (3.2) yields

$$a(t) = L \int_{t_0}^{t} a(\tau)\, \mathrm{d}\tau \;+\; L \int_{t_0}^{t} z(\tau) + \delta(\tau)\, \mathrm{d}\tau.$$

Hence, $a(\cdot)$ solves the inhomogeneous linear differential equation

$$\frac{\mathrm{d}a}{\mathrm{d}t}(t) = La(t) \;+\; L(z(t) + \delta(t))$$

for $t \in [t_0,t_\mathrm{f}]$ almost everywhere and initial value $a(t_0) = 0$. The well-known solution formula for linear differential equations yields

$$a(t) = L \int_{t_0}^{t} e^{L(t-\tau)}\, (z(\tau) + \delta(\tau))\, \mathrm{d}\tau$$

respectively

$$w(t) = z(t) + \delta(t) \;+\; L \int_{t_0}^{t} e^{L(t-\tau)}\, (z(\tau) + \delta(\tau))\, \mathrm{d}\tau.$$

Since $\delta(t) \leq 0$ the first assertion holds. If $z(\cdot)$ is essentially bounded we find

$$w(t) \leq \| \, z(\cdot) \, \| \left(1 + L \int_{t_0}^{t} e^{L(t-\tau)}\, \mathrm{d}\tau \right) = \| \, z(\cdot) \, \| \; e^{L(t-t_0)},$$

completing the proof. □

Assume now we are given an initial value problem that is of the form

$$\dot{x}(t) \;\; = \;\; A(t,x(t))\, \alpha(t), \quad x(0) = x_0. \tag{3.3}$$

Here $A(t,x(t))$ is a matrix in $\mathbb{R}^{n_x \times n_\omega}$ with entries depending on t and $x(t)$. We assume in the following that the function $A(\cdot)$ is differentiable with respect to time and fulfills certain requirements with respect to its argument x. Note that we leave away a term independent of $\alpha(\cdot)$, as it may be included easily by fixing one additional component of α to 1. The following theorem states how the difference of solutions to this initial value problem depends on the integrated difference between control functions and the difference between the initial values.

Theorem 3.2.2. *Let $x(\cdot)$ and $y(\cdot)$ be solutions of the initial value problems*

$$\dot{x}(t) \;=\; A(t,x(t))\cdot\alpha(t),\quad x(0)=x_0, \tag{3.4a}$$

$$\dot{y}(t) \;=\; A(t,y(t))\cdot\omega(t),\quad y(0)=y_0, \tag{3.4b}$$

with $t\in[0,t_f]$, for given measurable functions $\alpha,\omega:[0,t_f]\to[0,1]^{n_\omega}$ and a differentiable $A:\mathbb{R}^{n_x+1}\mapsto\mathbb{R}^{n_x\times n_\omega}$. If positive numbers $C,L\in\mathbb{R}^+$ exist such that for $t\in[0,t_f]$ almost everywhere it holds that

$$\left\| \frac{\mathrm{d}}{\mathrm{d}t}A(t,x(t)) \right\| \;\leq\; C, \tag{3.4c}$$

$$\|\, A(t,y(t)) - A(t,x(t)) \,\| \;\leq\; L\,\|\, y(t)-x(t) \,\|, \tag{3.4d}$$

and $A(\cdot,x(\cdot))$ is essentially bounded by $M\in\mathbb{R}^+$ on $[0,t_f]$, and it exists $\varepsilon\in\mathbb{R}^+$ such that for all $t\in[0,t_f]$

$$\left\| \int_0^t \alpha(\tau)-\omega(\tau)\,\mathrm{d}\tau \right\| \;\leq\; \varepsilon \tag{3.4e}$$

then it also holds

$$\|\, y(t)-x(t) \,\| \leq (\|\, x_0-y_0 \,\| + (M+Ct)\varepsilon)\, e^{Lt} \tag{3.4f}$$

for all $t\in[0,t_f]$.

Proof. Because both α and ω map to $[0,1]^{n_\omega}$ we have

$$\|\,\alpha(t)\,\| \;\leq\; 1,\quad \|\,\omega(t)\,\| \leq 1 \tag{3.5}$$

for all $t\in[0,t_\mathrm{f}]$. As ω and α are measurable and bounded functions, so is $\Delta w := \alpha-\omega$. We define Δa as $\Delta a(t) := \int_0^t \Delta w(\tau)\,\mathrm{d}\tau$. Note that it holds $\Delta a(0)=\int_0^0 \Delta w(\tau)\,\mathrm{d}\tau = 0$ and $\|\,\Delta a(t)\,\|\leq\varepsilon$. Because of (3.4a,3.4b) we can write

$$x(t) \;=\; x_0+\int_0^t A(\tau,x(\tau))\;\alpha(\tau)\,\mathrm{d}\tau,\quad y(t)=y_0+\int_0^t A(\tau,y(\tau))\;\omega(\tau)\,\mathrm{d}\tau$$

and obtain

$$\begin{aligned}\|\,x(t)-y(t)\,\| \;&\leq\; \|\,x_0-y_0\,\| + \left\| \int_0^t A(\tau,x(\tau))\;\alpha(\tau) - A(\tau,y(\tau))\;\omega(\tau)\,\mathrm{d}\tau \right\| \\ &\leq\; \|\,x_0-y_0\,\| + \left\| \int_0^t A(\tau,x(\tau))\;\omega(\tau) - A(\tau,y(\tau))\;\omega(\tau)\,\mathrm{d}\tau \right\| \\ &\qquad + \left\| \int_0^t A(\tau,x(\tau))\;\alpha(\tau) - A(\tau,x(\tau))\;\omega(\tau)\,\mathrm{d}\tau \right\|\end{aligned}$$

$$
\begin{aligned}
&= \;\| x_0 - y_0 \| + \left\| \int_0^t (A(\tau,x(\tau)) - A(\tau,y(\tau)))\; \omega(\tau)\; \mathrm{d}\tau \right\| \\
&\qquad + \left\| \int_0^t A(\tau,x(\tau))\; \Delta w(\tau)\; \mathrm{d}\tau \right\| \\
&= \;\| x_0 - y_0 \| + \left\| \int_0^t (A(\tau,x(\tau)) - A(\tau,y(\tau)))\; \omega(\tau)\; \mathrm{d}\tau \right\| \\
&\qquad + \left\| A(t,x(t))\Delta a(t) - \int_0^t \frac{\mathrm{d}}{\mathrm{d}\tau} A(\tau,x(\tau))\; \Delta a(\tau)\; \mathrm{d}\tau \right\| \\
&\leq \;\| x_0 - y_0 \| + L \int_0^t \| x(\tau) - y(\tau) \| \; \| \omega(\tau) \| \; \mathrm{d}\tau \\
&\qquad + \| A(t,x(t)) \| \, \varepsilon + \int_0^t \left\| \frac{\mathrm{d}}{\mathrm{d}t} A(\tau,x(\tau)) \right\| \; \| \Delta a(\tau) \| \; \mathrm{d}\tau \\
&\leq \;\| x_0 - y_0 \| + L \int_0^t \| x(\tau) - y(\tau) \| \; \mathrm{d}\tau \\
&\qquad + (\| A(t,x(t)) \| + Ct)\varepsilon.
\end{aligned}
$$

The functions

$$
w(t) := \| x(t) - y(t) \|, \qquad z(t) := \| x_0 - y_0 \| + (\| A(t,x(t)) \| + Ct)\varepsilon
$$

are integrable and $z(\cdot)$ is in $L^\infty([t_0,t_\mathrm{f}],\mathbb{R})$. Applying Lemma 3.2.1 yields the claim

$$
\| y(t) - x(t) \| \leq (\| x_0 - y_0 \| + (M + Ct)\varepsilon)\, e^{Lt}
$$

for all $t \in [0,t_\mathrm{f}]$. □

Note that assumptions (3.4c) and (3.4d) do not require global constants, but only for the two trajectories $x(\cdot)$ and $y(\cdot)$ under consideration. In our context the initial values x_0 and y_0 are identical. From the monotonicity $e^{Lt} \leq e^{Lt_\mathrm{f}}$ it follows that Theorem 3.2.2 states that we have an upper bound $\| y(t) - x(t) \| \leq c \cdot \varepsilon$ with constant $c \geq 0$ on the difference between differential states that depends linearly on the integrated difference between the two control functions. In the next section we investigate this term closer.

3.3 Approximating the integral over the controls by sum up rounding

We consider given measurable functions $\alpha_j : [0,t_\mathrm{f}] \mapsto [0,1]$ with $j = 1 \ldots n_\omega$ and a time grid $0 = t_0 < t_1 < \cdots < t_m = t_\mathrm{f}$ on which we want to approximate the control $\alpha(\cdot)$. We write $\Delta t_i := t_{i+1} - t_i$ and Δt for the maximum distance between two time points,

$$
\Delta t := \max_{i=0\ldots m-1} \Delta t_i = \max_{i=0\ldots m-1} \{t_{i+1} - t_i\}. \tag{3.6}
$$

Let then a function $\omega(\cdot) : [0,t_\mathrm{f}] \mapsto \{0,1\}^{n_\omega}$ be defined by

$$\omega_j(t) = p_{j,i}, \quad t \in [t_i, t_{i+1}) \tag{3.7}$$

where for $i = 0 \dots m-1$ the $p_{j,i}$ are binary values given by

$$p_{j,i} \quad = \quad \begin{cases} 1 & \text{if } \int_0^{t_{i+1}} \alpha_j(\tau)\mathrm{d}\tau - \sum_{k=0}^{i-1} p_{j,k}\Delta t_k \geq 0.5\Delta t_i \\ 0 & \text{else} \end{cases} . \tag{3.8}$$

See Figure 3.1 for an example. We have the following estimate on the integral over the difference between the control functions $\alpha(\cdot)$ and $\omega(\cdot)$.

Theorem 3.3.1. *Let a measurable function* $\alpha : [0,t_f] \mapsto [0,1]^{n_\omega}$ *and a function* $\omega : [0,t_f] \mapsto \{0,1\}^{n_\omega}$ *defined by (3.7, 3.8) be given. Then it holds*

$$\left\| \int_0^t \alpha(\tau) - \omega(\tau) \,\mathrm{d}\tau \right\| \leq 0.5\,\Delta t.$$

Proof. Let $0 \leq r \leq m-1$ be the index such that $t_r \leq t < t_{r+1}$. First observe that maximum or minimum values of the integrals

$$\int_0^t \alpha_j(\tau) - \omega_j(\tau)\,\mathrm{d}\tau = \int_0^{t_r} \alpha_j(\tau) - \omega_j(\tau)\,\mathrm{d}\tau + \int_{t_r}^t \alpha_j(\tau) - p_{j,r}\,\mathrm{d}\tau$$

are obtained on the time grid, as either $\alpha_j(\tau) \leq p_{j,r}$ or $\alpha_j(\tau) \geq p_{j,r}$ on $[t_r, t_{r+1}]$. Therefore it suffices to show the claim for all $t = t_r$. For $r = 0 \dots m$ we show by induction that

$$\left\| \int_0^{t_r} \alpha(\tau) - \omega(\tau)\,\mathrm{d}\tau \right\| \quad = \quad \max_j \left| \int_0^{t_r} \alpha_j(\tau)\mathrm{d}\tau - \sum_{i=0}^{r-1} p_{j,i}\Delta t_i \right| \leq 0.5\,\Delta t. \tag{3.9}$$

For $r = 0$ the claim follows trivially. So let us assume

$$\max_j \left| \int_0^{t_r} \alpha_j(\tau)\mathrm{d}\tau - \sum_{i=0}^{r-1} p_{j,i}\Delta t_i \right| \quad \leq \quad 0.5\,\Delta t \tag{3.10}$$

and show that also

$$\max_j \left| \int_0^{t_{r+1}} \alpha_j(\tau)\mathrm{d}\tau - \sum_{i=0}^{r} p_{j,i}\Delta t_i \right| \quad \leq \quad 0.5\,\Delta t.$$

For all $j = 1, \dots, n_\omega$ it holds that if $p_{j,r} = 1$, then because of (3.8) we have

$$\int_0^{t_{r+1}} \alpha_j(\tau)\mathrm{d}\tau - \sum_{i=0}^{r-1} p_{j,i}\Delta t_i \geq 0.5\Delta t_r$$

and by adding $-p_{j,r}\Delta t_r = -\Delta t_r$ on both sides

$$\int_0^{t_{r+1}} \alpha_j(\tau)\mathrm{d}\tau - \sum_{i=0}^{r} p_{j,i}\Delta t_i \geq -0.5\Delta t_r \geq -0.5\Delta t.$$

By induction hypothesis we also have

$$\underbrace{\int_0^{t_r} \alpha_j(\tau)\mathrm{d}\tau - \sum_{i=0}^{r-1} p_{j,i}\Delta t_i}_{\leq 0.5\Delta t} + \underbrace{\int_{t_r}^{t_{r+1}} \alpha_j(\tau) - 1 \;\mathrm{d}\tau}_{\leq 0} \leq 0.5\Delta t.$$

If $p_{j,r} = 0$, then because of (3.8) we have

$$\begin{aligned}\int_0^{t_{r+1}} \alpha_j(\tau)\mathrm{d}\tau - \sum_{i=0}^{r} p_{j,i}\Delta t_i &= \int_0^{t_{r+1}} \alpha_j(\tau)\mathrm{d}\tau - \sum_{i=0}^{r-1} p_{j,i}\Delta t_i \\ &< 0.5\Delta t_r \leq 0.5\Delta t.\end{aligned}$$

By induction hypothesis we also have

$$\underbrace{\int_0^{t_r} \alpha_j(\tau)\mathrm{d}\tau - \sum_{i=0}^{r-1} p_{j,i}\Delta t_i}_{\geq -0.5\Delta t} + \underbrace{\int_{t_r}^{t_{r+1}} \alpha_j(\tau) \;\mathrm{d}\tau}_{\geq 0} \geq -0.5\Delta t,$$

for all $j = 1, \dots, n_\omega$, completing the proof. □

3.4 Extension to the nonlinear case

To apply the above results to the more general nonlinear case, we convexify problem (2.1) with respect to the integer control functions $v(\cdot)$ as first suggested in [203]. We replace (2.1b) and (2.1e) by the partially convexified right hand side (2.2b) and the SOS1 constraint (2.2c). This *Outer Convexification* has shown very efficient in practice [147]. It allows us to generate a tight relaxation of the integer control problem - very similar as before for the affinely entering binary controls, but with one important modification, namely an additional linear constraint to ensure the controls form a Special Ordered Set (2.2c) at each instant in time.
There is obviously a bijection $v(t) = v^i \leftrightarrow \omega_i(t) = 1$ between solutions of problems

(2.1a,2.1b,2.1c,2.1d,2.1e,2.1f)

and

(2.1a,2.2b,2.1c,2.1d,2.2a,2.2c,2.1f),

compare [203]. This means that we can find a solution to the convexified problem that is affine in $\omega(\cdot)$ by applying the proposed Sum Up Rounding strategy to a solution of its relaxation and then deduce the optimal solution to the nonlinear binary problem (2.1) from it.
However, the Sum Up Rounding strategy (3.8) does not work for problems with the additional Special Ordered Set property (2.2c), as can be seen by the easy example of two functions with constant $\alpha_1(t) = \alpha_2(t) = 0.5$.

Remark 3.4.1. *The SOS1 constraint (2.2c) can be used to eliminate one control, e.g., $\omega_{n_\omega}(\cdot)$. One replaces*

$$\omega_{n_\omega}(t) = 1 - \sum_{i=1}^{n_\omega - 1} \omega_i(t)$$

for $t \in [0,t_f]$. Constraint (2.2c) is then always fulfilled. However, now the constraint $0 \leq \omega_{n_\omega}(t) \leq 1$ may be violated if the SUR strategy is applied (example: $\alpha_1(t) = \alpha_2(t) = 0.5$ and $\alpha_3(t) = 0$, substitute α_3). Furthermore, if $\alpha_i(t) < 0.5$ for all $i = 1 \dots n_\omega - 1$ then ω_{n_ω} is (implicitly) rounded up, even if $\omega_{n_\omega}(t) = 1 - \sum_{i=1}^{n_\omega-1} \omega_i(t)$ is small.
Substituting controls typically makes a difference concerning computational efficiency and is an interesting aspect to study. Whereas in linear programming this substitution is usually avoided to maintain sparsity, for control functions there might be good reasons for a substitution. In the diploma thesis [53] an adaptive replacement has been proposed that minimizes the effort of the underlying QP solver, dependent on the dimensions of null and image space. However, for our theoretical considerations we do not consider this case separately, all results can be transfered easily.

Therefore we propose a different technique for functions that have to fulfill this equality. Let us assume we are given a measurable function $\alpha(\cdot)$ that fulfills (2.2c). Again we define $\omega(\cdot)$ via (3.7), but with $p_{j,i}$ given by

$$\hat{p}_{j,i} = \int_0^{t_{i+1}} \alpha_j(\tau)\mathrm{d}\tau - \sum_{k=0}^{i-1} p_{j,k}\Delta t_k \tag{3.11a}$$

$$p_{j,i} = \begin{cases} 1 & \text{if } \hat{p}_{j,i} \geq \hat{p}_{k,i} \; \forall \; k \neq j \text{ and } j < k \; \forall \; k : \hat{p}_{j,i} = \hat{p}_{k,i} \\ 0 & \text{else} \end{cases} \tag{3.11b}$$

and not by (3.8). Again we have an estimation of the integral over $\alpha - \omega$ that depends on Δt of the underlying grid, compare (3.6).

Theorem 3.4.2. *Let a measurable function $\alpha : [0,t_f] \mapsto [0,1]^{n_\omega}$ that fulfills equation (2.2c) and a function $\omega : [0,t_f] \mapsto \{0,1\}^{n_\omega}$ defined by (3.7, 3.11) be given for $n_\omega \geq 2$. Then it holds*

$$\left\| \int_0^t \alpha(\tau) - \omega(\tau) \, \mathrm{d}\tau \right\| \leq (n_\omega - 1)\, \Delta t$$

and also $\omega(\cdot)$ fulfills (2.2c).

Proof. Note that $\omega(t)$ fulfills the Special Ordered Set type one property (2.2c) by construction, as exactly one entry is set to 1 and all others to 0. This is important for the proof, because it implies

$$\sum_{j=1}^{n_\omega} \int_0^t \alpha_j(\tau) - \omega_j(\tau) \, \mathrm{d}\tau = \int_0^t \sum_{j=1}^{n_\omega} (\alpha_j(\tau) - \omega_j(\tau)) \, \mathrm{d}\tau = 0 \tag{3.12}$$

for all $t \in [0, t_\mathrm{f}]$. As above we can restrict our proof to the case that $t = t_r$. For the sake of notational simplicity we define

$$k := \arg \max_{j=1 \ldots n_\omega} \left| \int_0^{t_r} \alpha_j(\tau) \mathrm{d}\tau - \sum_{i=0}^{r-1} p_{j,i} \Delta t_i \right|,$$

observing that $\int_0^{t_r} \omega_j(\tau)) \, \mathrm{d}\tau = \sum_{i=0}^{r-1} p_{j,i} \Delta t_i$. We assume that there exists an $r \in \{0 \ldots m\}$ such that the claim does not hold, i.e.,

$$\left| \int_0^{t_r} \alpha_k(\tau) \mathrm{d}\tau - \sum_{i=0}^{r-1} p_{k,i} \Delta t_i \right| \geq (n_\omega - 1) \, \Delta t$$

and contradict this assumption. We distinguish two cases. Let us first assume that

$$\int_0^{t_r} \alpha_k(\tau) \mathrm{d}\tau - \sum_{i=0}^{r-1} p_{k,i} \Delta t_i < -(n_\omega - 1) \Delta t. \tag{3.13}$$

Let $\hat{\imath}$ be the highest index for which the control k has been rounded up,

$$\hat{\imath} := \arg \max_{0 \leq i \leq r-1} \{ i : p_{k,i} = 1 \text{ and } p_{k,l} = 0 \, \forall \, l : i < l \leq r-1 \}.$$

Note that $\hat{\imath}$ is well defined, as there must be at least two i such that $p_{k,i} = 1$. Then it holds by assumption (3.13)

$$\begin{aligned} \sum_{i=0}^{\hat{\imath}} p_{k,i} \Delta t_i = \sum_{i=0}^{r-1} p_{k,i} \Delta t_i \quad &> \quad \int_0^{t_r} \alpha_k(\tau) \mathrm{d}\tau + (n_\omega - 1) \Delta t \\ &\geq \quad \int_0^{t_{\hat{\imath}+1}} \alpha_k(\tau) \mathrm{d}\tau + (n_\omega - 1) \Delta t \end{aligned}$$

and as k had the maximum value on interval $\hat{\imath}$,

$$\int_0^{t_{\hat{\imath}+1}} \alpha_j(\tau) \mathrm{d}\tau - \sum_{i=0}^{\hat{\imath}} p_{j,i} \Delta t_i < -(n_\omega - 1) \Delta t$$

for all $j = 1, \dots, n_\omega$. Summing up over all controls j yields

$$\sum_{j=1}^{n_\omega} \left(\int_0^{t_{\hat{i}+1}} \alpha_j(\tau)\mathrm{d}\tau - \sum_{i=0}^{\hat{i}} p_{j,i}\Delta t_i \right) < -\sum_{j=1}^{n_\omega} (n_\omega - 1)\Delta t$$

and because of (3.12) we have the contradiction $0 < n_\omega - n_\omega^2$.

Let us now assume that

$$\int_0^{t_r} \alpha_k(\tau)\mathrm{d}\tau - \sum_{i=0}^{r-1} p_{k,i}\Delta t_i > (n_\omega - 1)\Delta t. \tag{3.14}$$

Because of (3.12) it holds

$$\sum_{1=j\neq k}^{n_\omega} \left(\int_0^{t_r} \alpha_j(\tau)\mathrm{d}\tau - \sum_{i=0}^{r-1} p_{j,i}\Delta t_i \right) + \int_0^{t_r} \alpha_k(\tau)\mathrm{d}\tau - \sum_{i=0}^{r-1} p_{k,i}\Delta t_i = 0$$

and with assumption (3.14)

$$\sum_{1=j\neq k}^{n_\omega} \left(\int_0^{t_r} \alpha_j(\tau)\mathrm{d}\tau - \sum_{i=0}^{r-1} p_{j,i}\Delta t_i \right) + (n_\omega - 1)\Delta t < 0.$$

We can write the left hand side as the sum of $n_\omega - 1$ terms

$$\Delta t + \int_0^{t_r} \alpha_j(\tau)\mathrm{d}\tau - \sum_{i=0}^{r-1} p_{j,i}\Delta t_i.$$

Obviously at least one of them has to be negative, thus there exists an index $\hat{j}$ such that

$$\Delta t + \int_0^{t_r} \alpha_{\hat{j}}(\tau)\mathrm{d}\tau - \sum_{i=0}^{r-1} p_{\hat{j},i}\Delta t_i < 0.$$

Let $\hat{i}$ be the highest index for which the control $\hat{j}$ has been rounded up,

$$\hat{i} := \arg\max_{0\leq i\leq r-1} \{ i : p_{\hat{j},i} = 1 \text{ and } p_{\hat{j},l} = 0 \,\forall\, l : i < l \leq r-1 \}.$$

Note that $\hat{i}$ is well defined, as there must be at least two i such that $p_{\hat{j},i} = 1$. Then it holds

$$\int_0^{t_{\hat{i}+1}} \alpha_{\hat{j}}(\tau)\mathrm{d}\tau - \sum_{i=0}^{\hat{i}-1} p_{\hat{j},i}\Delta t_i \leq \Delta t + \int_0^{t_r} \alpha_{\hat{j}}(\tau)\mathrm{d}\tau - \sum_{i=0}^{r-1} p_{\hat{j},i}\Delta t_i < 0$$

and with

$$\hat{j} = \arg \max_{1 \le j \le n_\omega} \left\{ \int_0^{t_{\hat{i}+1}} \alpha_j(\tau) \mathrm{d}\tau - \sum_{i=0}^{\hat{i}-1} p_{j,i} \Delta t_i \right\}$$

which must hold because of the rounding decision at time $\hat{i}$ we have

$$\int_0^{t_{\hat{i}+1}} \alpha_j(\tau) \mathrm{d}\tau - \sum_{i=0}^{\hat{i}} p_{j,i} \Delta t_i \le \int_0^{t_{\hat{i}+1}} \alpha_j(\tau) \mathrm{d}\tau - \sum_{i=0}^{\hat{i}-1} p_{j,i} \Delta t_i < 0$$

for all $j = 1, \dots, n_\omega$ in contradiction to (3.12). □

3.5 Connection to the optimization problem

We connect the results to the optimization problem (2.1).

Corollary 3.5.1. *Let $(x, \alpha, u^*)(\cdot)$ be a feasible trajectory of the relaxed control problem*

(2.1c,2.1d,2.2b,2.2c)

with the measurable function $\alpha : [0, t_f] \to [0,1]^{n_\omega}$ replacing ω in (2.2b,2.2c).
Consider the trajectory $(y, \omega, u^*)(\cdot)$ which consists of a control $\omega(\cdot)$ determined via (3.7, 3.11) on a given time grid from $\alpha(\cdot)$ and differential states $y(\cdot)$ that are obtained by solving the initial value problem (2.1c,2.2b).
Assume that constants $C, L, M \in \mathbb{R}^+$ exist for the fixed measurable control $u^* \in \mathscr{U}$ and all $v^i \in \Omega$ such that the function $f(t, x(t), u^*(t), v^i)$ be differentiable with respect to time and it holds

$$\left\| \frac{\mathrm{d}}{\mathrm{d}t} f(t, x(t), u^*(t), v^i) \right\| \le C, \tag{3.15}$$

$$\| f(t, y(t), u^*(t), v^i) - f(t, x(t), u^*(t), v^i) \| \le L \| y(t) - x(t) \| \tag{3.16}$$

for $t \in [0, t_f]$ almost everywhere and $f(\cdot, x((\cdot), u^*(\cdot), v^i)$ is essentially bounded by M.
Then $(y, \omega, u^*)(\cdot)$ is a feasible trajectory for (2.1c,2.1d,2.2b,2.2c) and it holds

$$\| y(t) - x(t) \| \le ((M + Ct)\ c(n_\omega)\ \Delta t)\, e^{Lt} \tag{3.17}$$

for all $t \in [0, t_f]$ with constant $c(n_\omega)$.

Proof. We define the function $A : \mathbb{R}^{n_x+1} \mapsto \mathbb{R}^{n_x \times n_\omega}$ as a matrix with column i given by $f(t, x, u^*, v^i)$ for $i = 1, \dots, n_\omega$. Here both u^* and the feasible integer controls v^i are fixed. The ODE (2.1b) is then of the form (3.4a). Because $f(\cdot)$ is assumed to be differentiable with respect to time, bounded and fulfills a Lipschitz condition, this holds for $A(\cdot)$ as well. All assumptions of Theorem 3.2.2 and of either Theorem 3.3.1 or 3.4.2 are fulfilled. The constant $c(n_\omega)$ is given by $c(n_\omega) = n_\omega - 1$ if $n_\omega \ge 2$ and (2.2c) holds and $c(n_\omega) = 0.5$, else. □

The differentiability assumption on $f(\cdot)$ in Corollary 3.5.1 is quite strong, as it implies that the optimal control $u^*(\cdot)$ must be differentiable as well. However, this holds only almost everywhere, hence the important case of controls u^* with finitely many discontinuities is included.

Remark 3.5.2. *The important result of Corollary 3.5.1 is the linear convergence order with respect to Δt. However, also the constants may be interesting from a practical point of view.*
In Theorem 3.3.1 the estimation is sharp, as can be seen by investigating the constant function $\alpha(\cdot) = 0.5$.
In Theorem 3.4.2 we think the constant $(n_\omega - 1)$ can be improved. Without proof: Assume $[0, t_f]$ is partitioned in n_ω equidistant time intervals. The deviation from the constructed control $\omega(\cdot)$ and the control $\alpha(\cdot)$ is maximal, when the n_ω controls $\alpha(\cdot)$ are piecewise constant functions defined as

$$\alpha_j(t) = \begin{cases} \frac{1}{n_\omega - i} & j \geq i \\ 0 & j < i \end{cases} \quad t \in [t_i, t_{i+1}],\ i = 0, \dots, n_\omega - 1,\ j = 1, \dots, n_\omega$$

Applying (3.7, 3.11) results in

$$p_{n_\omega, i} = \begin{cases} 0 & i < n_\omega - 1 \\ 1 & i = n_\omega - 1 \end{cases}$$

The maximal deviation at time $t_{n_\omega - 1}$ is then the harmonic number

$$\sum_{i=0}^{n_\omega - 2} \alpha_{n_\omega}(t_i) = \sum_{i=0}^{n_\omega - 2} \frac{1}{n_\omega - i} = \sum_{i=2}^{n_\omega} \frac{1}{i}$$

which is approximately $ln(n_\omega)$.

Corollary 3.5.3. *Let the assumptions and definitions of Corollary 3.5.1 hold. Assume that the objective function $\Phi(\cdot)$ in* (2.1a) *and all constraints $c_i(\cdots)$ in* (2.1f) *are continuous functions. Then for any $\delta > 0$ there exists a grid size Δt such that*

$$|\Phi(x(t_f)) - \Phi(y(t_f))| \leq \delta, \tag{3.18}$$

$$|c_i(x(t), u^*(t)) - c_i(y(t), u^*(t))| \leq \delta, \quad i = 1, \dots, n_c. \tag{3.19}$$

Proof. Follows directly from the definition of continuity, $e^{Lt} \leq e^{Lt_\mathrm{f}}$ for all $t \in [0, t_\mathrm{f}]$, and (3.17).□

Remark 3.5.4. *For "first discretize, then optimize" methods that discretize $\alpha(\cdot)$ and $u(\cdot)$ by means of differentiable basis functions the assumptions of Corollary 3.5.3 are fulfilled. In particular there are only finitely many discontinuities in the optimal control $u^*(\cdot)$. The results can be transfered to more general problems than* (2.1)*. This is discussed in [214] and Chapter 2.*

Remark 3.5.5. *Note that Corollary 3.5.3 is not related to the issue of local or global optima. In fact, it holds for all feasible trajectories (x, α, u), hence also for globally and locally optimal trajectories. Naturally, the global lower bound for the integer problem can only be obtained when the relaxed problem is solved to global optimality, as discussed, e.g., in [63].*

The motivation for the estimation (3.17) was to obtain the exact lower bound for an optimal integer solution. But the result can also be interpreted in the sense of the Hausdorff distance between reachability sets.

Definition 3.5.6. *We define the Hausdorff distance between sets X and Y as*

$$d_{\mathrm{H}}(X,Y) = \max \left\{ \sup_{x \in X} \inf_{y \in Y} d(x,y), \sup_{y \in Y} \inf_{x \in X} d(x,y) \right\}.$$

The reachable set Y is defined as the set of all differential states $z \in \mathbb{R}^{n_x}$ for which a control function $\omega : [0,t_f] \mapsto \{0,1\}^{n_\omega}$ exists such that (2.2c) *holds and for a given function $u^*(\cdot)$ and initial value x_0 the solution $y(\cdot)$ of the ordinary differential equation (2.2b, 2.1b) fulfills $y(t_f) = z$. The set X is defined accordingly by taking the convex hull of the feasible control values, $\alpha : [0,t_f] \mapsto [0,1]^{n_\omega}$.*

Corollary 3.5.7. *Let the assumptions of Corollary 3.5.1 hold. Then a positive constant c exists such that*

$$d_{\mathrm{H}}(X,Y) \leq c\Delta t.$$

Proof. $[0,1]^{n_\omega}$ is a relaxation of $\{0,1\}^{n_\omega}$, hence $Y \subseteq X$. For any given trajectory $(x, u^*, \alpha)(\cdot)$ corresponding to a point in X, a trajectory $(y, u^*, \omega)(\cdot)$ can be found such that

$$\| \, y(t_{\mathrm{f}}) - x(t_{\mathrm{f}}) \, \| \leq c\Delta t,$$

as was shown in Corollary 3.5.1. □

Corollary 3.5.7 improves the results in [123] in two ways. First it provides the better order Δt instead of $\sqrt{\Delta t}$. Secondly, it allows the inclusion of the SOS1 constraint (2.2c), which allows the application to more general functions that are nonlinear in the control function $v(\cdot)$.

3.6 Numerical example

The Sum Up Rounding Strategy has been successfully applied to various applications by now. See Chapter 7 or [202] for an online description of most of them. To illustrate theoretical properties and the effect of the rounding strategy we investigate an academic example. In the following we simplify notation by leaving the argument (t) away, where convenient.

We want to solve the following nonlinear MIOCP,

$$\begin{aligned}
\min_{x,v} \quad & x_2(t_\mathrm{f}) \\
\text{s.t.} \quad & \dot{x}_0 = -\frac{x_0}{\sin(1)} \sin(v_1) + (x_0 + x_1)\, v_2^2 + (x_0 - x_1)\, v_3^3, \\
& \dot{x}_1 = (x_0 + 2x_1)\, v_1 + (x_0 - 2x_1)\, v_2 + (x_0 + x_1)\, v_3 \\
& \qquad + (x_0 x_1 - x_2)\, v_2^2 - (x_0 x_1 - x_2)\, v_2^3, \\
& \dot{x}_2 = x_0^2 + x_1^2, \\
& x(0) = (0.5, 0.5, 0)^T, \\
& x_1 \geq 0.4, \\
& v \in \{(1,0,0), (0,1,0), (0,0,1)\}
\end{aligned} \tag{3.20}$$

with $t \in [t_0, t_\mathrm{f}] = [0,1]$. This problem can be relaxed by requiring

$$\omega_1, \omega_2, \omega_3 \in [0,1], \quad \sum_{i=1}^{3} \omega_i = 1$$

instead of

$$v \in \{(1,0,0), (0,1,0), (0,0,1)\}.$$

We denote the solution of this relaxed problem with $(x^\mathrm{N}, \omega^\mathrm{N})$ to stress the nonlinear character. This relaxation naturally gives a lower bound, however the gap between this bound and integer solutions may be quite large.

The tightest relaxation is obtained, if an *outer convexification* of the integer components is applied. This results in the optimization problem

$$\begin{aligned}
\min_{x,\omega} \quad & x_2(t_\mathrm{f}) \\
\text{s.t.} \quad & \dot{x}_0 = -x_0\, \omega_1 + (x_0 + x_1)\, \omega_2 + (x_0 - x_1)\, \omega_3, \\
& \dot{x}_1 = (x_0 + 2x_1)\, \omega_1 + (x_0 - 2x_1)\, \omega_2 + (x_0 + x_1)\, \omega_3, \\
& \dot{x}_2 = x_0^2 + x_1^2, \\
& x(0) = (0.5, 0.5, 0)^T, \\
& x_1 \geq 0.4, \\
& \omega_i \in \{0,1\}, \quad \sum_{i=1}^{3} \omega_i = 1
\end{aligned} \tag{3.21}$$

with $t \in [t_0, t_\mathrm{f}] = [0,1]$. Note that this problem is (by construction) identical to the one investigated in [233] and originally in [82]. The only difference is the path constraint

$$x_1(t) \geq 0.4 \qquad t \in [t_0, t_\mathrm{f}] \tag{3.22}$$

that has been added to make the problem more interesting for our purposes. The relaxation of optimization problem (3.21) is obtained by replacing $\omega_i \in \{0,1\}$ by its convex hull $\omega_i \in [0,1]$. We denote the solution of this relaxation by $(x^{\mathrm{R}}, \omega^{\mathrm{R}})$ and the solution obtained with Sum Up Rounding by $(x^{\mathrm{SUR}}, \omega^{\mathrm{SUR}})$.

We solve all relaxed problems using the direct multiple shooting [44] based software package MUSCOD-II for different equidistant control discretization intervals. Figure 3.1 shows trajectories $(x^{\mathrm{N}}, \omega^{\mathrm{N}})$ as the solution of the relaxed nonlinear problem, $(x^{\mathrm{R}}, \omega^{\mathrm{R}})$ as the solution of the relaxed convexified problem, and $(x^{\mathrm{SUR}}, \omega^{\mathrm{SUR}})$ of the Sum Up Rounding solution obtained from ω^{R}. All depicted solutions are based on a control discretization with 80 equidistant time intervals.

In Table 3.1 objective function and infeasibility values for different grid sizes are given. The number of equidistant intervals m listed in the first column determines the interval length Δt as $t_{\mathrm{f}} = 1$ divided by m. The second and third columns show the objective function values of the relaxations of (3.20) and (3.21), denoted by $x_2^{\mathrm{N}}(t_{\mathrm{f}})$ and $x_2^{\mathrm{R}}(t_{\mathrm{f}})$, respectively.

The fourth column shows the objective function value $x_2^{\mathrm{SUR}}(t_{\mathrm{f}})$ obtained by applying the Sum Up Rounding strategy (3.7, 3.11) to the relaxed solution ω^{R}. The fifth column "infeasibility" contains the norm of the constraint violation of $x^{\mathrm{SUR}}(\cdot)$, which is the norm of the discretized path constraint vector corresponding to constraint (3.22).

The relaxed problems are only solved until a certain criterion on the progress in objective function values is fulfilled, in our case at $m = 80$. For all finer discretizations this solution is used for the SUR strategy (3.7, 3.11) in the interest of comparability of the objective function values. We define $x_2^*(t_{\mathrm{f}})$ to be the value of $x_2^{\mathrm{R}}(t_{\mathrm{f}}) = 0.995569$ for $m = 80$, as a sufficiently fine approximation of the infinite dimensional control problem. In the right-most column we list the deviation of $x_2^{\mathrm{SUR}}(t_{\mathrm{f}})$ from this value.

As can be observed, there is a linear dependence of both constraints and objective function value on the control grid size, as stated by Corollary 3.5.3. The deviation is not deterministic and especially for small m outliers are possible within the range of the bounds, but the asymptotic behavior can be clearly seen as m doubles from row to row. It can also be seen the gap between $x_2^{\mathrm{N}}(t_{\mathrm{f}})$ and the SUR integer solutions (which of course give the same objective function value for problem (3.20) as for problem (3.21)) is large, whereas it goes to zero with respect to $x_2^{\mathrm{R}}(t_{\mathrm{f}})$.

Looking again at Figure 3.1 we would like to stress that the SUR strategy needs to be applied to the solution of the relaxation of the (partially) convexified problem (3.21) and not of (3.20). If we apply it to the latter for $m = 80$ the objective function value would only be 1.108835 instead of 1.011600, and no theoretical guarantee can be given.

The discretization has been bisected for illustrative purposes. In practice more advanced adaptive schemes are used that neglect bang-bang arcs and take the goal to obtain approximate integral values into account, see [203]. The computational effort is low compared to enumerative schemes, such as Branch and Bound. In every step only a relaxed optimization problem has to be solved. The rounding procedure is almost for free and then a simple forward simulation has to be performed. The relaxed solution on a coarse grid is used to initialize the optimization variables

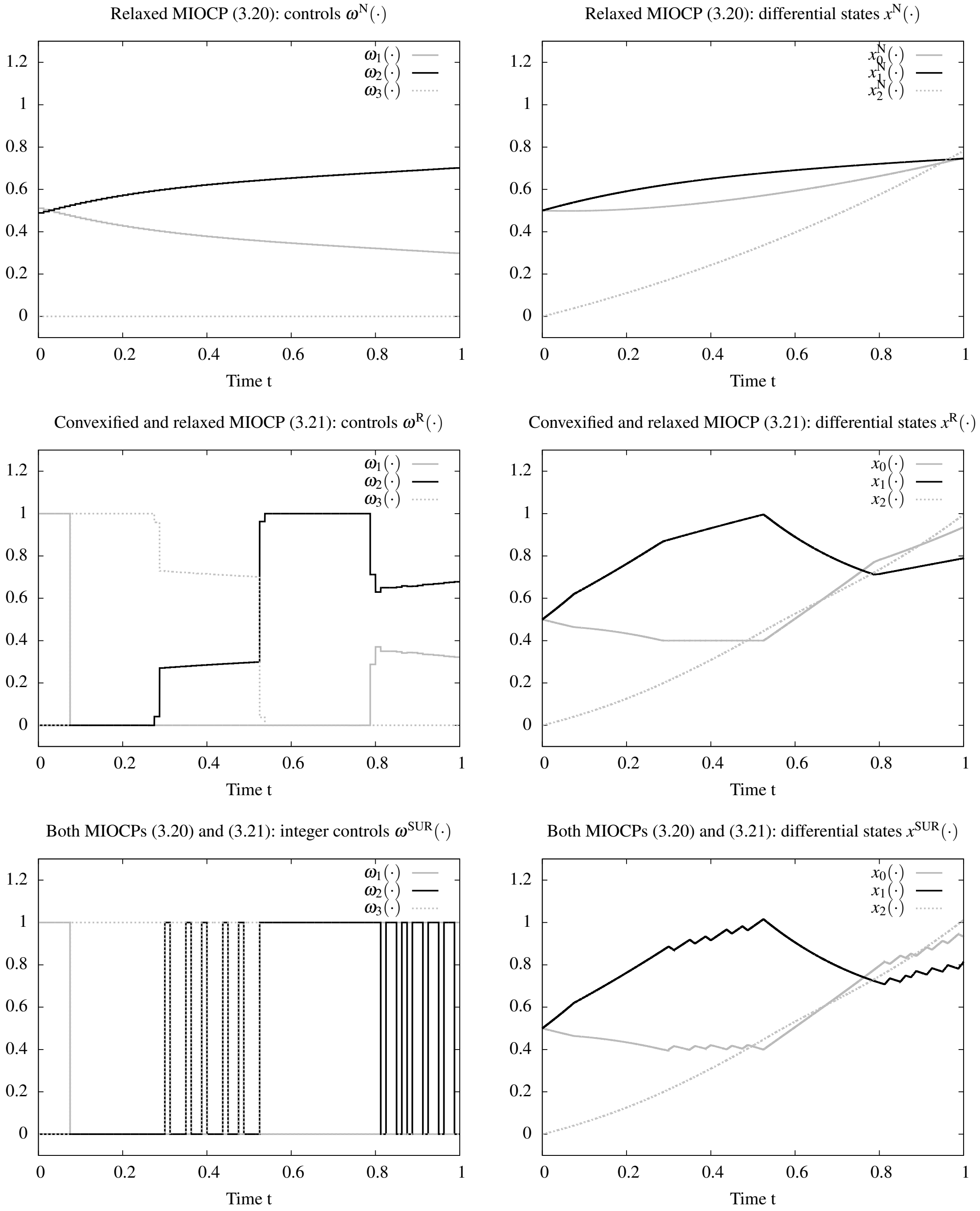


Figure 3.1: Different trajectories for $m = 80$ equidistant control intervals. First row: controls $v(\cdot)$ and states $x(\cdot)$ as the solution of the relaxation of problem (3.20). Second row: solution of the relaxation of convexified problem (3.21). Third row: the Sum Up Rounding solution, identical for both problems (3.20) and (3.21). Note the path-constrained arc for $x_1 \geq 0.4$ in row 4 at $t \approx [0.3, 0.5]$ and the constraint violation for the SUR solution.

m	$x_2^{\text{N}}(t_{\text{f}})$	$x_2^{\text{R}}(t_{\text{f}})$	$x_2^{\text{SUR}}(t_{\text{f}})$	infeasibility	$x_2^{\text{SUR}}(t_{\text{f}}) - x_2^*(t_{\text{f}})$
10	0.782278	0.999869	1.120181	6.30E-02	0.124612
20	0.782219	0.997646	1.132580	3.72E-02	0.137011
40	0.782204	0.995621	1.028741	1.45E-02	0.033172
80	0.782200	0.995569	1.011600	6.49E-03	0.016031
160	-	-	1.004031	3.26E-03	0.008462
320	-	-	1.000119	1.75E-03	0.004550
640	-	-	0.997933	8.19E-04	0.002364
1280	-	-	0.996706	4.61E-04	0.001137
2560	-	-	0.996154	2.03E-04	0.000585

Table 3.1: Numerical results for Egerstedt example.

on the finer grid, leading to fast convergence. An additional benefit of this approach is the fact that all previously calculated solutions can be stored and compared a posteriori to compare the trade off between frequent switching and a loss in the objective function.

3.7 Summary

We presented theoretical results with applications in mixed-integer nonlinear optimal control.
First, a novel proof was given that a trajectory with the strong property of integer feasibility exists that approximates the optimal relaxed solution arbitrarily close. Compared to previous studies it could be shown that a finite number of switches suffices.
Second, the role of the Sum Up Rounding strategy to obtain integer controls from continuous, relaxed ones, has been clarified. Previously, it has been described as a heuristic, similar to rounding methods in integer programming. We showed that it yields a constructive way to obtain an integer solution with a guaranteed bound on the performance loss, depending on the control discretization grid.
Third, we obtain an estimate of the Hausdorff distance between reachable sets. We improved previously known results in the sense that the approximation order is linear in the grid size Δt instead of the previously known result with order $\sqrt{\Delta t}$ [123], that we are able to include an SOS1 condition which allows for a transfer of the results to a more general, multi-dimensional and nonlinear case compared to the Theorems in [123, 190]. Hence, we proved Vladimir Veliov's conjecture [241], however with the additional assumption of differentiability.

4 Block Structured Quadratic Programming

The contents of this chapter are based on the paper

[145] C. Kirches, H.G. Bock, J.P. Schlöder, S. Sager. Block Structured Quadratic Programming for the Direct Multiple Shooting Method for Optimal Control. *Optimization Methods and Software*, 2010, Vol. 26(2):239–257.

Chapter Summary. We address the efficient solution of optimal control problems of dynamic processes with many controls. Such problems arise, e.g., from the outer convexification of integer control decisions. We treat this optimal control problem class using the direct multiple shooting method to discretize the optimal control problem. The resulting nonlinear problems are solved using sequential quadratic programming methods. We review the classical condensing algorithm that preprocesses the large but structured quadratic programs to obtain small but dense ones. We show that this approach leaves room for improvement when applied in conjunction with outer convexification. To this end, we present a new complementary condensing algorithm for quadratic programs with many controls. This algorithm is based on a hybrid null–space range–space approach to exploit the block structure of the quadratic programs that is due to direct multiple shooting. An assessment of the theoretical run time complexity reveals significant advantages of the proposed algorithm. We give a detailed account on the required number of floating point operations, depending on the process dimensions. Finally we demonstrate the merit of the new complementary condensing approach by comparing the behavior of both methods for a vehicle control problem in which the integer gear decision is convexified.

4.1 Introduction

Our main motivation are mixed-integer optimal control problems (MIOCPs) in ordinary differential equations (ODE) that are of the form (2.1), compare page 8. See Chapter 2 for a generic introduction to MIOC and the relation to other approaches in MIOC.

Direct methods, in particular *all–at–once approaches*, [34, 44, 191], have become the methods of choice for most practical optimal control problems. By direct method we refer to methods that *discretize first, then optimize* and work directly on the optimality conditions of the discretized control problem. *Indirect methods* for optimal control are methods that *optimize first, then discretize* by applying necessary conditions of optimality in function space, and then solving the control problem indirectly by solving the resulting boundary value problem numerically.

The drawback of direct methods with binary control functions obviously is that they lead to high–dimensional vectors of binary variables. Because of the exponentially growing complexity of the problem, techniques from mixed–integer nonlinear programming work only for small instances [238]. In Chapters 2 and 3 and past contributions [147, 203, 214] we proposed to use an *outer convexification* with respect to the binary controls, which has several main advantages over standard formulations or convexifications. A number of challenging mixed–integer optimal control problems has already been solved with this approach, see Chapter 7.

From the direct approach discretization of the MIOCP that is applied after outer convexification of the integer control, a highly structured Nonlinear Program (NLP) is obtained. For its solution, both Sequential Quadratic Programming (SQP) methods [34, 44, 191] and Interior Point (IP) methods [243] have become popular. In this contribution we consider SQP methods exclusively, which solve a sequence of Quadratic Programs (QPs) to obtain the NLP's solution and thus those of the discretized OCP. Here, the outer convexification approach results in QPs with *many control parameters*, one per possible discrete choice per discretization point in time.

Concerning the efficient solution of these QPs, active set methods are favored over interior point methods in an SQP context. This is due to the better performance of active set QP methods on a sequence of closely related QPs [24, 89]. Efficient exploitation of the problem structure found in the QP data is crucial for the efficiency of the QP solving procedure. Two possible approaches are thinkable here. First, in a preprocessing step the QP may be subjected to a reformulation that is tailored to the structures introduced by the discretization method. The classical *condensing* algorithm [191, 44] is reviewed here. Second, an active set QP code that directly exploits the block structure may be designed. This still is an active field of research, cf. [23, 131], where the difficulties lie with the efficient factorization of the QP's structured KKT system. In [229] a family of structure exploiting factorizations for systems arising in direct optimal control is studied systematically. Therein, the presented techniques are applied to KKT systems arising in interior point methods.

In this contribution, we present a new approach for solving QPs with block structure due to *direct multiple shooting*, named *complementary condensing*, based on prior work by [229]. This approach was first introduced in [144] and provides a factorization of the QP's KKT system tailored to the direct multiple shooting block structure. An evaluation of an ANSI C implementation of this approach and a comparison to classical condensing are presented for the first time. A detailed analysis of the required number of floating point operations is made, depending on the process dimensions. We apply the new complementary condensing approach for the first time to a vehicle control problem due to [105, 106] in which the integer gear decision is convexified as first proposed for this problem in [147]. We compare the obtained run times to the performance of the classical condensing algorithm, using the dense active set QP code QPOPT [111] for solving the condensed QPs. Classical condensing is shown to leave room for improvement if the QP has more control parameters than system states, as is the case for MIOCPs. In addition, we apply a general–purpose sparse symmetric indefinite factorization to the QP's KKT system, using the

HSL code MA57 [79]. We comment on the obtained run times and on numerical stability issues.

This chapter is structured as follows. Section 4.2 describes *direct multiple shooting* as our method of choice for discretizing the OCP and presents the structure of the block structured NLP. We briefly mention SQP methods and motivate the source of the QPs with block structure and many control parameters, to be dealt with in the two following sections. Section 4.3 reviews the classical condensing algorithm that reduces the large and block structured QP to a small but dense one in what can be seen as a preprocessing step. This condensed QP may be solved with any available QP code. Section 4.4 presents a the complementary condensing method as a new alternative approach at solving the block structured QPs. It exploits the block structure inside the QP solver. Section 4.5 describes an exemplary mixed–integer vehicle control problem with gear shift. The integer gear choice is treated by outer convexification, which is briefly explained and gives rise to an OCP with many controls. The classical condensing approach is applied to the example problem together with a dense active–set QP solver, and the resulting run times are discussed. Application of the proposed complementary condensing method to the example problem yields improved run times. A brief comparison to a sparse symmetric indefinite factorization of the KKT system as provided by HSL MA57 is made and run times as well as issues of numerical stability are discussed. In Section 4.6 further developments in block structured active set QP solving are discussed, in particular the possibility to further speed up the computations using update techniques. Section 4.7 summarizes this chapter.

4.2 Direct multiple shooting for optimal control

In this section we describe the direct multiple shooting method due to [191, 44] as an efficient tool for the discretization and parameterization of a general class of infinite dimensional optimal control problems (OCP). Using this method, we obtain from the OCP a highly structured NLP which we solve with an SQP method.
Details on Bock's direct multiple shooting method can be found in a number of recent publications and theses, e.g., in [168, 143].

4.2.1 Optimal control problem formulation

We consider the following general class (4.1) of optimal control problems

$$
\begin{aligned}
\min_{x(\cdot),u(\cdot)} \quad & l(x(\cdot),u(\cdot)) && && \text{(4.1a)}\\
\text{s.t.} \quad & \dot{x}(t) = f(t,x(t),u(t)) && \forall t \in \mathcal{T} && \text{(4.1b)}\\
& 0 \leq c(t,x(t),u(t)) && \forall t \in \mathcal{T} && \text{(4.1c)}\\
& 0 = r_i^{\text{eq}}(t_i,x(t_i)) && 0 \leq i \leq m && \text{(4.1d)}\\
& 0 \leq r_i^{\text{in}}(t_i,x(t_i)) && 0 \leq i \leq m && \text{(4.1e)}
\end{aligned}
$$

in which we strive to minimize objective function $l(\cdot)$ depending on the trajectory $x(\cdot)$ of a dynamic process described in terms of a system $f : \mathscr{T} \times \mathbb{R}^{n^{\mathrm{x}}} \times \mathbb{R}^{n^{\mathrm{u}}} \to \mathbb{R}^{n^{\mathrm{x}}}$ of ordinary differential equations (ODE), running on a time horizon $\mathscr{T} := [t_0, t_{\mathrm{f}}] \subset \mathbb{R}$, and governed by a control trajectory $u(\cdot)$ subject to optimization. The process trajectory $x(\cdot)$ and the control trajectory $u(\cdot)$ shall satisfy certain inequality path constraints $c : \mathscr{T} \times \mathbb{R}^{n^{\mathrm{x}}} \times \mathbb{R}^{n^{\mathrm{u}}} \to \mathbb{R}^{n^{\mathrm{c,in}}}$ on the time horizon $\mathscr{T}$, as well as equality and inequality point constraints $r_i^{\mathrm{eq}} : \mathscr{T} \times \mathbb{R}^{n^{\mathrm{x}}} \to \mathbb{R}^{n_i^{\mathrm{r,eq}}}$ and $r_i^{\mathrm{in}} : \mathscr{T} \times \mathbb{R}^{n^{\mathrm{x}}} \to \mathbb{R}^{n_i^{\mathrm{r,in}}}$ on a prescribed grid on $\mathscr{T}$ consisting of $m+1$ grid points

$$t_0 < t_1 < \ldots < t_{m-1} < t_m := t_{\mathrm{f}}, \qquad m \in \mathbb{N},\ m \geq 1. \tag{4.2}$$

In order to make this infinite dimensional optimal control problem computationally accessible, the direct multiple shooting method is applied to discretize the control trajectory $u(\cdot)$ subject to optimization.

4.2.2 The direct multiple shooting method

We introduce a discretization of the control trajectory $u(\cdot)$ by defining a *shooting grid*

$$t_0 < t_1 < \ldots < t_{m-1} < t_m := t_{\mathrm{f}}, \qquad m \in \mathbb{N},\ m \geq 1. \tag{4.3}$$

that shall be a superset of the constraint grid used in (4.1). For clarity, we assume in the following that the two grids coincide, though this is not a theoretical or algorithmic requirement. On each interval of the *shooting grid* (4.3) we introduce a vector $q_i \in \mathbb{R}^{n_i^{\mathrm{q}}}$ of *control parameters* together with an associated *control base function* $b_i : \mathscr{T} \times \mathbb{R}^{n_i^{\mathrm{q}}} \to \mathbb{R}^{n^{\mathrm{u}}}$ with local support,

$$u(t) := \textstyle\sum_{j=1}^{n_i^{\mathrm{q}}} b_{ij}(t, q_{ij}), \qquad t \in [t_i, t_{i+1}] \subseteq \mathscr{T},\ 0 \leq i \leq m-1. \tag{4.4}$$

The number and location of the shooting grid points and the choice of base functions obviously affects the approximation quality of the optimal solution of the discretized problem.

In addition, we introduce state vectors $s_i \in \mathbb{R}^{n^{\mathrm{x}}}$ in all shooting nodes serving as initial values for m IVPs

$$\dot{x}_i(t) = f(t, x_i(t), q_i), \quad x_i(t_i) = s_i, \qquad t \in [t_i, t_{i+1}] \subseteq \mathscr{T}, \quad 0 \leq i \leq m-1. \tag{4.5}$$

This parameterization of the process trajectory $x(\cdot)$ is in general discontinuous on $\mathscr{T}$. Continuity of the solution is ensured by introduction of additional *matching conditions*

$$x_i(t_{i+1};\ t_i, s_i, q_i) - s_{i+1} = 0, \qquad 0 \leq i \leq m-1, \tag{4.6}$$

where $x_i(t_{i+1};\ t_i, s_i, q_i)$ denotes the state trajectory's value $x_i(\cdot)$ in t_{i+1}, depending on the start time t_i, initial value s_i, and control parameters q_i on that interval.

The path constraints of problem (4.1) are enforced on the nodes of the shooting grid (4.3) only.

While in general it can be observed that this formulation already leads to a solution that satisfies the path constrains on the whole of $\mathscr{T}$, methods from semi–infinite programming exist [193] to ensure this in a rigorous fashion. For clarity we define the combined constraint functions $r_i : \mathscr{T} \times \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}^{n^\mathrm{u}} \to \mathbb{R}^{n_i^\mathrm{r}}$,

$$0 \leqq r_i(t_i, s_i, q_i), \quad 0 \le i \le m-1, \qquad 0 \leqq r_m(t_m, s_m) \tag{4.7}$$

with $n_i^\mathrm{r} := n^\mathrm{c} + n_i^{\mathrm{r,eq}} + n_i^{\mathrm{r,in}}$. These comprise all discretized path constraints as well as equality and inequality point constraints.

The objective function $l(x(\cdot), u(\cdot))$ shall be separable with respect to the shooting grid structure. In general, $l(\cdot)$ is a Mayer type function evaluated at the end of the horizon $\mathscr{T}$, or Lagrange type integral objective evaluated on the whole of $\mathscr{T}$. For both types, a separable formulation is easily found,

$$l(x(\cdot), u(\cdot)) = M(s_m) \quad \text{or} \quad l(x(\cdot), u(\cdot)) = \textstyle\sum_{i=0}^{m-1} \displaystyle\int_{t_i}^{t_{i+1}} L_i(x_i(t), q_i)\, \mathrm{d}t. \tag{4.8}$$

Summarizing, the discretized multiple shooting optimal control problem can be cast as a nonlinear problem

$$\begin{aligned} \min_{w} \quad & \textstyle\sum_{i=0}^{m} l_i(w_i) && && \text{(4.9a)} \\ \text{s.t.} \quad & 0 = x_i(t_{i+1};\, t_i, w_i) - s_{i+1}, && 0 \le i \le m-1 && \text{(4.9b)} \\ & 0 \leqq r_i(w_i), && 0 \le i \le m && \text{(4.9c)} \end{aligned}$$

with the vector of unknowns w being

$$\begin{aligned} w &:= (s_1, q_1, \ldots, s_{m-1}, q_{m-1}, s_m), && \text{(4.10a)} \\ w_i &:= (s_i, q_i),\ 0 \le i \le m-1, \qquad w_m := s_m, && \text{(4.10b)} \end{aligned}$$

where the evaluation of the matching condition constraint (4.9b) requires the solution of an initial value problem (4.5).

4.2.3 Structured quadratic subproblem

Sequential Quadratic Programming (SQP) methods are a long–standing and highly effective method for the solution of NLPs that also allow for much flexibility in exploiting the problem's special structure. First introduced by [124, 194], SQP methods iteratively progress towards a KKT point of the NLP by solving a linearly constrained local quadratic model of the NLP's Lagrangian [181]. For the NLP (4.9) arising from direct multiple shooting, this local quadratic model reads

$$\begin{aligned} \min_{\delta w} \quad & \tfrac{1}{2}\textstyle\sum_{i=0}^{m} \delta w_i' B_i \delta w_i + g_i' \delta w_i && && \text{(4.11a)} \\ \text{s.t.} \quad & 0 = X_i(w_i)\delta w_i - \delta s_{i+1} - h_i(w_i), && 0 \le i \le m-1, && \text{(4.11b)} \\ & 0 \leqq R_i(w_i)\delta w_i - r_i(w_i), && 0 \le i \le m, && \text{(4.11c)} \end{aligned}$$

with the following notations for vector of unknowns δw and its components

$$\delta w := (\delta s_1, \delta q_1, \dots, \delta s_{m-1}, \delta q_{m-1}, \delta s_m), \tag{4.12a}$$

$$\delta w_i := (\delta s_i, \delta q_i),\ 0 \le i \le m-1, \qquad \delta w_m := \delta s_m, \tag{4.12b}$$

similar to the notation used in (4.10a), and with vectors h_i denoting the matching conditions residuals,

$$h_i(w_i) := x_i(t_{i+1};\, t_i, w_i) - s_{i+1}. \tag{4.13}$$

The matrices B_i denote the node Hessians of the NLP's Lagrangian, or suitable approximations, cf. [44]. The vectors g_i denotes the node gradients of the NLP's objective function. Matrices X_i, R_i^{eq}, and R_i^{in} denote linearizations of the constraint functions obtained in w_i,

$$B_i \approx \frac{\mathrm{d}^2 l_i(w_i)}{\mathrm{d}w_i^2}, \qquad g_i := \frac{\mathrm{d}l_i(w_i)}{\mathrm{d}w_i}, \tag{4.14a}$$

$$R_i := \frac{\mathrm{d}r_i(w_i)}{\mathrm{d}w_i}, \qquad X_i := \frac{\partial x_i(t_{i+1};\, t_i, w_i)}{\partial w_i}. \tag{4.14b}$$

In particular, the computation of the *sensitivity matrices* X_i requires the computation of derivatives of the solution of IVP (4.5) with respect to the initial values w_i. To ensure consistency of the derivatives, this should be done according to the principle of *internal numerical differentiation* (IND) [7, 39], i.e. by computing nominal solution and its derivatives using the same discretization scheme.

4.3 Condensing to obtain a dense quadratic problem

In order to solve the QP (4.11) efficiently, one has to take advantage of its block structure that is due to multiple shooting. In view of the widespread availability and reliable performance of active–set QP codes, an obvious choice is to employ one of these solvers for that purpose. System (4.11) does not suit the majority of codes, though. They either do not exploit sparsity in the QP data, i.e. they are dense solvers [111, 218], or do exploit sparsity at a general–purpose level by employing linear algebra working on specially shaped dense data [23, 111], where the shape assumptions are not fulfilled by QP (4.11). Generic sparse data in triplets or column–compressed format is commonly accepted by interior–point QP solvers only, cf. [240, 109], which are not ideally suited for employment inside an SQP method. Only recently, some progress towards a general–purpose sparse active set solver has been made as presented in [131].

The block structure in QP (4.11) therefore is exploited in a preprocessing or *condensing* step that transforms the QP into a related, considerably smaller, and densely populated one. In this section we briefly review a condensing algorithm due to [191, 44] and presented to great detail in [168].

4.3.1 Reordering the structured quadratic problem

We start by reordering the constraint matrix of QP (4.11) to separate the additionally introduced node values $\delta v = (\delta s_1, \dots, \delta s_m)$ from the single shooting values $\delta u = (\delta s_0, \delta q_0, \dots, \delta q_{m-1})$ as shown below,

$$\left(\begin{array}{ccccc|ccccc}
X_0^{\mathrm{s}} & X_0^{\mathrm{q}} & & & & -I & & & & \\
 & & X_1^{\mathrm{q}} & & & X_1^{\mathrm{s}} & -I & & & \\
 & & & \ddots & & & \ddots & \ddots & & \\
 & & & & X_{m-1}^{\mathrm{q}} & & & & X_{m-1}^{\mathrm{s}} & -I \\
\hline
R_0^{\mathrm{s}} & R_0^{\mathrm{q}} & & & & & & & & \\
 & & R_1^{\mathrm{q}} & & & R_1^{\mathrm{s}} & & & & \\
 & & & \ddots & & & \ddots & & & \\
 & & & & R_{m-1}^{\mathrm{q}} & & & & R_{m-1}^{\mathrm{s}} & \\
 & & & & & & & & & R_m^{\mathrm{s}}
\end{array}\right). \tag{4.15}$$

4.3.2 Elimination using the matching conditions

We may now use the negative identity matrix blocks of the equality matching conditions as pivots to formally eliminate the additionally introduced multiple shooting state values $(\delta s_1, \dots, \delta s_m)$ from system (4.15), analogous to the usual Gaussian elimination method for triangular matrices. This elimination procedure was introduced in [44] and a detailed presentation can be found in

[168]. From this elimination procedure the dense constraint matrix

$$\begin{pmatrix} \overline{X} & -I \\ \overline{R} & 0 \end{pmatrix} = \left(\begin{array}{ccccc|ccc} X_0^{\mathrm{s}} & X_0^{\mathrm{q}} & & & & -I & & \\ X_1^{\mathrm{s}} X_0^{\mathrm{s}} & X_1^{\mathrm{s}} X_0^{\mathrm{q}} & X_1^{\mathrm{q}} & & & & -I & \\ \vdots & \vdots & \vdots & \ddots & & & & \ddots \\ \Pi_0^{m-1} & \Pi_1^{m-1} X_0^{\mathrm{q}} & \Pi_2^{m-1} X_1^{\mathrm{q}} & \cdots & X_{m-1}^{\mathrm{q}} & & & -I \\ \hline R_0^{\mathrm{s}} & R_0^{\mathrm{q}} & & & & & & \\ R_1^{\mathrm{s}} X_0^{\mathrm{s}} & R_1^{\mathrm{s}} X_0^{\mathrm{q}} & R_1^{\mathrm{q}} & & & & & \\ \vdots & \vdots & \vdots & \ddots & & & & \\ R_m^{\mathrm{s}} \Pi_0^{m-1} & R_m^{\mathrm{s}} \Pi_1^{m-1} X_0^{\mathrm{q}} & R_m^{\mathrm{s}} \Pi_2^{m-1} X_1^{\mathrm{q}} & \cdots & R_m^{\mathrm{s}} X_{m-1}^{\mathrm{q}} & & & \end{array}\right). \tag{4.16}$$

is obtained, with sensitivity matrix products Π_j^k defined to be

$$\Pi_j^k := \prod_{l=j}^{k} X_l^{\mathrm{s}},\ 0 \le j \le k \le m-1, \quad \Pi_j^k := I,\ j > k. \tag{4.17}$$

From (4.16) we deduce that, after this elimination step, the transformed QP in terms of the two unknowns δu and δv reads

$$\min_{\delta u, \delta v} \quad \frac{1}{2} \begin{pmatrix} \delta u \\ \delta v \end{pmatrix}' \overbrace{\begin{pmatrix} \overline{B}_{11} & \overline{B}_{12} \\ \overline{B}_{12}' & \overline{B}_{22} \end{pmatrix}}^{=B} \begin{pmatrix} \delta u \\ \delta v \end{pmatrix} + \overbrace{\begin{pmatrix} \overline{g}_1 \\ \overline{g}_2 \end{pmatrix}'}^{=g'} \begin{pmatrix} \delta u \\ \delta v \end{pmatrix} \tag{4.18a}$$

$$\text{s.t.} \quad 0 = \overline{X}\delta u - I\delta v - \overline{h} \tag{4.18b}$$

$$0 \leqq \overline{R}\delta u - \overline{r} \tag{4.18c}$$

with appropriate right hand side vectors $\overline{h}$ and $\overline{r}$ obtained by applying the Gaussian elimination steps to h and r.

4.3.3 Reduction to a single shooting sized system

System (4.18) easily lends itself to the elimination of the unknown δv. By this step we arrive at the final *condensed QP*

$$\min_{\delta u} \quad \tfrac{1}{2}\delta u' \overline{\overline{B}} \delta u + \overline{\overline{g}}' \delta u \tag{4.19a}$$

$$\text{s.t.} \quad 0 \le \overline{R}\delta u - \overline{r} \tag{4.19b}$$

with the following dense Hessian matrix and gradient obtained from substitution of δv in the objective (4.18a)

$$\overline{\overline{B}} = \overline{B}_{11} + \overline{B}_{12}\overline{X} + \overline{X}'\overline{B}'_{12} + \overline{X}'\overline{B}_{22}\overline{X}, \tag{4.20a}$$

$$\overline{\overline{g}} = \overline{g}_1 + \overline{X}'\overline{g}_2 - \overline{B}'_{12}\overline{h} - \overline{X}'\overline{B}_{22}\overline{h} \tag{4.20b}$$

The matrix multiplications required for the computation of these values are easily laid out to exploit the block structure of $\overline{X}$ and B. In addition, from the elimination steps of sections 4.3.2 and 4.3.3 one obtains relations that allow to recover $\delta v = (\delta s_1, \dots, \delta s_m)$ from the solution $\delta u = (\delta s_0, \delta q_0, \dots, \delta q_{m-1})$ of the condensed QP (4.19).

4.3.4 Solving the condensed quadratic problem

As the resulting condensed QP (4.19) no longer has a multiple shooting specific structure, it may be solved using any standard dense method for quadratic programming, which is what condensing aims for. Popular codes are the null space method QPSOL, available as subroutine E04NAF in the NAG library, and its successor QPOPT [111], available as subroutine E04NFF. An efficient code for parametric quadratic programming is qpOASES [89]. Further active set codes such as the Schur complement code QPSchur [23] and the QPKWIK code [218] exist. The primal–dual null–space solver BQPD [91] is also able to exploit sparsity remaining in the condensed QP to a certain extent. An extensive bibliography of existing QP methods and codes can be found in [116].

4.4 Block structured quadratic programming: “complementary condensing”

In this section we present a new approach of solving the KKT system of a QP with block structure due to multiple shooting that is suited for embedding in a standard active–set loop. This approach is based on related work by [229] and was first presented in [144]. It does not work as a preprocessing step but directly exploits the block structure inside the solver. We derive in detail the necessary elimination steps that shall ultimately retain the duals of the matching condition equalities only. In classical condensing, these were used for elimination, which gives rise to the name *complementary condensing* for our new method. An analysis of the run time complexity as well as a detailed account on the number of floating point operations spent in the various parts of the algorithm is presented.

4.4.1 The KKT system's block structure

For a given active set, the KKT system of the QP (4.11) to be solved for the primal step δw_i and the dual step $(\delta\lambda, \delta\mu)$ reads for $0 \le i \le m$

$$\begin{aligned} P_i'\delta\lambda_{i-1} + B_i(-\delta w_i) + R_i'\delta\mu_i + X_i'\delta\lambda_i &= B_i w_i + g_i &&=: \bar{g}_i, && (4.21a)\\ R_i(-\delta w_i) &= R_i w_i - r_i &&=: \bar{r}_i, && (4.21b)\\ X_i(-\delta w_i) + P_{i+1}(-\delta w_{i+1}) &= X_i w_i + P_{i+1}s_{i+1} - h_i &&=: \bar{h}_i. && (4.21c) \end{aligned}$$

with Lagrange multipliers $\delta\lambda \in \mathbb{R}^{n^{\mathrm{x}}}$ for the matching conditions (4.11b) and $\delta\mu \in \mathbb{R}^{n_i^{\mathrm{r}}}$ for the equality point constraints and the active subset of the discretized inequality path and point constraints (4.11c). The projection matrices P_i are defined as

$$P_i := \begin{pmatrix} -I & 0 \end{pmatrix} \in \mathbb{R}^{n^{\mathrm{x}}\times(n^{\mathrm{x}}+n^{\mathrm{q}})},\ 1 \le i \le m, \tag{4.22}$$

and for the first and last shooting nodes as

$$P_0 := 0 \in \mathbb{R}^{n^{\mathrm{x}}\times(n^{\mathrm{x}}+n^{\mathrm{q}})}, \qquad P_{m+1} := 0 \in \mathbb{R}^{n^{\mathrm{x}}\times n^{\mathrm{x}}}. \tag{4.23}$$

In the following, all matrices and vectors are assumed to comprise the components of the active set only. To avoid the need for repeated special treatment of the first and last shooting node, we introduce the following conventions that make equation (4.21) hold also for the border cases $i = 0$ and $i = m$:

$$\begin{aligned} &\delta\lambda_{-1} := 0 \in \mathbb{R}^{n^{\mathrm{x}}}, && \lambda_{-1} := 0 \in \mathbb{R}^{n_{\mathrm{x}}}, && \delta\lambda_m := 0 \in \mathbb{R}^{n^{\mathrm{x}}}, && \lambda_m := 0 \in \mathbb{R}^{n^{\mathrm{x}}}, && (4.24a)\\ &\delta w_{m+1} := 0 \in \mathbb{R}^{n^{\mathrm{x}}}, && w_{m+1} := 0 \in \mathbb{R}^{n^{\mathrm{x}}}, && h_m := 0 \in \mathbb{R}^{n^{\mathrm{x}}}, && X_m := 0 \in \mathbb{R}^{n^{\mathrm{x}}\times n^{\mathrm{x}}}. && (4.24b) \end{aligned}$$

As our presentation focuses on a single block of the KKT system only, we omit the subscript index i where unambiguous. System (4.21) can also be put in matrix form,

$$\begin{pmatrix} B_0 & R_0' & X_0' & & & & & & & \\ R_0 & & & & & & & & & \\ X_0 & & & P_1 & & & & & & \\ & & P_1' & B_1 & R_1' & X_1' & & & & \\ & & & R_1 & & & & & & \\ & & & X_1 & & & & & & \\ & & & & & & \ddots & & & \\ & & & & & & & B_m & R_m' \\ & & & & & & & R_m & \end{pmatrix} \begin{pmatrix} -\delta w_0 \\ \delta\mu_0 \\ \delta\lambda_0 \\ -\delta w_1 \\ \delta\mu_1 \\ \delta\lambda_1 \\ \vdots \\ -\delta w_m \\ \delta\mu_m \end{pmatrix} = \begin{pmatrix} \bar{g}_0 \\ \bar{r}_0 \\ \bar{h}_0 \\ \bar{g}_1 \\ \bar{r}_1 \\ \bar{h}_1 \\ \vdots \\ \bar{g}_m \\ \bar{r}_m \end{pmatrix} \tag{4.25}$$

4.4.2 Hessian projection step

Under the assumption that the number of active point constraints does not exceed the number of unknowns (i.e. the active set is not degenerate), we can perform QR decompositions of the linearized point constraints matrix R,

$$RQ = \begin{pmatrix} R^{\mathrm{R}'} & 0 \end{pmatrix}, \qquad Q =: \begin{pmatrix} Y & Z \end{pmatrix}. \tag{4.26}$$

We remind the reader that the subscript index i denoting the node is omitted to improve readability. Here Q are a unitary matrices and R^{R} are upper triangular. We partition δw into its range space part δy and its null space part δz, where the identity $\delta w = Y\delta y + Z\delta z$ holds. We find δy from the range space projection of (4.21b)

$$R(-\delta w) = -R^{\mathrm{R}}\delta y = \bar{r}. \tag{4.27}$$

We transform the remainder of the KKT system onto the null space of R by substituting $Y\delta y + Z\delta z$ for δw and solving for δz. First, we find for the matching conditions (4.21c)

$$-XZ\delta z - P_{i+1}Z\delta z_{i+1} = \bar{h} + XY\delta y + P_{i+1}Y\delta y_{i+1} \tag{4.28}$$

which can be solved for δw_i^Z once δw_{i+1}^Z is known. Second, from the stationarity conditions (4.21a) we find

$$\begin{aligned} Z'P'\delta\lambda_{i-1} - Z'BZ\delta z + Z'R'\mu + Z'X'\delta\lambda &= Z'\bar{g} + Z'BY\delta y, &(4.29a)\\ Y'R'\delta\mu &= -Y'(B\delta w + P'\delta\lambda_{i-1} - X'\delta\lambda + \bar{g}). &(4.29b)\end{aligned}$$

Herein, $Z'R' = 0$ and $Y'R' = R^{\mathrm{R}}$. Equation (4.29a) can be solved for $\delta\lambda_i$ once δw_i and $\delta\lambda_{i-1}$ are known, while (4.29b) can be used to determine the point constraints multipliers $\delta\mu$. Let thus null space projections be defined as follows, where we use a tilde to distinguish them from their full-space counterparts:

$$\begin{aligned} &\tilde{B} := Z'BZ, \qquad \tilde{X} := XZ, \qquad \tilde{P} := PZ, &(4.30a)\\ &\tilde{g} := Z'(\bar{g} + B(Y\delta y)), \qquad \tilde{h} := \bar{h} + X(Y\delta y) + P_{i+1}(Y_{i+1}\delta y_{i+1}). &(4.30b)\end{aligned}$$

With this notation the projection of the KKT system on the null space of the point constraints can be read from equations (4.28) and (4.29a) for $0 \leq i \leq m-1$ as

$$\begin{aligned} \tilde{P}'\delta\lambda_{i-1} + \tilde{B}(-\delta z) + \tilde{X}'\delta\lambda &= \tilde{g}, &(4.31a)\\ \tilde{X}(-\delta z) + \tilde{P}_{i+1}(-\delta z_{i+1}) &= \tilde{h}. &(4.31b)\end{aligned}$$

This again can be put in matrix form as

$$\begin{pmatrix} \tilde{B}_0 & \tilde{X}_0' & & & & \\ \tilde{X}_0 & & \tilde{P}_1 & & & \\ & \tilde{P}_1' & \tilde{B}_1 & \tilde{X}_1' & & \\ & & \tilde{X}_1 & & & \\ & & & & \ddots & \tilde{P}_m \\ & & & & \tilde{P}_m' & \tilde{B}_m \end{pmatrix} \begin{pmatrix} -\delta z_0 \\ \delta\lambda_0 \\ -\delta z_1 \\ \delta\lambda_1 \\ \vdots \\ -\delta z_m \end{pmatrix} = \begin{pmatrix} \tilde{g}_0 \\ \tilde{h}_0 \\ \tilde{g}_1 \\ \tilde{h}_1 \\ \vdots \\ \tilde{g}_m \end{pmatrix}. \tag{4.32}$$

4.4.3 Schur complement step

In (4.32) the elimination of δz is possible using a Schur complement step, provided that the reduced Hessians $\tilde{B}$ are positive definite. We find from (4.31a)

$$(-\delta z) = \tilde{B}^{-1}(\tilde{g} - \tilde{P}'\delta\lambda_{i-1} - \tilde{X}'\delta\lambda) \tag{4.33}$$

depending on the knowledge of $\delta\lambda$. Inserting into (4.31b) and collecting for $\delta\lambda$ yields

$$\begin{aligned} &\tilde{X}\tilde{B}^{-1}\tilde{P}'\delta\lambda_{i-1} + (\tilde{X}\tilde{B}^{-1}\tilde{X}' + \tilde{P}_{i+1}\tilde{B}_{i+1}^{-1}\tilde{P}_{i+1}')\delta\lambda + \tilde{P}_{i+1}\tilde{B}_{i+1}^{-1}\tilde{X}_{i+1}'\delta\lambda_{i+1} \\ =& -\tilde{h} + \tilde{X}\tilde{B}^{-1}\tilde{g} + \tilde{P}_{i+1}\tilde{B}_{i+1}^{-1}\tilde{g}_{i+1} \end{aligned} \tag{4.34}$$

With Cholesky decompositions $\tilde{B} = R^{\mathrm{B}\prime}R^{\mathrm{B}}$ we define the following symbols, where we use a hat to distinguish them from their full-space counterparts:

$$\hat{X} := \tilde{X}R^{\mathrm{B}^{-1}}, \qquad A := \tilde{X}\tilde{B}^{-1}\tilde{X}' + \tilde{P}_{i+1}\tilde{B}_{i+1}^{-1}\tilde{P}_{i+1}' \qquad = \hat{X}\hat{X}' + \hat{P}_{i+1}\hat{P}_{i+1}', \tag{4.35a}$$

$$\hat{P} := \tilde{P}R^{\mathrm{B}^{-1}}, \qquad B := \tilde{X}\tilde{B}^{-1}\tilde{P}' \qquad = \hat{X}\hat{P}', \tag{4.35b}$$

$$\hat{g} := R^{\mathrm{B}^{-T}}\tilde{g}, \qquad a := -\tilde{h} + \tilde{X}\tilde{B}^{-1}\tilde{g} + \tilde{P}_{i+1}\tilde{B}_{i+1}^{-1}\tilde{g}_{i+1} = -\tilde{h} + \hat{X}\hat{g} + \hat{P}_{i+1}\hat{g}_{i+1}. \tag{4.35c}$$

Equation (4.34) can be written in terms of these values for $0 \leq i \leq m-1$ as

$$B\delta\lambda_{i-1} + A\delta\lambda + B_{i+1}'\delta\lambda_{i+1} = a. \tag{4.36}$$

In matrix form, the remaining symmetric positive definite system reads

$$\begin{pmatrix} A_0 & B_1' & & \\ B_1 & A_1 & \ddots & \\ & \ddots & \ddots & B_{m-1}' \\ & & B_{m-1} & A_{m-1} \end{pmatrix} \begin{pmatrix} \delta\lambda_0 \\ \delta\lambda_1 \\ \vdots \\ \delta\lambda_{m-1} \end{pmatrix} = \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{m-1} \end{pmatrix}. \tag{4.37}$$

4.4.4 Solving the banded system

In the symmetric positive definite banded system (4.37), only the matching condition duals $\delta\lambda_i \in \mathbb{R}^{n_x}$ remain as unknowns. Since in classical condensing, exactly these matching conditions were used for elimination of a part of the primal unknowns, this new method is in a sense *complementary* to the classical condensing method. For optimal control problems with dimensions $n^q \geq n_x$, the presented approach obviously is computationally more favorable than retaining unknowns of dimension n^q. System (4.37) can be solved for $\delta\lambda$ by means of a tridiagonal block Cholesky decomposition [11] and two backsolves with the block Cholesky factors.

Once $\delta\lambda$ is known, the primal null space step δz can be recovered using equation (4.33). The full primal step δw is then obtained from $\delta w = Y\delta y + Z\delta z$. Finally, the decoupled point constraint multipliers step $\delta\mu$ can be recovered by insertion into (4.29b).

4.4.5 Computational complexity

In Table 4.1 a detailed list of the linear algebra operations required to carry out the individual steps of the complementary condensing method can be found. A clear distinction between matrix operations of quadratic or cubic runtime complexity on the one hand, and vector operations of linear or quadratic runtime complexity on the other hand has been made. The number of floating point operations required for the linear algebra operations, depending on the system dimensions $n = n_x + n^q$ and n_i^r, is given in Table 4.2. The numbers n^y and n^z with $n^y + n^z = n_i^r$ denote the range–space and null-space dimension resulting from the QR decomposition (4.26), respectively. All FLOP counts are given on a per shooting node basis. It's easy to see that the method's runtime complexity is $O(m)$, in sharp contrast to the classical condensing method, as the shooting grid length m does not appear as a dependency in Table 4.2. In addition, the run time of a significant part of the complementary condensing, the decomposition of the banded system (4.37), even is *independent* of the number n^q of discrete choices.

4.4.6 Pivoting

Both the classical condensing of Section 4.3 and the proposed complementary condesing KKT solver fix a part of the pivoting sequence. The question of improving numerical stability of the proposed method and its applicability to ill–conditioned systems therefore is of interest.

Several extensions are thinkable here. First, within the QR decomposition of the active decoupled point constraints one can of course make use of pivoting to improve the behavior for almost linear dependent active sets. Second, the Cholesky decompositions of both the null-space Hessian $\tilde{B}$ and the diagonal blocks of system (4.37) can employ pivoting strategies. Probably the most interesting strategy is a block pivoting strategy for the decomposition of system (4.37). Based on cheap estimates of the conditions of the blocks A_i and the norms of the blocks B_i, block row and column interchanges can be determined. The block fill-in can be shown to introduce at most one additional nonzero block subdiagonal.

	Matrix				Vector		
Action	dec	bs	mul	add	bs	mul	add
Decompose R_i	1	–	–	–			
Solve for δy, $Y\delta y$					1	1	–
Build $\tilde{B}_i$	–	–	2	–			
Build $\tilde{X}_i$, $\tilde{P}_i$	–	–	2	–			
Build $\tilde{g}_i$, $\tilde{h}_i$					–	4	3
Decompose $\tilde{B}_i$	1	–	–	–			
Build $\hat{X}_i$, $\hat{P}_i$	–	2	–	–			
Build A_i, B_i	–	–	3	1			
Build $\hat{g}_i$, a_i					1	2	2
Decompose (4.37)	1	1	1	–			
Solve for $\delta\lambda_i$					2	2	2
Solve for δz_i, $Z\delta z_i$					1	3	2
Solve for $\delta\mu_i$					1	4	3

Table 4.1: Number of matrix and vector operations per node required for the individual parts of the proposed block structured QP solver, separated into decompositions (dec), backsolves (bs), multiplications (mul), and additions (add).

4.5 Example: a vehicle control problem with gear shift

In this section we review a vehicle control problem that is due to [105, 106] as a test bed for both presented approaches to solving the block structured quadratic problems within a direct multiple shooting method for optimal control. The same benchmark problem is described in Section 7.9.

4.5.1 Vehicle model

We consider a single–track model of a vehicle as depicted in Figure 4.1 whose dynamics are modeled by a system of ordinary differential equations (ODEs) described in Section 7.9 with 7 states as briefly listed in Table 4.3. As this is a time optimal problem, an additional differential state representing the current time t is introduced for the transformation from a fixed time horizon $\tau \in [0,1]$ to the one of variable length $t \in [0,t_\mathrm{f}]$. The length t_f of the time horizon is a global model parameter subject to optimization, which for simplicity of the implementation is introduced as a constant differential state as well. Finally, together with the objective function of Lagrangian type, the problem has a total of $n_\mathrm{x} = 10$ differential states.

The driver, in our case the optimal control problem solver, exercises control over the steering wheel, the pedal, the brakes, and the choice of the gear, as listed in Table 4.4. A more exten-

Action	Floating point operations
Decompose R_i	$n_i^{\mathrm{r}2} n$
Solve for δy, $Y\delta y$	$n_i^{\mathrm{r}} n^{\mathrm{y}} + n^{\mathrm{y}} n$
Build $\tilde{B}_i$	$n^{\mathrm{z}2} n + n^{\mathrm{z}} n^2$
Build $\tilde{X}_i$, $\tilde{P}_i$	$2n_{\mathrm{x}} n^{\mathrm{z}} n$
Build $\tilde{g}_i$, $\tilde{h}_i$	$2n_{\mathrm{x}} n + n^{\mathrm{z}} n + n^2 + 2n_{\mathrm{x}} + n$
Decompose $\tilde{B}_i$	$\frac{1}{3} n^{\mathrm{z}3}$
Build $\hat{X}_i$, $\hat{P}_i$	$2n_{\mathrm{x}} n^{\mathrm{z}2}$
Build A_i, B_i	$3{n_{\mathrm{x}}}^2 n^{\mathrm{z}} + {n_{\mathrm{x}}}^2$
Build $\hat{g}_i$, a_i	$n^{\mathrm{z}2} + 2n_{\mathrm{x}} n^{\mathrm{z}} + 2n_{\mathrm{x}}$
Decompose (4.37)	$\frac{7}{3} {n_{\mathrm{x}}}^3$
Solve for $\delta\lambda_i$	$4{n_{\mathrm{x}}}^2 + 2n_{\mathrm{x}}$
Solve for δz_i, $Z\delta z_i$	$n^{\mathrm{z}2} + 2n_{\mathrm{x}} n^{\mathrm{z}} + n^{\mathrm{z}} n + 2n^{\mathrm{z}}$
Solve for $\delta\mu_i$	$n_i^{\mathrm{r}} n^{\mathrm{y}} + n^{\mathrm{y}} n + 2n_{\mathrm{x}} n + n^2 + 3n$

Table 4.2: Number of floating point operations (FLOPs) per shooting node required for the individual parts of the proposed block structured QP solver. One FLOP comprises one scalar floating point multiplication and addition. The numbers n^{y} and n^{z} with $n^{\mathrm{y}} + n^{\mathrm{z}} = n_i^{\mathrm{r}}$ denote the range–space and null-space dimension resulting from the QR decomposition (4.26), respectively. Further, we use $n := n_{\mathrm{x}} + n^{\mathrm{q}}$ to denote the system's dimension.

sive description of this optimal control problem, its differential equations, model parameters, objective function, and constraints can be found in [147] together with optimal solutions and computation times for a test driving scenario.

4.5.2 Outer Convexification of the integer control

We treat the integer gear choice $\mu(t) \in \{1, \dots, n^{\mu}\}$, wherein n^{μ} denotes the number of available gears, by outer convexification as detailed in Chapters 2 and 3 and in [213, 147]. Reasonable choices for n^{μ} range from 4 up to 24 in heavy–duty trucks, cf. [125, 237]. Outer convexification basically amounts to replacing the right hand side $f(\cdot)$ of the car model's ODE system

$$\dot{x}(t) = f(t, x(t), u(t), \mu(t)) \tag{4.38}$$

wherein $x = \left(c_{\mathrm{x}}, c_{\mathrm{y}}, v, \delta, \beta, \psi, w_{\mathrm{z}}\right)$ and $u = (w_{\delta}, F_{\mathrm{B}}, \phi)$, by its outer convexified reformulation

$$\dot{x}(t) = \sum_{i=1}^{n^{\mu}} w_i(t) \cdot f(t, x(t), u(t), \mu_i), \quad \sum_{i=1}^{n^{\mu}} w_i(t) = 1, \qquad \forall\, t \in \mathscr{T}. \tag{4.39}$$

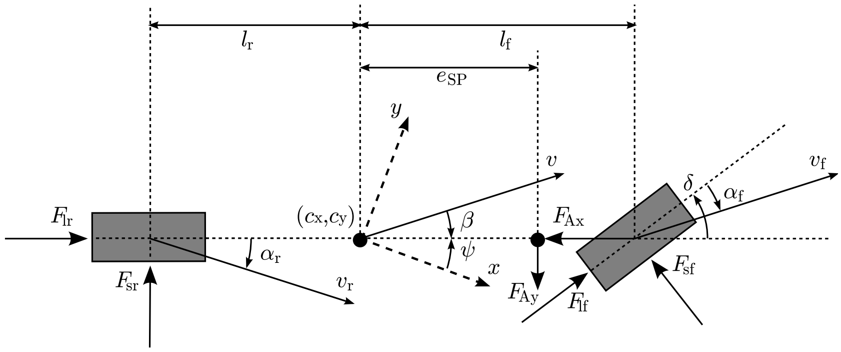


Figure 4.1: Coordinates, angles, and forces in the single–track car model used as a test bed.

State	Unit	Description
c_x	m	Horizontal position of the car
c_y	m	Vertical position of the car
v	$\frac{m}{s}$	Magnitude of directional velocity of the car
δ	rad	Steering wheel angle
β	rad	Side slip angle
ψ	rad	Yaw angle
w_z	$\frac{rad}{s}$	Yaw angle velocity

Table 4.3: Coordinates and states used in the single–track car model.

For each element $\mu_i \in \{1,\dots,n^\mu\}$ a separate binary control $w_i(\cdot) \in \{0,1\}$ is introduced, subject to the Special Ordered Set 1 (SOS1) constraint ensuring that for all $t \in [t_0,t_f]$ exactly one of the choices is attained. The same is done for every constraint function that involves $\mu(\cdot)$. The total number of control parameters for this car model then is $n^q = 3 + n^\mu$. Note that this formulation is still equivalent to the original one. The optimal control problem is solved with relaxed controls $w_i(t) \in [0,1] \subset \mathbb{R}$, making the $w_i(t)$ convex multipliers. We refer to [147, 204] for a discussion of the favorable properties of the obtained relaxed solution as well as a detailed presentation of possibilities to reconstruct an integer solution from the relaxed one.

4.5.3 Classical condensing for the example problem

The application of the classical condensing algorithm of Section 4.3 to the exemplary vehicle control problem reveals some shortcomings of the condensing algorithm for OCPs with many controls, e.g. due to outer convexification of integer controls. Clearly from Table 4.5 it can already be deduced that the classical condensing algorithm is suitable for problems with limited grid lengths m and with considerably more states than controls, i.e. $n_x \gg n^q$, which is exactly contrary to the situation encountered for MIOCPs. Nonetheless, using this approach we could

Control	Unit	Description
w_δ	$\frac{\mathrm{rad}}{\mathrm{s}}$	Steering wheel angular velocity
F_B	N	Total braking force
ϕ	–	Accelerator pedal position
μ	–	Selected gear

Table 4.4: Controls used in the single–track car model.

Classical Condensing and Dense QP Solver	
Computing the Hessian $\overline{\overline{B}}$	$O(m^2{n_\mathrm{x}}^3)+O(m^2{n_\mathrm{x}}^2n^\mathrm{q})$
Computing the Constraints $\overline{X}$, $\overline{R}$	$O(m^2{n_\mathrm{x}}^3)+O(m^2{n_\mathrm{x}}^2n^\mathrm{q})$
Dense QP solver on (4.19), startup	$O((mn^\mathrm{q}+n_\mathrm{x})^3)$
Dense QP solver on (4.19), per iteration	$O((mn^\mathrm{q}+n_\mathrm{x})^2)$
Recovering δv	$O(m{n_\mathrm{x}}^2)$

Complementary Condensing	
Startup	$O(mn^2)+O(m{n_\mathrm{x}}^3)$
Per Iteration	$O(mn^2)+O(m{n_\mathrm{x}}^3)$

Table 4.5: Run time complexity of the presented algorithms, given in terms of the optimal control problem dimensions. The symbol m denotes the shooting grid length, while $n = n_\mathrm{x} + n^\mathrm{q}$ is the total number of unknowns per shooting node.

solve several challenging mixed–integer optimal control problems to optimality with little computational effort, as reported in [147, 214, 204].

In Table 4.6 the dimensions and amount of sparsity present in the Hessian and constraints matrices are given for the exemplary vehicle control problem for $n^\mu = 4$ and $n^\mu = 16$ available gears. Here, the shooting grid lengths of $m = 20$ and $m = 50$ intervals were examined. As can be seen in the left part of the table, the block structured QP is only sparsely populated with the number of nonzero matrix entries never exceeding 3 percent. The number of nonzero elements per row of system (4.25) ranges from approximately 15 for the smallest example to 27 for the largest one. After the condensing step, the sparsity of both Hessian and constraints has been lost almost completely, as expected. This would be of no concern if the overall dimension of the QP had reduced considerably, as is the case for optimal control problems with $n_\mathrm{x} \gg n^\mathrm{q}$. For the case of an outer convexified MIOCP, however, this is not achieved. Worse yet, the dense active set method is unable to exploit what sparsity remains in the condensed constraints matrix, impairing the QP solver's performance further.

The results shown in Table 4.6 indicate that for larger values of m or n^μ, a considerable increase

			Block structured		Condensed		Dense QP solver
m	n^{μ}	Matrix	Size	nnz	Size	nnz	nnz seen
20	4	Hess.	330×330	$5,136$	130×130	$16,900$	$16,900$ ($3.3\times$)
		Constr.	264×330	$2,005$	64×130	$3,116$	$8,320$ ($4.1\times$)
50	4	Hess.	810×810	$12,816$	310×310	$96,100$	$96,100$ ($7.5\times$)
		Constr.	654×810	$4,756$	154×310	$16,767$	$47,740$ ($10.0\times$)
20	16	Hess.	570×570	$15,624$	370×370	$136,900$	$136,900$ ($8.8\times$)
		Constr.	264×570	$3,585$	64×370	$8,876$	$23,680$ ($6.6\times$)
50	16	Hess.	1410×1410	$39,144$	910×910	$828,100$	$828,100$ ($21.2\times$)
		Constr.	654×1410	$8,956$	154×910	$49,167$	$140,140$ ($15.6\times$)

Table 4.6: Comparison of dimensions and number of nonzero elements (nnz) of the Hessian and constraints matrix of QPs (4.11) and (4.19) for the exemplary vehicle control problem. All numbers for $n_{\mathrm{x}} = 10$, $n^{\mathrm{q}} = 3 + n^{\mu}$, m and n^{μ} varied. The last column gives the number of nonzero elements seen by the dense QP solver. In parentheses the increase compared to the number of nonzero elements in the block structured QP is given.

of the run time is to be expected. The matrices' size has been reduced only marginally, while the number of matrix entries treated by the dense QP solver has, for the largest instance examined, risen by a more than a factor of 15 when compared to the block structured QP.

This concern is supported by the results shown in tables 4.7 and 4.8. Here we list the run times in milliseconds of the classical condensing algorithm and of a single iteration of the dense null–space active–set QP solver QPOPT [111]. Averages have been taken over the all SQP iterations required to solve the optimal control problem to a precision of 10^{-6}. All run times have been obtained for an ANSI C99 (direct multiple shooting, condensing) and Fortran 77 (QPOPT) implementation running under Ubuntu Linux 9.04 on *a single core* of an Intel Core i7 920 machine at 2.67 GHz. BLAS linear algebra operations were done by ATLAS [246] in all parts of the implementation.

While the condensing algorithm's quadratic run time growth with the number m of multiple shooting nodes is acceptable for small systems, it becomes very noticeable for a larger number of integer decisions n^{μ}. The cubic complexity of the dense QP solver's initial setup with respect to m is clearly visible. The run time per iteration grows quadratically with both problem dimensions. When many active set iterations are required to find the QP's solution, this quickly becomes the bottleneck of the entire optimal control problem solution process as m or n^{μ} grow.

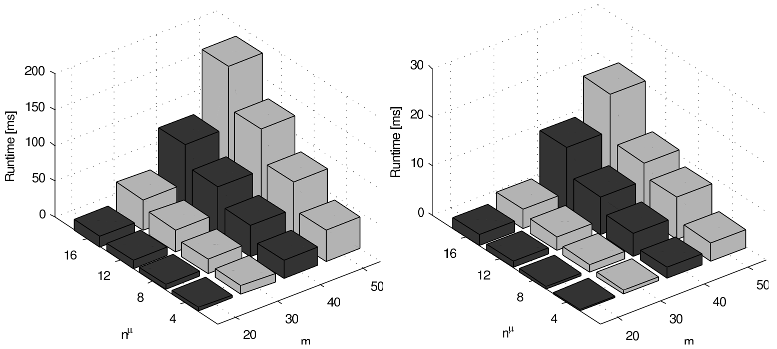
Summarizing the results presented in this section, we have seen that the dense QP solver's performance on the unnecessarily large QP is worse than what could be achieved by a suitable exploitation of the block structure for the case $n^{\mathrm{q}} \geq n_{\mathrm{x}}$.

	Number of nodes m			
n^{μ}	20	30	40	50
4	4	12	25	45
8	7	20	44	81
12	11	31	68	126
16	15	43	97	183

Table 4.7: Run times in milliseconds of the classical condensing algorithm of Section 4.3 for the presented vehicle control problem with increasing number of shooting nodes m and number of gears n^{μ}.

	Number of nodes m			
n^{μ}	20	30	40	50
4	0.3	0.9	2.0	3.7
8	0.6	1.6	4.7	8.7
12	1.2	2.9	7.6	11.1
16	2.2	3.9	13.2	20.7

Table 4.8: Average run times in milliseconds per iteration of QPOPT running on the condensed QPs for the presented vehicle control problem with increasing number of shooting nodes m and number of gears n^{μ}.



(a) Classical condensing runtime

(b) Average run time per iteration of the dense active–set QP solver.

Figure 4.2: Average run times in milliseconds for the presented vehicle control problem when solved using the dense null–space active–set QP solver QPOPT running on the condensed QPs obtained from the classical condensing algorithm of Section 4.3.

4.5.4 Complementary condensing for the example problem

In this final section we apply the proposed complementary condensing technique to the introductory vehicle control example. Table 4.9 lists the run times obtained for an ANSI C99 implementation of a primal active set QP code using the presented complementary condensing technique for the factorization of the KKT system. We compare the run times obtained for the exemplary vehicle control problem for different values of the shooting grid length m and the number n^{μ} of available gears.

The claimed run time complexity of $O(m)$ is easily seen in Figure 4.3(a), while the $O({n^{\mathrm{q}}}^2)$ complexity is not noticeable for the examined instances, as the computationally demanding parts of the complementary condensing approach are independent of the number n^{q} of discrete choices. The growth of the run time of a single QP iteration with growing problem dimensions is very small. The total number of required QP iterations still grows, though. For the largest instance examined, we find an average per iteration speedup of more than a factor of 20 when comparing the proposed block structured active set solver to the dense active–set solver QPOPT running on the condensed QP. In addition, the run time required for condensing, up to 200 milliseconds per SQP iteration, is saved entirely.

In a model–predictive control setup, giving fast feedback close to the controlled process' reference trajectory enables an active–set QP solver to complete in one or very few iterations, cf. [88, 89]. In this case, the proposed algorithm substitutes condensing plus one dense QP iteration for one block structured QP iteration, and the achievable speedup is as high as a factor of $(200\text{ ms} + 20.7\text{ ms})/1.0\text{ ms} > 200$.

Table 4.10 compares the performance of the complementary condensing approach, which effectively proposes a factorization of the KKT system with special block structure, to the highly efficient multifrontal symmetric indefinite factorization subroutine MA57 [79, 80] available from the Harwell Subroutine Library (HSL). Diagonal pivoting based on the minimum degree criterion as provided by MA57A was used. Our proposed method has a performance advantage of up to a factor of 8 for the largest examined instance. Our method does not currently exploit model–inherent sparsity, i.e. structures of the KKT matrix induced by the model rather than by the multiple shooting discretization. It should be noted, though, that the computation times listed include the necessary reassembly of the KKT matrix in triplets storage format. This overhead could be avoided. In addition, as noted in Section 4.5.3 the number of nonzero elements per row increases as the problem's dimensions grow, which explains the increasing performance gap between MA57 and our method. Finally, MA57 is likely to be numerically more stable, as our method does not currently make use of pivoting.

4.6 Extensions

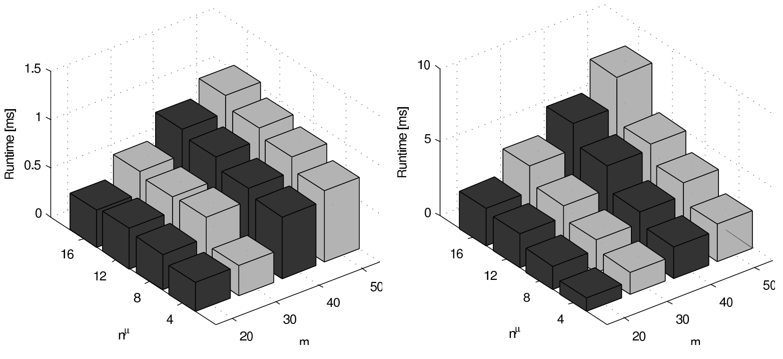
Additional work on the complementary condensing algorithm presented in Section 4.4 includes the following topics.

	Number of nodes m			
n^μ	20	30	40	50
4	0.3	0.3	0.6	0.7
8	0.4	0.6	0.7	0.9
12	0.4	0.6	0.8	0.9
16	0.4	0.6	0.9	1.0

Table 4.9: Average computation time in milliseconds per iteration of the proposed structure exploiting block structured QP solver.

	Number of nodes m			
n^μ	20	30	40	50
4	0.9	1.5	2.1	2.5
8	1.5	2.2	3.0	3.8
12	2.3	3.1	4.7	5.0
16	2.6	4.3	6.0	8.1

Table 4.10: Average computation time in milliseconds per iteration of a symmetric indefinite factorization of the QP's KKT system using HSL MA57.



(a) Average run time per iteration of the proposed block structured solver.

(b) Average run time per iteration when solving the KKT system using HSL MA57.

Figure 4.3: Average run times in milliseconds per iteration of the proposed block structured active set QP solver. The KKT system was solved with the proposed factorization (a) and with the highly efficient sparse symmetric indefinite factorization code HSL MA57 (b).

A first improvement to the presented approach inside an active–set loop is the exploitation of so–called *simple bounds* $\underline{w}_i \leq \delta w_i \leq \overline{w}_i$ on the unknowns by introducing the notion of *free* and *fixed unknowns*. Since the block structure of system (4.25) can be maintained, this effectively reduces the size of the matrices B_i, R_i, and X_i. Decompositions and multiplications during solution of the KKT system on the smaller matrices can be expected to perform faster.

The formulation of the optimal control problem (4.1) can be extended to include more general classes of constraints, such as periodicity, boundary conditions, or fully coupled constraints. Such extensions destroy the block structure of system (4.25) to some extent, though, and prevent the factorizations to work on a per-block level. Certain subclasses, such as periodicity conditions, can still be treated by introduction of artificial constraints taking the role of residuals.

The proposed method's stability on severely ill-conditioned systems has to be investigated. Pivoting strategies improving the stability can work on both the block level and inside the block-local factorizations. To this end, several pivoting strategies have been briefly mentioned in Section 4.4.6.

Within the active–set loop of the QP solver, all decompositions have to be recomputed whenever a point constraint enters or leaves the active set. When exploiting simple bounds, the same holds true whenever an unknown hits or leaves one of its bounds. From dense null–space and range–space methods it is common knowledge that certain decompositions can be updated during an active set change in $O(n^2)$ time [110]. Such update techniques would essentially remove all matrix decompositions and matrix–matrix operations listed in Table 4.1 from the active–set loop. This yields an $O(mn^2)$ block structured active set method with only an initial factorization in $O(mn^3)$ time.
Details on these more specific issues can be found in the PhD thesis of Christian Kirches [143] and in the publication [146].

4.7 Summary

We have considered the solution of mixed–integer optimal control problems in ordinary differential equations. We treated the integer control by outer convexification [203] and reviewed the direct multiple shooting method [191, 44, 168] to obtain a discretized optimal control problem. Sequential quadratic programming methods have been our motivation to investigate the solution of the highly structured quadratic subproblems. We reviewed the classical condensing algorithm [191, 44, 168] that works as a preprocessing step for the quadratic subproblems, and enables the efficient usage of a wealth of available dense quadratic programming codes. Application of this approach to an exemplary vehicle control problem with gear shift revealed that for longer horizons or larger numbers of choices for the integer control, the classical condensing algorithm leaves room for improvement. To address this issue, we presented a new approach at solving the highly structured quadratic program by devising a new factorization of the QP's KKT matrix that respects the block structure introduced by direct multiple shooting. We employed this new method to solve the exemplary vehicle control problem and compared it a) to the classical condensing approach, and b) to the highly efficient sparse symmetric indefinite factorization code MA57 which was used as an alternative means to obtain a factorization of the highly structured QP's KKT matrix. The presented computational results indicate that the proposed method is able to deliver promising run times for all examined instances of the vehicle control problem.

We derived an $O(mn^3)$ runtime complexity for our method, in contrast to $O(m^2n^3)$ for the classical condensing, where m is the number of nodes and n is the number of process states and control parameters per node. A speedup of a factor of 20 was obtained for the largest instance of the example problem examined, and we proposed a speedup of a factor of 200 for a special model predictive control scenario.

The mentioned extensions in Section 4.6 allow a further reduction. Especially the updating of factorizations reduces the runtime complexity per iteration to $O(mn^2)$.

5 Combinatorial Integral Approximation

The contents of this chapter are based on the paper

[212] S. Sager, M. Jung, C. Kirches. Combinatorial Integral Approximation. *Mathematical Methods for Operations Research*, 2011, Vol. 73(3):363–380.

Chapter Summary. We are interested in structures and efficient methods for mixed-integer nonlinear programs (MINLP) that arise from a *first discretize, then optimize* approach to time-dependent mixed-integer optimal control problems (MIOCPs). In this study we focus on combinatorial constraints, in particular on restrictions on the number of switches on a fixed time grid.

We propose a novel approach that is based on a decomposition of the MINLP into a NLP and a MILP. We discuss the relation of the MILP solution to the MINLP solution and formulate bounds for the gap between the two, depending on Lipschitz constants and the control discretization grid size. The MILP solution can also be used for an efficient initialization of the MINLP solution process.

The speedup of the solution of the MILP compared to the MINLP solution is considerable already for general purpose MILP solvers. We analyze the structure of the MILP that takes switching constraints into account and propose a tailored Branch and Bound strategy that outperforms *cplex* on a numerical case study and hence further improves efficiency of our novel method.

5.1 Introduction

Again, our main motivation are mixed-integer optimal control problems in ordinary differential equations that are of the form (2.1), compare page 8. However, we look at a more specific problem formulation that allows to formulate constraints on the number of switches that take place.

We assume that one of the controls needs to take binary values and can only change these values on a prefixed time grid

$$0 = t_1 < \ldots < t_{n_t+1} = t_f, \tag{5.1}$$

which we use for a discretization of the control in a *first discretize, then optimize* approach, compare Chapter 4. For the sake of notational simplicity we consider a problem with linearly entering piecewise constant binary control functions,

$$\omega_k(t) = p_{k,i}, \quad t \in [t_i, t_{i+1}],\ k = 1 \dots n_\omega,\ i = 1 \dots n_\mathrm{t} \tag{5.2}$$

with $p_{k,i} \in \{0,1\}$. We want to minimize a Mayer term

$$\min_{x,p} \Phi(x(t_\mathrm{f})) \tag{5.3a}$$

over the differential states $x(\cdot)$ and the discretized binary control p subject to the n_x-dimensional ODE system

$$\dot{x}(t) \quad = \quad f_0(x(t)) + \sum_{k=1}^{n_\omega} f_k(x(t))\, p_{k,i}, \quad t \in [t_i, t_{i+1}], \tag{5.3b}$$

fixed initial values

$$x(0) \quad = \quad x_0, \tag{5.3c}$$

integrality of the control function $\omega(\cdot)$

$$p_{k,i} \quad \in \quad \{0,1\}, \quad k = 1 \dots n_\omega,\ i = 1 \dots n_\mathrm{t}, \tag{5.3d}$$

and switching constraints

$$\sum_{i=1}^{n_\mathrm{t}-1} |p_{k,i+1} - p_{k,i}| \leq \sigma_{k,\max}, \quad k = 1 \dots n_\omega. \tag{5.3e}$$

Note that the generalization towards the more general case in which $\omega(\cdot)$ enters in a nonlinear way into the right-hand side can be achieved by means of an SOS1 constraint. Also additional continuous controls, path constraints, or multi-stage formulations can be included, compare the results in [214, 204] and Chapter 2. For the sake of notational simplicity, however, we concentrate on the special case stated above.

Although in practice we use a simultaneous approach, e.g., collocation [136] or direct multiple shooting [168], we consider the differential states as dependent variables in the theoretical part that can be determined uniquely, whenever the controls are fixed. This transforms (5.3) into a MINLP with finitely many degrees of freedom. The difference to MIOCPs as they are defined, e.g., in [214, 204] are the additional switching restriction (5.3e) and the fixed time grid (5.1) which do not allow the usage of a switching time optimization. More remotely related is the question of the maximum number of switches for equivalent reachable sets. For a special case of a switched system it is shown in [225] that 4 switches are enough. A counterexample based on Fuller's phenomenon is given in [172]. However, these approaches are based on continuous time, not on fixed switching grids. Therefore we focus on combinatorial approaches, i.e., integer programming.

Progress in mixed-integer linear programming (MILP) started with the fundamental work of Dantzig and coworkers on the Traveling Salesman problem in the 1950s. Since then, enormous progress has been made in areas such as *linear programming* (and especially in the *dual simplex* method that is the core of almost all MILP solvers because of its restart capabilities), in the understanding of *branching rules* and more powerful selection criteria such as *strong branching*, the derivation of tight *cutting planes*, novel *preprocessing* and *bound tightening procedures*, and of course the computational advances roughly following Moore's law. For specific problem classes problems with millions of integer variables can now be routinely solved [13]. Also generic problems can often be solved very efficiently in practice, despite the known exponential complexity from a theoretical point of view [38].

The situation is different in the field of Mixed-Integer Nonlinear Programming (MINLP). Only at first sight many properties of MILP seem to carry over to the nonlinear case. Restarting nonlinear continuous relaxations within branching trees is essentially more difficult than restarting linear relaxations (which some global solvers also use for nonlinear problems), as no dual algorithm comparable to the dual simplex is available in the general case. Nonconvexities lead to local minima and do not allow for easy calculation of subtrees, which is important to avoid an explicit enumeration. Additionally, nonlinear solvers are slower and less robust than LP solvers. However, the last decade saw great progress triggered by cross-disciplinary work of integer and nonlinear optimizers, resulting in generic MINLP solvers, e.g., [1, 45], or efficient heuristics such as the Feasibility Pump [46]. Most of them, however, still require the underlying functions to be convex. Comprehensive surveys on algorithms and software for convex MINLPs are given in [120, 47]. Recent progress in the solution of nonconvex MINLPs is in most cases based on methods from global optimization, in particular convex under- and overestimation. See, e.g., [30, 92, 235] for references on general under– and overestimation of functions and sets. In our study we use the solver *Bonmin* [45] for comparison and show how important it is to exploit problem-class specific structures.

The basic idea of our new approach to solve problem (5.3) consists of a decomposition of the MINLP into an NLP and an MILP, which we can both solve comparatively efficiently. This idea is related to ideas of [54]. The authors reformulate the MIOCP as a large-scale, structured nonlinear program (NLP) and solve a small scale linear integer program on a second level to approximate the calculated continuous aggregated output of all pumps in a water works. However, their decomposition is tailored to the special structure of the water network application, while our approach targets generic problems of the form (5.3).

To guarantee error bounds on the obtained solution compared to the MINLP solution, we revise some theoretical results in Section 5.2. In Section 5.3 we discuss our new method that is based on a combinatorial approximation of the integral over control deviations. In Section 5.4 we analyze the structure of the MILP and provide a structure exploiting Branch and Bound algorithm. In Section 5.5 we present results for a numerical benchmark example. Finally, we conclude and give an outlook in Section 5.6.

5.2 Approximation results

We often leave the argument (t) away for the sake of notational simplicity. In the following $\|\cdot\|$ denotes the maximum norm.

For our error-bounded decomposition approach we need the results from Chapter 3. Theorem 3.2.2 on page 36 postulates an upper bound on the difference between differential states that depends on the value η in

$$\left\| \int_0^t \alpha(\tau) - \omega(\tau)\, \mathrm{d}\tau \right\| \quad \leq \quad \eta. \tag{5.4}$$

We repeat the definition of the Sum Up Rounding strategy that was first given in (3.7, 3.8). We assume we have an optimal trajectory $(x^*(\cdot), \alpha(\cdot))$ with

$$\alpha_k(t) = q_{k,i}, \quad k = 1 \dots n_\omega,\ t \in [t_i, t_{i+1}]. \tag{5.5}$$

Again, we write $\Delta t_i := t_{i+1} - t_i$ and Δt for the maximum distance between two time points, $\Delta t := \max_{i=1\dots n_t} \Delta t_i = \max_{i=1\dots n_t}\{t_{i+1} - t_i\}$. Let then a function $\omega(\cdot) : [0, t_f] \mapsto \{0,1\}^{n_\omega}$ be defined by

$$\omega_k(t) = p^{\text{SUR}}_{k,i}, \quad k = 1 \dots n_\omega,\ t \in [t_i, t_{i+1}] \tag{5.6}$$

where the $p^{\text{SUR}}_{k,i}$ are binary values given for $k = 1 \dots n_\omega$ by

$$p^{\text{SUR}}_{k,i} \quad = \quad \begin{cases} 1 & \text{if } \sum_{j=1}^{i} q_{k,j}\Delta t_j - \sum_{j=1}^{i-1} p^{\text{SUR}}_{k,j}\Delta t_j \geq 0.5\Delta t_i \\ 0 & \text{else} \end{cases}. \tag{5.7}$$

Additionally, we define σ^{SUR}_k to be the minimal number for which inequality (5.3e) holds for p^{SUR}_k. For convenience we repeat Theorem 3.3.1.

Theorem 5.2.1. *Let the functions* $\alpha : [0, t_f] \mapsto [0,1]^{n_\omega}$ *and* $\omega : [0, t_f] \mapsto \{0,1\}^{n_\omega}$ *be given by (5.5) and (5.6, 5.7), respectively. Then it holds*

$$\left\| \int_0^t \alpha(\tau) - \omega(\tau)\, \mathrm{d}\tau \right\| \leq \eta$$

with $\eta = 0.5\,\Delta t$.

Note that for the more general case in which the integer control functions enter in a nonlinear way into the differential equations the SUR strategy can be modified to incorporate the SOS1 constraint, see Theorem 3.4.2. Theorem 5.2.1 still holds with a constant η which is a function of n_ω multiplied by Δt.

5.3 Approximating the integral over the controls by MILP techniques

The results of Section 5.2 have been used in several ways. Most importantly they imply that, if the control discretization grid is fine enough, no integer gap exists [214], because Δt can be chosen arbitrarily small and the estimation carries over to continuous objective and constraint functions. Also, the specific way of constructing a binary solution (5.6,5.7) can be used, e.g., in the adaptive algorithm MINTOC, Section 2.7.4. However, both uses require that the constructed binary control is feasible for the original problem. This is not a problem if only constraints on the differential states are present when $\Delta t \to 0$, but constraints of the type (5.3e) are typically violated if Δt is small.

Therefore we propose to change the point of view: while before it was argued that the difference between integer and relaxed solution becomes arbitrarily small if $\Delta t \to 0$, we now consider Δt to be fixed and allow a larger constant to obtain a feasible solution.

To be able to include constraint (5.3e) we determine p not by (5.6,5.7), but as the solution of the MILP

$$\begin{aligned}
&\min_{p} \max_{k=1\ldots n_\omega} \max_{i=1\ldots n_\mathrm{t}} \left| \sum_{j=1}^{i} (q_{k,j} - p_{k,j}) \Delta t_j \right| \\
&\text{subject to} \\
&\qquad \begin{array}{lcll}
\sigma_{k,\max} & \geq & \sum_{i=1}^{n_\mathrm{t}-1} |p_{k,i} - p_{k,i+1}|, & k = 1\ldots n_\omega, \\
p_{k,i} & \in & \{0,1\}, & k = 1\ldots n_\omega,\ i = 1\ldots n_\mathrm{t}.
\end{array}
\end{aligned} \tag{5.8}$$

To get rid of the min max formulation and the absolute values, we introduce slack variables $\eta \in \mathbb{R}$ and $s \in [0,1]^{n_\omega \times (n_\mathrm{t}-1)}$ and obtain

$$\begin{aligned}
&\min_{\eta,s,p} \ \eta \\
&\text{subject to} \\
&\qquad \begin{array}{lcll}
\eta & \geq & \sum_{j=1}^{i} (q_{k,j} - p_{k,j}) \Delta t_j, & k = 1\ldots n_\omega,\ i = 1\ldots n_\mathrm{t}, \\
\eta & \geq & -\sum_{j=1}^{i} (q_{k,j} - p_{k,j}) \Delta t_j, & k = 1\ldots n_\omega,\ i = 1\ldots n_\mathrm{t}, \\
s_{k,i} & \geq & p_{k,i} - p_{k,i+1}, & k = 1\ldots n_\omega,\ i = 1\ldots n_\mathrm{t}-1, \\
s_{k,i} & \geq & -p_{k,i} + p_{k,i+1}, & k = 1\ldots n_\omega,\ i = 1\ldots n_\mathrm{t}-1, \\
\sigma_{k,\max} & \geq & \sum_{i=1}^{n_\mathrm{t}-1} s_{k,i}, & k = 1\ldots n_\omega, \\
p_{k,i} & \in & \{0,1\}, & k = 1\ldots n_\omega,\ i = 1\ldots n_\mathrm{t},
\end{array}
\end{aligned} \tag{5.9}$$

for fixed control values q that stem from the solution of the relaxed problem (5.3) and given upper bounds on the number of switches, $\sigma_{k,\max}$.

Although problem (5.9) is a MILP and thus typically hard to solve, for certain values $\sigma_{k,\max}$ the solution can be calculated in polynomial time using the Sum Up Rounding strategy (5.6, 5.7). This is the content of the following theorem. In analogy to the maximal interval length $\Delta t := \max_{i=1\dots n_\mathrm{t}} \Delta t_i$ we also define the minimal one, $\delta t := \min_{i=1\dots n_\mathrm{t}} \Delta t_i$.

Theorem 5.3.1. *Assume p^{SUR} to be the solution obtained by Sum Up Rounding (5.6, 5.7). The following claims hold for the optimal solution (η^*, s^*, p^*) of the MILP* (5.9)*:*

$$\begin{aligned} & (a)\ \eta^* < 0.5\ \delta t = 0.5 \min_{i=1\dots n_t} \Delta t_i \\ \Rightarrow\quad & (b)\ p^* = p^{SUR} \\ \Rightarrow\quad & (c)\ \sigma_{k,\max} \geq \sigma_k^{SUR} \quad \forall\, k = 1\dots n_\omega \\ \Rightarrow\quad & (d)\ \eta^* \leq 0.5\ \Delta t = 0.5 \max_{i=1\dots n_t} \Delta t_i \end{aligned}$$

where the solution $p^* = p^{SUR}$ in (b) is unique.

Proof. "(a) ⇒ (b)". Assume first $\eta^* < 0.5\ \delta t$ and $p^* \neq p^{\mathrm{SUR}}$. Then there must exist indices $k \in \{1, \dots, n_\omega\}$ and $i \in \{1, \dots, n_\mathrm{t}\}$ such that $p^*_{k,j} = p^{\mathrm{SUR}}_{k,j}$ for all $j < i$ and $p^*_{k,i} \neq p^{\mathrm{SUR}}_{k,i}$.

We have two cases for the binary variables $p^*_{k,i} \neq p^{\mathrm{SUR}}_{k,i}$. If $p^*_{k,i} = 0$ and $p^{\mathrm{SUR}}_{k,i} = 1$, then from (5.7) it follows that

$$\sum_{j=1}^{i} q_{k,j} \Delta t_j - \sum_{j=1}^{i-1} p^{\mathrm{SUR}}_{k,j} \Delta t_j \geq 0.5\ \Delta t_i$$

and hence

$$\sum_{j=1}^{i} (q_{k,j} - p^*_{k,j}) \Delta t_j = \sum_{j=1}^{i} q_{k,j} \Delta t_j - \sum_{j=1}^{i-1} p^{\mathrm{SUR}}_{k,j} \Delta t_j \geq 0.5\ \Delta t_i. \tag{5.10}$$

Equivalently, if $p^*_{k,i} = 1$ and $p^{\mathrm{SUR}}_{k,i} = 0$ then

$$\sum_{j=1}^{i} q_{k,j} \Delta t_j - \sum_{j=1}^{i-1} p^{\mathrm{SUR}}_{k,j} \Delta t_j < 0.5\ \Delta t_i$$

and therefore

$$\sum_{j=1}^{i} (q_{k,j} - p^*_{k,j}) \Delta t_j = -\Delta t_i + \sum_{j=1}^{i} q_{k,j} \Delta t_j - \sum_{j=1}^{i-1} p^{\mathrm{SUR}}_{k,j} \Delta t_j < -0.5\ \Delta t_i. \tag{5.11}$$

As (η^*, s^*, p^*) is a feasible solution of (5.9), with (5.10) and (5.11) we have the contradiction

$$\eta^* \geq \left| \sum_{j=1}^{i} (q_{k,j} - p^*_{k,j}) \Delta t_j \right| \geq 0.5\ \Delta t_i \geq 0.5\ \delta t \tag{5.12}$$

to the assumption $\eta^* < 0.5\ \delta t$. Therefore $p^* = p^{\text{SUR}}$.

"(b) ⇒ (c)". Assume now $p^* = p^{\text{SUR}}$. As (η^*, s^*, p^*) is a feasible solution of (5.9), the number of switches of p^{SUR} given by $\sigma^{\text{SUR}}_{\max}$ is necessarily at least $\sigma_{\max}$, componentwise.

"(c) ⇒ (d)". If it holds that $\sigma_{k,\max} \geq \sigma^{\text{SUR}}_k$ for all $k = 1 \dots n_\omega$, then the vector given by

$$\begin{aligned} \eta &= 0.5\Delta t, \\ p &= p^{\text{SUR}}, \\ s_{k,i} &= |p_{k,i} - p_{k,i+1}|, \quad k = 1 \dots n_\omega,\ i = 1 \dots n_{\text{t}} - 1 \end{aligned}$$

is a feasible solution of (5.9) as follows from Theorem 5.2.1, and yields hence an upper bound on the objective function value η^*. □

Remark 5.3.2. *The asymmetry in Theorem 5.3.1 even for an equidistant grid with $\delta t = \Delta t = \Delta t_i$ is due to the degenerate case where $\eta^* = 0.5\Delta t$. While p^{SUR} always yields a solution with $\eta^{SUR} \leq 0.5\Delta t$, this solution might switch more often than another control resulting in $\eta^* = 0.5\Delta t$. The easiest example is $q_k = (0.5, 0, \dots, 0)$, which results in $p^{SUR}_k = (1, 0, \dots, 0)$ with one switch. The same value of $\eta^* = 0.5\Delta t$ is obtained by $p_k = (0, 0, \dots, 0)$. This is also the optimal solution of the MILP instance with q_k and $\sigma_{k,\max} = 0$ for which p^{SUR} is infeasible, but still $\eta^* = 0.5\Delta t$.*

Remark 5.3.3. *It holds $\eta^{SUR} \leq 0.5\Delta t$, and therefore also the optimal objective function values η^* of MILP* (5.9) *decrease, as Δt is decreased. However, this is not necessarily strictly monotonic, as the amount of reduction depends heavily on the values of q and Δt_i.*

Theorem 5.3.1 is particularly interesting, as we know from Corollary 3.5.3 that if $n_{\text{t}} \to \infty$, then $\Phi(x^{\text{SUR}}) \to \Phi^*$, i.e., the solution obtained with Sum Up Rounding, p^{SUR}, yields an objective function value that converges against the lower bound Φ^* obtained by solving the relaxed version of (5.3).

However, the solution p^{SUR} may violate the switching constraint (5.3e). Hence, solving the MILP yields a compromise between the approximation of the control integral, which has been shown to imply convergence towards the objective's lower bound if the control discretization is refined, and the incorporation of switching constraints — and possibly all other types of linear constraints on p — by means of a mixed-integer linear program.

5.4 Solving the MILP

The mixed-integer linear program (5.9) can be solved with standard solvers, such as *cplex*. However, as the structure is generic for all MIOCPs with switching restrictions, we have a closer look at the facets of the convex hull of all feasible points in Section 5.4.1. To speed up computational runtimes we also propose a tailored Branch and Bound strategy in Section 5.4.2.

5.4.1 Facet defining inequalities

Important insight can be gained by investigating the feasibility polytope. An investigation of min down/up polytopes, for example, can be found in [166].

The MILP (5.9) has a specific structure, partly independent of the values of q and $\sigma_{\max}$. To identify the structure – especially the facets – of the convex hull of all feasible points of MILP (5.9) we use the software-package *PORTA* [64, 65]. The following constraints define facets of this polytope,

$$\begin{aligned}
s_{k,i} &\geq p_{k,i} - p_{k,i+1}, & k = 1 \dots n_\omega,\; i = 1 \dots n_\mathrm{t} - 1,\\
s_{k,i} &\geq -p_{k,i} + p_{k,i+1}, & k = 1 \dots n_\omega,\; i = 1 \dots n_\mathrm{t} - 1,\\
s_{k,i} &\leq p_{k,i} + p_{k,i+1}, & k = 1 \dots n_\omega,\; i = 1 \dots n_\mathrm{t} - 1,\\
s_{k,i} &\leq 2 - p_{k,i} - p_{k,i+1}, & k = 1 \dots n_\omega,\; i = 1 \dots n_\mathrm{t} - 1.
\end{aligned} \tag{5.13}$$

Depending on whether the $\sigma_{k,\max}$ are fixed to a certain value or not, the corresponding facets are different. If $\sigma_{k,\max}$ is free, they read as

$$\sigma_{k,\max} \geq \textstyle\sum_{i=1}^{n_\mathrm{t}-1} s_{k,i}, \quad k = 1 \dots n_\omega. \tag{5.14a}$$

If $\sigma_{k,\max}$ is fixed to an even value, then as

$$\begin{aligned}
\sigma_{k,\max} &\geq p_{k,1} - p_{k,n_\mathrm{t}} + \textstyle\sum_{i=1}^{n_\mathrm{t}-1} s_{k,i}, \quad k = 1 \dots n_\omega,\\
\sigma_{k,\max} &\geq p_{k,n_\mathrm{t}} - p_{k,1} + \textstyle\sum_{i=1}^{n_\mathrm{t}-1} s_{k,i}, \quad k = 1 \dots n_\omega,
\end{aligned} \tag{5.14b}$$

and alternatively if $\sigma_{k,\max}$ is fixed to an odd value as

$$\begin{aligned}
\sigma_{k,\max} &\geq 1 - p_{k,1} - p_{k,n_\mathrm{t}} + \textstyle\sum_{i=1}^{n_\mathrm{t}-1} s_{k,i}, \quad k = 1 \dots n_\omega,\\
\sigma_{k,\max} &\geq p_{k,1} + p_{k,n_\mathrm{t}} - 1 + \textstyle\sum_{i=1}^{n_\mathrm{t}-1} s_{k,i}, \quad k = 1 \dots n_\omega.
\end{aligned} \tag{5.14c}$$

	Control values $q_{1,j}$ fixed to:								
n_t	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
4	16	18	23	33	29	33	23	18	16
5	21	31	87	189	54	189	87	31	21
6	30	60	745	612	248	612	745	60	30
7	47	150	4838	4840	922	4840	4838	150	47
8	83	899	37470	29884	4212	29884	37470	899	83

Table 5.1: Number of all facets for problems with only one control and all the given $q_{1,j}$ fixed to a certain value, all the $\sigma_{k,\max}$ are free.

Unfortunately, the facets arising from the approximation inequalities

$$\begin{aligned} \eta &\geq \quad \textstyle\sum_{j=1}^{i}(q_{k,j}-p_{k,j})\Delta t_j, \quad k=1\ldots n_\omega,\ i=1\ldots n_t, \\ \eta &\geq -\textstyle\sum_{j=1}^{i}(q_{k,j}-p_{k,j})\Delta t_j, \quad k=1\ldots n_\omega,\ i=1\ldots n_t. \end{aligned} \tag{5.15}$$

cannot be expressed as easily, as far as we know. They mainly depend on the values of q and generally are dense in both $p_{k,i}$ and $s_{k,i}$. Additionally, their number strongly increases with the size of the problem, as can be seen in Table 5.1. Therefore it is hard to identify structures in the corresponding facets which would possibly enable cutting plane methods.

5.4.2 Solving the MILPs efficiently

As an alternative to cutting planes we implemented a structure exploiting pure Branch and Bound algorithm. It uses the structure of the approximation inequalities (5.15) that model the min max formulation. We branch on controls p and determine s as dependent variables. We branch in increasing order of the time index i in $p_{k,i}$. This way $2\,n_\omega$ inequalities are fixed for each i, i.e., all variables $s_{k,j}$ and $p_{k,j}$ with $j \leq i$ are fixed and we can give a new bound on η using constraints (5.15). Because of this lower bound it is not necessary to solve an LP relaxation.

We present a short outline of the algorithm. Each node of the branching tree contains the four components

- depth d of the node, i.e., the number of timesteps for which the controls are fixed,
- the fixed control variables $p_{k,j}$ for $j \leq d$,
- the fixed slack variables $s_{k,j}$ for $j \leq d$,
- the corresponding lower bound on η.

The priority queue in Algorithm 5.1 models the search strategy, in our case a best-first search (if two nodes have the same objective value, the deeper one is preferred). Note that the algorithm does not solve any relaxed linear programs, but is purely based on efficient branching and constraint/objective evaluation.

Algorithm 5.1: Combinatorial Branch and Bound

```
Input  : Relaxed controls $q$, time grid $\{t_i\}, i = 1 \dots n_\mathrm{t}$, max. numbers of switches
         $\sigma_{k,\max}, k = 1 \dots n_\omega$.
Result : Optimal solution $(\eta^*, s^*, p^*)$ of (5.9).
begin
    Create empty priority queue $Q$ ordered by $a.\eta$ (non-decreasing), if equal by $a.d$
    (non-increasing).
    Push an empty node (0, {}, {}, 0.0) into the queue.
    while $Q$ is not empty do
        $a$ = top node of $Q$ and remove the node from $Q$.
        /* 1st solution found is optimal since best-first search is used      */
        if $a.d = n_\mathrm{t}$ then
            Return optimal solution $(a.\eta, a.s, a.p)$.
        end
        /* Create child nodes, use strong branching.                          */
        else
            forall possible permutations $\phi$ of $\{0,1\}^{n_\omega}$ do
                Create new node $n$ with $n.d = d + 1$, $n.p = a.p$, $n.s = a.s$.
                Set $n.p_{k,d+1} = \phi_k$, calculate $n.s_{k,d+1}$.
                if $n.s$ fulfills switching constraint (5.3e) until time $d + 1$ then
                    $n.\eta = \max\left\{a.\eta, \max_{k=1}^{n_\omega}\{\pm\sum_{j=1}^{d+1}(q_{k,j} - p_{k,j})\Delta t_j\}\right\}$
                    Push $n$ into $Q$.
                end
            end
        end
    end
end
```

5.5 Numerical results

An open online benchmark library for the problem class of MIOCPs is available at [202]. Here we present numerical results for the Lotka-Volterra benchmark fishing problem from Section 7.4 extended with an additional switching constraint (5.3e),

$$
\begin{aligned}
\min_{x,w} \quad & x_2(t_\mathrm{f}) && (5.16\text{a})\\
\text{subject to} \quad & \dot{x}_0(t) = x_0(t) - x_0(t)x_1(t) - \; c_0 x_0(t)\; w(t), && (5.16\text{b})\\
& \dot{x}_1(t) = -x_1(t) + x_0(t)x_1(t) - \; c_1 x_1(t)\; w(t), && (5.16\text{c})\\
& \dot{x}_2(t) = (x_0(t)-1)^2 + (x_1(t)-1)^2, && (5.16\text{d})\\
& x(0) = (0.5, 0.7, 0)^T, && (5.16\text{e})\\
& w(t) = p_i \in \{0,1\},\; t \in [t_i, t_{i+1}], && (5.16\text{f})\\
& \sigma_{\max} \geq \sum_{i=1}^{n_\mathrm{t}-1} |p_{i+1} - p_i|, && (5.16\text{g})
\end{aligned}
$$

with $c_0 = 0.4$, $c_1 = 0.2$, $t_\mathrm{f} = 12$, and different equidistant grids $\{t_1, \ldots, t_{n_\mathrm{t}+1}\}$. This problem is particularly suited for our study, because the optimal relaxed solution contains a singular arc [203].

The differential equations have been discretized with an implicit Euler method and 10000 equidistant time steps, independent of the control discretization. All computational times refer to a two core Intel CPU with 3GHz and 8GB RAM run under Ubuntu 9.10. We used *Bonmin* 1.2 trunk revision 1601[1] and *cplex* 8.1 with standard options, respectively.

Numerical results are shown in Tables 5.2 and 5.4. Here τ is the computing time in seconds, Φ denotes the objective function value. The number of switches of a solution is given by σ, while η is the maximum deviation of the integrated difference between relaxed and integer control over the time horizon. Note, however, that the values of η have been scaled by $\frac{t_\mathrm{f}}{n_\mathrm{t}}$ for better comparability. The upper script *rel* refers to the relaxed version of the OCP (5.3), *milp* to the solution of the MILP (5.9) obtained with either *cplex* 8.1 or with our own code (*bb*) as described in Section 5.4.2, and *minlp* to the solution of the MINLP resulting from a discretization of (5.16) and solution with *Bonmin* 1.2.

We used an upper time limit of 1800 seconds, indicated by an * in Table 5.2 whenever active. If no feasible solution could be found within this upper time limit, this is indicated by an *, otherwise the value of the upper bound feasible solution is listed. In Table 5.4 a value for $\sigma_{\max}$ and a * indicate that no better solution than the one from the MILP initialization could be found.

[1] using Cbc 2.4stable and Ipopt 3.8stable

5.5.1 MILP and MINLP solutions

Numerical results for the solution of problem (5.16) with different upper limits on the number of switchings of p_i between 0 and 1 and different equidistant discretizations (5.1) are shown in Table 5.2. The first rows show the behavior of the solution of the relaxed MINLP (5.16). As predicted by theory, compare Section 5.2, the objective function values of the relaxed problem Φ^{rel} and the Sum Up Rounding solutions Φ^{SUR} converge towards a Φ^* that is the solution of the non-discretized, relaxed optimal control problem. However, the number of switches $\sigma_{\max}$ of the SUR solution increases significantly. All values of $\frac{1}{\Delta t}\,\eta^{\text{SUR}} = \frac{n_{\text{t}}}{t_{\text{f}}}\,\eta^{\text{SUR}}$ are below 0.5, as predicted by Theorem 5.2.1. It is interesting to observe that these values approach 0.5 as n_{t} increases, due to the increased probability to find a maximum close to the upper bound 0.5.

The next blocks show results for the solutions of MILPs and MINLPs corresponding to different upper limits $\sigma_{\max}$. If this limit is large enough, then in accordance with Theorem 5.3.1 the MILP and SUR solutions coincide (e.g., $n_{\text{t}} = 25, \sigma_{\max} \geq 4$). If not, the value of $\frac{1}{\Delta t}\,\eta^{\text{milp}}$ necessarily increases above 0.5. The objective function values Φ^{milp} and Φ^{minlp} *both* converge against the value of Φ^{rel}, as $n_{\text{t}} \to \infty$ and $\sigma_{\max}$ large enough. If switching constraints are active, the objective function value is bounded by a constant multiple of η. Although the MILP is not necessarily optimal for the MINLP, it has the advantage to be feasible, to have asymptotic properties, and to be a priori bounded.

As can be observed, the CPU times for the Branch and Bound algorithm are below those of *cplex* (τ^{bb} vs. τ^{cplex}), which in turn are considerably below those of the MINLP solver (τ^{cplex} vs. τ^{bonmin}). For all larger problems *Bonmin* violated the upper time limit of 1800 seconds.

Remark 5.5.1. *It is interesting to observe that, as $\sigma_{\max}$ increases for given n_t, the computational effort increases, due to the fact that more Branch and Bound subtrees need to be evaluated. However, once the value $\sigma_{\max}$ reaches σ^{SUR}, the solution of MILP* (5.9) *can be determined in linear time with the Sum Up Rounding strategy, compare Theorem 5.3.1.*

5.5.2 Using the MILP solution for cutoff in the MINLP tree

The MILP solution can itself be used as a solution that gets arbitrarily close to the lower bound, if n_{t} and $\sigma_{\max}$ are large enough. If the global solution on a given grid is an issue, and MINLP solvers have to be used, the solution can still be used to obtain a reduction in the MINLP Branch and Bound tree. The MILP solution is a feasible solution that respects the switching constraint (5.3e). *Bonmin* provides a bonmin.cutoff option that can be used to eliminate branches with a lower bound exceeding this value. In Table 5.4 numerical results are presented that show the effect of this additional information.

It results either in a reduction of the overall computation time (up to approximately 50%) when comparing $\tau^{\text{milp\_init}}$ to τ^{scratch}, or in better solutions $\Phi^{\text{milp\_init}}$ compared to Φ^{scratch}, if the computation time is bounded. For the rightmost column with $n_{\text{t}} = 200$ all results obtained by making use of the information from the MILP solution resulted in a better solution.

n_t	10	20	25	50	80	100	200
τ^{rel}	2.59616	2.61616	2.75217	2.18814	2.25214	2.33214	1.89612
Φ^{rel}	1.34915	1.34741	1.34718	1.34683	1.34659	1.34649	1.34626
τ^{SUR}	0	0	0	0	0	0	0
Φ^{SUR}	1.60251	1.40651	1.37175	1.38366	1.35234	1.35561	1.35328
σ^{SUR}	2	4	4	8	10	14	24
$\frac{n_\mathrm{t}}{t_\mathrm{f}}\eta^{\mathrm{SUR}}$	0.316526	0.47577	0.492702	0.499711	0.483694	0.497768	0.49956
Maximum of $\sigma_{\max} = 3$ switches:							
τ^{bb}	0.00	0.00	0.00	0.00	0.00	0.01	0.14
τ^{cplex}	0.008001	0.016001	0.020002	0.100006	0.412026	0.584037	5.54835
Φ^{milp}	1.60251	1.60251	1.52323	1.67052	1.48912	1.55515	1.70474
σ^{milp}	2	2	3	2	2	3	3
$\frac{n_\mathrm{t}}{t_\mathrm{f}}\eta^{\mathrm{milp}}$	0.316526	0.753893	0.807746	0.970736	1.55474	1.85555	3.49649
τ^{bonmin}	63.212	134.204	164.106	420.134	998.922	1600.61	1800*
Φ^{minlp}	1.60251	1.57489	1.52323	1.38746	1.39481	1.38741	*
σ^{minlp}	2	3	3	2	3	3	*
Maximum of $\sigma_{\max} = 4$ switches:							
τ^{bb}	0.00	0.00	0.00	0.00	0.00	0.01	0.09
τ^{cplex}	0.008	0.016001	0.020001	0.080006	0.428027	1.00806	4.8443
Φ^{milp}	1.60251	1.40651	1.37175	1.36718	1.4576	1.39684	1.40632
σ^{milp}	2	4	4	4	4	4	4
$\frac{n_\mathrm{t}}{t_\mathrm{f}}\eta^{\mathrm{milp}}$	0.316526	0.47577	0.492702	0.671702	0.951219	1.17408	1.98732
τ^{bonmin}	62.8599	106.903	145.381	610.482	1800*	1800*	1800*
Φ^{minlp}	1.60251	1.40651	1.37175	1.35883	1.36079	1.35643	3.36001
σ^{minlp}	2	4	4	4	4	4	4
Maximum of $\sigma_{\max} = 5$ switches:							
τ^{bb}	0.00	0.00	0.00	0.00	0.00	0.02	0.56
τ^{cplex}	0.008001	0.016001	0.020001	0.088006	1.36809	3.1562	32.378
Φ^{milp}	1.60251	1.40651	1.37175	1.36718	1.4576	1.41056	1.40632
σ^{milp}	2	4	4	4	4	5	4
$\frac{n_\mathrm{t}}{t_\mathrm{f}}\eta^{\mathrm{milp}}$	0.316526	0.47577	0.492702	0.671702	0.951219	1.17408	1.98732
τ^{bonmin}	60.0358	114.095	153.706	979.285	1800*	1800*	1800*
Φ^{minlp}	1.60251	1.40651	1.37175	1.35883	1.37073	1.35896	*
σ^{minlp}	2	4	4	4	5	5	*

Table 5.2: Results for Lotka Volterra fishing problem with MILP (5.9) solved by our structure exploiting Branch and Bound algorithm (*bb*) or *cplex*. For reference the original MINLP is solved relaxed (*rel*), with Sum Up Rounding (*SUR*), and with *Bonmin*. τ CPU time, Φ MINLP objective, η MILP objective, σ number of switches. To be continued in Table 5.3.

n_t	10	20	25	50	80	100	200
Maximum of $\sigma_{\max} = 6$ switches:							
τ^{bb}	0.00	0.00	0.00	0.00	0.00	0.01	0.71
τ^{cplex}	0.012001	0.016001	0.016002	0.096007	0.884056	2.92418	41.9546
Φ^{milp}	1.60251	1.40651	1.37175	1.3654	1.45852	1.39149	1.39471
σ^{milp}	2	4	4	6	6	6	6
$\frac{n_t}{t_f}\eta^{\text{milp}}$	0.316526	0.47577	0.492702	0.505287	0.793561	0.86204	1.50351
τ^{bonmin}	59.7637	114.447	147.777	374.347	1800*	1800*	1800*
Φ^{minlp}	1.60251	1.40651	1.37175	1.35233	1.35122	1.35211	1.90096
σ^{minlp}	2	4	4	6	6	6	6
Maximum of $\sigma_{\max} = 7$ switches:							
τ^{bb}	0.00	0.00	0.00	0.00	0.00	0.02	2.39
τ^{cplex}	0.008	0.016001	0.020001	0.096006	1.73611	6.59241	250.428
Φ^{milp}	1.60251	1.40651	1.37175	1.36533	1.45852	1.35481	1.39471
σ^{milp}	2	4	4	7	6	7	6
$\frac{n_t}{t_f}\eta^{\text{milp}}$	0.316526	0.47577	0.492702	0.50359	0.793561	0.858368	1.50351
τ^{bonmin}	57.8996	111.763	147.473	364.447	1800*	1800*	1800*
Φ^{minlp}	1.60251	1.40651	1.37175	1.35233	1.35439	1.3539	*
σ^{minlp}	2	4	4	6	6	6	*
Maximum of $\sigma_{\max} = 8$ switches:							
τ^{bb}	0.00	0.00	0.00	0.00	0.00	0.01	1.15
τ^{cplex}	0.008	0.008	0.016002	0.084005	0.780049	5.34833	388.74
Φ^{milp}	1.60251	1.40651	1.37175	1.38366	1.34997	1.38437	1.35297
σ^{milp}	2	4	4	8	8	8	8
$\frac{n_t}{t_f}\eta^{\text{milp}}$	0.316526	0.47577	0.492702	0.499711	0.602692	0.725728	1.23938
τ^{bonmin}	57.8636	112.363	139.653	356.37	1800*	1800*	1800*
Φ^{minlp}	1.60251	1.40651	1.37175	1.35233	1.34964	1.34956	1.43779
σ^{minlp}	2	4	4	6	8	8	8

Table 5.3: Continuation of Table 5.2.

n_t	10	20	25	50	80	100	200
Maximum of $\sigma_{\max} = 3$ switches:							
τ^{scratch}	63.212	134.204	164.106	420.134	998.922	1600.61	1800*
Φ^{scratch}	1.60251	1.57489	1.52323	1.38746	1.39481	1.38741	*
σ^{scratch}	2	3	3	2	3	3	*
$\tau^{\text{milp\_init}}$	32.75	118.675	128.58	425.671	912.821	1391.44	1800*
$\Phi^{\text{milp\_init}}$	1.60251	1.57489	1.52323	1.38746	1.39481	1.38741	1.70474
$\sigma^{\text{milp\_init}}$	2	3	3	2	3	3	3*
Maximum of $\sigma_{\max} = 4$ switches:							
τ^{scratch}	62.8599	106.903	145.381	610.482	1800*	1800*	1800*
Φ^{scratch}	1.60251	1.40651	1.37175	1.35883	1.36079	1.35643	3.36001
σ^{scratch}	2	4	4	4	4	4	4
$\tau^{\text{milp\_init}}$	31.522	73.2166	89.9296	522.173	1800*	1800*	1800*
$\Phi^{\text{milp\_init}}$	1.60251	1.40651	1.37175	1.35883	1.36079	1.35643	1.40632
$\sigma^{\text{milp\_init}}$	2	4	4	4	4	4	4*
Maximum of $\sigma_{\max} = 5$ switches:							
τ^{scratch}	60.0358	114.095	153.706	979.285	1800*	1800*	1800*
Φ^{scratch}	1.60251	1.40651	1.37175	1.35883	1.37073	1.35896	*
σ^{scratch}	2	4	4	4	5	5	*
$\tau^{\text{milp\_init}}$	30.0979	79.965	119.463	824.568	1800*	1800*	1800*
$\Phi^{\text{milp\_init}}$	1.60251	1.40651	1.37175	1.35883	1.36917	1.35896	1.40632
$\sigma^{\text{milp\_init}}$	2	4	4	4	5	5	4*

Table 5.4: Results for Lotka Volterra fishing problem as MINLP resulting from (5.3). Solutions and computation times for Bonmin runs without initialization (*scratch*) as in Table 5.2 and using the solution Φ^{milp} of (5.9) for initial cutoff in the Branch & Bound tree (*milp_init*). To be continued in Table 5.5.

n_{t}	10	20	25	50	80	100	200
Maximum of $\sigma_{\max} = 6$ switches:							
τ^{scratch}	59.7637	114.447	147.777	374.347	1800*	1800*	1800*
Φ^{scratch}	1.60251	1.40651	1.37175	1.35233	1.35122	1.35211	1.90096
$\sigma^{\mathrm{scratch}}$	2	4	4	6	6	6	6
$\tau^{\mathrm{milp\_init}}$	30.3499	80.8731	129.78	377.04	1800*	1800*	1800*
$\Phi^{\mathrm{milp\_init}}$	1.60251	1.40651	1.37175	1.35233	1.35122	1.35098	1.38382
$\sigma^{\mathrm{milp\_init}}$	2	4	4	6	6	6	6
Maximum of $\sigma_{\max} = 7$ switches:							
τ^{scratch}	57.8996	111.763	147.473	364.447	1800*	1800*	1800*
Φ^{scratch}	1.60251	1.40651	1.37175	1.35233	1.35439	1.3539	*
$\sigma^{\mathrm{scratch}}$	2	4	4	6	6	6	*
$\tau^{\mathrm{milp\_init}}$	29.9899	78.4889	114.931	350.93	1800*	1800*	1800*
$\Phi^{\mathrm{milp\_init}}$	1.60251	1.40651	1.37175	1.35233	1.35354	1.35471	1.39471
$\sigma^{\mathrm{milp\_init}}$	2	4	4	6	6	6	6*
Maximum of $\sigma_{\max} = 8$ switches:							
τ^{scratch}	57.8636	112.363	139.653	356.37	1800*	1800*	1800*
Φ^{scratch}	1.60251	1.40651	1.37175	1.35233	1.34964	1.34956	1.43779
$\sigma^{\mathrm{scratch}}$	2	4	4	6	8	8	8
$\tau^{\mathrm{milp\_init}}$	30.1859	79.313	112.163	359.162	1800*	1800*	1800*
$\Phi^{\mathrm{milp\_init}}$	1.60251	1.40651	1.37175	1.35233	1.34952	1.34977	1.35297
$\sigma^{\mathrm{milp\_init}}$	2	4	4	6	8	8	8*

Table 5.5: Continuation of Table 5.4.

5.6 Summary

We presented a novel method to solve optimal control problems including control functions with a discrete feasible set and switching constraints. The approach is based on a *first discretize, then optimize* approach which results in MINLPs that need to be solved. To avoid the high computational burden of solving the MINLP with standard methods, we propose to decompose the problem into a NLP and a MILP.

Although the MILP solution is not necessarily optimal for the MINLP, it has the advantage to be feasible, to have asymptotic properties as n_t increases, and to be a priori bounded. We proved that it converges against the solution of the nonlinear mixed-integer optimal control problem, if the switching constraint does not become active and the time discretization is refined. If the switching constraint is active, knowledge of system properties, such as the Lipschitz constant of the right-hand side function of the differential equation, allows to formulate an upper bound on the deviation of the MILP based solution from the solution of the relaxed optimal control problem. This upper bound depends linearly on the objective function value of the MILP.

We furthermore analyzed the structure of the convex hull of feasible points to the MILP and discussed why tailored cutting planes are not likely to be computationally beneficial. We presented a tailored Branch and Bound algorithm to cope with this specific structure. We presented numerical results for a benchmark problem in nonlinear mixed-integer optimal control that illustrate the efficiency of our approach.

Future work may concentrate on related optimization problems, such as the minimization of the number of switches subject to a maximal deviation from the optimal solution without switching constraints, or a weighted sum between penalization of switching and performance with respect to the objective. Open questions include also an efficient determination of model-dependent constants that are needed for the error estimations, and the question of reusage of information in adaptive or moving horizon schemes.

6 Uncertainty and Delays in a Conspicuous Consumption Model

The contents of this chapter are based on the paper

[128] T. Huschto, G. Feichtinger, P. Kort, R.F. Hartl, S. Sager, A. Seidl. Numerical Solution of a Conspicuous Consumption Model with Constant Control Delay. *Automatica*, 2011, DOI 10.1016/j.automatica.2011.06.004.

Chapter Summary. We derive optimal pricing strategies for conspicuous consumption products in periods of recession. To that end, we formulate and investigate a two-stage economic optimal control problem that takes uncertainty of the recession period length and delay effects of the pricing strategy into account.

This non-standard optimal control problem is difficult to solve analytically, and solutions depend on the variable model parameters. Therefore, we use a numerical result-driven approach. We propose a structure-exploiting direct method for optimal control to solve this challenging optimization problem. In particular, we discretize the uncertainties in the model formulation by using scenario trees and target the control delays by introduction of slack control functions.

Numerical results illustrate the validity of our approach and show the impact of uncertainties and delay effects on optimal economic strategies. During the recession, delayed optimal prices are higher than the non-delayed ones. In the normal economic period, however, this effect is reversed and optimal prices with a delayed impact are smaller compared to the non-delayed case.

This chapter is special, because the control problem under consideration does not include integer controls. However, it could be easily extended by requiring that the prices need to be from a finite set, as is often the case for airlines, hotels, and so on. Hence, this control problem can be seen as the relaxed control problem, for which an integer solution can be determined in a second step. The optimal control problem is interesting, because it includes an uncertain scenario in which an expected value is optimized subject to worst case constraints. Additionally, it contains delays in the control functions.

6.1 Introduction

We are interested in optimal pricing strategies for conspicuous consumption products in periods of recession, such as the credit crunch recession that started in 2007. Besides a reduction in demand, which is quite usual for a recession, in the credit crunch recession capital markets cease to function. Hence, firms cannot borrow or issue new shares to finance their operations. They need to self-finance their investments [81]:

"...the only option is to try to ride out the recession. But companies can do this only if they have enough liquidity..."

For conspicuous goods demand does not only depend on price, but in addition it depends on the good's reputation, which increases in price. The product's reputation as being expensive allows people to signal their wealth to observers, which in turn increases the reputation of the consumer. Examples of conspicuous goods are luxury hotels [239], expensive cars, or fashionable clothes. The topic of how to price conspicuous goods is treated in [9, 10, 159].

We treat the management of conspicuous goods during the credit crunch recession. The conspicuous goods' manager faces the following trade off. To keep future demand at a high level the manager likes to keep the price of its conspicuous good high. However, during the recession demand as such is low and pricing the good high makes demand even lower. This has detrimental effects for the firm's cash flow, which can bring it into bankruptcy problems, because during the recession capital markets do not function so that the firm needs to have a positive cash level in order to prevent bankruptcy. In [60, 61] this problem was extensively analyzed.

The present chapter extends [60, 61] by establishing a new numerical methodology and by considering a delayed effect of the current price on the firm's reputation. This implies that the good's reputation, which has been built up in the past, is not immediately affected by a price decrease. It takes some time for consumers to get used to the new situation, before a price change really starts to have an effect on the good's reputation.

The very first paper including a delay in an economic model was [135] treating a descriptive business cycle model. Much later, [83] analyzed an optimal growth model with time lags. Starting with the nineties several so-called time-to-build (investment gestation lag) models have been dealt with. Continuous-time deterministic optimal growth models have been enriched by assuming that production occurs with a delay while new capital is installed; see [14, 48, 18, 19, 67]. The methodological background are functional differential equations; for a modified version of Pontryagin's Maximum Principle compare [153]. Additionally, in [247, 249] some related results are presented. In [67] economic models characterized by advanced or delayed time arguments in both the states and controls are discussed. The authors present an algorithm combining the method of steps and a specially tailored shooting method.

It turns out that introducing this delayed effect has considerable qualitative implications for pricing the conspicuous good. In particular, the delayed consumer reaction makes that it is optimal for the firm to set a higher price during the recession and a lower one during the normal period.

We formulate and investigate a two-stage economic optimal control problem that takes uncertainty of the recession period length and delay effects of the pricing strategy into account. This non-standard optimal control problem is difficult to solve analytically, and solutions depend on the variable model parameters. Therefore we use a numerical result-driven approach. We propose a structure-exploiting direct method for optimal control to solve this challenging optimization problem. In particular, we discretize the uncertainties in the model formulation by using scenario trees and target the control delays by introduction of slack control functions.

This chapter is organized as follows: In Section 6.2 we take a closer look at the model. We specify the underlying dynamics for each of the economic stages and deduce the objective function. In Section 6.3 we first collect the algorithmic approaches used to solve a standard multi-stage optimal control problem numerically. Then we reformulate the model using a scenario tree approach and rearrange the emerging scheme to improve performance and simplify the incorporation of the delay via slack control functions. Section 6.4 treats analytical and numerical results and their economic interpretations in detail.

6.2 Model formulation

We consider an economic setting with a recession period followed by a normal economic period. In the following, the value τ denotes the endpoint of the crisis, compare Figure 6.1.

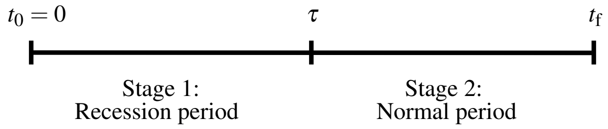


Figure 6.1: Stages $[t_0, \tau]$ and $[\tau, t_\mathrm{f}]$ of the recession model.

The dynamics of our model includes two states. The brand image A of the firm evolves in both periods according to the differential equation

$$\dot{A}(t) = \kappa(\gamma p(t-\sigma) - A(t)) \tag{6.1}$$

with a possible constant control delay $\sigma \geq 0$ in the dynamics of the reputation $A(\cdot)$, retarding the connection between changing the price $p(\cdot)$ and its consequence on the development of $A(\cdot)$. Equation (6.1) covers that, as usual with conspicuous goods, the reputation of the brand goes up with the price, which works positively on demand. Compared to the literature, the delay is a new feature, which captures the fact that consumers first have to get used to a new situation before they adjust their purchase decisions. In particular, if a good is known to be exclusive, a sudden price reduction at first instance does not change this perception. However, after a while consumers "forget" the old situation, implying that they start recognizing that the good is less exclusive, and reputation starts to decrease. Note that if the recession ends at time τ, we still have the direct influence of the price set during the final time interval of length σ of the recession. For a fixed price $\bar{p}$ equation (6.1) yields a steady state of $\bar{A} = \gamma\,\bar{p}$. The available cash $B(\cdot)$ depends on the gains $p(\cdot)\,D(\cdot)$, fixed costs C, and the short-time interest δ, leading to

$$\dot{B}(t) = p(t)D(A(t), p(t)) - C + \delta B(t).$$

Therein the demand D is driven by the brand image and the pricing strategy $p(\cdot)$, which is the control of our problem. It is essentially influenced by the economic stage, i.e., in the normal period (N) we have

$$D_{\mathrm{N}}(A(t),p(t)) = m - \frac{p(t)}{A(t)^{\beta}}, \tag{6.2a}$$

whereas in the recession (R) demand is reduced to

$$D_{\mathrm{R}}(A(t),p(t)) = D_{\mathrm{N}}(A(t),p(t)) - \alpha. \tag{6.2b}$$

The positive constant α measures the strength of the crisis, the parameter $0 < \beta < 1$ is given, and m corresponds to the potential market size.

The objective of the company is to maximize the expected value of profit over the finite or infinite time horizon $[0,t_{\mathrm{f}}]$ of interest. The profit is composed of two parts: the gains of the normal economic period $(\tau,t_{\mathrm{f}}]$ and an impulse dividend of the cash reserve at the end of the recession phase, $B(\tau)$. This dividend is included as the capital market is assumed to become functional again in the normal economic period and firms can freely borrow and lend cash there. Thus, the firm does not need a positive $B(\cdot)$ on $(\tau,t_{\mathrm{f}}]$. For a fixed τ and a given discount rate r, the objective function is calculated as

$$\Phi(\tau) := \mathrm{e}^{-r\tau}B(\tau) + \int_{\tau}^{t_{\mathrm{f}}} \mathrm{e}^{-rt}\left(p(t)D_{\mathrm{N}}(A(t),p(t)) - C\right)\,\mathrm{d}t, \tag{6.3}$$

being the sum of these two components, resulting in the optimal control problem

$$\begin{aligned}
&\max_{p(\cdot)}\ \Phi(\tau) \\
&\text{s.t.}\quad \dot{A}(t) = \kappa(\gamma p(t-\sigma) - A(t)), && t \in [0,t_{\mathrm{f}}], \\
&\qquad\ p(t) = \eta(t), && t \in [-\sigma,0], \\
&\qquad\ \dot{B}(t) = p(t)D_{\mathrm{R}}(A(t),p(t)) - C + \delta B(t), && t \in [0,\tau], \\
&\qquad\ A(0) = A_0, \quad B(0) = B_0, \\
&\qquad\quad 0 \le D_{\mathrm{R/N}}(A(t),p(t)), && t \in [0,t_{\mathrm{f}}], \\
&\qquad\ p(t) \ge 0, && t \in [0,t_{\mathrm{f}}], \\
&\qquad\ B(t) \ge 0, && t \in [0,\tau]
\end{aligned} \tag{6.4}$$

with $D_{\mathrm{R/N}}(A(t),p(t))$ given as in (6.2) and $B(t)$ negligible in the normal period $(\tau,t_{\mathrm{f}}]$. However, typically the recession length τ is not known beforehand to decision makers. An individual firm also has no influence on when the recession ends. Therefore, we assume that the length of the recession period τ is an exponentially distributed random variable. The goal is to maximize the expectation value of the net present value (NPV) at time τ, i.e., the objective function Φ

weighted by the exponential probability density function with rate parameter λ,

$$\max_{p(\cdot)} \mathbb{E}\left[\mathrm{NPV}(\tau)\right] := \max_{p(\cdot)} \int_0^{t_\mathrm{f}} \lambda\, \mathrm{e}^{-\lambda\tau}\, \Phi(\tau)\, \mathrm{d}\tau \tag{6.5}$$

subject to the constraints given in (6.4) for all $0 \le \tau \le t_\mathrm{f}$.
This problem is a non-standard optimal control problem in the sense that uncertainty and control delays are present, making analytical investigations difficult.[1] Therefore, we propose a different approach in the next section.

6.3 Numerical treatment

We propose to use reformulations to transfer the optimal control problem (6.5) into a more standard form that can be efficiently solved. In Section 6.3.1 we present such a standard multi-stage formulation that is more general than the one in Section 4.2.1 and give references to Bock's direct multiple shooting method. In Section 6.3.2 we present a discretization of the uncertainty, and in Section 6.3.3 a reformulation of the time delays. In both cases alternatives are discussed.

6.3.1 The Direct Multiple Shooting Approach

Efficient numerical methods have been developed to solve multi-stage, nonlinear optimal control problems of the following form

$$\begin{aligned}
\max_{x_i(\cdot),u_i(\cdot),q,t_i} \quad & \sum_{i=0}^{M-1} \int_{t_i}^{t_{i+1}} L_i(x_i(t),u_i(t),q)\, \mathrm{d}t + E_i(x(t_{i+1}),q) && (6.6a)\\
\text{s.t.} \quad & \dot{x}_i = f_i(x_i(t),u_i(t),q), && (6.6b)\\
& x_{i+1}(t_{i+1}) = f_{\mathrm{tr},i}(x_i(t_{i+1}),q), && (6.6c)\\
& 0 \le c_i(x_i(t),u_i(t),q) && (6.6d)\\
& 0 = r_\mathrm{eq}(x_0(t_0),x_1(t_1),\dots,q), && (6.6e)\\
& 0 \le r_\mathrm{ineq}(x_0(t_0),x_1(t_1),\dots,q), && (6.6f)
\end{aligned}$$

with $t \in [t_i,t_{i+1}]$ and $i = 0,\dots,M-1$. The optimization problem (6.6) couples M model stages via explicit transitions (6.6c) and interior point constraints (6.6e-6.6f). The differential states $x_i : [t_0,t_M] \mapsto \mathbb{R}^{n_{x_i}}$ and the control functions $u_i : [t_0,t_M] \mapsto \mathbb{R}^{n_{u_i}}$ and control values $q \in \mathbb{R}^{n_q}$ need to be feasible for the path- and control constraints (6.6d) and the ordinary differential equations (ODEs) (6.6b).

[1] In [62] it is shown that an important class of models with delays can be transformed into equivalent problems without delays. However, the present model does not fit in this family. This is because the control p appears with a delay in one state equation and without in the other one. Hence, it is not possible to eliminate the delay using a time transformation.

An overview of different methods can be found, e.g., in [37]. We propose to use Bock's direct multiple shooting method to solve problems of type (6.6). It transforms the optimal control problem into a Nonlinear Program (NLP) by discretizing the space of admissible control functions $u(\cdot)$ and the path constraints (6.6d). The solutions of the ODEs (6.6b) are obtained by a decoupled integration on a multiple shooting grid, starting from artificial intermediate variables. Continuity of the differential states is assured by means of an inclusion of matching conditions into the NLP.

For details on this method we refer as in Section 4.2.1 to [44, 167, 168, 143]. At this place we would only like to remind the reader of one of the advantages of the direct multiple shooting method. As control functions, constraints, and multiple shooting variables are discretized on a common time grid, the Hessian of the Lagrangian is block structured for linearly coupled point constraints $r_.(\cdot)$. For $i \neq j$ we have

$$\frac{\nabla^2 \mathscr{L}(w_1, \dots, w_N)}{\partial w_i \, \partial w_j} = 0 \tag{6.7}$$

for variable vectors w_i that subsume all variables of the i-th multiple shooting interval. This allows applying Broyden–Fletcher–Goldfarb–Shanno (BFGS) updates to every single one of the N multiple shooting blocks [44]. These high-rank updates typically lead to a fast accumulation of higher order information and thus to fast convergence [181]. This feature becomes important in the context of the following reformulations of problem (6.5).

6.3.2 Discretizing the probability density function

To solve problem (6.5) at least approximatively, we need to reformulate it. We discretize the exponential distribution of the random variable τ by defining a time grid

$$0 = \tau_0 < \tau_1 < \ldots < \tau_n < t_\text{f}.$$

In the following, switches from recession period to normal stage are only possible at these times τ_i with $i = 1, \dots, n$. The recession ends at τ_i with probability $\mathbb{P}_i$. We use an equidistant discretization, resulting in a geometric distribution

$$\mathbb{P}_i = \int_{\tau_{i-1}}^{\tau_i} \lambda \, \mathrm{e}^{-\lambda t} \, \mathrm{d}t = \, \mathrm{e}^{-\lambda \tau_{i-1}} - \, \mathrm{e}^{-\lambda \tau_i}, \tag{6.8a}$$

for $i = 1, \dots, n-1$, and

$$\mathbb{P}_n = 1 - \sum_{j=1}^{n-1} \mathbb{P}_j. \tag{6.8b}$$

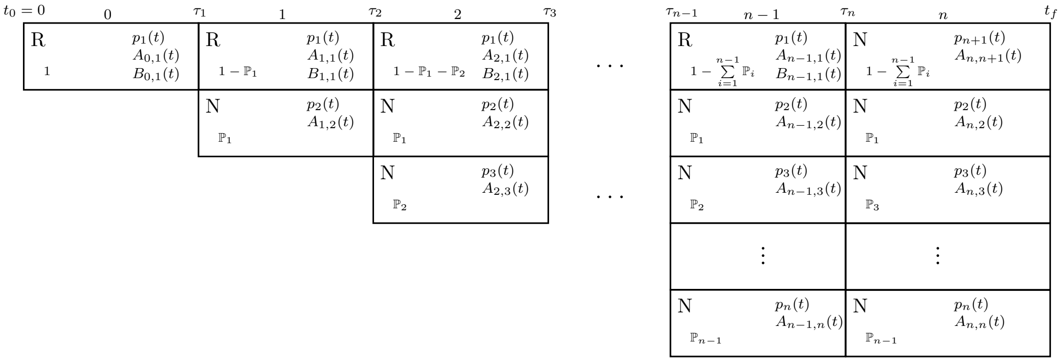


Figure 6.2: Controls and variables in the multi-stage formulation of problem (6.4) with associated probabilities and in a (R)ecession or a (N)ormal period.

The discretized distribution can be used to reformulate the maximization of the expected value as a multi-stage optimal control problem of type (6.6), by using a scenario tree. However, this formulation is not unique. One possibility is to use a staircase-like approach, increasing the number of variables as the number of possible recession ends τ_i increases. This approach is illustrated schematically in Figure 6.2 and results in $M = n+1$ model stages, where n is the number of discretizations of the probability density function. The dimensions $n_{x_i} = 2+i$ of differential states and $n_{u_i} = 1+i$ of control functions, $i = 0,\dots,M-1$, are different on the model stages. The transition functions (6.6c) are defined by

$$A_{i,j}(\tau_i) = A_{i-1,j}(\tau_i), \quad 1 \le j \le i, \tag{6.9a}$$

$$A_{i,i+1}(\tau_i) = A_{i-1,1}(\tau_i), \tag{6.9b}$$

$$B_{i,1}(\tau_i) = B_{i-1,1}(\tau_i), \tag{6.9c}$$

for all model stages $i = 1,\dots,n-1$, and

$$A_{n,n+1}(\tau_n) = A_{n-1,1}(\tau_n). \tag{6.9d}$$

At each τ_i one has to distinguish between transitions (6.9a), (6.9c) of the brand image A and the cash B for the ongoing recession and the initialization (6.9b), (6.9d) of the additional differential states $A_{i,i+1}$ for the normal period beginning at τ_i, compare Figure 6.2.

The second possibility is to use linearly coupled point constraints of type (6.6e) instead of transitions to initialize the new variables. All possible scenarios at τ_i are concatenated, resulting in $M = 2n$ model stages. This “flat” arrangement of stages is shown in Figure 6.3.

In contrast to the first formulation, the model stage dimensions $n_{x_i} = 2$ for $i = 0,\dots,n-1$ and $n_{x_i} = 1$ for $i = n,\dots,M-1$ of differential states and $n_{u_i} = 1$ for $i = 0,\dots,M-1$ of controls are

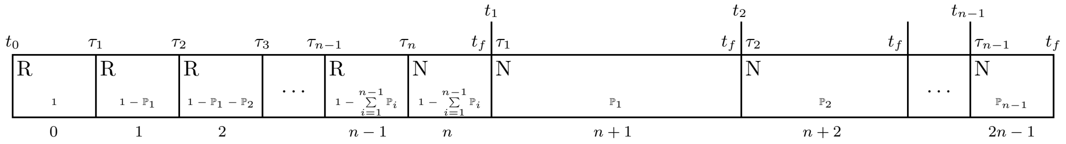


Figure 6.3: Rearranged scheme for the discretization of the random end time τ of the recession. Again, the symbols denote the (R)ecession and (N)ormal stage, as well as the appropriate probabilities.

(almost) constant. The coupled point constraints (6.6e) are given by

$$A_{i,1}(t_{i-n}) = A_{i-n-1,1}(\tau_{i-n}), \quad n+1 \le i \le 2n-1. \tag{6.10a}$$

The first n stages are recession periods with continuous transitions of all states. They differ in the objective function. The transition from the last recession stage n to the subsequent normal period that starts at $t = \tau_n$ is continuous, too. However, the model stage lengths of this approach vary. While all n recession stages have the constant duration $h = \tau_i - \tau_{i-1}$, the n normal period stages have a length of $t_\mathrm{f} - \tau_i$, $i = 1, \dots, n$.

Then we obtain for the staircase-like approach to discretize the probability density function, $k = 1$, the objective function

$$\begin{aligned}
&\Phi_i^1(\tau_i, A_{i,\cdot}(t), B_{i-1,1}(\tau_i), p(t), \bar{\mathbb{P}}_i) \\
&\quad = \mathbb{P}_i\, \mathrm{e}^{-r\tau_i} B_{i-1,1}(\tau_i) + \sum_{j=1}^{i} \mathbb{P}_j \int_{\tau_i}^{\tau_{i+1}} \mathrm{e}^{-rt} \left(p(t) D_\mathrm{N}(A_{i,j+1}(t), p(t)) - C\right) \mathrm{d}t
\end{aligned}$$

for $i = 1, \dots, n$, the transition (tr) functions

$$f^1_{\mathrm{tr}A,i}(A_{i-1,j}(\tau_i)) = \begin{cases} A_{i-1,j}(\tau_i); & 1 \le i \le n-1,\ 1 \le j \le i, \\ A_{i-1,1}(\tau_i); & 1 \le i \le n,\ j = i+1, \end{cases} \tag{6.11a}$$

$$f^1_{\mathrm{tr}B,i}(B_{i-1,1}(\tau_i)) = B_{i-1,1}(\tau_i),\ 1 \le i \le n-1, \tag{6.11b}$$

and the coupled point constraints functions

$$r^1_{\mathrm{eq},i} \equiv 0, \tag{6.11c}$$

where $\bar{\mathbb{P}}_i = (\mathbb{P}_1, \mathbb{P}_2, \dots, \mathbb{P}_i)$.

The concatenated approach, $k = 2$, is defined by the respective functions

$$\begin{aligned}
&\Phi_i^2(\tau_i, A_{n+i,1}(t), B_{i-1,1}(\tau_i), p(t), \mathbb{P}_i) \\
&\quad = \mathbb{P}_i\, \mathrm{e}^{-r\tau_i} B_{i-1,1}(\tau_i) + \mathbb{P}_i \int_{\tau_i}^{t_\mathrm{f}} \mathrm{e}^{-rt} \left(p(t) D_\mathrm{N}(A_{n+i,1}(t), p(t)) - C\right)\ \mathrm{d}t,
\end{aligned}$$

for $i = 1,\dots,n$,

$$f^2_{\mathrm{tr}A,i}(A_{i-1,1}(\tau_i)) = A_{i-1,1}(\tau_i),\ 1 \le i \le n, \tag{6.12a}$$

$$f^2_{\mathrm{tr}B,i}(B_{i-1,1}(\tau_i)) = B_{i-1,1}(\tau_i),\ 1 \le i \le n-1, \tag{6.12b}$$

and

$$r^2_{\mathrm{eq},i}(A_{i,1}(t_{i-n}),A_{i-n-1,1}(\tau_{i-n})) = A_{i,1}(t_{i-n}) - A_{i-n-1,1}(\tau_{i-n}),\ n+1 \le i \le M-1. \tag{6.12c}$$

6.3.3 Reformulation of the time delays

In [49] two possibilities are given to reformulate an optimal control problem with delayed equation of motion as in (6.4) into an instantaneous problem.

The first approach splits the time horizon t_f into m parts of length σ and formulates the system dynamics separately on each of the resulting intervals. By interpreting them as independent and introducing new state and control variables we can formulate a system of m differential equations on the time horizon $[0,\sigma]$. This can be used to reformulate the original optimal control problem. Furthermore, one has to introduce coupled boundary conditions to ensure the continuity of the state variable. The approach may give additional insight from an analytical point of view, compare [49]. However, it requires the determination of $m-1$ control paths in the interval $[0,\sigma]$. For small values of the delay σ this results in a large number of state and control functions.

Therefore, we prefer a different reformulation. We introduce a second control function $u_2(t) = p(t)$ that denotes the unretarded control at time t, whereas $u_1(t) = p(t-\sigma)$ characterizes the delayed one. They are coupled via equalities $u_1(t) = u_2(t-\sigma)$ for $t \ge \sigma$ and $u_1(t) = \eta(t-\sigma)$ for $0 \le t \le \sigma$.

Taking either staircase (6.11) or flat (6.12) discretization of uncertainty presented in the previous Section, $k = 1,2$, we obtain

$$\max_{u_1(\cdot),u_2(\cdot)} \sum_{i=1}^{n} \Phi_i^k(\tau_i, A_{\chi^k(i),\cdot}(t), B_{i-1,1}(\tau_i), u_2(t), \bar{\mathbb{P}}_i) \tag{6.13a}$$

$$\begin{aligned}
\text{s.t. } \dot{A}_{i,j}(t) &= \kappa(\gamma u_1(t) - A_{i,j}(t)), && t \in [0,t_\mathrm{f}], 0 \le i \le M-1,\ j \in J^k, && (6.13\mathrm{b})\\
\dot{B}_{i,1}(t) &= u_2(t) D_\mathrm{R}(A_{i,1}(t),u_2(t)) - C + \delta B_{i,1}(t), && t \in [0,\tau_i], 0 \le i \le n-1, && (6.13\mathrm{c})\\
u_1(t) &= \eta(t-\sigma), && t \in [0,\sigma], && (6.13\mathrm{d})\\
u_1(t) &= u_2(t-\sigma), && t \in [\sigma,t_\mathrm{f}], && (6.13\mathrm{e})\\
A_{0,1}(0) &= A_0, \quad B_{0,1}(0) = B_0, && && \\
0 &\le D_{\mathrm{R,N}}(A_{i,j}(t),u_2(t)), && t \in [0,t_\mathrm{f}], && (6.13\mathrm{f})\\
u_1(t) &\ge 0, \quad u_2(t) \ge 0, && t \in [0,t_\mathrm{f}], && (6.13\mathrm{g})\\
B_{i,1}(t) &\ge 0, && t \in [0,\tau_i], 1 \le i \le n-1, && (6.13\mathrm{h})
\end{aligned}$$

$$A_{i,j}(\tau_i) = f^k_{\mathrm{tr}A,i}(A_{i-1,j}(\tau_i)), 1 \leq i \leq n,\ j \in J^k, \tag{6.13i}$$

$$B_{i,1}(\tau_i) = f^k_{\mathrm{tr}B,i}(B_{i-1,1}(\tau_i)), \qquad 1 \leq i \leq n-1, \tag{6.13j}$$

$$0 = r^k_{\mathrm{eq},i}(A_{i,1}(t_{i-n}), A_{i-n-1,1}(\tau_{i-n})), \qquad n+1 \leq i \leq M-1, \tag{6.13k}$$

where $\chi^1(i) := i$, $\chi^2(i) := n+i$, $J^1 := \{j \,|\, 1 \leq j \leq i+1\}$, $J^2 := \{j \,|\, j = 1\}$.

This problem still contains a delayed term, but it is not apparent in the system dynamics anymore. It has moved to a constraint (6.13e) on the controls. This can be efficiently dealt with the multiple shooting method we introduced in Section 6.3.1 for the special case of a constant delay.

6.4 Results

As suggested in [60, 61], we use the following set of parameters in our numerical treatment:

$$\begin{array}{llll} \kappa = 2.0, & \gamma = 5.0, & C = 7.5, & \delta = 0.05, \\ m = 3.0, & \beta = 0.5, & r = 0.1, & \lambda = 0.5, \\ \alpha_1 = 0.7, & \alpha_2 = 0.836, & \alpha_3 = 1.25. & \end{array} \tag{6.14a}$$

The choice for parameters r, δ, and λ is based on the assumption that we measure time in years and that the expected duration of the recession is two years. We set β assuming that an increase in reputation will influence less and less customers. The more fashionable the product is, the more specialized is its market niche. See [61] for a motivation of the remaining parameters.

A key result of [61] was that the authors were able to distinguish three different types of recessions corresponding to the severity of the demand reduction and the resulting optimal strategy. Following their results, the values of the parameter α indicate a mild ($\alpha_1 = 0.7$), intermediate ($\alpha_2 = 0.836$), and severe ($\alpha_3 = 1.25$) economic crisis.

Due to the discretization of τ we need to further specify the last possible endpoint of the recession,

$$\tau_n = 20. \tag{6.14b}$$

This implies that in this context the probability that the recession persists longer than that is low, i.e., $\mathbb{P}[\tau > 20] = 4.54 \cdot 10^{-5}$. For the control delay we choose

$$\sigma = 0.25. \tag{6.14c}$$

To accomplish this, two equidistant discretization step lengths are applied, first with $n_1 = 20$, i.e., $h = \tau_i - \tau_{i-1} = 1.0$, and $n_2 = 40$, i.e., $h = 0.5$. Each of them is combined with four shooting nodes per time unit, i.e., per year. Then condition (6.13e) can be implemented via interior point constraints applied on the shooting nodes.

For convenience, the overall final time t_f is chosen to be

$$t_\mathrm{f} = 21 \text{ (years)}, \tag{6.14d}$$

so that we definitely have a small normal period of one year in all possible stages.

Finally, in the subsequent sections we provide some computational results. They are obtained with the following combinations of number of discretization points n, recession parameter α, initial values (A_0, B_0), and initial price paths η for the delayed model, cf. Table 6.1.

In Section 6.4.1 we analyze the computational performance of the various reformulations presented in the previous section. In Section 6.4.2 we derive some analytical insight into the problem structure. More economic insight can be gained from the computational results in Section 6.4.3.

6.4.1 Computational performance

As discussed in Sections 6.3.2 and 6.3.3 different mathematically equivalent reformulations of the optimal control problem (6.4) exist. However, they are by no means equivalent from a computational point of view.

Table 6.2 compares the computational performance of the two different approaches to discretize the uncertainty. With the staircase formulation (6.11) (Figure 6.2) the overall time horizon is quite small. However, the number of state variables is increased compared to the concatenated arrangement, leading to more steps of the error-controlled, adaptive integrator. More significant, however, is the impact of more blocks in the Hessian of the Lagrangian. They are used for high-rank updates, compare Section 6.3.1. This leads to a drastic increase in local convergence and hence to a decrease of the number of sequential quadratic programming (SQP) iterations [168] and overall computation time, as can be seen in Table 6.2 for the case $\sigma = 0$. These results carry over to the case with $\sigma > 0$, therefore we concentrate on the formulation (6.12) visualized in Figure 6.3.

As already observed in [49], the first approach suggested in Section 6.3.3 to handle time lags σ is computationally inferior to the second one, although it might be interesting from an analytical point of view. E.g., for scenarios 4–12 the number of 1800 additional state and 1799 control functions needs to be included. Therefore, we use the second formulation in the following for our calculations. Table 6.3 gives an overview over the moderate increase in the dimension of the resulting nonlinear program.

Table 6.4 gives an indication of the computational expense for including delays. The main part of the computation is needed for the condensing algorithm, see [44, 167], which is almost identical for both cases, as the state dimension is independent of σ. The main extra cost is solving the quadratic programs, as the runtime depends crucially on the number of control variables. Therefore, asymptotically for $\sigma > 0$ getting smaller and smaller, the quadratic programming (QP) runtime becomes more and more dominant.

Scenario	n	α	A_0	B_0	η
1	20	0.7	10.0	5.0	-
2	20	0.836	20.0	5.0	-
3	20	1.25	100.0	100.0	-
4	40	0.7	10.0	5.0	7.406785
5	40	0.7	0.1	5.0	4.296460
6	40	0.7	10.0	2.0	7.088001
7	40	0.7	$\bar{A}_{\mathrm{d}}^{\mathrm{N}}$	5.0	$\bar{p}_{\mathrm{d}}^{\mathrm{N}}$
8	40	0.7	$\bar{A}_{\mathrm{d}}^{\mathrm{N}}$	1.0	$\bar{p}_{\mathrm{d}}^{\mathrm{N}}$
9	40	0.7	$\bar{A}_{\mathrm{d}}^{\mathrm{N}}$	0.1	$\bar{p}_{\mathrm{d}}^{\mathrm{N}}$
10	40	0.836	0.1	10.0	3.917962
11	40	0.836	0.1	10.0	3.5
12	40	0.836	0.1	10.0	3.0
13	40	0.836	0.1	10.0	2.5
14	40	0.836	20.0	5.0	8.153575
15	40	0.836	0.1	8.0	3.917948
16	40	0.836	25.0	3.5	8.671824
17	40	0.836	$\bar{A}_{\mathrm{d}}^{\mathrm{N}}$	1.0	$\bar{p}_{\mathrm{d}}^{\mathrm{N}}$
18	40	0.836	0.1	7.05	-
19	40	0.836	63.0	0.05	-
20	40	0.836	0.1	9.8	3.5
21	40	0.836	73.5	0.1	12.517549
22	40	1.25	100.0	100.0	10.751307
23	40	1.25	0.1	100.0	2.924618
24	40	1.25	40.0	80.0	7.855208
25	40	1.25	80.0	50.0	9.922934
26	40	1.25	0.1	60	2.924617
27	40	1.25	$\bar{A}_{\mathrm{d}}^{\mathrm{N}}$	50.0	$\bar{p}_{\mathrm{d}}^{\mathrm{N}}$
28	40	1.25	$\bar{A}_{\mathrm{d}}^{\mathrm{N}}$	70.0	-
29	40	1.25	0.1	76.0	-
30	40	1.25	$\bar{A}_{\mathrm{d}}^{\mathrm{N}}$	71.5	$\bar{p}_{\mathrm{d}}^{\mathrm{N}}$
31	40	1.25	0.1	79.5	2.924580

Table 6.1: Different scenarios used for computational performance tests and visualizations. Note that some of these scenarios are used in both a delayed ($\sigma = 0.25$) and undelayed model ($\sigma = 0$), others in only one of them. In undelayed settings η is obsolete and denoted by "-".

	Scheme (6.11)		Scheme (6.12)	
Scenario	# of SQP	t (s)	# of SQP	t (s)
1	846	5259	51	1341
2	829	1312	35	835
3	858	1411	102	2969
4	1254	67131	102	21443
14	1716	93773	48	9615
22	915	47285	102	24163

Table 6.2: Comparison of the different schemes for discretizing τ, see (6.11), (6.12), and Figures 6.2, 6.3, respectively. The results correspond to the undelayed case, i.e., $\sigma = 0$. The faster convergence of (6.12) (recognizable in SQP iterations and runtime) is due to the high-rank updates mentioned in Section 6.3.1. The scenarios are listed in Table 6.1.

	Undelayed model		Delayed model	
	$n = 20$	$n = 40$	$n = 20$	$n = 40$
discr. points	940	1840	940	1840
variables	3797	7437	4738	9278
eq. constraints	2855	5595	3797	7437
ineq. constraints	7594	14874	9476	18556

Table 6.3: Comparison of the size of the resulting NLP for the delayed and the undelayed model.

	Undelayed model		Delayed Model	
Scenario	# of SQP	t (s)	# of SQP	t (s)
6	71	14103	60	20238
7	102	24515	98	28422
16	70	12896	102	28787
17	69	14796	82	24466
24	81	18114	81	22166
27	101	24456	101	29404

Table 6.4: Number of iterations and CPU time for undelayed and delayed scenarios. The computational effort is moderately higher, when delays are taken into account.

6.4.2 Analytical results

We deduce analytical results that help us to obtain a better insight into the qualitative changes related to the introduction of the time lag σ. We investigate the steady state in the normal period of our model (6.4) and compare it with the result of the undelayed case, i.e., $\sigma = 0$.

The integral term of $\Phi(\cdot)$ in (6.3) corresponds to the normal economic period, where the capital markets are working again and we are not using the cash state B anymore. Let $\bar{A}^{\mathrm{N}}_{\mathrm{d/nd}}$ and $\bar{p}^{\mathrm{N}}_{\mathrm{d/nd}}$ denote the normal period's steady state brand image and price in the (d)elayed and the u(nd)elayed case, respectively.

By using Pontryagin's Maximum Principle [118] we calculate

$$\bar{A}^{\mathrm{N}}_{\mathrm{nd}} = \left(\frac{\gamma m(r+\kappa)}{2(r+\kappa)-\beta\kappa}\right)^{\frac{1}{1-\beta}}, \quad \bar{p}^{\mathrm{N}}_{\mathrm{nd}} = \frac{\bar{A}^{\mathrm{N}}_{\mathrm{nd}}}{\gamma}. \tag{6.15a}$$

In the model's delayed version the maximum principle is far more complex, see [83]. However, in the normal period the stationary state of the corresponding one-dimensional problem can be derived using the results in [248]. We substitute

$$F(t) := F(A(t), p(t)) = p(t)\left(m - \frac{p(t)}{A(t)^{\beta}}\right) - C$$

and obtain the Hamiltonian

$$\mathscr{H} = \mathrm{e}^{-rt}F(t) + \mu(t+\sigma)\cdot\kappa\gamma p(t) - \mu(t)\cdot\kappa A(t)$$

with the co-state variable $\mu(t)$. This induces the system

$$\begin{aligned}
\dot{A}(t) &= \kappa(\gamma p(t-\sigma) - A(t))\\
\dot{p}(t) &= \frac{1}{F_{pp}(t)}\big((r+\kappa)F_p(t) + \kappa\gamma\,\mathrm{e}^{-r\sigma}F_A(t+\sigma) - F_{pA}(t)\dot{A}(t)\big)
\end{aligned}$$

that directly gives us the stationary price $\bar{p}^{\mathrm{N}}_{\mathrm{d}}$. Further on, it yields

$$\frac{(r+\kappa)\,\mathrm{e}^{r\sigma}}{\kappa\gamma} = -\frac{F_A(t+\sigma)}{F_p(t)}$$

and, therefore, the equality

$$(r+\kappa)\,\mathrm{e}^{r\sigma} = -\frac{\beta\kappa(\bar{A}^{\mathrm{N}}_{\mathrm{d}})^{1-\beta}}{\gamma m - 2(\bar{A}^{\mathrm{N}}_{\mathrm{d}})^{1-\beta}}$$

that determines the stationary state of the brand image

$$\bar{A}_{\mathrm{d}}^{\mathrm{N}} = \left(\frac{\gamma m(r+\kappa)\,\mathrm{e}^{r\sigma}}{2(r+\kappa)\,\mathrm{e}^{r\sigma} - \beta\kappa} \right)^{\frac{1}{1-\beta}}, \quad \bar{p}_{\mathrm{d}}^{\mathrm{N}} = \frac{\bar{A}_{\mathrm{d}}^{\mathrm{N}}}{\gamma}. \tag{6.15b}$$

The latter result obviously includes the special case (6.15a). Our parameters (6.14) determine the values

$$\bar{A}_{\mathrm{nd}}^{\mathrm{N}} = 96.899414, \qquad \bar{p}_{\mathrm{nd}}^{\mathrm{N}} = 19.379883, \tag{6.16a}$$

$$\bar{A}_{\mathrm{d}}^{\mathrm{N}} = 95.421259, \qquad \bar{p}_{\mathrm{d}}^{\mathrm{N}} = 19.084252. \tag{6.16b}$$

Those coincide with the numerical results we obtained. One can see the impact of the delay very clearly. The benefit of keeping the price up is obtained later in the delayed world, while the benefit of reducing it (with instantaneous profit) is still obtained immediately.

In the recession period the verification and calculation of steady states cannot be done this straightforwardly. Further on, the so-called weak Skiba curves[2] play an important role. While the authors of [60] were able to derive several results of the non-delayed case analytically, for the delayed model this is impeded much more.

6.4.3 Computational results

In our approach to discretize problem (6.4) we assume a finite and discrete grid of possible switching times τ_i. We think that this transformation to the finite-time case is well justified, as the influence of the errors caused by the discretization are small. The intervals between τ_i are short and the probability (6.8b) for switching the stage at the last possible time τ_n is only marginally higher than it would be in the infinite case.

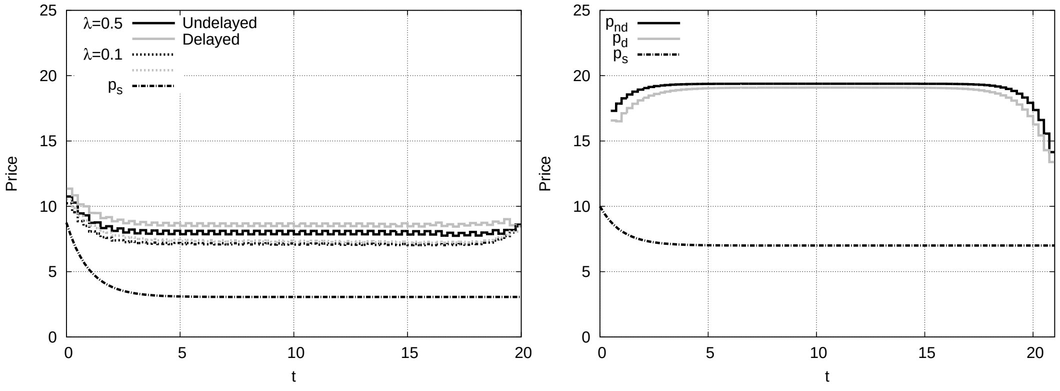
In [60] possible pricing strategies in recession periods are explained depending on the value of α. Additionally, the impact of these pricing policies on the development of the reputation A and the cash B is depicted. In the delayed world the behavior of the firm is qualitatively similar. In a severe crisis ($\alpha_3 = 1.25$) the brand image and/or cash required to avoid bankruptcy are particularly large. The milder the crisis is the less reputation/cash is needed. In all cases the cash state diverges to infinity if the firm survives with certainty.

The main result of our analysis of problem (6.4) is the relation

$$p_{\mathrm{d}}(t) > p_{\mathrm{nd}}(t),\ 0 \le t \le \tau, \quad p_{\mathrm{d}}(t) < p_{\mathrm{nd}}(t),\ \tau \le t \le t_{\mathrm{f}}, \tag{6.17}$$

which can be seen in Figure 6.4.

[2] Also known as threshold or weak DNSS curve referring to early contributions of [71], [220, 221], and [226]; see also [118]. Weak Skiba refers to the threshold property of this curve separating different long-term solutions. Which strategy has to be applied is history-dependent and, thus, particularly depends on the initial state values.



(a) Recession in $[0, \tau_n]$ (b) Normal period in $(\tau_1, t_f]$

Figure 6.4: Exemplary price paths of
(a) a recession period lasting until τ_n (using Scenario 22). During the recession $p_\mathrm{d} > p_\mathrm{nd}$ holds, but the difference in between depends on the size of the rate parameter λ.
(b) a normal economic stage for the same scenario setting. By way of better illustration this figure shows price paths of a normal period beginning already at time τ_1. Note that neither λ nor the strength α of the recession have any influence on these paths.
For comparison, p_s shows the static optimization price.

The optimal solution of the normal period follows the results of Section 6.4.2. Due to the delay σ there is a less direct effect of the price p_d on the dynamics of the brand image $\dot{A}$. This reduces the incentive to set a high price, as a lower price raises revenues, which consequently raises the value of the objective function immediately.

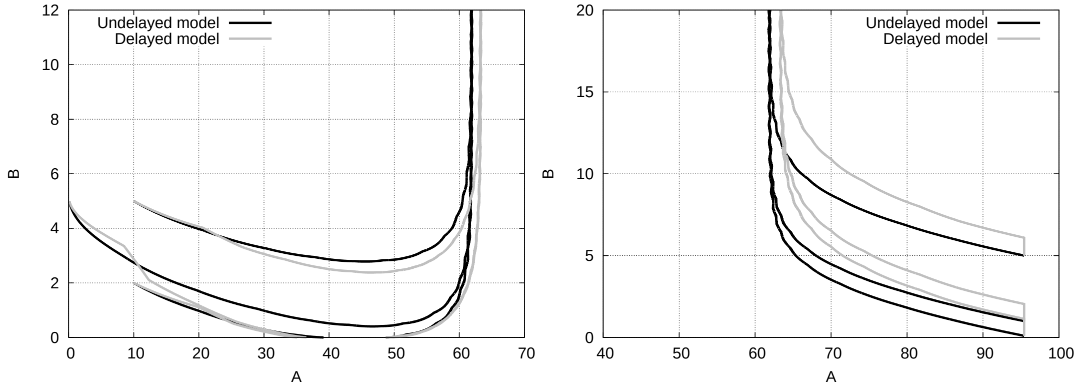
In the recession period, however, the opposite relation holds. A direct consequence of this is visible in Figures 6.5 and 6.6: The vertical line indicating the divergence of the cash state B in an infinite horizon setting is shifted to a value $\bar{A}_\mathrm{d}^\mathrm{R}$ of reputation that is higher than the respective value $\bar{A}_\mathrm{nd}^\mathrm{R}$ in the non-delayed case.

While the negative effect of smaller revenues with higher prices (independent of the economic period) is the same for both the delayed and the undelayed case, there are also two positive aspects of increasing the price p_d.

The first effect is that the brand image A increases as well during the recession, implying that the bankruptcy probability reduces. This effect is stronger the less the delay σ is. Hence, this first impact is the strongest in the non-delayed case.

Given that the recession terminates somewhere during the next time interval of duration σ, the second effect of increasing p_d is that the reputation goes up after the recession, implying that the revenue of the normal period rises. This effect occurs with the probability $\mathbb{P}[\tau \in [t, t+\sigma]]$ that the recession will be over during the next interval of length σ, hence, it is stronger the larger the delay is. But it is completely absent in the undelayed case.

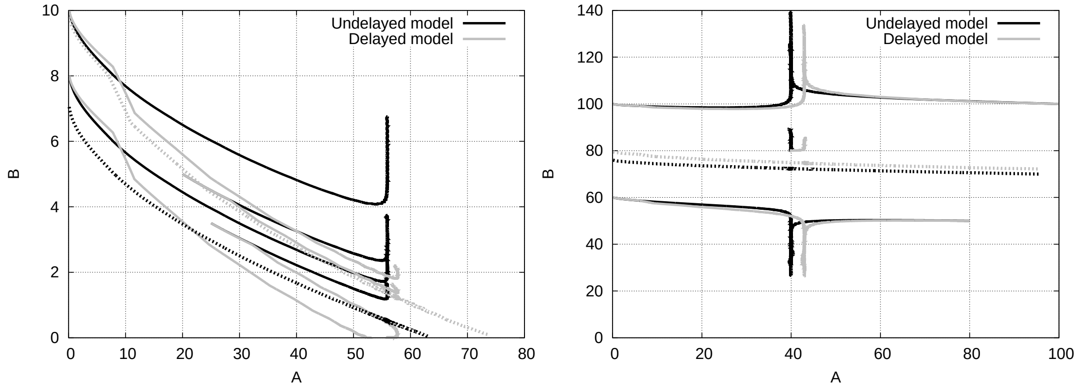
According to the first effect, which is comparable to the impact in the normal period, it holds



(a)

(b)

Figure 6.5: Evolution of optimal trajectories over time in a phase diagram with brand image $A(\cdot)$ and capital $B(\cdot)$. They start in (A_0, B_0) according to Table 6.1 and evolve until $(A(\tau_n), B(\tau_n))$. Optimal solutions of a delayed ($\sigma = 0.25$) and the undelayed ($\sigma = 0$) model are shown for a mild recession ($\alpha_1 = 0.7$), if we assume that for $t \in [-\sigma, 0]$
(a) the recession has been present (Scenarios 4–6 (from top to bottom)),
(b) a steady state normal economic period existed, i.e., $A_0 = \bar{A}_\mathrm{d}^\mathrm{N}$, $\eta = \bar{p}_\mathrm{d}^\mathrm{N}$ (Scenarios 7–9).
Due to the introduction of the delay the recession's steady state of the brand image $\bar{A}_\mathrm{d}^\mathrm{R}$ (and correspondingly $\bar{p}_\mathrm{d}^\mathrm{R}$) is greater than in the undelayed case.



(a) $\alpha_2 = 0.836$

(b) $\alpha_3 = 1.25$

Figure 6.6: Phase diagram as in Figure 6.5(a) for an intermediate and severe recession.
(a) Scenarios 10, 14–16, (b) Scenarios 22–26.
In analogy to weak Skiba curves, the dotted lines based on Scenarios (a) 18–21, (b) 28–31 indicate the initial values which separate the state space into the ones (above) that do not lead to bankruptcy and the ones (below) that do. After the introduction of the time lag σ the bankruptcy region becomes larger. This results in an upwards-adjustment of the weak Skiba curve in the delayed case.

that $p_{\mathrm{d}} < p_{\mathrm{nd}}$ then. The second effect implies the opposite relation during the recession stage. Note that this second impact only occurs with $\mathbb{P}[\tau \in [t, t+\sigma]]$, i.e., it depends on the size of σ and the probability density function.

In our case (with $\sigma = 0.25$) the second effect dominates, meaning that the mentioned probability is large enough. For the first effect to dominate we have to decrease this probability by either reducing the time lag or end of recession probability parameter λ. The results of the latter possibility can be seen in Figure 6.4(a).

In a more vivid way we can interpret this second effect by assuming that the crisis ends at time $\hat{\tau}$. In the undelayed case the firm can start building up their reputation immediately after the realization of $\hat{\tau}$ by charging higher prices (supposing that it has survived). The effect on A comes directly. If $\sigma > 0$ the impact of rising prices after $\hat{\tau}$ only starts to have a positive outcome from time $\hat{\tau}+\sigma$ onwards. In the initial phase of the normal period $[\hat{\tau}, \hat{\tau}+\sigma]$ the demand is directly influenced by the price set in the last interval of the recession. Hence, increasing prices in $[\hat{\tau}-\sigma, \hat{\tau}]$ leads to a higher reputation σ time units later. That is, the demand is also higher in the period $[\hat{\tau}, \hat{\tau}+\sigma]$, which generates higher revenues during the first phase of the normal period. As the firm does not know beforehand when the recession is over, there is always a positive probability that the current time t is located in the period $[\hat{\tau}-\sigma, \hat{\tau}]$. Keeping this in mind, the firm has an additional incentive to keep prices up in recession periods when a delay is apparent, avoiding damaging the reputation too much. Otherwise their product will still perceived to be comparatively cheap for some time period after the recession is over.

Another important result can be observed in Figure 6.6. As observed in [61], in cases of an intermediate or severe recession there is a weak Skiba curve separating the regions of possible bankruptcy and certain survival. If $\sigma > 0$ this curve is adjusted upwards to some extent. With the incorporation of the delay in our model it is less easy for the firm to survive the crisis because the effect of changing the price p on the brand image is less direct. This explains why the bankruptcy region becomes larger.

At the end of this Section we want to remark that the condition (6.13d) causes two main scenarios we have to distinguish in the delayed model. The economic stage that is apparent in the time prior to the planning period $[0, t_{\mathrm{f}}]$ can either be a normal or a recession stage. We consider two slightly simplified cases.

In the first one we assume a steady state corresponding to the normal economic period in the interval $[-\sigma, 0]$, i.e., we have already one “switching” occurrence at the beginning of the horizon. We initialize the retarded control with $\eta = \bar{p}_{\mathrm{d}}^{\mathrm{N}}$ and the brand image with $A_0 = \bar{A}_{\mathrm{d}}^{\mathrm{N}}$. Then the system evolves as shown in Figure 6.5(b). The non-smooth behavior of the trajectories there is quite natural. At $t = 0$ the recession begins and the demand is reduced immediately due to the influence of α. Hence, prices will drop and the firm’s cash decreases. However, the brand image in the time interval $[0, \sigma]$ develops according to the high steady state price $\bar{p}_{\mathrm{d}}^{\mathrm{N}}$, i.e., it remains at its level. Only thereafter the condition (6.13e) becomes active and the reputation reacts to the lower prices.

The second case is more complicated. If we suppose a persisting recession stage, it is very hard

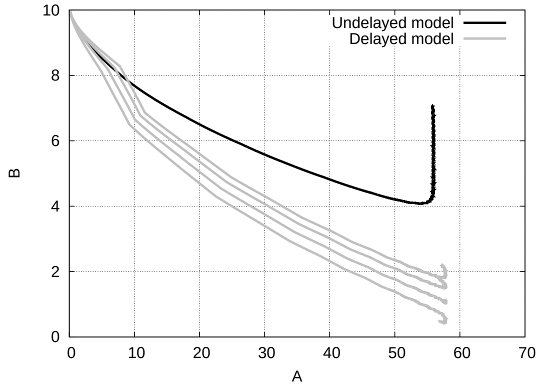


Figure 6.7: Phase diagram as in Figure 6.5(a) for Scenarios 10–13 (gray lines from top to bottom). It is obvious that the initial control path η has a considerable influence on the firm's future situation.

to find a satisfying initialization η for the retarded price in the interval $[0,\sigma]$. In our calculations we started with the optimal price obtained in the first interval of the non-delayed model. This causes the kink in the initial part of the trajectories in Figures 6.5 and 6.6. Experiments of varying the value of η changed the amplitude of this deformation slightly, see Figure 6.7. In this special scenario the different initializations also had a qualitative influence on the bankruptcy probability of the firm. If the combination of brand image and cash moves below the weak Skiba curve, the firm has to face bankruptcy in the long run. This happens for small initial prices, whereas high ones lead to certain survival.

6.5 Summary

We showed that a constant control delay in a two-stage model of a firm selling conspicuous consumption goods has a qualitative influence on the optimal pricing strategy the firm should apply in periods of economic uncertainties. In the recession stage of the delayed case the firm should use higher prices than in the corresponding scenario in the undelayed world, whereas in the normal economic stage after the recession is over the pricing policy is optimal if the reversed relation is true. This behavior is strongly depending on the probability that the recession ends during the next σ periods, i.e., on the size of the delay and the rate parameter λ. We also showed that the bankruptcy region is larger if $\sigma > 0$.

Our approach to solve this non-standard optimal control problem by a scenario tree approach deduced from the discretization of the random variable τ as the end point of the crisis and combining this with the introduction of a slack control function to incorporate price delays has proven to be successful. The application of structure-exploiting direct numerical methods is an adequate means to gain insight into solution structures of complex economical systems, also and especially if additional analytical studies are required.

Possible extensions of our model can include state equations with delays in both the control and the state [67], the inclusion of quality as additional control, or an reversion of the order of stages, i.e., beginning with a normal period followed by a recession. Another variant can be obtained by a redefinition of the brand image

$$A(t) = \int_{t-\sigma}^{t} p(z)\,\mathrm{d}z,$$

yielding $\dot{A}(t) = p(t) - p(t-\sigma) = p(t) - \delta(t)A(t)$, where the depreciation rate $\delta(t)$ depends on the delayed price [48, 67]. Further on, the recession parameter α might be regarded as a random variable as well, possibly even as a random process.

Airlines or hotels often use a company-internal staircase system for their prices. This motivates integer requirements on the control function. They can be addressed with theory and methods that have been presented in Chapters 2 and 3. Integer controls can be easily deduced by applying either the Sum Up Rounding strategy, or if combinatorial constraints are present, the decomposition approach from Chapter 5. Hence, we showed how multiple setpoint scenarios, worst case constraints, and control delays can be treated in the framework of mixed-integer optimal control as well.

7 A MIOC Benchmark Library

The contents of this chapter are based on the paper

[205] S. Sager. A benchmark library of mixed-integer optimal control problems. *Proceedings MINLP09 IMA Minneapolis*, accepted.

Chapter Summary. Numerical algorithm developers need standardized test instances for empirical studies and proofs of concept. There are several libraries available for finite-dimensional optimization, such as the netlib or the miplib. However, for mixed-integer optimal control problems (MIOCP) this is not yet the case. One explanation for this is the fact that no dominant standard format has been established yet. In many cases instances are used in a discretized form, but without proper descriptions on the modeling assumptions and discretizations that have been applied. In many publications crucial values, such as initial values, parameters, or a concise definition of all constraints are missing.

We intend to establish the basis for a benchmark library of mixed-integer optimal control problems that is meant to be continuously extended online on the open community web page located at http://mintoc.de. The guiding principles are comprehensiveness, a detailed description of where a model comes from and what the underlying assumptions are, a clear distinction between problem and method description (such as a discretization in space or time), reproducibility of solutions and a standardized problem formulation. Also, the problems are classified according to model and solution characteristics. We do not benchmark MIOCP solvers, but provide a library infrastructure and sample problems as a basis for future studies.

A second objective is to formulate mixed-integer nonlinear programs (MINLPs) originating from these MIOCPs. The snag is of course that we need to apply one out of several possible method-specific discretizations in time and space in the first place to obtain a MINLP. Yet the resulting MINLPs originating from control problems with an indication of the currently best known solution are hopefully a valuable test set for developers of generic MINLP solvers. The problem specifications can also be downloaded from http://mintoc.de.

7.1 Introduction

The algorithms that have been presented in the previous chapters have been applied to a variety of mixed integer optimal control problems. In this section we collect them in short form as a reference for algorithm developers.

For empirical studies and proofs of concept, developers of optimization algorithms need standardized test instances. There are several libraries available, such as the netlib for linear programming (LP) [16], the Schittkowski library for nonlinear programming (NLP) [215], the miplib [173] for mixed-integer linear programming (MILP), or more recently the MINLPLib [57] and the CMU-IBM Cyber-Infrastructure for for mixed-integer nonlinear programming (MINLP) collaborative site [66]. Further test libraries and related links can be found on [58]. A comprehensive testing environment is *CUTEr* [115]. The solution of these problems with different solvers is facilitated by the fact that standard formats such as the *standard input format* (SIF) or the *Mathematical Programming System* format (MPS) have been defined.

Collections of optimal control problems (OCPs) in ordinary differential equations (ODE) and in differential algebraic equations (DAE) have also been set up. The PROPT (a matlab toolkit for dynamic optimization using collocation) homepage states over 100 test cases from different applications with their results and computation time, [132]. With the software package dsoa [87] come currently 77 test problems. The ESA provides a test set of global optimization spacecraft trajectory problems and their best putative solutions [4].

This is a good starting point. However, no standard has evolved yet as in the case of finite-dimensional optimization. The specific formats for which only few optimization / optimal control codes have an interface, insufficient information on the modeling assumptions, or missing initial values, parameters, or a concise definition of all constraints make a transfer to different solvers and environments very cumbersome. The same is true for hybrid systems, which incorporate MIOCPs as defined in this thesis as a special case. Two benchmark problems have been defined at [182].

Although a general open library would be highly desirable for optimal control problems, we restrict ourselves here to the case of MIOCPs, in which some or all of the control values and functions need to take values from a finite set. MIOCPs are of course more general than OCPs as they include OCPs as a special case, however the focus in this library is on integer aspects. We want to be general in our formulation, without becoming too abstract. It allows to incorporate ordinary and partial differential equations, as well as algebraic constraints. Most hybrid systems can be formulated by means of state-dependent switches. Closed-loop control problems are on a different level, because a unique and comparable scenario would include well-defined external disturbances. We try to leave our approach open to future extensions to nonlinear model predictive control (NMPC) problems, but do not incorporate them yet. The formulation allows for different kinds of objective functions, e.g., time minimal or of tracking type, and of boundary constraints, e.g., periodicity constraints. Abstract problem formulations, together with a proposed categorization of problems according to model, objective, and solution characteristics are given in Section 7.2.

As discussed in Chapter 2, there is a large variety of different reformulations, solvers, and methods that have been proposed to analyze and solve MIOCPs. We would like to point out that they all discretize the optimization problem in function space in a different manner, and hence result in different mathematical problems that are actually solved on a computer.

We have two objectives. First, we intend to establish the basis for a benchmark library of mixed-integer optimal control problems that is meant to be continuously extended online on the open community web page http://mintoc.de. The guiding principles are comprehensiveness, a detailed description of where a model comes from and what the underlying assumptions are, a clear distinction between problem and method description (such as a discretization in space or time), reproducibility of solutions and a standardized problem formulation that allows for an easy transfer, once a method for discretization has been specified, to formats such as AMPL or GAMS. Also, the problems are classified according to model and solution characteristics.

Although our focus is on formulating MIOCPs before any irreversible reformulation and numerical solution strategy has been applied, a second objective is to provide specific MINLP formulations as benchmarks for developers of MINLP solvers. Powerful commercial MILP solvers and advances in MINLP solvers as described in the other contributions to this book make the usage of general purpose MILP/MINLP solvers more and more attractive. Please be aware however that the MINLP formulations we provide here are only one out of many possible ways to formulate the underlying MIOCP problems.

In Section 7.2 a classification of problems is proposed. Sections 7.3 to 7.11 describe the respective control problems and currently best known solutions. In Section 7.12 two specific MINLP formulations are presented for illustration. Section 7.13 gives a conclusion and an outlook.

7.2 Classifications

The MIOCPs in our benchmark library have different characteristics. In this section we describe these general characteristics, so we can simply list them later on where appropriate. Beside its origins from application fields such as mechanical engineering, aeronautics, transport, systems biology, chemical engineering and the like, we propose three levels to characterize a control problem. First, characteristics of the model from a mathematical point of view, second the formulation of the optimization problem, and third characteristics of an optimal solution from a control theory point of view. We address these three in the following subsections.

Although we strive for a standardized problem formulation, we do not formulate a specific generic formulation as such. Such a formulation is not even agreed upon for PDEs, let alone the possible extensions in the direction of algebraic variables, network topologies, logical connections, multi-stage processes, MPEC constraints, multiple objectives, functions including higher-order derivatives and much more that might come in. Therefore we chose to start with a very abstract formulation, formulate every control problem in its specific way as is adequate and to connect the two by using a characterization. On the most abstract level, we want to solve an optimization problem that can be written as

$$\begin{aligned} \min_{x,u,v} \quad & \Phi[x,u,v]) \\ \text{s.t.} \quad & 0 = F[x,u,v], \\ & 0 \leq C[x,u,v], \\ & 0 = \Gamma[x]. \end{aligned} \tag{7.1}$$

Here $x(\cdot) : \mathbb{R}^{\mathrm{d}} \mapsto \mathbb{R}^{n_x}$ denotes the differential-algebraic states[1] in a d-dimensional space. Until now, for most applications we have $d = 1$ and the independent variable time $t \in [t_0, t_f]$, the case of ordinary or algebraic differential equations. $u(\cdot) : \mathbb{R}^{\mathrm{d}} \mapsto \mathbb{R}^{n_u}$ and $v(\cdot) : \mathbb{R}^{\mathrm{d}} \mapsto \Omega$ are controls, where $u(\cdot)$ are continuous values that map to $\mathbb{R}^{n_u}$, and $v(\cdot)$ are controls that map to a finite set Ω. We allow also constant-in-time or constant-in-space control values rather than distributed controls.

We also use the term *integer control* for $v(\cdot)$, while *binary control* refers to $\omega(t) \in \{0,1\}^{n_\omega}$ that shall be introduced later. We use the expression *relaxed*, whenever a restriction $v(\cdot) \in \Omega$ is relaxed to a convex control set, which is typically the convex hull, $v(\cdot) \in \operatorname{conv} \Omega$.

Basically two different kinds of switching events are at the origin of hybrid systems, controllable and state-dependent ones. The first kind is due to degrees of freedom for the optimization, in particular with controls that may only take values from a finite set. The second kind is due to state-dependent switches in the model equations, e.g., ground contact of a robot leg or overflow of weirs in a distillation column. The focus in the benchmark library is on the first kind of switches, whereas the second one is of course important for a classification of the model equations, as for certain MIOCPs both kinds occur.

The model equations are described by the functional $F[\cdot]$, to be specified in Section 7.2.1. The objective functional $\Phi[\cdot]$, the constraints $C[\cdot]$ that may include control- and path-constraints, and the interior point constraints $\Gamma[x]$ that specify also the boundary conditions are classified in Section 7.2.2. In Section 7.2.3 characteristics of an optimal solution from a control theory point of view are listed.

The formulation of optimization problems is typically not unique. Sometimes, as in the case of MPEC reformulations of state-dependent switches [26], disjunctive programming [120], or outer convexification [214], reformulations may be seen as part of the solution approach in the sense of the *modeling for optimization paradigm* [184]. Even in obvious cases, such as a Mayer term versus a Lagrange term formulation, they may be mathematically, but not necessarily algorithmically equivalent. We propose to use either the original or the most adequate formulation of the optimization problem and list possible reformulations as variants.

[1] Note that we use the notation common in control theory with x as differential states and u as controls, not the PDE formulation with x as independent variable and u as differential states.

7.2.1 Model classification

This Section addresses possible realizations of the state equation

$$0 \quad = \quad F[x,u,v]. \tag{7.2}$$

We assume throughout that the differential-algebraic states x are uniquely determined for appropriate boundary conditions and fixed (u,v).

ODE model. This category includes all problems constrained by the solution of explicit ordinary differential equations (ODE). In particular, no algebraic variables and derivatives with respect to one independent variable only (typically time) are present in the mathematical model. Equation (7.2) reads

$$\dot{x}(t) \quad = \quad f(x(t),u(t),v(t)), \quad t \in [0,t_\mathrm{f}], \tag{7.3}$$

for $t \in [t_0,t_f]$ almost everywhere. We often leave the argument (t) away for notational convenience.

DAE model. If the model includes algebraic constraints and variables, for example from conversation laws, a problem is categorized as a DAE model. Equality (7.2) then includes both differential equations and algebraic constraints that determine the algebraic states in dependence of the differential states and the controls. A more detailed classification includes the index of the algebraic equations.

PDE model. If $d > 1$ the model equation (7.2) becomes a partial differential equation (PDE). Depending on whether convection or diffusion prevails, a further classification into hyperbolic, elliptic, or parabolic equations is necessary. A more elaborate classification shall evolve as more PDE constrained MIOCPs are described on http://mintoc.de. In this work one PDE-based instance is presented in Section 7.11.

Outer convexification. For time-dependent and space- independent integer controls often another formulation is beneficial, e.g., [147]. For every element v^i of Ω a binary control function $\omega_i(\cdot)$ is introduced. Equation (7.2) can then be written as

$$0 \quad = \quad \sum_{i=1}^{n_\omega} F[x,u,v^i]\ \omega_i(t), \quad t \in [0,t_\mathrm{f}]. \tag{7.4}$$

If we impose the special ordered set type one condition

$$\sum_{i=1}^{n_\omega} \omega_i(t) \quad = \quad 1, \quad t \in [0,t_\mathrm{f}], \tag{7.5}$$

there is a bijection between every feasible integer function $v(\cdot) \in \Omega$ and an appropriately chosen binary function $\omega(\cdot) \in \{0,1\}^{n_\omega}$, compare [214]. The relaxation of $\omega(t) \in \{0,1\}^{n_\omega}$ is given by $\omega(t) \in [0,1]^{n_\omega}$. We refer to (7.4) and (7.5) as *outer convexification* of (7.2). This characteristic applies to the control problems in Sections 3.6, 7.3, 7.6, 7.9, 7.10, and 7.11.

State-dependent switches. Many processes are modelled by means of state-dependent switches that indicate, e.g., model changes due to a sudden ground contact of a foot or a weir overflow in a chemical process. Mathematically, we write

$$0 \;=\; F_i[x,u,v] \quad \text{if } \sigma_i(x(t)) \geq 0. \tag{7.6}$$

with well defined switching functions $\sigma_i(\cdot)$ for $t \in [0,t_\mathrm{f}]$. This characteristic applies to the control problems in Sections 7.6 and 7.8.

Boolean variables. Discrete switching events can also be expressed by means of Boolean variables and logical implications. E.g., by introducing logical functions $\delta_i : [0,t_\mathrm{f}] \mapsto \{\text{true},\text{false}\}$ that indicate whether a model formulation $F_i[x,u,v]$ is active at time t, both state-dependent switches and outer convexification formulations may be written as *disjunctive programs*, i.e., optimization problems involving Boolean variables and logical conditions. Using disjunctive programs can be seen as a more natural way of modeling discrete events and has the main advantage of resulting in tighter relaxations of the discrete decisions, when compared to integer programming techniques. More details can be found in [120, 183, 184].

Multistage processes. Processes of interest are often modelled as multistage processes. At transition times the model can change, sometimes in connection with a state-dependent switch. The equations read as

$$0 \;=\; F_i[x,u,v] \quad t \in [t_i,t_{i+1}] \tag{7.7}$$

on a time grid $\{t_i\}_i$. With smooth transfer functions also changes in the dimension of optimization variables can be incorporated, [167].

Unstable dynamics. For numerical reasons it is interesting to keep track of instabilities in process models. As small changes in inputs lead to large changes in outputs, challenges for optimization methods arise. This characteristic applies to the control problems in Sections 7.3 and 7.7.

Network topology. Complex processes often involve an underlying network topology, such as in the control of gas or water networks [174, 55] . The arising structures should be exploited by efficient algorithms.

7.2.2 Classification of the optimization problem

The optimization problem (7.1) is described by means of an objective functional $\Phi[\cdot]$ and inequality constraints $C[\cdot]$ and equality constraints $\Gamma[\cdot]$. The constraints come in form of multipoint constraints that are defined on a time grid $t_0 \leq t_1 \leq \cdots \leq t_m = t_f$, and of path-constraints that need to hold almost everywhere on the time horizon. The equality constraints $\Gamma[\cdot]$ often fix the initial values or impose a periodicity constraint. In this classification we assume all functions to be sufficiently often differentiable.

In the future, the classification shall also include problems with nondifferentiable objective functions, multiple objectives, online control tasks including feedback, indication of nonconvexities, and more characteristics that allow for a specific choice of test instances.

Minimum time. This is a category with all control problems that seek for time-optimal solutions, e.g., reaching a certain goal or completing a certain process as fast as possible. The objective function is of Mayer type, $\Phi[\cdot] = t_\mathrm{f}$. This characteristic applies to the control problems in Sections 7.3, 7.9, and 7.10.

Minimum energy. This is a category with all control problems that seek for energy-optimal solutions, e.g., reaching a certain goal or completing a certain process with a minimum amount of energy. The objective function is of Lagrange type and sometimes proportional to a minimization of the squared control (e.g., acceleration) $u(\cdot)$, e.g., $\Phi[\cdot] = \int_{t_0}^{t_\mathrm{f}} u^2 \, \mathrm{d}t$. Almost always an upper bound on the free end time t_f needs to be specified. This characteristic applies to the control problems in Sections 7.6 and 7.8.

Tracking problem. This category lists all control problems in which a tracking type Lagrange functional of the form

$$\Phi[\cdot] \quad = \quad \int_{t_0}^{t_f} ||x(\tau) - x^{\mathrm{ref}}||_2^2 \, \mathrm{d}\tau. \tag{7.8}$$

is to be minimized. This characteristic applies to the control problems in Sections 3.6, 7.4, 7.5, and 7.7.

Optimum Experimental Design. This category lists all control problems in which a function of either the Fisher information matrix $F(\cdot)$ or the covariance matrix $C(\cdot)$ of an underlying parameter estimation problem is to be minimized. See Chapter 9 for details. This characteristic applies to the control problems in Sections 9.6.1, and 9.6.2.

Periodic processes. This is a category with all control problems that seek periodic solutions, i.e., a condition of the kind

$$\Gamma[x] \quad = \quad P(x(t_f)) - x(t_0) = 0, \tag{7.9}$$

has to hold. $P(\cdot)$ is an operation that allows, e.g., for a perturbation of states (such as needed for the formulation of Simulated Moving Bed processes, Section 7.11, or for offsets of angles by a multiple of 2π such as in driving on closed tracks, Section 7.10). This characteristic applies to the control problems in Sections 7.8, 7.10, and 7.11.

Equilibrium constraints. This category contains mathematical programs with equilibrium constraints (MPECs). An MPEC is an optimization problem constrained by a variational inequality, which takes for generic variables / functions y_1, y_2 the following general form:

$$\begin{aligned} \min_{y_1,y_2} \quad & \Phi(y_1,y_2) \\ \text{s.t.} \quad & 0 = F(y_1,y_2), \\ & 0 \le C(y_1,y_2), \\ & 0 \le (\mu - y_2)^T \, \phi(y_1,y_2), \quad y_2 \in Y(y_1), \; \forall \mu \in Y(y_1) \end{aligned} \tag{7.10}$$

where $Y(y_1)$ is the feasible region for the variational inequality and given function $\phi(\cdot)$. Variational inequalities arise in many domains and are generally referred to as equilibrium constraints. The variables y_1 and y_2 may be controls or states.

Complementarity constraints. This category contains optimization problems with complementarity constraints (MPCCs), for generic variables / functions y_1, y_2, y_3 in the form of

$$\begin{aligned} \min_{y_1,y_2,y_3} \quad & \Phi(y_1,y_2,y_3) \\ \text{s.t.} \quad & 0 = F(y_1,y_2,y_3), \\ & 0 \le C(y_1,y_2,y_3), \\ & 0 \le y_1 \perp y_2 \ge 0 \end{aligned} \tag{7.11}$$

The complementarity operator $\perp$ implies the disjunctive behavior

$$y_{1,i} = 0 \quad \text{OR} \quad y_{2,i} = 0 \qquad \forall\, i = 1 \dots n_y.$$

MPCCs may arise from a reformulation of a bilevel optimization problem by writing the optimality conditions of the inner problem as variational constraints of the outer optimization problem, or from a special treatment of state-dependent switches, [26]. Note that all MPCCs can be reformulated as MPECs.

Vanishing constraints. This category contains mathematical programs with vanishing constraints (MPVCs). The problem

$$\begin{aligned} \min_{y} \quad & \Phi(y) \\ \text{s.t.} \quad & 0 \ge g_i(y) h_i(y), \quad i \in \{1, \dots, m\} \\ & 0 \le h(y) \end{aligned} \tag{7.12}$$

with smooth functions $g, h : \mathbb{R}^{n_y} \mapsto \mathbb{R}^m$ is called MPVC. Note that every MPVC can be transformed into an MPEC [3, 133]. Examples for vanishing constraints are engine speed constraints that are only active if the corresponding gear control is nonzero. This characteristic applies to the control problems in Sections 7.9, and 7.10.

7.2.3 Solution classification

The classification that we propose for switching decisions is based on insight from Pontryagin's maximum principle, [192], applied here only to the relaxation of the binary control functions $\omega(\cdot)$, denoted by $\alpha(\cdot) \in [0,1]^{n_\omega}$. In the analysis of linear control problems one distinguishes three cases: bang-bang arcs, sensitivity-seeking arcs, and path-constrained arcs, [228], where an arc is defined to be a nonzero time-interval. Of course a problem's solution can show two or even all three behaviors at once on different time arcs.

Bang-bang arcs. Bang-bang arcs are time intervals on which the control bounds are active, i.e., $\alpha_i(t) \in \{0,1\} \; \forall \, t$. The case where the optimal solution contains only bang-bang arcs is in a sense the easiest. The solution of the relaxed MIOCP is integer feasible, if the control discretization grid is a superset of the switching points of the optimal control. Hence, the main goal is to adapt the control discretization grid such that the solution of the relaxed problem is already integer. Also on fixed time grids good solutions are easy to come up with, as rounded solutions approximate the integrated difference between relaxed and binary solution very well.

A prominent example of this class is time-optimal car driving, see Section 7.9 and see Section 7.10. Further examples of "bang-bang solutions" include free switching of ports in Simulated Moving Bed processes, see Section 7.11, unconstrained energy-optimal operation of subway trains see Section 7.6, a simple F-8 flight control problem see Section 7.3, and phase resetting in biological systems, such as in Section 7.7.

Path–constrained arcs. Whenever a path constraint is active, i.e., it holds $c_i(x(t)) = 0 \; \forall \, t \in [t^{\text{start}}, t^{\text{end}}] \subseteq [0, t_\text{f}]$, and no continuous control $u(\cdot)$ can be determined to compensate for the changes in $x(\cdot)$, naturally $\alpha(\cdot)$ needs to do so by taking values in the interior of its feasible domain. An illustrating example has been given in [214], where velocity limitations for the energy-optimal operation of New York subway trains are taken into account, see Section 7.6. The optimal integer solution does only exist in the limit case of infinite switching (Zeno behavior), or when a tolerance is given. Another example is compressor control in supermarket refrigeration systems, see Section 7.8. Note that all applications may comprise path-constrained arcs, once path constraints need to be added, as has been done in Section 3.6.

Sensitivity–seeking arcs. We define sensitivity–seeking (also compromise–seeking) arcs in the sense of Srinivasan and Bonvin, [228], as arcs which are neither bang–bang nor path–constrained and for which the optimal control can be determined by time derivatives of the Hamiltonian. For control–affine systems this implies so-called singular arcs.

A classical small-sized benchmark problem for a sensitivity-seeking (singular) arc is the Lotka-Volterra Fishing problem, see Section 7.4. Also in Section 3.6 a sensitivity–seeking arc is present. The treatment of sensitivity–seeking arcs is very similar to the one of path–constrained arcs. As above, an approximation up to any a priori specified tolerance is possible, probably at the price of frequent switching.

Chattering arcs. Chattering controls are bang–bang controls that switch infinitely often in a finite time interval $[0, t_\text{f}]$. An extensive analytical investigation of this phenomenon can be found in

[252]. An example for a chattering arc solution is the famous example of Fuller, see Section 7.5.
Sliding Mode. Solutions of model equations with state-dependent switches as in (7.6) may show a sliding mode behavior in the sense of Filippov systems [90]. This means that at least one of the functions $\sigma_i(\cdot)$ has infinitely many zeros on the finite time interval $[0, t_\mathrm{f}]$. In other words, the right hand side switches infinitely often in a finite time horizon.
The two examples with state-dependent switches in Sections 7.6 and 7.8 do not show sliding mode behavior.

7.3 F-8 flight control

The F-8 aircraft control problem is based on a very simple aircraft model. The control problem was introduced by Kaya and Noakes [140] and aims at controlling an aircraft in a time-optimal way from an initial state to a terminal state. The mathematical equations form a small-scale ODE model. The interior point equality conditions fix both initial and terminal values of the differential states. The optimal, relaxed control function shows bang bang behavior. The problem is furthermore interesting as it should be reformulated equivalently. Despite the reformulation the problem is nonconvex and exhibits multiple local minima.

7.3.1 Model and optimal control problem

The F-8 aircraft control problem is based on a very simple aircraft model in ordinary differential equations, introduced by Garrard [104]. The differential states consist of x_0 as the angle of attack in radians, x_1 as the pitch angle, and x_2 as the pitch rate in rad/s. The only control function $w = w(t)$ is the tail deflection angle in radians. The control objective is to control the airplane from one point in space to another in minimum time. For $t \in [0, T]$ almost everywhere the mixed-integer optimal control problem is given by

$$\begin{aligned}
\min_{x,w,T} \quad & T \\
\text{s.t.} \quad & \dot{x}_0 = -0.877\, x_0 + x_2 - 0.088\, x_0\, x_2 + 0.47\, x_0^2 - 0.019\, x_1^2 \\
& \qquad - x_0^2\, x_2 + 3.846\, x_0^3 \\
& \qquad - 0.215\, w + 0.28\, x_0^2\, w + 0.47\, x_0\, w^2 + 0.63\, w^3 \\
& \dot{x}_1 = x_2 \\
& \dot{x}_2 = -4.208\, x_0 - 0.396\, x_2 - 0.47\, x_0^2 - 3.564\, x_0^3 \\
& \qquad - 20.967\, w + 6.265\, x_0^2\, w + 46\, x_0\, w^2 + 61.4\, w^3 \\
& x(0) = (0.4655, 0, 0)^T, \quad x(T) = (0, 0, 0)^T, \\
& w(t) \in \{-0.05236, 0.05236\}, \quad t \in [0, T].
\end{aligned} \tag{7.13}$$

In the control problem, both initial and terminal values of the differential states are fixed. The control w(t) is restricted to take values from a finite set only. Hence, the control problem can be reformulated equivalently to

$$\begin{aligned}
\min_{x,w,T}\quad & T \\
\text{s.t.}\quad & \dot{x}_0 = -0.877\,x_0 + x_2 - 0.088\,x_0\,x_2 + 0.47\,x_0^2 - 0.019\,x_1^2 \\
& \qquad - x_0^2\,x_2 + 3.846\,x_0^3 \\
& \qquad + 0.215\,\xi - 0.28\,x_0^2\,\xi + 0.47\,x_0\,\xi^2 - 0.63\,\xi^3 \\
& \qquad - \left(0.215\,\xi - 0.28\,x_0^2\,\xi - 0.63\,\xi^3\right)\,2w \\
& \dot{x}_1 = x_2 \\
& \dot{x}_2 = -4.208\,x_0 - 0.396\,x_2 - 0.47\,x_0^2 - 3.564\,x_0^3 \\
& \qquad + 20.967\,\xi - 6.265\,x_0^2\,\xi + 46\,x_0\,\xi^2 - 61.4\,\xi^3 \\
& \qquad - \left(20.967\,\xi - 6.265\,x_0^2\,\xi - 61.4\,\xi^3\right)\,2w \\
& x(0) = (0.4655, 0, 0)^T, \quad x(T) = (0, 0, 0)^T, \\
& w(t) \in \{0, 1\}, \quad t \in [0, T]
\end{aligned} \tag{7.14}$$

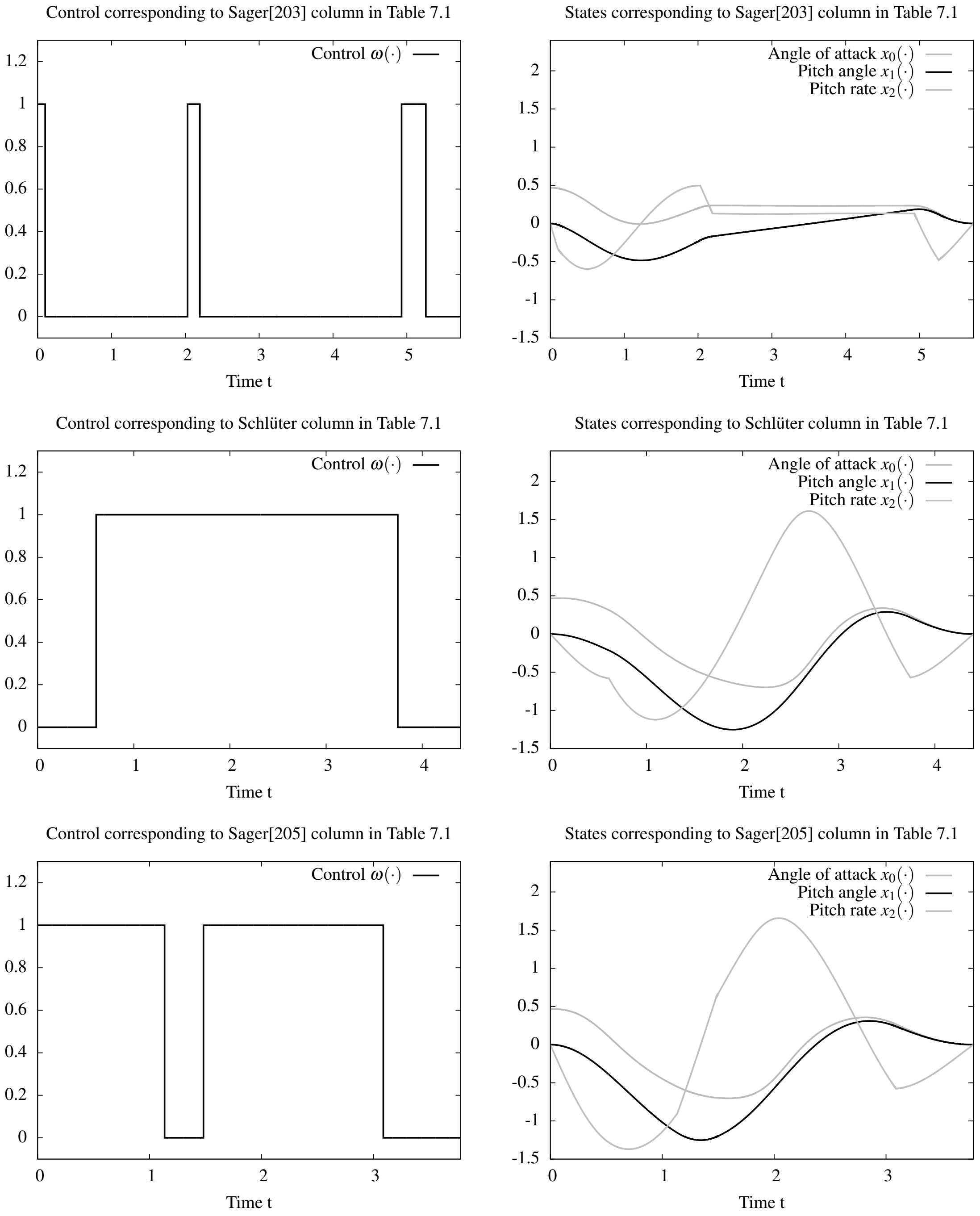
with $\xi = 0.05236$. Note that there is a bijection between optimal solutions of the two problems, and that the second formulation is an outer convexification, compare Section 7.2.1.

7.3.2 Results

We provide in Table 7.1 a comparison of different solutions reported in the literature. The numbers show the respective lengths $t_i - t_{i-1}$ of the switching arcs with the value of $w(t)$ on the upper or lower bound (given in the second column). The infeasibility shows values obtained by a simulation with a Runge-Kutta-Fehlberg method of 4th/5th order and an integration tolerance of 10^{-8}. The best known optimal objective value of this problem given is given by $T = 3.78086$. The trajectories are shown in Figure 7.1.

Arc	$w(t)$	Lee[165]	Kaya[140]	Sager[203]	Schlüter	Sager[205]
1	1	0.00000	0.10292	0.10235	0.0	1.13492
2	0	2.18800	1.92793	1.92812	0.608750	0.34703
3	1	0.16400	0.16687	0.16645	3.136514	1.60721
4	0	2.88100	2.74338	2.73071	0.654550	0.69169
5	1	0.33000	0.32992	0.32994	0.0	0.0
6	0	0.47200	0.47116	0.47107	0.0	0.0
Infeasibility		1.75E-3	1.64E-3	5.90E-6	3.29E-6	2.21E-7
Objective		6.03500	5.74218	5.72864	4.39981	3.78086

Table 7.1: Results for the F-8 flight control problem. The solution in the second last column is a personal communication by Martin Schlüter and Matthias Gerdts.



Control corresponding to Sager[203] column in Table 7.1

States corresponding to Sager[203] column in Table 7.1

Control corresponding to Schlüter column in Table 7.1

States corresponding to Schlüter column in Table 7.1

Control corresponding to Sager[205] column in Table 7.1

States corresponding to Sager[205] column in Table 7.1

Figure 7.1: Trajectories for the F-8 flight control problem from Table 7.1. Top: corresponding to the Sager[203] column. Middle: Schlüter column. Bottom: Sager[205] column.

7.3.3 Variants

The F-8 control problem has recently been reformulated as a mixed-integer optimal control benchmark for an index 1 DAE system [107]. The authors use it as an example of a boundary value problem that is deduced directly from the necessary conditions of optimality and follow a *first optimize, then discretize* approach. The authors reformulated (7.13) with the help of two artificial algebraic variables. This is not necessarily helpful from a computational point of view, but allows to compare the results to the ones above and it is clear that the index one assumption is valid for this MIOCP in DAE:

$$
\begin{aligned}
\min_{x,w,T} \quad & T \\
\text{s.t.} \quad & \dot{x}_0 = y_0 x_0 + x_2 - 0.019 x_1^2 - 0.215 w + 0.63 w^3 \\
& \dot{x}_1 = x_2 \\
& \dot{x}_2 = -4.208 x_0 - 0.396 x_2 - 0.47 x_0^2 - 3.564 x_0^3 \\
& \qquad - 20.967 w + 6.265 x_0 y_1 + 46 y_1 w + 61.4 w^3 \\
& \;\; 0 = -y_0 - 0.877 - 0.088 x_2 + 0.47 x_0 - x_0 x_2 \\
& \qquad + 3.846 x_0^2 + 0.28 y_1 + 0.47 w^2 \\
& \;\; 0 = -y_1 + x_0 w \\
& x(0) = (0.4655, 0, 0)^T, \quad x(T) = (0,0,0)^T, \\
& w(t) \in \{-0.05236, 0.05236\}, \quad t \in [0,T].
\end{aligned}
\tag{7.15}
$$

The problem formulation (7.13) can be easily regained by substituting first $y_1(\cdot)$ and then $y_0(\cdot)$ back in the differential equations. With $\ell_0 = 1$, the Hamiltonian of Problem (7.15) is given by

$$
\begin{aligned}
\mathscr{H}(\cdot) &= 1 + \lambda_f^\top f(x,y,u) + \lambda_g^\top g(x,y,u) \\
&= 1 + \lambda_{f0}(y_0 x_0 + x_2 - 0.019 x_1^2 - 0.215 w + 0.63 w^3) \\
&\quad + \lambda_{f1} x_2 \\
&\quad + \lambda_{f2}(-4.208 x_0 - 0.396 x_2 - 0.47 x_0^2 - 3.564 x_0^3 \\
&\qquad -20.967 w + 6.265 x_0 y_1 + 46 y_1 w + 61.4 w^3) \\
&\quad + \lambda_{g0}(-y_0 - 0.877 - 0.088 x_2 + 0.47 x_0 \\
&\qquad -x_0 x_2 + 3.846 x_0^2 + 0.28 y_1 + 0.47 w^2) \\
&\quad + \lambda_{g1}(-y_1 + x_0 w)
\end{aligned}
\tag{7.16}
$$

The adjoint equations read

$$
\begin{aligned}
-\dot{\lambda_{f0}} = \frac{\partial \mathscr{H}(\cdot)}{\partial x_0} &= \lambda_{f2}(-4.208 - 2 \cdot 0.47 x_0 - 3 \cdot 3.564 x_0^2 + 6.265 y_1) \\
&\quad + \lambda_{f0} y_0 + \lambda_{g0}(0.47 - x_2 + 2 \cdot 3.846 x_0) + \lambda_{g1} w
\end{aligned}
\tag{7.17}
$$

$$
\begin{aligned}
-\dot{\lambda_{f1}} = \frac{\partial \mathscr{H}(\cdot)}{\partial x_1} &= -0.038\lambda_{f0}x_1 && (7.18)\\
-\dot{\lambda_{f2}} = \frac{\partial \mathscr{H}(\cdot)}{\partial x_2} &= \lambda_{f0} + \lambda_{f1} - 0.396\lambda_{f2} + \lambda_{g0}(-0.088 - x_0) && (7.19)\\
0 = \frac{\partial \mathscr{H}(\cdot)}{\partial y_0} &= \lambda_{f0}x_0 - \lambda_{g0} && (7.20)\\
0 = \frac{\partial \mathscr{H}(\cdot)}{\partial y_1} &= \lambda_{f2}(6.265x_0 + 46w) + 0.28\lambda_{g0} - \lambda_{g1} && (7.21)
\end{aligned}
$$

with the transversality conditions

$$
\lambda_{fi}(t_0) = -\frac{\partial \varphi(\cdot) + \sigma^\top \psi(\cdot)}{\partial x_i(t_0)} = \sigma_{0i}, \quad \lambda_{fi}(t_\mathrm{f}) = -\frac{\partial \varphi(\cdot) + \sigma^\top \psi(\cdot)}{\partial x_i(t_\mathrm{f})} = \sigma_{fi}, \tag{7.22}
$$

for $i = 0 \dots 2$. In other words, the initial and terminal values of the differential adjoint states are free, because all initial and terminal values of the differential states $x_i(\cdot)$ are fixed. However, the value of the Hamiltonian at the free end time is fixed to 0,

$$
\mathscr{H}(\cdot, t_\mathrm{f}) = 0. \tag{7.23}
$$

The value of the optimal control $w(t)$ is determined according to the global minimum principle in [107] as the pointwise minimizer

$$
w^*(t) = \arg\min\{\mathscr{H}(\cdot, w \equiv -0.05236), \mathscr{H}(\cdot, w \equiv 0.05236)\}. \tag{7.24}
$$

It results in an optimal integer control

$$
w^*(t) = \begin{cases} 0.05236 & \text{if } t \in [0, \tau_1] \cup [\tau_2, \tau_3] \\ -0.05236 & \text{if } t \in [\tau_1, \tau_2] \cup [\tau_3, t_\mathrm{f}] \end{cases}
$$

with $\tau_1 = 1.135007, \tau_2 = 1.482512, \tau_3 = 3.088809, t_\mathrm{f} = 3.780858$. Figures 7.2 and 7.3 show the optimal trajectories and the competing Hamiltonians. The solution is similar to the bottom row in Figure 7.1 which has been used for initialization of the boundary value problem.

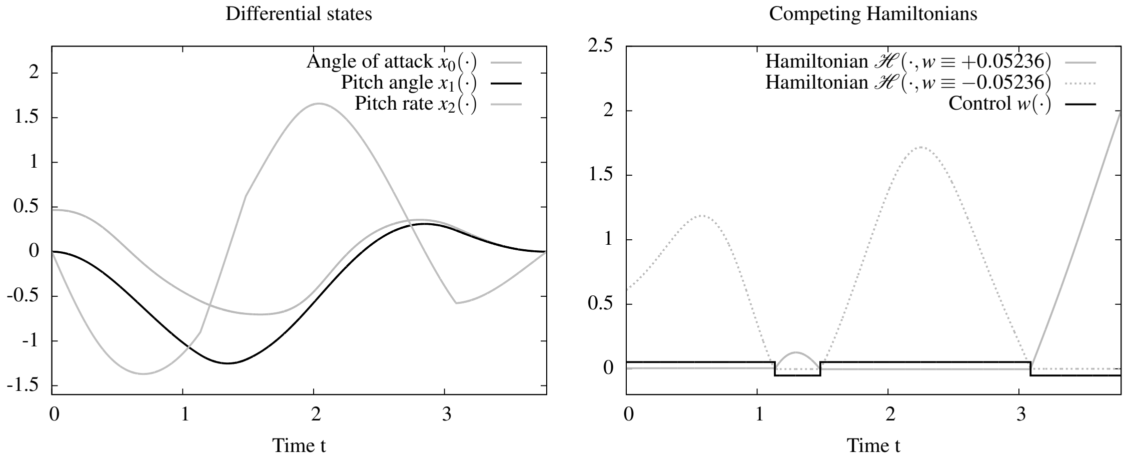


Figure 7.2: Left: differential states of the optimal trajectory. Right: the two competing Hamiltonians and the optimal control. Note that the minimizing Hamiltonian (7.24) is identical 0, in accordance to its end value (7.23) and the requirement to be constant.

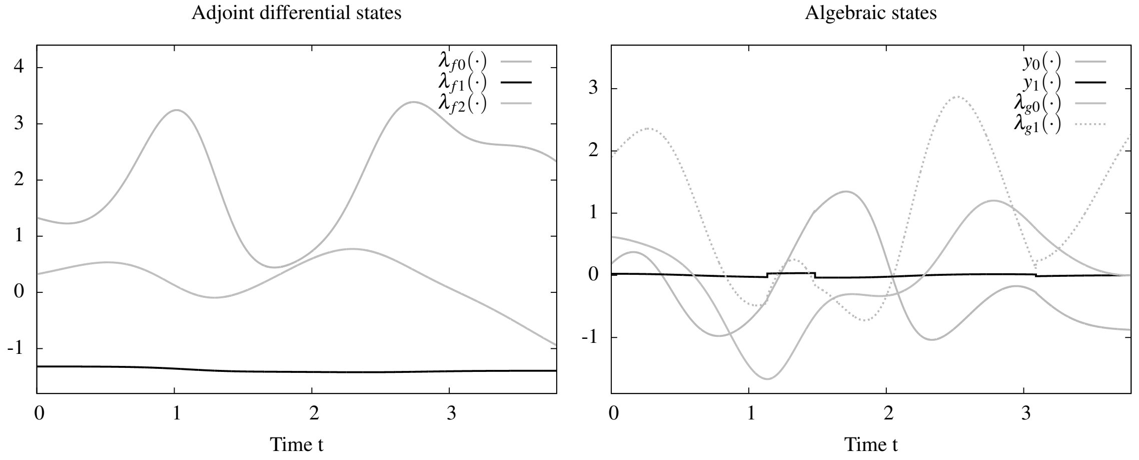


Figure 7.3: Left: the adjoint states plotted over time. Right: the algebraic states. The discontinuity in $w^*(\cdot)$ is accounted for by jumps in $y_1(\cdot)$.

7.4 Lotka Volterra fishing problem

The Lotka Volterra fishing problem seeks an optimal fishing strategy to be performed on a fixed time horizon to bring the biomasses of both predator as prey fish to a prescribed steady state. The problem was set up as a small-scale benchmark problem in [209] and has since been used for the evaluation of algorithms, e.g., [233].

The mathematical equations form a small-scale ODE model. The interior point equality conditions fix the initial values of the differential states. The optimal integer control shows chattering behavior, making the Lotka Volterra fishing problem an ideal candidate for benchmarking of algorithms.

7.4.1 Model and optimal control problem

The biomasses of two fish species — one predator, the other one prey — are the differential states of the model, the binary control is the operation of a fishing fleet. The optimization goal is to penalize deviations from a steady state,

$$\begin{aligned}
\min_{x,w} \quad & \int_{t_0}^{t_f} (x_0 - 1)^2 + (x_1 - 1)^2 \, \mathrm{d}t \\
\text{s.t.} \quad & \dot{x}_0 = x_0 - x_0 x_1 - \ c_0 x_0 \, w \\
& \dot{x}_1 = -x_1 + x_0 x_1 - \ c_1 x_1 \, w, \\
& x(0) = (0.5, 0.7)^T, \\
& w(t) \in \{0, 1\}, \quad t \in [0, t_\mathrm{f}],
\end{aligned} \tag{7.25}$$

with $t_\mathrm{f} = 12$, $c_0 = 0.4$, and $c_1 = 0.2$.

7.4.2 Results

If the problem is relaxed, i.e., we demand that $w(\cdot)$ be in the continuous interval $[0,1]$ instead of the binary choice $\{0,1\}$, the optimal solution can be determined by means of Pontryagin's maximum principle [192]. The optimal solution contains a singular arc, [209].

The optimal objective value of this relaxed problem is $\Phi = 1.34408$. As follows from the results of Chapter 3, this is the best lower bound on the optimal value of the original problem with the integer restriction on the control function. In other words, this objective value can be approximated arbitrarily close, if the control only switches often enough between 0 and 1. As no optimal solution exists, a suboptimal one is shown in Figure 7.4.

7.4.3 Variants

There are several alternative formulations and variants of the above problem, in particular

- a prescribed time grid for the control function [209],

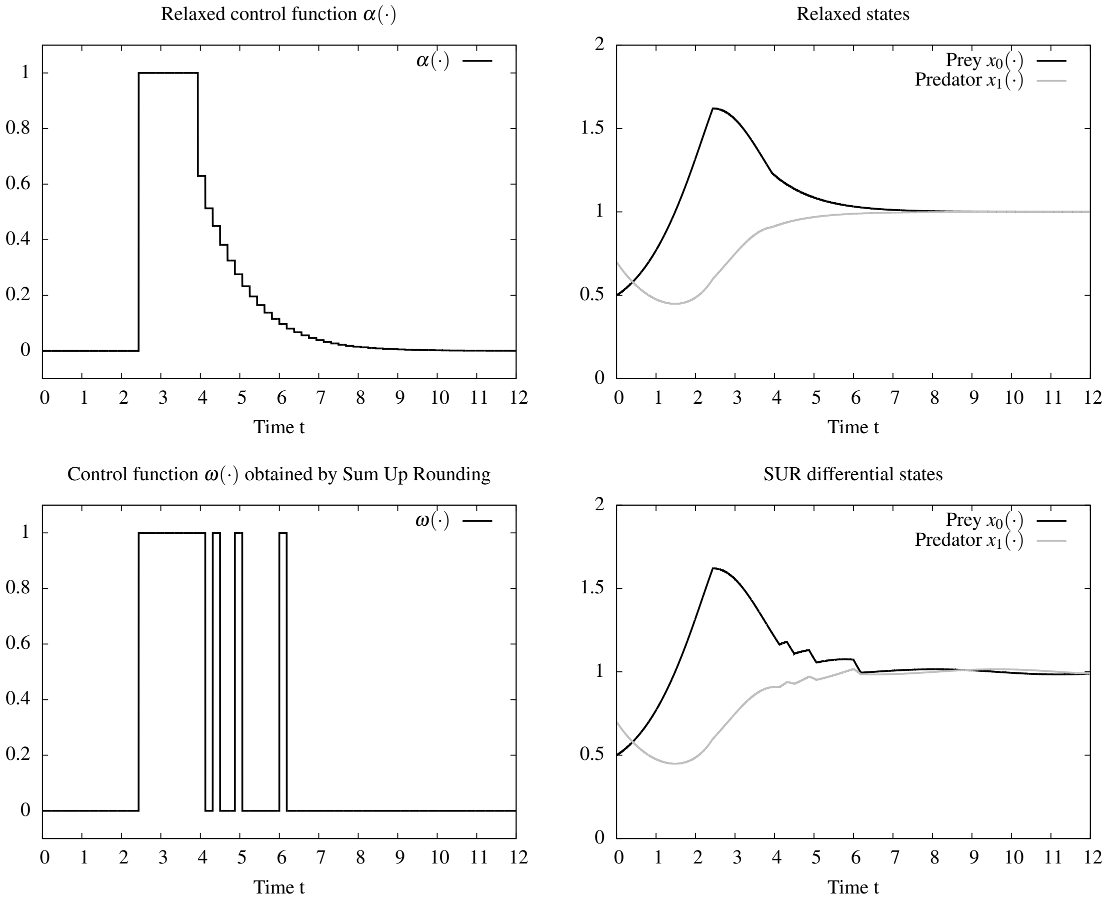


Figure 7.4: Trajectories for the Lotka Volterra Fishing problem. Top: optimal relaxed solution on grid with 64 intervals. Bottom: feasible integer solution.

- a time-optimal formulation to get into a steady-state [203],
- the usage of a different target steady-state, as the one corresponding to $w(\cdot) = 1$ which is $(1 + c_1, 1 - c_0)$,
- different fishing control functions for the two species,
- different parameters and start values,
- the usage as an optimum experimental design problem, see Chapter 9.

7.5 Fuller's problem

The first control problem with an optimal chattering solution was given by [97]. An optimal trajectory does exist for all initial and terminal values in a vicinity of the origin. As Fuller showed, this optimal trajectory contains a bang-bang control function that switches infinitely often. The mathematical equations form a small-scale ODE model. The interior point equality conditions fix initial and terminal values of the differential states, the objective is of tracking type.

7.5.1 Model and optimal control problem

The MIOCP reads

$$\begin{aligned} \min_{x,w} \quad & \int_0^1 x_0^2 \, \mathrm{d}t \\ \text{s.t.} \quad & \dot{x}_0 = x_1 \\ & \dot{x}_1 = 1 - 2\,w \\ & x(0) = (0.01, 0)^T, \quad x(T) = (0.01, 0)^T, \\ & w(t) \in \{0, 1\}, \quad t \in [0, 1]. \end{aligned} \tag{7.26}$$

7.5.2 Results

The optimal trajectories for the relaxed control problem on different equidistant grids $\mathscr{G}^0$ with $n_{\text{ms}} = 20, 30, 60$ are shown in Figure 7.5. Note that this solution is not bang–bang due to the discretization of the control space. Even if this discretization is made very fine, a trajectory with $w(\cdot) = 0.5$ on an interval in the middle of $[0,1]$ will be found as a minimum.

The application of Algorithm 2.1 yields an objective value of $\Phi = 1.52845 \cdot 10^{-5}$, which is better than the limit of the relaxed problems, $\Phi^{20} = 1.53203 \cdot 10^{-5}$, $\Phi^{30} = 1.53086 \cdot 10^{-5}$, and $\Phi^{60} = 1.52958 \cdot 10^{-5}$. A sample integer solution is also shown in Figure 7.5.

7.5.3 Variants

An extensive analytical investigation of this problem and a discussion of the ubiquity of Fuller's problem can be found in [252].

7.6 Subway ride

The optimal control problem we treat in this section goes back to work of [43] for the city of New York. In an extension, also velocity limits that lead to path–constrained arcs appear. The aim is to minimize the energy used for a subway ride from one station to another, taking into account boundary conditions and a restriction on the time.

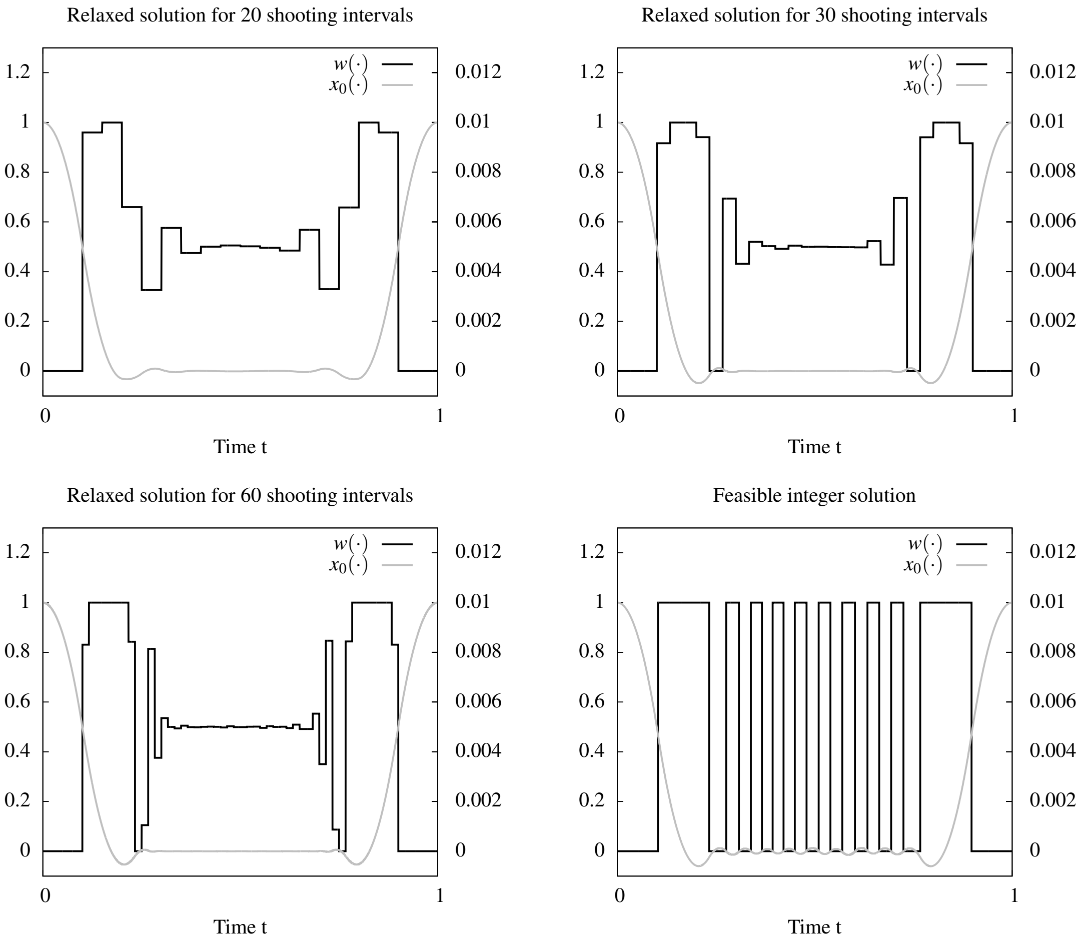


Figure 7.5: Trajectories for Fuller's problem for different discretizations. Bottom right shows a feasible integer solution. The scales for the differential state $x_0(\cdot)$ are given on the right axis.

7.6.1 Model and optimal control problem

The MIOCP reads

$$\begin{aligned} \min_{x,w} \quad & \int_0^{t_\text{f}} L(x,w)\,\text{d}t \\ \text{s.t.} \quad & \dot{x}_0 = x_1 \\ & \dot{x}_1 = f_1(x,w) \\ & x(0) = (0,0)^T, \quad x(t_\text{f}) = (2112,0)^T, \\ & w(t) \in \{1,2,3,4\}, \quad t \in [0,t_\text{f}]. \end{aligned} \tag{7.27}$$

The terminal time $t_\text{f} = 65$ denotes the time of arrival of a subway train in the next station. The differential states $x_0(\cdot)$ and $x_1(\cdot)$ describe position and velocity of the train, respectively. The train can be operated in one of four different modes, $w(\cdot) = 1$ series, $w(\cdot) = 2$ parallel, $w(\cdot) = 3$

coasting, or $w(\cdot) = 4$ braking that accelerate or decelerate the train and have different energy consumption. Acceleration and energy comsumption are velocity-dependent. Hence, we will need switching functions $\sigma_i(x_1) = v_i - x_1$ for given velocities $v_i, i = 1..3$. The Lagrange term reads

$$\begin{aligned} L(x,1) &= \begin{cases} e\, p_1 & \text{if } \sigma_1 \geq 0 \\ e\, p_2 & \text{else if } \sigma_2 \geq 0 \\ e\, \sum_{i=0}^{5} c_i(1) \left(\frac{1}{10}\gamma\, x_1\right)^{-i} & \text{else} \end{cases} && (7.28) \\ L(x,2) &= \begin{cases} \infty & \text{if } \sigma_2 \geq 0 \\ e\, p_3 & \text{else if } \sigma_3 \geq 0 \\ e\, \sum_{i=0}^{5} c_i(2) \left(\frac{1}{10}\gamma\, x_1 - 1\right)^{-i} & \text{else} \end{cases} && (7.29) \\ L(x,3) &= L(x,4) = 0. && (7.30) \end{aligned}$$

The right hand side function $f_1(x,w)$ reads is

$$\begin{aligned} f_1(x,1) &= \begin{cases} f_1^{1A} := \frac{g\, e\, a_1}{W_{\text{eff}}} & \text{if } \sigma_1 \geq 0 \\ f_1^{1B} := \frac{g\, e\, a_2}{W_{\text{eff}}} & \text{else if } \sigma_2 \geq 0 \\ f_1^{1C} := \frac{g\, (e\, T(x_1,1) - R(x_1))}{W_{\text{eff}}} & \text{else} \end{cases} && (7.31) \\ f_1(x,2) &= \begin{cases} 0 & \text{if } \sigma_2 \geq 0 \\ f_1^{2B} := \frac{g\, e\, a_3}{W_{\text{eff}}} & \text{else if } \sigma_3 \geq 0 \\ f_1^{2C} := \frac{g\, (e\, T(x_1,2) - R(x_1))}{W_{\text{eff}}} & \text{else} \end{cases} && (7.32) \\ f_1(x,3) &= -\frac{g\, R(x_1)}{W_{\text{eff}}} - C, && (7.33) \\ f_1(x,4) &= -u = -u_{\max}. && (7.34) \end{aligned}$$

The braking deceleration $u(\cdot)$ can be varied between 0 and a given $u_{\max}$. It can be shown that for problem (7.27) only maximal braking can be optimal, hence we fixed $u(\cdot)$ to $u_{\max}$ without loss of generality. Occurring forces are

$$\begin{aligned} R(x_1) &= ca\, \gamma^2 {x_1}^2 + bW\gamma x_1 + \frac{1.3}{2000} W + 116, && (7.35) \\ T(x_1,1) &= \sum_{i=0}^{5} b_i(1) \left(\frac{1}{10}\gamma x_1 - 0.3\right)^{-i}, && (7.36) \\ T(x_1,2) &= \sum_{i=0}^{5} b_i(2) \left(\frac{1}{10}\gamma x_1 - 1\right)^{-i}. && (7.37) \end{aligned}$$

Parameters are listed in Table 7.2, while $b_i(w)$ and $c_i(w)$ are given by

Symbol	Value	Unit	Symbol	Value	Unit
W	78000	lbs	v_1	0.979474	mph
W_{eff}	85200	lbs	v_2	6.73211	mph
S	2112	ft	v_3	14.2658	mph
S_4	700	ft	v_4	22.0	mph
S_5	1200	ft	v_5	24.0	mph
γ	$\frac{3600}{5280}$	$\frac{\text{sec}}{\text{h}} / \frac{\text{ft}}{\text{mile}}$	a_1	6017.611205	lbs
a	100	ft^2	a_2	12348.34865	lbs
n_{wag}	10	-	a_3	11124.63729	lbs
b	0.045	-	$u_{\max}$	4.4	ft / sec^2
C	0.367	-	p_1	106.1951102	-
g	32.2	$\frac{\text{ft}}{\text{sec}^2}$	p_2	180.9758408	-
e	1.0	-	p_3	354.136479	-

Table 7.2: Parameters used for the subway MIOCP and its variants.

$b_0(1)$	$-0.1983670410E02,$	$c_0(1)$	$0.3629738340E02,$
$b_1(1)$	$0.1952738055E03,$	$c_1(1)$	$-0.2115281047E03,$
$b_2(1)$	$0.2061789974E04,$	$c_2(1)$	$0.7488955419E03,$
$b_3(1)$	$-0.7684409308E03,$	$c_3(1)$	$-0.9511076467E03,$
$b_4(1)$	$0.2677869201E03,$	$c_4(1)$	$0.5710015123E03,$
$b_5(1)$	$-0.3159629687E02,$	$c_5(1)$	$-0.1221306465E03,$
$b_0(2)$	$-0.1577169936E03,$	$c_0(2)$	$0.4120568887E02,$
$b_1(2)$	$0.3389010339E04,$	$c_1(2)$	$0.3408049202E03,$
$b_2(2)$	$0.6202054610E04,$	$c_2(2)$	$-0.1436283271E03,$
$b_3(2)$	$-0.4608734450E04,$	$c_3(2)$	$0.8108316584E02,$
$b_4(2)$	$0.2207757061E04,$	$c_4(2)$	$-0.5689703073E01,$
$b_5(2)$	$-0.3673344160E03,$	$c_5(2)$	$-0.2191905731E01.$

Details about the derivation of this model and the assumptions made can be found in [43] or in [160].

7.6.2 Results

The optimal trajectory for this problem has been calculated by means of an indirect approach in [43, 160], and based on the direct multiple shooting method in [214]. The resulting trajectory is listed in Table 7.3.

Time t	$w(\cdot)$	$f_1 =$	x_0 [ft]	x_1 [mph]	x_1 [ftps]	Energy
0.00000	1	f_1^{1A}	0.0	0.0	0.0	0.0
0.63166	1	f_1^{1B}	0.453711	0.979474	1.43656	0.0186331
2.43955	1	f_1^{1C}	10.6776	6.73211	9.87375	0.109518
3.64338	2	f_1^{2B}	24.4836	8.65723	12.6973	0.147387
5.59988	2	f_1^{2C}	57.3729	14.2658	20.9232	0.339851
12.6070	1	f_1^{1C}	277.711	25.6452	37.6129	0.93519
45.7827	3	$f_1(3)$	1556.5	26.8579	39.3915	1.14569
46.8938	3	$f_1(3)$	1600	26.5306	38.9115	1.14569
57.1600	4	$f_1(4)$	1976.78	23.5201	34.4961	1.14569
65.0000	-	$-$	2112	0.0	0.0	1.14569

Table 7.3: Optimal trajectory for the subway MIOCP as calculated in [43, 160, 214].

7.6.3 Variants

The given parameters have to be modified to match different parts of the track, subway train types, or amount of passengers. A minimization of travel time might also be considered.

The problem becomes more challenging, when additional point or path constraints are considered. First we consider the point constraint

$$x_1 \le v_4 \text{ if } x_0 = S_4 \tag{7.38}$$

for a given distance $0 < S_4 < S$ and velocity $v_4 > v_3$. Note that the state $x_0(\cdot)$ is strictly monotonically increasing with time, as $\dot{x}_0 = x_1 > 0$ for all $t \in (0, T)$.

The optimal order of gears for $S_4 = 1200$ and $v_4 = 22/\gamma$ with the additional interior point constraints (7.38) is $1,2,1,3,4,2,1,3,4$. The stage lengths between switches are 2.86362, 10.722, 15.3108, 5.81821, 1.18383, 2.72451, 12.917, 5.47402, and 7.98594 with $\Phi = 1.3978$. For different parameters $S_4 = 700$ and $v_4 = 22/\gamma$ we obtain the gear choice 1, 2, 1, 3, 2, 1, 3, 4 and stage lengths 2.98084, 6.28428, 11.0714, 4.77575, 6.0483, 18.6081, 6.4893, and 8.74202 with $\Phi = 1.32518$.

A more practical restriction are path constraints on subsets of the track. We will consider a problem with additional path constraints

$$x_1 \le v_5 \;\text{ if }\; x_0 \ge S_5. \tag{7.39}$$

The additional path constraint changes the qualitative behavior of the relaxed solution. While all solutions considered this far were bang–bang and the main work consisted in finding the switching points, we now have a path–constrained arc. The optimal solutions for refined grids yield a series of monotonically decreasing objective function values, where the limit is the best value that can be approximated by an integer feasible solution. In our case we obtain

$$1.33108, 1.31070, 1.31058, 1.31058, \ldots \tag{7.40}$$

Figure 7.6 shows a possible integer realization. Note that the solutions approximate the optimal driving behavior (a convex combination of two operation modes) by switching between the two and causing a touching of the velocity constraint from below as many times as we switch. Hence, there is a trade-off between energy consumption and number of switches.

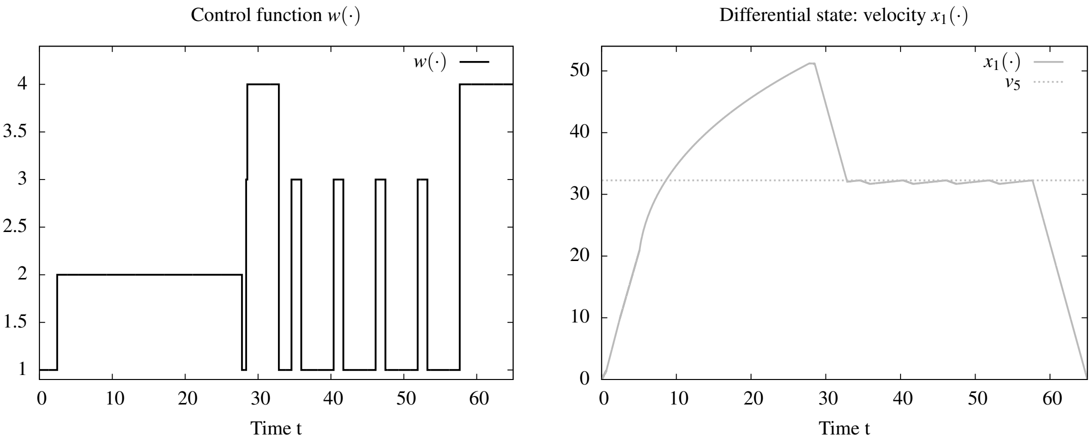


Figure 7.6: Left: the operation mode $w(t) \in \{1,2,3,4\}$. Right: the differential state *velocity* of a subway train over time. The dotted horizontal line indicates the maximum velocity after S_5. The energy-optimal solution needs to stay as close as possible to the maximum velocity on this time interval to avoid even higher energy-intensive accelerations in the start-up phase to match the terminal time constraint $t_f \leq 65$ to reach the next station.

7.7 Resetting calcium oscillations

The aim of the control problem is to identify strength and timing of inhibitor stimuli that lead to a phase singularity which annihilates intra-cellular calcium oscillations. This is formulated as an objective function that aims at minimizing the state deviation from a desired unstable steady state, integrated over time. A calcium oscillator model describing intra-cellular calcium spiking in hepatocytes induced by an extracellular increase in adenosine triphosphate (ATP) concentration is described. The calcium signaling pathway is initiated via a receptor activated G-protein inducing the intra-cellular release of inositol triphosphate (IP3) by phospholipase C. The IP3 triggers the opening of endoplasmic reticulum and plasma membrane calcium channels and a subsequent inflow of calcium ions from intra-cellular and extracellular stores leading to transient calcium spikes.

The mathematical equations form a small-scale ODE model. The interior point equality conditions fix the initial values of the differential states. The problem is, despite of its low dimension, very hard to solve, as the target state is unstable.

7.7.1 Model and optimal control problem

The model for the calcium oscillator comprises four system states $x_i \in C^1[0,T]$. They describe the concentrations of activated G-protein $x_1(t)$, active phospholipase C $x_2(t)$, intracellular calcium $x_3(t)$ and intra-ER calcium $x_4(t)$. The model takes into account well known feedback-regulations of the pathway, in particular CICR (calcium induced calcium release) and active transport of calcium from the cytoplasm across both ER-membrane and plasma membrane via SERCA (sarco-endoplasmic reticulum Ca^{2+}-ATPase) and PMCA (plasma membrane Ca^{2+}-ATPase) pumps.

We introduce control functions $u, v \in L^\infty[0,T]$ representing the influence of drug stimuli. The control function $u(\cdot)$ represents the temporally varying concentration of an uncompetitive inhibitor of the PMCA ion pump. The control function $v(\cdot)$ is the inhibitor of PLC activation.

In the interest of notational simplicity we write v, u, x_i for $v(t), u(t), x_i(t)$ and formulate the initial value problem for given $u(\cdot)$ and $v(\cdot)$ as $\dot{x}(t) = f^{\mathrm{Ca}}(x(t), u(t), v(t))$:

$$\dot{x_1} = k_1 + k_2 x_1 - \frac{k_3 x_1 x_2}{x_1 + K_4} - \frac{k_5 x_1 x_3}{x_1 + K_6} \tag{7.41a}$$

$$\dot{x_2} = (1-v)\cdot k_7 x_1 - \frac{k_8 x_2}{x_2 + K_9} \tag{7.41b}$$

$$\dot{x_3} = \frac{k_{10} x_2 x_3 x_4}{x_4 + K_{11}} + k_{12} x_2 + k_{13} x_1 - \frac{k_{14} x_3}{u\, x_3 + K_{15}} - \frac{k_{16} x_3}{x_3 + K_{17}} + \frac{x_4}{10} \tag{7.41c}$$

$$\dot{x_4} = -\frac{k_{10} x_2 x_3 x_4}{x_4 + K_{11}} + \frac{k_{16} x_3}{x_3 + K_{17}} - \frac{x_4}{10}, \tag{7.41d}$$

$$x(0) = x_0 := (0.03966, 1.09799, 0.00142, 1.65431)^T, x_0, \tag{7.41e}$$

for $t \in [0,T]$ with fixed initial values x_0 and parameter values $k_1 = 0.09, k_2 = 2.30066, k_3 = 0.64, K_4 = 0.19, k_5 = 4.88, K_6 = 1.18, k_7 = 2.08, k_8 = 32.24, K_9 = 29.09, k_{10} = 5.0, K_{11} = 2.67, k_{12} = 0.7, k_{13} = 13.58, k_{14} = 153.0, K_{15} = 0.16, k_{16} = 4.85, K_{17} = 0.05$.

The uncontrolled case $u(t) \equiv 1, v(t) \equiv 0$ with $\bar{u} = 1.3$ gives rise to bursting-type limit cycle oscillations, compare Figure 7.7 left. The control aim is to identify drug stimuli $u(t), v(t)$ leading to a phase singularity which annihilates the intracellular calcium oscillations. This is equivalent to driving the system into an unstable steady state. We fix the control function $v(\cdot)$ to zero in this scenario. A solution with both inhibitors can be found in [163], along with references to the origin of the model.

For practical reasons of applicability, we are interested in a $\{\bar{u}^{\min}, \bar{u}^{\max}\}$ valued solution for the control function $u(\cdot)$, where the upper bound $\bar{u}^{\max}$ may be subject to optimization, but will be a constant-in time value for the time-dependent control function. This corresponds to a calibration of technical equipment to a fixed dosage, that needs to stay constant over time once it has been a priori calibrated.

Again, a partial outer convexification with respect to the integer control functions is beneficial. Applying it for $u(t) \in \{\bar{u}^{\min}, \bar{u}^{\max}\}$, we obtain instead of (7.41c) the equation

$$\begin{aligned} \dot{x_3} &= \frac{k_{10}x_2x_3x_4}{x_4+K_{11}} + k_{12}x_2 + k_{13}x_1 - \frac{k_{16}x_3}{x_3+K_{17}} + \frac{x_4}{10} \\ &\quad - \left(\omega \frac{k_{14}x_3}{\bar{u}^{\max}\, x_3+K_{15}} + (1-\omega)\frac{k_{14}x_3}{\bar{u}^{\min}\, x_3+K_{15}} \right) \end{aligned} \tag{7.42}$$

with a new control function $\omega(\cdot) \in \{0,1\}$, whereas the equations (7.41a, 7.41b, 7.41d) stay the same. Let in the following $f^{\mathrm{OC}}(x(t), \omega(t), v(t) \equiv 0)$ denote the function that consists of the right hand sides of (7.41a,7.41b,7.42,7.41d) for $v(\cdot)$ fixed to zero. Note that by construction it holds

$$\begin{aligned} f^{\mathrm{OC}}(x(t), \omega(t) \equiv 1, v(t)) &= f^{\mathrm{Ca}}(x(t), u(t) \equiv \bar{u}^{\max}, v(t)) \\ f^{\mathrm{OC}}(x(t), \omega(t) \equiv 0, v(t)) &= f^{\mathrm{Ca}}(x(t), u(t) \equiv \bar{u}^{\min}, v(t)). \end{aligned}$$

We formulate a control problem with the minimization of the system state deviation from the desired steady state and a penalization of control functions integrated over time as objective function,

$$\min_{x,\omega,\bar{u}^{\max}} J(x, \bar{u}^{\max}, p) := \int_0^T \sum_{i=1}^4 \left(\frac{x_i(\tau)-x_i^s}{x_i^s} \right)^2 + 100\omega(t)\, \mathrm{d}\tau \tag{7.43a}$$

subject to

$$\begin{aligned} \dot{x}(t) &= f^{\mathrm{OC}}(x(t), \omega, v \equiv 0), && (7.43\text{b}) \\ x(0) &= x_0, && (7.43\text{c}) \\ x_i(t) &\geq 0, && (7.43\text{d}) \\ \bar{u}^{\min} &= 1.0, && (7.43\text{e}) \\ \bar{u}^{\max} &\in [1.1, 1.3], && (7.43\text{f}) \\ \omega(t) &\in \{0,1\} && (7.43\text{g}) \end{aligned}$$

with $\omega \in L^\infty[0,T]$, a fixed time horizon $T = 22$, $t \in [0,T]$, and initial values and parameters as above. The constants

$$x_1^s = 6.78677, x_2^s = 22.65836, x_3^s = 0.38431, x_4^s = 0.28977$$

refer to the concentrations corresponding to the unstable steady state of the uncontrolled system surrounded by the limit cycle.

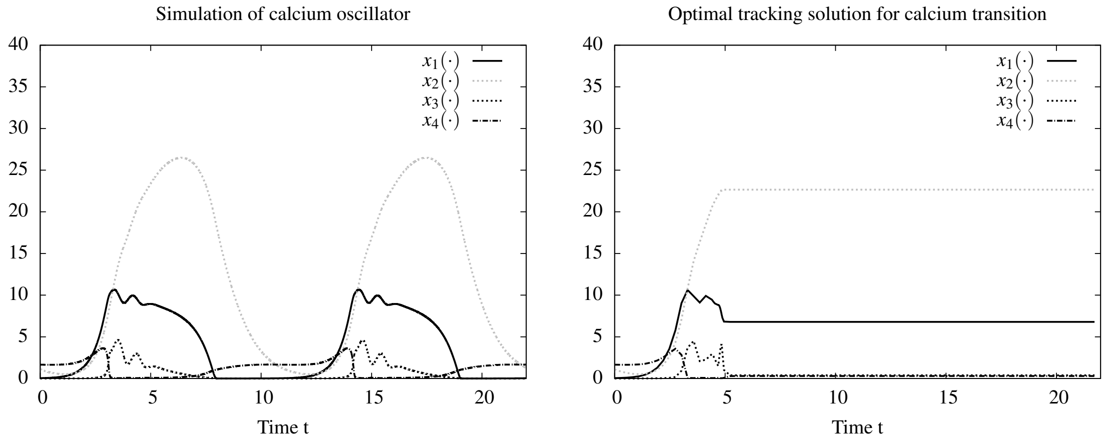


Figure 7.7: Left: simulation of periodic calcium oscillations on a stable orbit. The plot shows the four differential states on the fixed time horizon $[0,22]$ for $u(\cdot) \equiv 1, v(\cdot) \equiv 0$. Right: optimal tracking transition of calcium oscillations towards the unstable steady state. The plot shows the four differential states corresponding to the control from Figure 7.8.

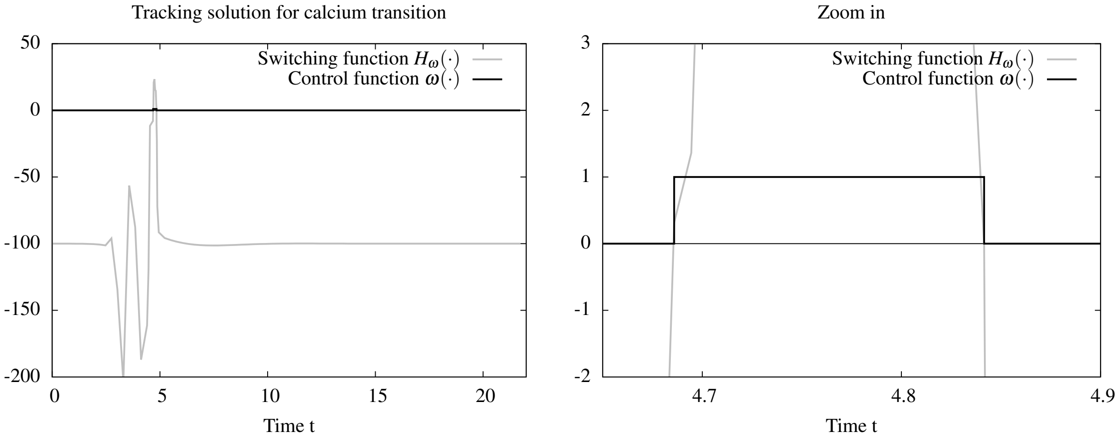


Figure 7.8: Transition of calcium oscillations towards the unstable steady state via the tracking-type objective (7.43a). The plot shows the bang-bang solution and the corresponding switching function for $\omega(\cdot)$ given by (7.44). The right hand side plot shows a zoom into the area where $\omega(\cdot) = 1$. The switching function has been calculated a posteriori for illustration.

7.7.2 Results

The optimal trajectory for problem (7.43) is depicted in Figures 7.8 and 7.7. Figure 7.8 shows the bang-bang solution for the control function $\omega(\cdot)$ that consists of a pulse of length 0.149103 with $\omega(t) \equiv 1$ for $t \in [4.694485, 4.843588]$. As was already shown and visualized in [203, pp. 152], the sensitivity of the objective function with respect to the timing is extremly high, because of the instability of the target steady-state. The gray curve in Figure 7.8 shows the switching function

$$H_\omega = -100 - \lambda_3(t) \left(\frac{k_{14} x_3}{\bar{u}^{\max}\, x_3 + K_{15}} - \frac{k_{14} x_3}{\bar{u}^{\min}\, x_3 + K_{15}} \right) \tag{7.44}$$

that is the partial derivative of the Hamiltonian $H = -L(t) + \lambda(t)^T f^{\mathrm{OC}}(x(t), \omega, v \equiv 0)$.We use a direct method for optimal control and do not need the switching function. It has been calculated a posteriori for visualization, making use of the Lagrangian multipliers of the matching conditions as approximations of the continuous adjoints $\lambda(\cdot)$. As can be seen, the timings for $H_v(t) > 0$ and $\omega(t) = 1$ coincide, as expected from a control problem that is linear in $\omega(\cdot)$.

Figure 7.7 shows the four corresponding differential states $x_i(\cdot)$. As can be seen, the control goal to drive the system into the unstable steady state and to keep it there on the time horizon is accomplished.

For the switching time solution we get a Hessian approximation of

$$H = \begin{pmatrix} 6336620841 & 13780350694 & -100.76 \\ 13780350694 & 85604174121 & -622.76 \\ -100.76 & -622.76 & 105.12 \end{pmatrix}$$

where our variables are the three arc lengths $t_1, t_2 - t_1$ and $T - t_2$. All other variables are at their simple bounds or fixed. The linearized constraints (fixed end time $T = 22$) are

$$g_x = \begin{pmatrix} -1 & -1 & -1 \end{pmatrix}$$

If we eliminate the first stage ($y = t_1$) and $z = (t_2 - t_1, t_{\mathrm{f}} - t_2)$, we get a reduced Hessian of

$$H^{\mathrm{red}} = \begin{pmatrix} -g_y^{-1} g_z \\ I \end{pmatrix}^T H \begin{pmatrix} -g_y^{-1} g_z \\ I \end{pmatrix}$$

with positive eigenvalues $6.53 \cdot 10^{10}$ and $0.54 \cdot 10^{10}$. Hence the reduced Hessian is positive definite and the second order sufficient conditions are fulfilled.

7.7.3 Variants

One may also be interested in a time-minimal formulation to reach the unstable steady state. Here we fix the control $u(\cdot)$ that represents the temporally varying concentration of an uncompetitive

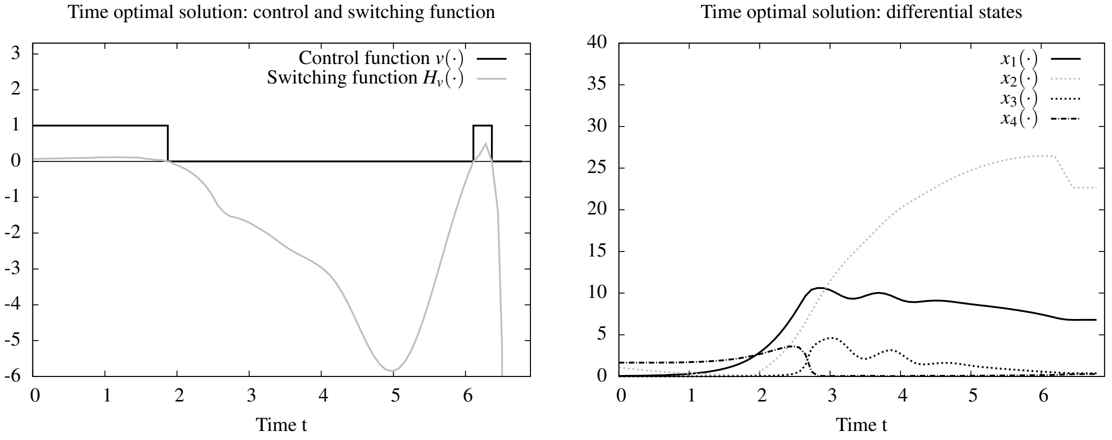


Figure 7.9: Time-optimal transition of calcium oscillations towards the unstable steady state. Left: the bang-bang solution and the corresponding switching function $H_v = -\lambda_2(t) \cdot k_7 \cdot x_1(t)$. Right: the four corresponding differential states.

inhibitor of the PMCA ion pump, instead of the control function $v(\cdot)$ which is the inhibitor of PLC activation.

These two changes give rise to the optimal control problem

$$\min_{x,v,T} \quad T \tag{7.45a}$$

subject to

$$\begin{aligned}
\dot{x}(t) &= f^{\mathrm{OC}}(x(t), u \equiv \bar{u}^{\min}, v), && (7.45b)\\
x(0) &= x_0, && (7.45c)\\
x(T) &\leq x^s + \varepsilon e, && (7.45d)\\
x(T) &\geq x^s - \varepsilon e, && (7.45e)\\
x_i(t) &\geq 0, && (7.45f)\\
v(t) &\in \{0,1\} && (7.45g)
\end{aligned}$$

for $t \in [0,T]$, the all ones vector $e = (1,1,1,1)^T$ and a small tolerance $\varepsilon = 10^{-5}$ for the terminal constraint $x(t) = x^s$.

The optimal trajectory for problem (7.45) is depicted in Figure 7.9. The left plot shows the bang-bang solution for the control function $\omega(\cdot)$ that consists of two pulses with $v(t) \equiv 1$ for $t \in [0, 1.914173]$ and $t \in [6.195274, 6.450467]$. The minimum time is $T = 6.896168$, which is obviously slower than what can be achieved with the control $u(\cdot)$, compare Figure 7.7 right.

For the switching time solution we get a Hessian approximation of

$$H = 10^3 \cdot \begin{pmatrix} -1.7312 & -1.7376 & -0.6022 & -0.0190 \\ -1.7376 & -1.7431 & -0.6090 & -0.0191 \\ -0.6022 & -0.6090 & 0.8103 & 0.0044 \\ -0.0190 & -0.0191 & 0.0044 & 0.0158 \end{pmatrix}$$

where our variables are the four arc lengths $t_1, t_2 - t_1, t_3 - t_2$, and $T - t_3$. All other variables are at their simple bounds or fixed. There are three active constraints in the solution, the constraints (7.45e) for components x_1, x_2 and x_4. Their linearizations yield

$$g_x = \begin{pmatrix} -4.9141 & -4.9636 & 5.3257 & -1.710^{-6} \\ -3.9168 & -3.8881 & -9.8988 & -3.210^{-5} \\ 0.6144 & 0.6188 & -0.2928 & 0.0003 \end{pmatrix}$$

If we eliminate the first three stages ($y = (t_1, t_2 - t_1, t_3 - t_2)$ and $z = T - t_3$), we get a reduced Hessian of

$$H^{\text{red}} = \begin{pmatrix} -g_y^{-1} g_z \\ 1 \end{pmatrix}^T H \begin{pmatrix} -g_y^{-1} g_z \\ 1 \end{pmatrix} = 1.29910^9$$

which is obviously positive definite.

Alternatively, also the annihilation of calcium oscillations with PLC activation inhibition, i.e., the use of two control functions is possible, compare [163]. Of course, results depend very much on the scaling of the deviation in the objective function.

7.8 Supermarket refrigeration system

This benchmark problem was formulated first within the European network of excellence HYCON, [182] by Larsen et. al, [162]. The formulation lacks however a precise definition of initial values and constraints, which are only formulated as “soft constraints”. The task is to control a refrigeration system in an energy optimal way, while guaranteeing safeguards on the temperature of the showcases. This problem would typically be a moving horizon online optimization problem, here it is defined as a fixed horizon optimization task.

The mathematical equations form a periodic ODE model.

7.8.1 Model and optimal control problem

The MIOCP reads

$$
\begin{aligned}
\min_{x,w,t_\mathrm{f}} \quad & \tfrac{1}{t_\mathrm{f}} \textstyle\int_0^{t_\mathrm{f}} (w_2 + w_3) \cdot 0.5 \cdot \eta_{vol} \cdot V_{sl} \cdot f \,\mathrm{d}t \\
\text{s.t.} \quad & \dot{x}_0 = \frac{\Big(x_4\big(x_2 - T_e(x_0)\big) + x_8\big(x_6 - T_e(x_0)\big)\Big)}{V_{suc} \cdot \frac{d\rho_{suc}}{dP_{suc}}(x_0)} \cdot \frac{UA_{wrm}}{M_{rm} \cdot \Delta h_{lg}(x_0)} \\
& \qquad + \frac{M_{rc} - \eta_{vol} \cdot V_{sl} \cdot 0.5\;(w_2 + w_3)\,\rho_{suc}(x_0)}{V_{suc} \cdot \frac{d\rho_{suc}}{dP_{suc}}(x_0)} \\
& \dot{x}_1 = -\frac{UA_{goods-air}\,(x_1 - x_3)}{M_{goods} \cdot C_{p,goods}} \\
& \dot{x}_2 = \frac{UA_{air-wall}\,(x_3 - x_2) - \dfrac{UA_{wrm}}{M_{rm}} x_4\big(x_2 - T_e(x_0)\big)}{M_{wall} \cdot C_{p,wall}} \\
& \dot{x}_3 = \frac{UA_{goods-air}\,(x_1 - x_3) + \dot{Q}_{airload} - UA_{air-wall}\,(x_3 - x_2)}{M_{air} \cdot C_{p,air}} \\
& \dot{x}_4 = \left(\frac{M_{rm} - x_4}{\tau_{fill}}\right) w_0 - \frac{UA_{wrm}(1 - w_0)}{M_{rm} \cdot \Delta h_{lg}(x_0)} x_4\big(x_2 - T_e(x_0)\big) \\
& \dot{x}_5 = -\frac{UA_{goods-air}\,(x_5 - x_7)}{M_{goods} \cdot C_{p,goods}} \\
& \dot{x}_6 = \frac{UA_{air-wall}\,(x_7 - x_6) - \dfrac{UA_{wrm}}{M_{rm}} x_8\big(x_6 - T_e(x_0)\big)}{M_{wall} \cdot C_{p,wall}} \\
& \dot{x}_7 = \frac{UA_{goods-air}\,(x_5 - x_7) + \dot{Q}_{airload} - UA_{air-wall}\,(x_7 - x_6)}{M_{air} \cdot C_{p,air}} \\
& \dot{x}_8 = \left(\frac{M_{rm} - x_8}{\tau_{fill}}\right) w_1 - \frac{UA_{wrm}(1 - w_1)}{M_{rm} \cdot \Delta h_{lg}(x_0)} x_8\big(x_6 - T_e(x_0)\big) \\
& x(0) = x(t_\mathrm{f}), \\
& 650 \le t_\mathrm{f} \le 750, \\
& x_0 \le 1.7,\; 2 \le x_3 \le 5,\; 2 \le x_7 \le 5 \\
& w(t) \in \{0,1\}^4, \quad t \in [0, t_\mathrm{f}].
\end{aligned}
$$

The differential state x_0 describes the suction pressure in the suction manifold (in bar). The next three states model temperatures in the first display case (in C). x_1 is the goods' temperature, x_2 the one of the evaporator wall and x_3 the air temperature surrounding the goods. x_4 then models the mass of the liquefied refrigerant in the evaporator (in kg). x_5 to x_8 describe the corresponding states in the second display case. w_0 and w_1 describe the inlet valves of the first two display cases, respectively. w_2 and w_3 denote the activity of a single compressor.

The model uses the parameter values listed in Table 7.4 and the polynomial functions obtained

Symbol	Value	Unit	Description
$\dot{Q}_{airload}$	3000.00	$\frac{J}{s}$	Disturbance, heat transfer
$\dot{m}_{rc}$	0.20	$\frac{kg}{s}$	Disturbance, constant mass flow
M_{goods}	200.00	kg	Mass of goods
$C_{p,goods}$	1000.00	$\frac{J}{kg\cdot K}$	Heat capacity of goods
$UA_{goods-air}$	300.00	$\frac{J}{s\cdot K}$	Heat transfer coefficient
M_{wall}	260.00	kg	Mass of evaporator wall
$C_{p,wall}$	385.00	$\frac{J}{kg\cdot K}$	Heat capacity of evaporator wall
$UA_{air-wall}$	500.00	$\frac{J}{s\cdot K}$	Heat transfer coefficient
M_{air}	50.00	kg	Mass of air in display case
$C_{p,air}$	1000.00	$\frac{J}{kg\cdot K}$	Heat capacity of air
UA_{wrm}	4000.00	$\frac{J}{s\cdot K}$	Maximum heat transfer coefficient
τ_{fill}	40.00	s	Filling time of the evaporator
T_{SH}	10.00	K	Superheat in the suction manifold
M_{rm}	1.00	kg	Maximum mass of refrigerant
V_{suc}	5.00	m^3	Total volume of suction manifold
V_{sl}	0.08	$\frac{m^3}{s}$	Total displacement volume
η_{vol}	0.81	$-$	Volumetric efficiency

Table 7.4: Parameters used for the supermarket refrigeration problem.

from interpolations:

$$\begin{aligned} T_e(x_0) &= -4.3544x_0^2 + 29.224x_0 - 51.2005, \\ \Delta h_{lg}(x_0) &= (0.0217x_0^2 - 0.1704x_0 + 2.2988)\cdot 10^5, \\ \rho_{suc}(x_0) &= 4.6073x_0 + 0.3798, \\ \frac{d\rho_{suc}}{dP_{suc}}(x_0) &= -0.0329{x_0}^3 + 0.2161{x_0}^2 - 0.4742x_0 + 5.4817. \end{aligned}$$

7.8.2 Results

For the relaxed problem the optimal solution is $\Phi = 12072.45$. The integer solution plotted in Figure 7.10 is feasible, but yields an increased objective function value of $\Phi = 12252.81$, a compromise between effectiveness and a reduced number of switches.

7.8.3 Variants

Since the compressors are parallel connected one can introduce a single control $w_2 \in \{0,1,2\}$ instead of two equivalent controls. The same holds for scenarios with n parallel connected compressors.

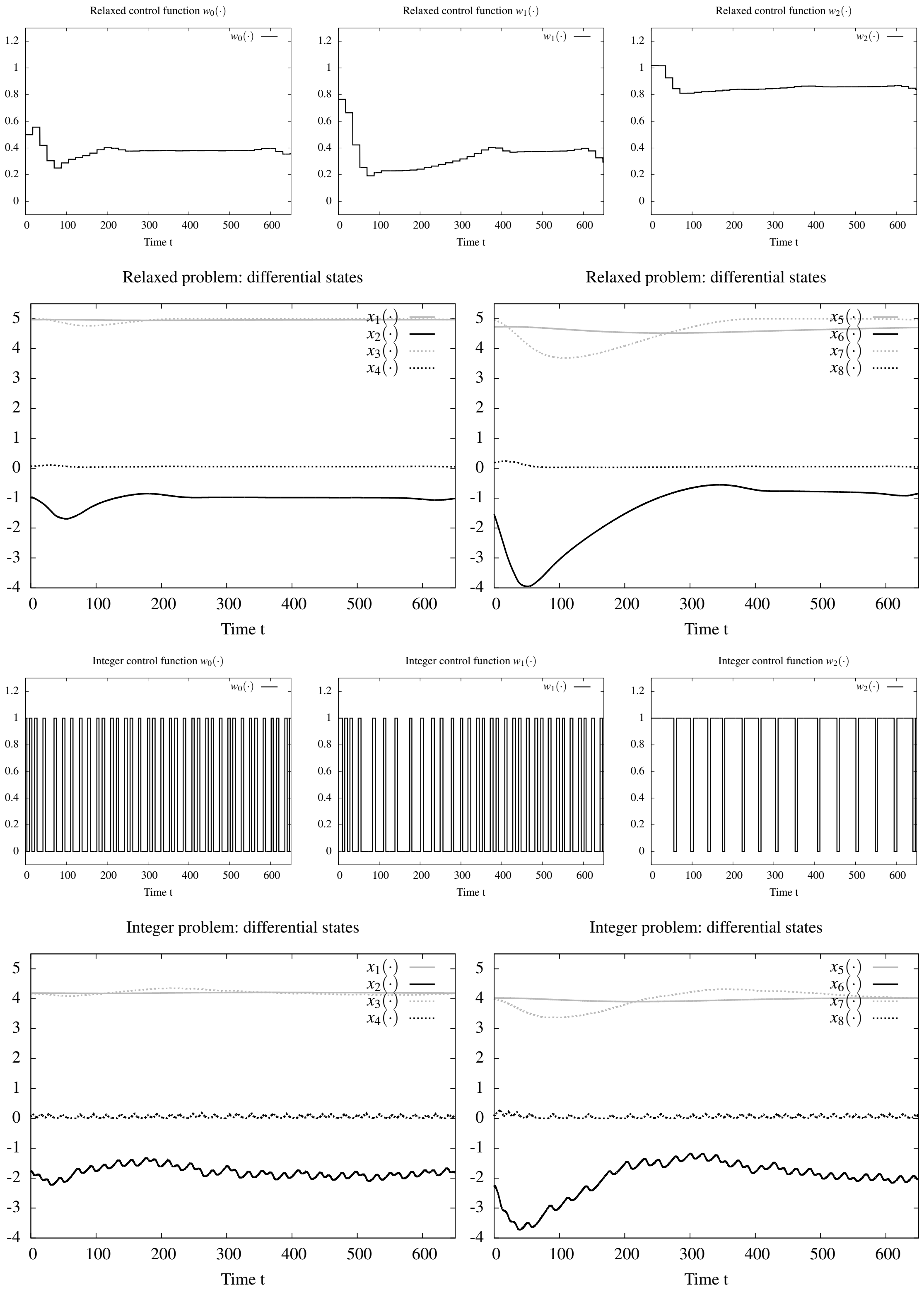


Figure 7.10: Periodic trajectories for optimal relaxed (top 2 rows) and integer feasible controls (bottom rows).

In [162], the problem was stated slightly different:

- The temperature constraints weren't hard bounds but there was a penalization term added to the objective function to minimize the violation of these constraints.
- The differential equation for the mass of the refrigerant had another switch, if the valve (e.g. w_0) is closed. It was formulated as

$$\dot{x_4} = \begin{cases} \dfrac{M_{rm} - x_4}{\tau_{fill}} & \text{if } w_0 = 1 \\ -\dfrac{UA_{wrm}}{M_{rm} \cdot \Delta h_{lg}(x_0)} x_4\big(x_2 - T_e(x_0)\big) & \text{if } w_0 = 0 \text{ and } x_4 > 0 \\ 0 & \text{if } w_0 = 0 \text{ and } x_4 = 0 \end{cases}$$

  This additional switch is redundant because the mass itself is a factor on the right hand side and so the complete right hand side is 0 if $x_4 = 0$.

- A night scenario with two different parameters was given. At night the following parameters change their value to $\dot{Q}_{airload} = 1800.00\frac{J}{s}$ and $\dot{m}_{rc} = 0.00\frac{kg}{s}$. Additionally the constraint on the suction pressure $x_0(t)$ is softened to $x_0(t) \leq 1.9$.
- The number of compressors and display cases is not fixed. Larsen also proposed the problem with 3 compressors and 3 display cases. This leads to a change in the compressor rack's performance to $V_{sl} = 0.095\frac{m^3}{s}$. Unfortunately this constant is only given for these two cases although Larsen proposed scenarios with more compressors and display cases.

7.9 Elchtest testdrive

We consider a time-optimal car driving maneuver to avoid an obstacle with small steering effort. At any time, the car must be positioned on a prescribed track. This control problem was first formulated in [105] and used for subsequent studies [106, 147]. It has also been investigated in Section 4.5.
The mathematical equations form a small-scale ODE model. The interior point equality conditions fix initial and terminal values of the differential states, the objective is of minimum-time type.

7.9.1 Model and optimal control problem

We consider a car model derived under the simplifying assumption that rolling and pitching of the car body can be neglected. Only a single front and rear wheel is modelled, located in the virtual center of the original two wheels. Motion of the car body is considered on the horizontal plane only.

The MIOCP reads

$$
\begin{aligned}
&\min_{t_\mathrm{f}, x(\cdot), u(\cdot)} \quad t_\mathrm{f} + \int_0^{t_\mathrm{f}} w_\delta^2(t)\ \mathrm{d}t && (7.46\mathrm{a})\\
&\text{s.t.} \qquad \dot{c}_\mathrm{x} = v\ \cos\big(\psi - \beta\big) && (7.46\mathrm{b})\\
&\qquad\qquad \dot{c}_\mathrm{y} = v\ \sin\big(\psi - \beta\big) && (7.46\mathrm{c})\\
&\qquad\qquad \dot{v} = \frac{1}{m}\Big(\big(F_\mathrm{lr}^{\mu} - F_\mathrm{Ax}\big)\cos\beta + F_\mathrm{lf}\cos\big(\delta + \beta\big) && (7.46\mathrm{d})\\
&\qquad\qquad\qquad - \big(F_\mathrm{sr} - F_\mathrm{Ay}\big)\sin\beta - F_\mathrm{sf}\sin\big(\delta + \beta\big)\Big) &&\\
&\qquad\qquad \dot{\delta} = w_\delta && (7.46\mathrm{e})\\
&\qquad\qquad \dot{\beta} = w_\mathrm{z} - \frac{1}{m\,v}\Big(\ \big(F_\mathrm{lr} - F_\mathrm{Ax}\big)\ \sin\beta + F_\mathrm{lf}\ \sin\big(\delta + \beta\big) && (7.46\mathrm{f})\\
&\qquad\qquad\qquad + \big(F_\mathrm{sr} - F_\mathrm{Ay}\big)\ \cos\beta + F_\mathrm{sf}\ \cos\big(\delta + \beta\big)\Big) &&\\
&\qquad\qquad \dot{\psi} = w_\mathrm{z} && (7.46\mathrm{g})\\
&\qquad\qquad \dot{w}_\mathrm{z} = \frac{1}{I_\mathrm{zz}}\big(F_\mathrm{sf}\, l_\mathrm{f}\ \cos\delta - F_\mathrm{sr}\, l_\mathrm{r} - F_\mathrm{Ay}\ e_\mathrm{SP} + F_\mathrm{lf}\, l_\mathrm{f}\ \sin\delta\big) && (7.46\mathrm{h})\\
&\qquad\qquad c_\mathrm{y}(t) \in \big[P_\mathrm{l}(c_\mathrm{x}(t)) + \tfrac{B}{2}, P_\mathrm{u}(c_\mathrm{x}(t)) - \tfrac{B}{2}\big] && (7.46\mathrm{i})\\
&\qquad\qquad w_\delta(t) \in [-0.5, 0.5], \quad F_\mathrm{B}(t) \in [0, 1.5 \cdot 10^4], \quad \phi(t) \in [0,1] && (7.46\mathrm{j})\\
&\qquad\qquad \mu(t) \in \{1, \dots, 5\} && (7.46\mathrm{k})\\
&\qquad\qquad x(t_0) = \big(-30, \text{free}, 10, 0, 0, 0, 0\big)^{\mathcal{T}}, \quad (c_\mathrm{x}, \psi)(t_\mathrm{f}) = (140, 0) && (7.46\mathrm{l})
\end{aligned}
$$

for $t \in [t_0, t_\mathrm{f}]$ almost everywhere. The four control functions contained in $u(\cdot)$ are steering wheel angular velocity w_δ, total braking force F_B, the accelerator pedal position ϕ and the gear μ. The differential states contained in $x(\cdot)$ are horizontal position of the car c_x, vertical position of the car c_y, magnitude of directional velocity of the car v, steering wheel angle δ, side slip angle β, yaw angle ψ, and the yaw angle velocity w_z.

The model parameters are listed in Table 7.5, while the forces and expressions in (7.46b) to (7.46h) are given for fixed μ by

$$
\begin{aligned}
F_\mathrm{sf,sr}(\alpha_\mathrm{f,r}) &:= D_\mathrm{f,r}\ \sin\Big(C_\mathrm{f,r}\ \arctan\big(B_\mathrm{f,r}\ \alpha_\mathrm{f,r}\\
&\qquad\qquad - E_\mathrm{f,r}(B_\mathrm{f,r}\ \alpha_\mathrm{f,r} - \arctan(B_\mathrm{f,r}\ \alpha_\mathrm{f,r}))\big)\Big),\\
\alpha_\mathrm{f} &:= \delta(t) - \arctan\left(\frac{l_\mathrm{f}\ \dot{\psi}(t) - v(t)\ \sin\beta(t)}{v(t)\ \cos\beta(t)}\right)\\
\alpha_\mathrm{r} &:= \arctan\left(\frac{l_\mathrm{r}\ \dot{\psi}(t) + v(t)\ \sin\beta(t)}{v(t)\ \cos\beta(t)}\right),\\
F_\mathrm{lf} &:= -F_\mathrm{Bf} - F_\mathrm{Rf},
\end{aligned}
$$

$$\begin{aligned}
F_{\mathrm{lr}}^{\mu} &:= \frac{i_{\mathrm{g}}^{\mu}\, i_{\mathrm{t}}}{R}\, M_{\mathrm{mot}}^{\mu}(\phi) - F_{\mathrm{Br}} - F_{\mathrm{Rr}},\\
M_{\mathrm{mot}}^{\mu}(\phi) &:= f_1(\phi)\, f_2(w_{\mathrm{mot}}^{\mu}) + (1 - f_1(\phi))\, f_3(w_{\mathrm{mot}}^{\mu}),\\
f_1(\phi) &:= 1 - \exp(-3\,\phi),\\
f_2(w_{\mathrm{mot}}) &:= -37.8 + 1.54\, w_{\mathrm{mot}} - 0.0019\, w_{\mathrm{mot}}^2,\\
f_3(w_{\mathrm{mot}}) &:= -34.9 - 0.04775\, w_{\mathrm{mot}},\\
w_{\mathrm{mot}}^{\mu} &:= \frac{i_{\mathrm{g}}^{\mu}\, i_{\mathrm{t}}}{R}\, v(t),\\
F_{\mathrm{Bf}} &:= \frac{2}{3}\, F_{\mathrm{B}}, \qquad F_{\mathrm{Br}} := \frac{1}{3}\, F_{\mathrm{B}},\\
F_{\mathrm{Rf}}(v) &:= f_{\mathrm{R}}(v)\, \frac{m\, l_{\mathrm{r}}\, g}{l_{\mathrm{f}} + l_{\mathrm{r}}},\ F_{\mathrm{Rr}}(v) := f_{\mathrm{R}}(v)\, \frac{m\, l_{\mathrm{f}}\, g}{l_{\mathrm{f}} + l_{\mathrm{r}}},\\
f_{\mathrm{R}}(v) &:= 9 \cdot 10^{-3} + 7.2 \cdot 10^{-5}\, v + 5.038848 \cdot 10^{-10}\, v^4,\\
F_{\mathrm{Ax}} &:= \frac{1}{2}\, c_{\mathrm{w}}\, \rho\, A\, v^2(t), \qquad F_{\mathrm{Ay}} := 0.
\end{aligned}$$

The test track is described by setting up piecewise cubic spline functions $P_{\mathrm{l}}(x)$ and $P_{\mathrm{r}}(x)$ modeling the top and bottom track boundary, given a horizontal position x.

$$P_{\mathrm{l}}(x) := \begin{cases}
0 & \text{if} \quad \phantom{44.5 <}\ x \le 44,\\
4\, h_2\, (x-44)^3 & \text{if} \quad 44 < x \le 44.5,\\
4\, h_2\, (x-45)^3 + h_2 & \text{if} \quad 44.5 < x \le 45,\\
h_2 & \text{if} \quad 45 < x \le 70,\\
4\, h_2\, (70-x)^3 + h_2 & \text{if} \quad 70 < x \le 70.5,\\
4\, h_2\, (71-x)^3 & \text{if} \quad 70.5 < x \le 71,\\
0 & \text{if} \quad 71 < x.
\end{cases} \tag{7.47}$$

$$P_{\mathrm{u}}(x) := \begin{cases}
h_1 & \text{if} \quad \phantom{15.5 <}\ x \le 15,\\
4\,(h_3 - h_1)\,(x-15)^3 + h_1 & \text{if} \quad 15 < x \le 15.5,\\
4\,(h_3 - h_1)\,(x-16)^3 + h_3 & \text{if} \quad 15.5 < x \le 16,\\
h_3 & \text{if} \quad 16 < x \le 94,\\
4\,(h_3 - h_4)\,(94-x)^3 + h_3 & \text{if} \quad 94 < x \le 94.5,\\
4\,(h_3 - h_4)\,(95-x)^3 + h_4 & \text{if} \quad 94.5 < x \le 95,\\
h_4 & \text{if} \quad 95 < x.
\end{cases} \tag{7.48}$$

where $B = 1.5$ m is the car's width and

$$h_1 := 1.1\, B + 0.25, \quad h_2 := 3.5, \quad h_3 := 1.2\, B + 3.75, \quad h_4 := 1.3\, B + 0.25.$$

	Value	Unit	Description
m	$1.239 \cdot 10^3$	kg	Mass of the car
g	9.81	$\frac{\text{m}}{\text{s}^2}$	Gravity constant
l_f	1.19016	m	Front wheel distance to c.o.g.
l_r	1.37484	m	Rear wheel distance to c.o.g.
R	0.302	m	Wheel radius
I_zz	$1.752 \cdot 10^3$	kg m^2	Moment of inertia
c_w	0.3	–	Air drag coefficient
ρ	1.249512	$\frac{\text{kg}}{\text{m}^3}$	Air density
A	1.4378946874	m^2	Effective flow surface
i_g^1	3.09	–	Gear 1 transmission ratio
i_g^2	2.002	–	Gear 2 transmission ratio
i_g^3	1.33	–	Gear 3 transmission ratio
i_g^4	1.0	–	Gear 4 transmission ratio
i_g^5	0.805	–	Gear 5 transmission ratio
i_t	3.91	–	Engine torque transmission
B_f	$1.096 \cdot 10^1$	–	Pacejka coeff. (stiffness)
B_r	$1.267 \cdot 10^1$	–	
$C_\text{f,r}$	1.3	–	Pacejka coefficients (shape)
D_f	$4.5604 \cdot 10^3$	–	Pacejka coefficients (peak)
D_r	$3.94781 \cdot 10^3$	–	
$E_\text{f,r}$	-0.5	–	Pacejka coefficients (curv.)

Table 7.5: Parameters of the car model.

7.9.2 Results

In [105, 106, 147] numerical results for the benchmark problem have been deduced. Table 7.6 gives the optimal gear choice and the resulting objective function value (the end time) for different numbers N of control discretization intervals, which were also used for a discretization of the path constraints.

The outer convexification approach (Section 2.6.5) led to a tremendous speed-up compared to the published reference benchmark solution for a fixed control discretization grid by several orders of magnitude as shown in Table 7.7. In [147] one can also find an explanation why a bang-bang solution for the relaxed and convexified gear choices has to be optimal.

N	$\mu = 1$	$\mu = 2$	$\mu = 3$	$\mu = 4$	$\mu = 5$	t_f
10	0.0	0.435956	2.733326	–	–	6.764174
20	0.0	0.435903	2.657446	6.467723	–	6.772046
40	0.0	0.436108	2.586225	6.684504	–	6.782052
80	0.0	0.435796	2.748930	6.658175	–	6.787284

Table 7.6: Gear choice depending on discretization in time N. Times when gear becomes active.

	Inner convexification and Branch&Bound		**Outer convexification and MS MINTOC**	
m	t_f	CPU Time	t_f	CPU Time
20	6.779751	00:23:52	6.779035	00:00:24
40	6.786781	232:25:31	6.786730	00:00:46
80	–	–	6.789513	00:04:19

Table 7.7: Comparison of computational times for a Branch&Bound approach on a Pentium III machine with 750 MHz, [105] (left), and for MS MINTOC on an AMD Athlon XP 3000+ with 2.166 GHz, [147] (right). m denotes the number of control discretization intervals, t_f is the optimal objective function value. The path constraints are discretized on the same grid, hence the non-monotonicity of t_f in m. CPU times are given in hh:min:sec. Note that the results based on MS MINTOC were obtained on a computer that is approximately 4 times faster than the Pentium III machine, which would normally make a comparison of computation times highly suspect. However, here the computation times vary by at least 2 orders of magnitude with a difference growing in m, which is clearly a significant improvement even with the difference in machines.

7.10 Elliptic track testdrive

This control problem is very similar to the one in Section 7.9. However, instead of a simple lane change maneuver the time-optimal driving on an elliptic track with periodic boundary conditions is considered, [213].

7.10.1 Model and optimal control problem

With the notation of Section 7.9 the MIOCP reads

$$\begin{aligned}
\min_{t_\mathrm{f},x(\cdot),u(\cdot)} \quad & t_\mathrm{f} \\
\text{s.t.} \quad & (7.46b-7.46h),(7.46\mathrm{j}),(7.46\mathrm{k}), \\
& (c_\mathrm{x},c_\mathrm{y}) \in \mathscr{X}, \\
& x(t_0) = x(t_\mathrm{f}) - (0,0,0,0,0,2\pi,0)^T, \\
& c_\mathrm{y}(t_0) = 0, \\
& 0 \leq r^{\mathrm{eng}}(v,\mu),
\end{aligned} \tag{7.49a}$$

for $t \in [t_0, t_\mathrm{f}]$ almost everywhere.

The set $\mathscr{X}$ describes an elliptic track with axes of $a = 170$ meters and $b = 80$ meters respectively, centered in the origin. The track's width is $W = 7.5$ meters, five times the car's width $B = 1.5$ meters,

$$\mathscr{X} = \left\{ \left[(a+r)\cos\eta, (b+r)\sin\eta\right] \middle| \, r \in [-W/2, W/2] \subset \mathbb{R} \right\},$$

with $\eta = \arctan \frac{c_\mathrm{y}}{c_\mathrm{x}}$. Note that the special case $c_\mathrm{x} = 0$ leading to $\eta = \pm\frac{\pi}{2}$ requires separate handling.

The model in Section 7.9 has a shortcoming, as switching to a low gear is possible also at high velocities, although this would lead to an unphysically high engine speed. Therefore we extend it by additional constraints on the car's engine speed

$$800 =: n_{\mathrm{eng}}^{\mathrm{MIN}} \leq n_{\mathrm{eng}} \leq n_{\mathrm{eng}}^{\mathrm{MAX}} := 8000, \tag{7.50}$$

in the form of equivalent velocity constraints

$$\frac{\pi n_{\mathrm{eng}}^{\mathrm{MIN}} R}{30 i_\mathrm{t} i_\mathrm{g}^{\mu}} \leq \quad v \quad \leq \frac{\pi n_{\mathrm{eng}}^{\mathrm{MAX}} R}{30 i_\mathrm{t} i_\mathrm{g}^{\mu}} \tag{7.51}$$

for all $t \in [0, t_\mathrm{f}]$ and the active gear μ. We write this as $r^{\mathrm{eng}}(v,\mu) \geq 0$.

7.10.2 Results

Parts of the optimal trajectory from [213] are shown in Figures 7.11 and 7.12. The order of gears is $(2,3,4,3,2,1,2,3,4,3,2,1,2)$. The gear switches take place after 1.87, 5.96, 10.11, 11.59, 12.21, 12.88, 15.82, 19.84, 23.99, 24.96, 26.10, and 26.76 seconds, respectively. The final time is $t_\mathrm{f} = 27.7372$ s.

As can be seen in Fig. 7.12, the car uses the track width to its full extent, leading to active path constraints. As was expected, the optimal gear increases in an acceleration phase. When the

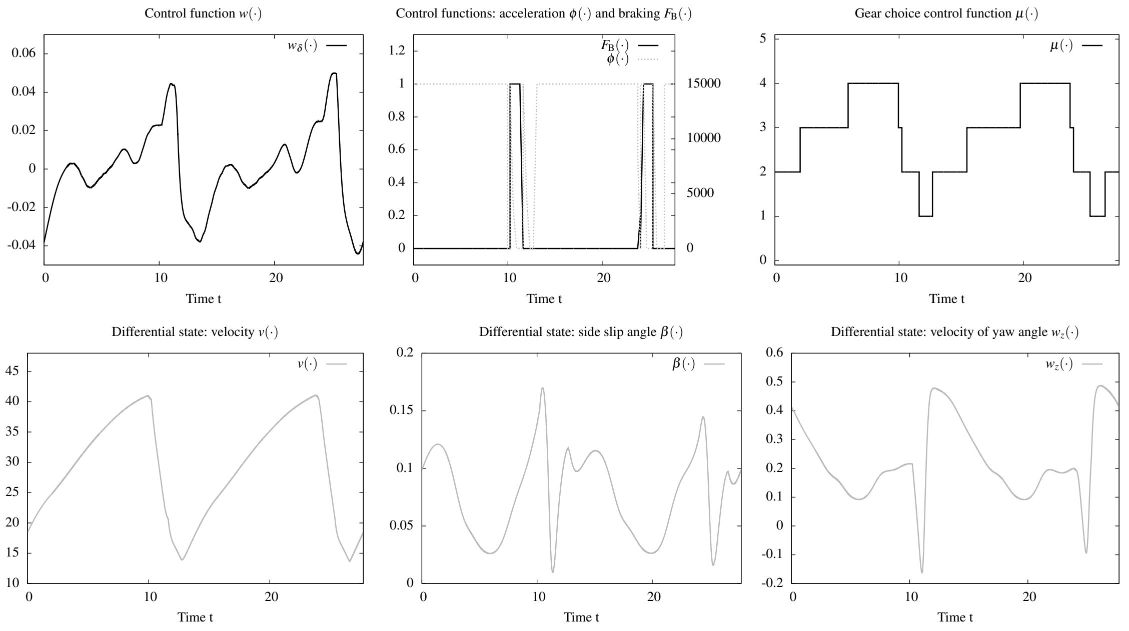


Figure 7.11: The control functions (top row), and selected differential states of the optimal solution: directional velocity, side slip angle β, and velocity of yaw angle w_z plotted over time.

velocity has to be reduced, a combination of braking, no acceleration, and engine brake is used. The result depends on the engine speed constraint $r^{\text{eng}}(v, \mu)$ that becomes active in the braking phase. If the constraint is omitted, the optimal solution switches directly from the fourth gear into the first one to maximize the effect of the engine brake. For $n_{\text{eng}}^{\text{MAX}} = 15000$ braking occurs in the gear order $4, 2, 1$.

Although this was left as a degree of freedom, the optimizer yields a symmetric solution with respect to the upper and lower parts of the track for all scenarios we considered.

7.10.3 Variants

By a more flexible use of Bezier patches more general track constraints can be specified, e.g., of formula 1 race courses.

7.11 Simulated moving bed

We consider a simplified model of a Simulated Moving Bed (SMB) chromatographic separation process that contains time–dependent discrete decisions. SMB processes have been gaining increased attention lately, see [84, 137, 211] for further references. The related optimization problems are challenging from a mathematical point of view, as they combine periodic nonlinear optimal control problems in partial differential equations (PDE) with time–dependent discrete decisions.

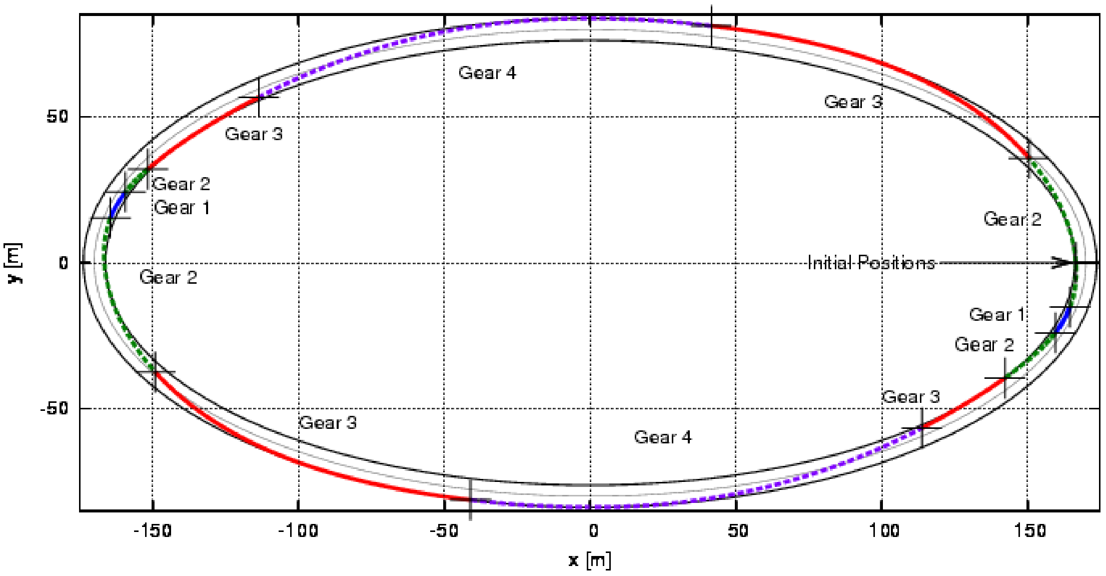


Figure 7.12: Elliptic race track seen from above with optimal position and gear choices of the car. Note the exploitation of the slip (sliding) to change the car's orientation as fast as possible, when in first gear. The gear order changes when a different maximum engine speed is imposed.

7.11.1 Model and optimal control problem

SMB chromatography finds various industrial applications such as sugar, food, petrochemical and pharmaceutical industries. A SMB unit consists of multiple columns filled with solid absorbent. The columns are connected in a continuous cycle. There are two inlet streams, *desorbent* (De) and *feed* (Fe), and two outlet streams, *raffinate* (Ra) and *extract* (Ex). The continuous counter-current operation is simulated by switching the four streams periodically in the direction of the liquid flow in the columns, thereby leading to better separation. This is visualized in Figure 7.13.

Due to this discrete switching of columns, SMB processes reach a cyclic or periodic steady state, i.e., the concentration profiles at the end of a period are equal to those at the beginning shifted by one column ahead in direction of the fluid flow. A number of different operating schemes have been proposed to further improve the performance of SMB.

The considered SMB unit consists of $N_{\text{col}} = 6$ columns. The flow rate through column i is denoted by Q_{i}, $i \in I := \{1, \ldots, N_{\text{col}}\}$. The raffinate, desorbent, extract and feed flow rates are denoted by Q_{Ra}, Q_{De}, Q_{Ex} and Q_{Fe}, respectively. The (possibly) time–dependent value $w_{i\alpha}(t) \in \{0,1\}$ denotes if the port of flow $\alpha \in \{\text{Ra}, \text{De}, \text{Ex}, \text{Fe}\}$ is positioned at column $i \in I$. As in many

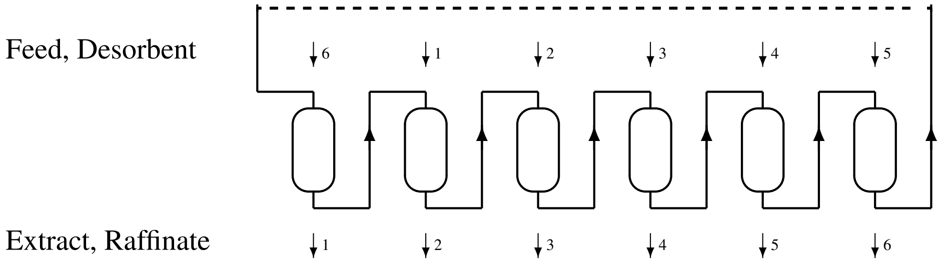


Figure 7.13: Scheme of SMB process with 6 columns.

practical realizations of SMB processes only one pump per flow is available and the ports are switched by a 0–1 valve, we obtain the additional *special ordered set type one* restriction

$$\sum_{i\in I} w_{i\alpha}(t) \quad = \quad 1, \ \ \forall \ t \in [0,T], \ \alpha \in \{\text{Ra},\text{De},\text{Ex},\text{Fe}\}. \tag{7.52}$$

The flow rates Q_1, Q_{De}, Q_{Ex} and Q_{Fe} enter as control functions $u(\cdot)$ resp. time–invariant parameters p into the optimization problem, depending on the operating scheme to be optimized. The remaining flow rates are derived by mass balance as

$$Q_{\text{Ra}} \quad = \quad Q_{\text{De}} - Q_{\text{Ex}} + Q_{\text{Fe}} \tag{7.53}$$

$$Q_i \quad = \quad Q_{i-1} - \sum_{\alpha\in\{\text{Ra,Ex}\}} w_{\text{i}\alpha} Q_\alpha + \sum_{\alpha\in\{\text{De,Fe}\}} w_{\text{i}\alpha} Q_\alpha \tag{7.54}$$

for $i = 2, \dots N_{\text{col}}$. The feed contains two components A and B dissolved in desorbent, with concentrations $c_{\text{Fe}}^{\text{A}} = c_{\text{Fe}}^{\text{B}} = 0.1$. The concentrations of A and B in desorbent are $c_{\text{De}}^{\text{A}} = c_{\text{De}}^{\text{B}} = 0$.

A simplified equilibrium model is described in Diehl and Walther [75]. It can be derived from an equilibrium assumption between solid and liquid phases along with a simple spatial discretization. The mass balance in the liquid phase for $K = \text{A},\text{B}$ is given by:

$$\varepsilon_b \frac{\partial c_i^K(x,t)}{\partial t} + (1-\varepsilon_b)\frac{\partial q_i^K(x,t)}{\partial t} + u_i(t)\frac{\partial c_i^K(x,t)}{\partial x} = 0 \tag{7.55}$$

with equilibrium between the liquid and solid phases given by a linear isotherm:

$$q_i^K(x,t) = C_K c_i^K(x,t). \tag{7.56}$$

Here ε_b is the void fraction, $c_i^K(x,t)$ is the concentration in the liquid phase of component K in column i, q_i^K is the concentration in the solid phase. Also, i is the column index and N_{Column} is the number of columns. We can combine (7.55) and (7.56) and rewrite the model as:

$$\frac{\partial c_i^K(x,t)}{\partial t} = -(u_i(t)/\bar{K}_K)\frac{\partial c_i^K(x,t)}{\partial x} \tag{7.57}$$

where $\bar{K}_K = \varepsilon_b + (1-\varepsilon_b)C_K$. Dividing the column into N_{FEX} compartments and applying a simple backward difference with $\Delta x = L/N_{FEX}$ leads to:

$$\frac{dc_{i,j}^K}{dt} = \frac{u_i(t)N_{FEX}}{\bar{K}_K L}[c_{i,j-1}^K(t) - c_{i,j}^K(t)] = k^K[c_{i,j-1}^K(t) - c_{i,j}^K(t)] \tag{7.58}$$

for $j = 1,\dots,N_{FEX}$, with $k^A = 2N_{FEX}$, $k^B = N_{FEX}$, and $c_{i,j}^K(t)$ is a discretization of $c_i^K(j\Delta x, t)$ for $j = 0,\dots,N_{FEX}$.

This simplified model for the dynamics in each column considers axial convection and axial mixing introduced by dividing the respective column into N_{dis} perfectly mixed compartments. Although this simple discretization does not consider all effects present in the advection–diffusion equation for the time and space dependent concentrations, the qualitative behavior of the concentration profiles moving at different velocities through the respective columns is sufficiently well represented. We assume that the compartment concentrations are constant. We denote the concentrations of A and B in the compartment with index i by c_i^{A}, c_i^{B} and leave away the time dependency. For the first compartment $j = (i-1)N_{\text{dis}} + 1$ of column $i \in I$ we have by mass transfer for $K = \text{A},\text{B}$

$$\frac{\dot{c}_j^K}{k^K} = Q_{i^-}c_{j^-}^K - Q_i c_j^K - \sum_{\alpha\in\{\text{Ra},\text{Ex}\}} w_{i\alpha}Q_\alpha c_{j^-}^K + \sum_{\alpha\in\{\text{De},\text{Fe}\}} w_{i\alpha}Q_\alpha C_\alpha^K \tag{7.59}$$

where i^- is the preceding column, $i^- = N_{\text{col}}$ if $i = 1$, $i^- = i-1$, else and equivalently $j^- = N$ if $j = 1$, $j^- = j-1$, else. k^K denotes the axial convection in the column, $k^{\text{A}} = 2N_{\text{dis}}$ and $k^{\text{B}} = N_{\text{dis}}$. Component A is less adsorbed, thus travels faster and is prevailing in the raffinate, while B travels slower and is prevailing in the extract. For interior compartments j in column i we have

$$\frac{\dot{c}_j^K}{k^K} = Q_{i^-}c_{j^-}^K - Q_i c_j^K. \tag{7.60}$$

The compositions of extract and raffinate, $\alpha \in \{\text{Ex},\text{Ra}\}$, are given by

$$\dot{M}_\alpha^K = Q_\alpha \sum_{i\in I} w_{i\alpha} c_{j(i)}^K \tag{7.61}$$

with $j(i)$ the last compartment of column i^-. The feed consumption is

$$\dot{M}_{\text{Fe}} = Q_{\text{Fe}}. \tag{7.62}$$

These are altogether $2N+5$ differential equations for the differential states $x = (x_A, x_B, x_M)$ with $x_A = (c_0^{\text{A}},\dots,c_N^{\text{A}})$, $x_B = (c_0^{\text{B}},\dots,c_N^{\text{B}})$, and finally $x_M = (M_{\text{Ex}}^{\text{A}}, M_{\text{Ex}}^{\text{B}}, M_{\text{Ra}}^{\text{A}}, M_{\text{Ra}}^{\text{B}}, M_{\text{Fe}})$. They can be summarized as

$$\dot{x}(t) = f(x(t),u(t),w(t),p). \tag{7.63}$$

We define a linear operator $P : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ that shifts the concentration profiles by one column and sets the auxiliary states to zero, i.e.,

$$\begin{aligned} x \mapsto Px &:= (P_A x_A, P_B x_B, P_M x_M) \quad \text{with} \\ P_A x_A &:= (c^{\text{A}}_{N_{\text{dis}}+1}, \dots, c^{\text{A}}_N, c^{\text{A}}_1, \dots, c^{\text{A}}_{N_{\text{dis}}}), \\ P_B x_B &:= (c^{\text{B}}_{N_{\text{dis}}+1}, \dots, c^{\text{B}}_N, c^{\text{B}}_1, \dots, c^{\text{B}}_{N_{\text{dis}}}), \\ P_M x_M &:= (0,0,0,0,0). \end{aligned}$$

Then we can impose periodicity after the unknown cycle duration T by requiring $x(0) = Px(T)$. The purity of component A in the raffinate at the end of the cycle must be higher than $p_{\text{Ra}} = 0.95$ and the purity of B in the extract must be higher than $p_{\text{Ex}} = 0.95$, i.e., we impose the terminal purity conditions

$$M^{\text{A}}_{\text{Ex}}(T) \quad \leq \quad \frac{1-p_{\text{Ex}}}{p_{\text{Ex}}} M^{\text{B}}_{\text{Ex}}(T), \tag{7.64}$$

$$M^{\text{B}}_{\text{Ra}}(T) \quad \leq \quad \frac{1-p_{\text{Ra}}}{p_{\text{Ra}}} M^{\text{A}}_{\text{Ra}}(T). \tag{7.65}$$

We impose lower and upper bounds on all external and internal flow rates,

$$0 \leq Q_{\text{Ra}}, Q_{\text{De}}, Q_{\text{Ex}}, Q_{\text{Fe}}, Q_1, Q_2, Q_3, Q_4, Q_5, Q_6 \leq Q_{\max} = 2. \tag{7.66}$$

To avoid draining inflow into outflow streams without going through a column,

$$Q_i - w_{i\text{De}} Q_{\text{De}} - w_{i\text{Fe}} Q_{\text{Fe}} \quad >= \quad 0 \tag{7.67}$$

has to hold for all $i \in I$. The objective is to maximize the feed throughput $M_{\text{Fe}}(T)/T$. Summarizing, we obtain the following MIOCP

$$\begin{array}{lll} \max\limits_{x(\cdot),u(\cdot),w(\cdot),p,T} & M_{\text{Fe}}(T)/T & \\ \text{s.t.} & \dot{x}(t) & = f(x(t),u(t),w(t),p), \\ & x(0) & = Px(T), \\ & (7.64-7.67), & \\ & \sum_{i \in I} w_{i\alpha}(t) & = 1, \quad \forall\ t \in [0,T], \\ & w(t) & \in \{0,1\}^{4N_{\text{col}}}, \quad \forall\ t \in [0,T]. \end{array} \tag{7.68}$$

with $\alpha \in \{\text{Ra}, \text{De}, \text{Ex}, \text{Fe}\}$.

Process	Time	1	2	3	4	5	6
SMB fix	0.00 – 0.63	De	Ex		Fe		Ra
SMB relaxed	0.00 – 0.50	De,Ex	Ex		Fe		Ra
PowerFeed	0.00 – 0.56	De	Ex		Fe		Ra
VARICOL	0.00 – 0.18	De	Ex		Fe		Ra
	0.18 – 0.36	De		Ex	Fe		Ra
	0.36 – 0.46	De,Ra		Ex	Fe		
	0.46 – 0.53	De,Ra		Ex		Fe	
Superstruct	0.00 – 0.10	Ex					De
	0.10 – 0.18		De,Ex				
	0.18 – 0.24	De					Ra
	0.24 – 0.49	De		Ex	Fe		Ra
	0.49 – 0.49		De,Ex				

Table 7.8: Fixed or optimized port assignment $w_{i\alpha}$ and switching times of the process strategies.

7.11.2 Results

We optimized different operation schemes that fit into the general problem formulation (7.68): **SMB fix.** The $w_{i\alpha}$ are fixed as shown in Table 7.8. The flow rates $Q_{\cdot}$ are constant in time, i.e., they enter as optimization parameters p into (7.68). Optimal solution $\Phi = 0.7345$. **SMB relaxed.** As above. But the $w_{i\alpha}$ are free for optimization and relaxed to $w_{i\alpha} \in [0,1]$, allowing for a ”splitting” of the ports. $\Phi = 0.8747$. In **PowerFeed** the flow rates are modulated during one period, i.e., the $Q_{\cdot}$ enter as control functions $u(\cdot)$ into (7.68). $\Phi = 0.8452$. **VARICOL.** The ports switch asynchronically, but in a given order. The switching times are subject to optimization. $\Phi = 0.9308$. **Superstruct.** This scheme is the most general and allows for arbitrary switching of the ports. The flow rates enter as continuous control functions, but are found to be bang–bang by the optimizer (i.e., whenever the port is given in Table 7.8, the respective flow rate is at its upper bound). $\Phi = 1.0154$.

7.12 Discretizations to MINLPs

In this section we provide AMPL code for two discretized variants of the control problems from Sections 7.3 and 7.4 as an illustration of the discretization of MIOCPs to MINLPs. More examples will be collected in the future on http://mintoc.de.

7.12.1 General AMPL code

In Listings 7.1 and 7.2 we provide two AMPL input files that can be included for MIOCPs with one binary control $w(t)$.

Listing 7.1: Generic settings AMPL model file to be included

```
param T      > 0;     # End time
param nt     > 0;     # Number of discretization points in time
param nu     > 0;     # Number of control discretization points
param nx     > 0;     # Dimension of differential state vector
param ntperu > 0;     # nt / nu
set I:= 0..nt;
set U:= 0..nu-1;
param uidx {I}; param fix_w; param fix_dt;

var w {U} >= 0, <= 1 binary;     # control function
var dt {U} >= 0, <= T;           # stage length vector
```

Listing 7.2: Generic settings AMPL data file to be included

```
if ( fix_w > 0 )  then { for {i in U} { fix w[i]; } }
if ( fix_dt > 0 ) then { for {i in U} { fix dt[i]; } }

# Set indices of controls corresponding to time points
for {i in 0..nu-1} {
  for {j in 0..ntperu-1} { let uidx[i*ntperu+j] := i; }
}
let uidx[nt] := nu-1;
```

7.12.2 Lotka Volterra Fishing Problem

The AMPL code in Listings 7.3 and 7.4 shows a discretization of the problem(7.25) with piecewise constant controls on an equidistant grid of length T/n_u and with an implicit Euler method. Note that for other MIOCPs, especially for unstable ones as in Section 7.7, more advanced integration methods such as Backward Differentiation Formulae need to be applied.

Listing 7.3: AMPL model file for Lotka Volterra Fishing Problem

```
var x {I, 1..nx} >= 0;
param c1 > 0; param c2 > 0; param ref1 > 0; param ref2 > 0;

minimize Deviation:
 0.5 * (dt[0]/ntperu) * ( (x[0,1]-ref1)^2 + (x[0,2]-ref2)^2 )
 + 0.5 * (dt[nu-1]/ntperu) * ((x[nt,1]-ref1)^2 + (x[nt,2]-ref2)^2)
 + sum {i in I diff {0,nt} } ( (dt[uidx[i]]/ntperu) *
   ( (x[i,1] - ref1)^2 + (x[i,2] - ref2)^2 ) ) ;

subj to ODE_DISC_1 {i in I diff {0}}:
 x[i,1] = x[i-1,1] + (dt[uidx[i]]/ntperu) *
  ( x[i,1] - x[i,1]*x[i,2] - x[i,1]*c1*w[uidx[i]] );

subj to ODE_DISC_2 {i in I diff {0}}:
 x[i,2] = x[i-1,2] + (dt[uidx[i]]/ntperu) *
```

```
  ( - x[i,2] + x[i,1]*x[i,2] - x[i,2]*c2*w[uidx[i]] );

subj to overall_stage_length:
 sum {i in U} dt[i] = T;
```

Listing 7.4: AMPL dat file for Lotka Volterra Fishing Problem

```
# Algorithmic parameters
param ntperu := 100; param nu := 100;  param nt := 10000;
param nx := 2;       param fix_w := 0; param fix_dt := 1;

# Problem parameters
param T := 12.0;     param c1 := 0.4;  param c2 := 0.2;
param ref1 := 1.0;   param ref2 := 1.0;

# Initial values differential states
let x[0,1] := 0.5; let x[0,2] := 0.7;
fix x[0,1]; fix x[0,2];

# Initial values control
let {i in U} w[i] := 0.0;
for {i in 0..(nu-1) / 2} { let w[i*2] := 1.0; }
let {i in U} dt[i] := T / nu;
```

Note that the constraint overall_stage_length is only necessary, when the value for fix_dt is zero, a switching time optimization.
The solution calculated by *Bonmin* (subversion revision number 1453, default settings, 3 GHz, Linux 2.6.28-13-generic, with ASL(20081205)) has an objective function value of $\Phi = 1.34434$, while the optimum of the relaxation is $\Phi = 1.3423368$. *Bonmin* needs 35301 iterations and 2741 nodes (4899.97 seconds). The intervals on the equidistant grid on which $w(t) = 1$ holds, counting from 0 to 99, are 20–32, 34, 36, 38, 40, 44, 53.

7.12.3 F-8 flight control

The main difficulty in calculating a time-optimal solution for the problem in Section 7.3 is the determination of the correct switching structure and of the switching points. If we want to formulate a MINLP, we have to slightly modify this problem. Our aim is not a minimization of the overall time, but now we want to get as close as possible to the origin $(0,0,0)$ in a prespecified time $t_\mathrm{f} = 3.78086$ on *an equidistant time grid*. As this time grid is not a superset of the one used for the time-optimal solution in Section 7.3, one can not expect to reach the target state exactly. Listings 7.5 and 7.6 show the AMPL code.

Listing 7.5: AMPL model file for F-8 Flight Control Problem

```
var x {I, 1..nx};
param xi   > 0;

minimize Deviation: sum {i in 1..3} x[nt,i]*x[nt,i];

subj to ODE_DISC_1 {i in I diff {0}}:
```

```
x[i,1] = x[i-1,1] + (dt[uidx[i]]/ntperu) * (
 - 0.877*x[i,1] + x[i,3] - 0.088*x[i,1]*x[i,3] + 0.47*x[i,1]*x[i,1]
 - 0.019*x[i,2]*x[i,2]
 - x[i,1]*x[i,1]*x[i,3] + 3.846*x[i,1]*x[i,1]*x[i,1]
 + 0.215*xi - 0.28*x[i,1]*x[i,1]*xi + 0.47*x[i,1]*xi^2 - 0.63*xi^2
 - 2*w[uidx[i]] * (0.215*xi - 0.28*x[i,1]*x[i,1]*xi - 0.63*xi^3));

subj to ODE_DISC_2 {i in I diff {0}}:
   x[i,2] = x[i-1,2] + (dt[uidx[i]]/ntperu) * x[i,3];

subj to ODE_DISC_3 {i in I diff {0}}:
x[i,3] = x[i-1,3] + (dt[uidx[i]]/ntperu) * (
 - 4.208*x[i,1] - 0.396*x[i,3] - 0.47*x[i,1]*x[i,1]
 - 3.564*x[i,1]*x[i,1]*x[i,1]
 + 20.967*xi - 6.265*x[i,1]*x[i,1]*xi + 46*x[i,1]*xi^2 - 61.4*xi^3
 - 2*w[uidx[i]]*(20.967*xi - 6.265*x[i,1]*x[i,1]*xi - 61.4*xi^3));
```

Listing 7.6: AMPL dat file for F-8 Flight Control Problem

```
# Parameters
param ntperu := 500;  param nu := 60;    param nt := 30000;
param nx := 3;        param fix_w := 0;  param fix_dt := 1;
param xi := 0.05236;  param T := 8;

# Initial values differential states
let x[0,1] := 0.4655;
let x[0,2] := 0.0;
let x[0,3] := 0.0;
for {i in 1..3} { fix x[0,i]; }

# Initial values control
let {i in U} w[i] := 0.0;
for {i in 0..(nu-1) / 2} { let w[i*2] := 1.0; }
let {i in U} dt[i] := 3.78086 / nu;
```

The solution calculated by *Bonmin* has an objective function value of $\Phi = 0.022108$, while the optimum of the relaxation calculated with *Ipopt* is $\Phi = 0.021788$. *Bonmin* needs 85702 iterations and 7031 nodes (64282 seconds). The intervals on the equidistant grid on which $w(t) = 1$ holds, counting from 0 to 59, are 0, 1, 31, 32, 42, 52, and 54. Both solutions are shown in Figure 7.14.

7.13 Summary

We presented a collection of mixed-integer optimal control problem descriptions. These descriptions comprise details on the model and a specific instance of control objective, constraints, parameters, and initial values that yield well-posed optimization problems that allow for reproducibility and comparison of solutions. Furthermore, specific discretizations in time and space are applied with the intention to supply benchmark problems also for MINLP algorithm developers. The descriptions are complemented by references and best known solutions. All problem formulations are available for download at http://mintoc.de in a suited format, such as optimica or AMPL.

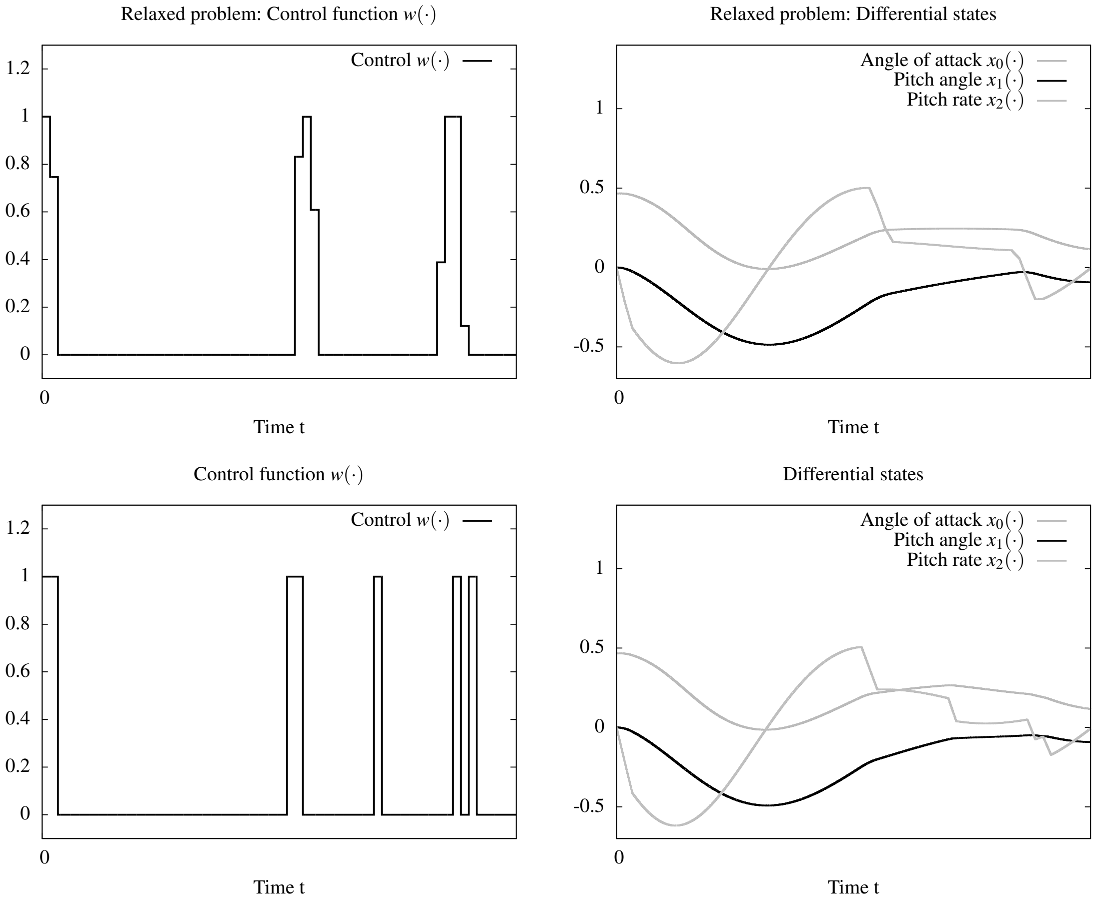


Figure 7.14: Trajectories for the discretized F-8 flight control problem. Top row: relaxed problem, bottom row: integer feasible solution. Left: control. Right: corresponding differential states. Compare also Figure 7.1 for solutions for the related problems with fixed terminal values on a free control discretization grid.

The author hopes to achieve at least two things. First, to provide a benchmark library that is of use for both MIOC and MINLP algorithm developers. Second, to motivate others to contribute to the extension of this library. For example, challenging and well-posed instances from water or gas networks [55, 174], traffic flow [122, 96], supply chain networks [114], submarine control [199], distributed autonomous systems [2], and chemical engineering [138, 227] would be highly interesting for the community.

8 Optimization as an Analysis Tool for Human Complex Problem Solving

The contents of this chapter are based on the paper

[207] S. Sager, C.M. Barth, H. Diedam, M. Engelhart, J. Funke. Optimization as an Analysis Tool for Human Complex Problem Solving. *SIAM Journal on Optimization*, accepted.

Chapter Summary. We present a problem class of mixed-integer nonlinear programs (MINLPs) with nonconvex continuous relaxations which stem from economic test scenarios that are used in the analysis of human complex problem solving. In a round based scenario participants hold an executive function. A posteriori a performance indicator is calculated and correlated to personal measures such as intelligence, working memory, or emotion regulation.

The MINLPs can be interpreted as time-discrete optimal control problems with integer-valued decisions, hence a special case of MIOCPs in which the control discretization grid is fixed.

We investigate altogether 2088 optimization problems that differ in size and initial conditions, based on real world experimental data from 12 rounds of 174 participants. The goals are twofold: first, from the optimal solutions we gain additional insight into a complex system, which facilitates the analysis of a participant's performance in the test. Second, we propose a methodology to automatize this process by providing a new criterion based on the solution of a series of optimization problems.

By providing a mathematical optimization model and this methodology, we disprove the assumption that the "fruit fly of complex problem solving", the *Tailorshop* scenario that has been used for dozens of published studies, is not mathematically accessible — although it turns out to be extremely challenging even for advanced state-of-the-art global optimization algorithms and we were not able to solve all instances to global optimality in reasonable time in this study.

By providing a detailed mathematical description and the computational tool *Tobago* [210] for an optimization-based analysis we hope to foster further interdisciplinary research between psychologists and applied mathematicians.

The publicly available computational tool *Tobago* [210] can be used to automatically generate problem instances of various complexity, contains interfaces to *AMPL* and *GAMS* and is hence ideally suited as a testbed for different kinds of algorithms and solvers. Computational practice is reported with respect to the influence of integer variables, problem dimension, and local versus global optimization with different optimization codes. The submission not merely de-

scribes an application, but opens up a whole new *application area* for optimization, which to our knowledge is yet completely unexplored.

8.1 Introduction

The methodology *optimization* has a long record of successful improvements in many technological and scientific areas, being used for tasks such as design, scheduling, business control rules, process control, and the like. More recently, optimization has also been successfully applied in the context of inverse problems, e.g., for the choice and calibration of mathematical models, or as a modeling paradigm for biological systems. In this work we propose to use numerical optimization as an analysis tool for the understanding of human problem solving, which to our knowledge has not yet received much attention.

Complex problem solving is defined as a *high-order cognitive process*. The complexity may result from one or several different characteristics, such as a coupling of subsystems, nonlinearities, dynamic changes, intransparency, or others [78]. The main intention of the research field *complex problem solving* of human beings is the desire to understand how certain *variables* influence a solution process. In general, *personal and situational variables* are differentiated. The most typical and frequently analyzed personal variable is *intelligence*. It is an ongoing debate how intelligence influences complex problem solving [251]. Other interesting personal variables are *working memory* [200], *amount of knowledge* [150], and *emotion regulation* [187]. Situational variables like the impact of *goal specificity and observation* [186], *feedback* [50], and *time constraints* [112] attracted less attention.

Psychologists have been working in the research fields of problem solving for approximately 80 years. One of the groundbreaking works by Ewert and Lambert in 1932 [86] was based on the disk problem, more commonly known as the *Tower of Hanoi*. Since the 1970s and 1980s also computer-based test scenarios are in use, e.g., LEARN [119], Moro [230], FSYS 2.0 [244], and *Tailorshop*, which is the basis for this study. The *Tailorshop* is sometimes referred to as the "Drosophila" for problem solving researchers [101] and thus a prominent example for a computer-based test scenario. All mentioned scenarios try to reflect the characteristics of real-life problems by simulating a microworld [113].

The overall idea, compared to early works in problem solving, is still the same: one evaluates the performance of a participant by calculating an *indicator function* and either correlates it to personal attributes, such as the intelligence quotient [130], or analyzes the influence of different experimental conditions for groups of participants [21]. The main difference is that for the early test scenarios the correct solution is known at every stage. For more complex scenarios the performance evaluation is not so straightforward.

We address the question how to get a reliable performance indicator for the *Tailorshop* scenario. The *Tailorshop* has been used in a large number of studies, e.g., [198, 151, 149, 179, 21, 22]. Also comprehensive reviews on studies and results in connection with the *Tailorshop* have been published, see [95, 99, 100, 102, 101], in which also more information on the psychological background can be found.

In *Tailorshop*, participants have to take economic decisions to maximize the overall balance of a small company, specialized in the production and sales of shirts. To measure performance within the *Tailorshop* scenario different indicator functions have been proposed in the literature. To use a comparison of accumulated capital at the final month 12 between all participants was proposed in [126]. This criterion seems natural, as this is what the participants are requested to maximize. However, it cannot yield insight into the temporal process and is not objective in the sense that the performance depends on what other participants achieved. Analyzing the temporal evolution of state variables has also been proposed. In [197, 231] the evolution of profit, equivalent to the evolution of capital after interest x_k^{CA}, was proposed. In [98, 22] the evolution of the overall worth of the tailorshop x_k^{OB} was used. An obvious drawback of comparing the results of several rounds with one another is that the main goal of the participant is to maximize the value at the end of the test, not necessarily in between. Thinking about the analogy of maximizing the amplitude of a pendulum with a hair dryer, in certain scenarios "going back" to gain momentum is obviously better than pushing it all the time in the desired direction. The same is true for the *Tailorshop* scenario. It may be better to invest into infrastructure at the beginning (which is actually decreasing the overall capital as infrastructure looses value over time) to have a higher pay-off towards the last rounds of the test.Hence it might happen that decisions are analyzed to be bad, while they are actually good ones and vice versa. To overcome this problem, we propose to compare the decisions to mathematically optimal solutions. For a recent review on *Tailorshop* success criteria, see [70].

Because all previously used indicators have unknown reliability and validity, we propose to compare the decisions to mathematically optimal solutions. Hussy [129, p. 62] writes in 1985[1]

> *"Only when it will be possible, e.g., by means of mathematical optimization methods, to determine the objectively optimal solution process to compare the process chosen by the proband with it, will this severe problem be overcome."*

The availability of an objective performance indicator is an obstacle for analysis and it has often been argued that inconsistent findings are due to the fact that

> *"... it is impossible to derive valid indicators of problem solving performance for tasks that are not formally tractable and thus do not possess a mathematically optimal solution. Indeed, when different dependent measures are used in studies using the same scenario (i.e., Tailorshop [98, 231, 197]), then the conclusions frequently differ."*

[1] author's translation from the German original: "Erst wenn es gelingt, z.B. durch mathematische Optimierungsverfahren, den objektiv besten Lösungsweg zu bestimmen, um daran den tatsächlich gewählten Lösungsweg der Pbn messen zu können, wird dieses ernste Problem [die objektive Bestimmung der Problemlösegüte] aus dem Weg zu räumen sein."

as stated by Wenke and Frensch [245, p.95]. Based on a mathematical model of the *Tailorshop*, an optimization is performed for every round of the participant's data, starting with exactly the same conditions as the participant. By comparing these optimal values that indicate *How much is still possible* if all future decisions were made perfectly, an analysis of at what rounds potential has been lost by decisions can be obtained. Based on optimization theory, even further insight into what decisions were decisive for bad or good performance can be obtained by analyzing Lagrange multipliers.

To our knowledge, numerical optimization methods have only scarcely been used for the analysis of participants' decisions in complex environments like *Tailorshop*. Cognitive psychologists and economists have been using simulation methods for finding optimal solutions for simple tasks within strongly constrained environments. Also in the context of *experimental economics* studies have been performed, however to our knowledge not with explicit mathematical representations of the scenarios, including nonlinearities and integer variables. The general approach to compare performance to optimal solutions has been discussed by [152]. However, the authors do not provide a mathematical model for their test scenario *EPEX*. Hence, they need to use the software as a black box for brute-force simulation or derivative free strategies, such as Nelder-Mead [180] or genetic algorithms. Such strategies result in significantly higher computational runtimes, give less insight, and have poor theoretical convergence properties. Our approach formulates the simulation task as equality constraints of the optimization problem and allows thus to apply modern optimization techniques, including simultaneous strategies that solve simulation and optimization task at the same time. They have shown to be superior in many cases, compare, e.g., [37, 35, 8].

It turns out that the optimization problems that need to be solved in the context of the *Tailorshop* scenario are mixed-integer nonlinear programs with nonconvex continuous relaxations. Whenever optimization problems involve variables of continuous and discrete nature, the term mixed-integer is used. In our case they can be interpreted as discretized optimal control problems. See Chapter 2 for a review of algorithms to treat continuous-time mixed-integer optimal control problems. However, as the time grid is fixed, the applicability of such methods is limited, and we focus on combinatorial methods.

Progress in mixed-integer linear programming (MILP) started with the fundamental work of Dantzig and coworkers on the Traveling Salesman problem in the 1950s. Since then, enormous progress has been made in areas such as *linear programming* (and especially in the *dual simplex* method that is the core of almost all MILP solvers because of its restart capabilities), in the understanding of *branching rules* and more powerful selection criteria such as *strong branching*, the derivation of tight *cutting planes*, novel *preprocessing* and *bound tightening procedures*, and of course the computational advances roughly following Moore's law. For specific problem classes problems with millions of integer variables can now be routinely solved [13]. Also generic problems can often be solved very efficiently in practice, despite the known exponential complexity from a theoretical point of view [38].

The situation is different in the field of Mixed-Integer Nonlinear Programming (MINLP). Only at first sight many properties of MILP seem to carry over to the nonlinear case. Restarting nonlinear continuous relaxations within branching trees is essentially more difficult than restarting linear relaxations (which, e.g., *BARON* and *Couenne* also use for nonlinear problems), as no dual algorithm comparable to the dual simplex is available in the general case. Nonconvexities lead to local minima and do not allow for easy calculation of subtrees, which is important to avoid an explicit enumeration. Additionally, nonlinear solvers are slower and less robust than LP solvers. However, the last decade saw great progress triggered by cross-disciplinary work of integer and nonlinear optimizers, resulting in generic MINLP solvers, e.g., [1, 45]. Most of them, however, still require the underlying functions to be convex. Comprehensive surveys on algorithms and software for convex MINLPs are given in [120, 47]. Recent progress in the solution of nonconvex MINLPs is in most cases based on methods from global optimization, in particular convex under- and overestimation. See, e.g., [92, 235] for references on general under– and overestimation of functions and sets.

Our intention is to foster interdisciplinary research between psychologists and applied mathematicians. We provide the research community in the field of complex problem solving with the open source software tool *Tobago* [210]. This data generation and analysis tool can be hooked to a variety of optimization solvers. Currently the software supports *AMPL* [93] and *GAMS* [69] interfaces. This allows for the usage of solvers from the *COIN-OR* initiative, which are also available under a public license. In this study we used the global solvers *Couenne* [29] and the local solvers *Bonmin* [45] and *Ipopt* [243]. In addition, we ran the global solver *BARON* [236] on the NEOS server.

It turns out, however, that the size and complexity of the problems leads to extremely long runtimes of the global solvers and can only be used on a small subset of the problems. We present a problem-specific lower bound to avoid bad local maxima and guarantee monotonicity of the analysis function that builds on the locally optimal objective function values. However, additional future work in several mathematical areas is needed to address all demands of researchers in *complex problem solving*.

This chapter is organized as follows. In Section 8.2 we explain the test scenario and derive a mathematical model for the *Tailorshop*. In Section 8.3 details concerning the software implementation and solution of the series of optimization problems are given, together with numerical results. The implications for a psychological study we performed are mentioned in Section 8.4. We give a summary and an outlook to future work in Section 8.5.

8.2 Tailorshop MINLP model

The *Tailorshop* has been developed and implemented as a test scenario in the 1980s by Dörner [78]. It has been used in numerous studies, e.g., [198, 151, 149, 179, 21, 22]. Also comprehensive reviews have been published, see [95, 99, 100, 102, 101], in which also more information on the psychological background can be found.

A participant has to take economic decisions to maximize the overall balance of a small company, specialized in the production and sales of shirts. The scenario comprises twelve rounds (months), in which the participant can modify infrastructure (employees, machines, distribution vans), financial settings (wages, maintenance, prices), and logistical decisions (shop location, buying raw material). As feedback he gets some key indicators in the next round, such as the current number of sold shirts, machines, employees, and the like. Arrows next to the indicators show if the value increased or decreased with respect to the previous round.

There are two different kinds of machines to produce either 50 or 100 shirts per month. Workers need to specialize for work on either one of them. The machines need to be maintained and equipped with raw material to actually produce something. The possible production depends furthermore on the satisfaction of the workers, linked to the controls wages and social expenses. Vans influence the demand in a positive way. Furthermore, advertisement, location of the sales shop, and shirt pricing decisions can be used to maximize profit.

We derive a mathematical formulation as an optimization problem. The basic idea is that for different initial values (the current state in round n_s of a participant's test run) the optimal solution for the remaining $N - n_\mathrm{s}$ rounds can be calculated. The optimal solution can then either be used for a manual comparison and analysis of the participant's decisions, Section 8.3, or for an automated indicator function, as discussed in Section 8.4.

The *Tailorshop* has been developed as a computer-based test scenario in GW-Basic code in the early eighties. This implementation was the starting point for the mathematical modeling process. Figure 8.8 in the Appendix shows a short extract of this file. The scenario as it is implemented in GW-Basic has several shortcomings and assumptions one might disagree about. However, this implementation and similar ones have been used over years and at the point where the interdisciplinary cooperation started, already most of the data of the 174 participants had been evaluated in a cumbersome procedure. Hence the formulation of test scenarios that have better mathematical properties has been postponed to future work, and the mathematical model which we derive from the GW-Basic code can be considered as given, even if it is not in all aspects close to reality.

On the basis of the GW-Basic code we derived a mathematical optimization problem for a participant and month $0 \le n_\mathrm{s} < N$ as

$$\begin{array}{llcll} \max\limits_{x,u,s} & F(x_N) & & & \\ \text{s.t.} & x_{k+1} & = & G(x_k, u_k, s_k, p), & k = n_\mathrm{s} \dots N-1, \\ & 0 & \le & H(x_k, x_{k+1}, u_k, s_k, p), & k = n_\mathrm{s} \dots N-1, \\ & u_k & \in & \Omega, & k = n_\mathrm{s} \dots N-1, \\ & x_{n_\mathrm{s}} & = & x^p_{n_\mathrm{s}}. & \end{array} \tag{8.1}$$

The model is dynamic with a discrete time $k = 0 \dots N$, where $N = 12$ is the number of rounds. The control vector $u_k = u(k)$ has 15 (or 13 when van purchase is fixed, compare Section 8.6.2) entries for each $k = 0 \dots N-1$ corresponding to the decisions the participant can make in the test. The vector of dependent state variables $x_k = x(k)$ comprises 16 entries. Both are listed in Table 8.1 (note that units of control and state variables are only given implicitly depending on how they enter the model equations and constraints). The vector s_k denotes slack variables we introduced to reformulate $\min - \max$ expressions by standard techniques using the constraints (8.27)–(8.31). For details on these and further reformulations, see Section 8.6.2. We define

$$(x^{\mathrm{p}}, u^{\mathrm{p}}) = (x_0^{\mathrm{p}}, \dots, x_N^{\mathrm{p}}, u_0^{\mathrm{p}}, \dots, u_{N-1}^{\mathrm{p}})$$

to be the vector of decisions and state variables for all months of a participant. Certain entries $x_{n_s}^p$ enter (8.1) as fixed initial values. Participant independent initial values $x_0^{\mathrm{p}} = p^{x_0}$ are given alongside fixed parameters p in Table 8.4 in the Appendix. Random values ξ appear in the equations, e.g., line 2810 in Figure 8.8. However, a detailed analysis of the compiled code revealed that the random values are only dependent on an initialization (seed) within the GW-Basic code, hence they are identical for all participants and can be fixed in the optimization problem, see Table 8.5 in the Appendix.

The goal is to find decisions u_k that maximize the overall balance at the end of the time horizon. The objective function is given by

$$F(x_N) = x_N^{OB}.$$

Whenever we use the expression *relaxed optimization problem* this refers to the case in which the sets of points in (8.21–8.24) are replaced by their convex hulls. The state propagation law $G(\cdot)$ is determined by the following set of equations for all $k \in \{0, \dots, 11\}$. For the sake of readability we omit the implicitly given units in the equations.

The number of machines for 50 and 100 shirts per month depends on buying and selling of machines. Note that there is a difference between buying and selling in the base capital equation so that two independent controls are needed here:

$$x_{k+1}^{M_{50}} = x_k^{M_{50}} + u_k^{\Delta M_{50}} - u_k^{\delta M_{50}}, \tag{8.2}$$

$$x_{k+1}^{M_{100}} = x_k^{M_{100}} + u_k^{\Delta M_{100}} - u_k^{\delta M_{100}}. \tag{8.3}$$

For the workers a single control which stands for hiring and firing workers is sufficient since there is no such difference (one might even avoid the state variable if the control was the current number of workers, but we stick to the hiring control for historical reasons):

$$x_{k+1}^{W_{50}} = x_k^{W_{50}} + u_k^{\Delta W_{50}}, \tag{8.4}$$

$$x_{k+1}^{W_{100}} = x_k^{W_{100}} + u_k^{\Delta W_{100}}. \tag{8.5}$$

Demand depends on a time dependent parameter p_k^{DE} as well as on the advertisement expenses

Decision	u_k	unit*	**State**	x_k	unit*
advertisement	u_k^{AD}	MU	machines 50	$x_k^{M_{50}}$	machines
shirt price	u_k^{SP}	MU	machines 100	$x_k^{M_{100}}$	machines
buy raw material	$u_k^{\Delta MS}$	shirts	workers 50	$x_k^{W_{50}}$	workers
workers 50	$u_k^{\Delta W_{50}}$	workers	workers 100	$x_k^{W_{100}}$	workers
workers 100	$u_k^{\Delta W_{100}}$	workers	demand	x_k^{DE}	shirts
buy machines 50	$u_k^{\Delta M_{50}}$	machines	vans	x_k^{VA}	vans
buy machines 100	$u_k^{\Delta M_{100}}$	machines	shirts sales	x_k^{SS}	shirts
sell machines 50	$u_k^{\delta M_{50}}$	machines	shirts stock	x_k^{ST}	shirts
sell machines 100	$u_k^{\delta M_{100}}$	machines	possible production	x_k^{PP}	shirts
maintenance	u_k^{MA}	MU	actual production	x_k^{AP}	shirts
wages	u_k^{WA}	MU	material stock	x_k^{MS}	shirts
social expenses	u_k^{SC}	MU	satisfaction	x_k^{SA}	—
buy vans	$u_k^{\Delta VA}$	vans	machine capacity	x_k^{MC}	shirts
sell vans	$u_k^{\delta VA}$	vans	base capital	x_k^{BC}	MU
choose site	u_k^{CS}	—	capital after interest	x_k^{CA}	MU
			overall balance	x_k^{OB}	MU

Table 8.1: Controls and states in the *Tailorshop* optimization problem with $k \in \{0,\dots,11\}$ for controls respectively $k \in \{0,\dots,12\}$ for states. Note that units are only given implicitly in the test scenario. * MU means money units.

and the number of vans multiplied by a factor depending on the site, $f^1(u_k^{CS})$ (see Section 8.6.2),

$$x_{k+1}^{DE} = 100p_k^{DE} - 50 + \left(\frac{u_k^{AD}}{5} + 100(x_k^{VA} + u_k^{\Delta VA} - u_k^{\delta VA})\right) f^1(u_k^{CS}). \tag{8.6}$$

While the influence of advertisement is bounded, see below and Section 8.6.2, the effect of vans is unbounded. This leads to unboundedness of the whole problem. In Section 8.6.2 our approach to generate reasonable solutions anyway is described.

For the vans again, two controls for buying and selling are needed due to differences in the base capital. Shirt sales are determined by the slack variable s_k^{SS} and shirts in stock depend on the slack variables for actual production s_k^{PP} and shirt sales s_k^{SS},

$$x_{k+1}^{VA} = x_k^{VA} + u_k^{\Delta VA} - u_k^{\delta VA}, \tag{8.7}$$

$$x_{k+1}^{SS} = s_k^{SS}, \tag{8.8}$$

$$x_{k+1}^{ST} = x_k^{ST} + s_k^{PP} - s_k^{SS}. \tag{8.9}$$

In the possible production equation, the part representing machine and worker dependence consists of a term for each machine type with slack variables $s_k^{M_{50}}$ and $s_k^{M_{100}}$, which are used to replace min expressions of workers and machines, multiplied by a machine capacity term (machines for 100 shirts have double machine capacity). This part is multiplied by the square root of workers' satisfaction. The actual production is determined by a slack variable.

$$\begin{aligned} x_{k+1}^{PP} = &\left(s_k^{M_{50}}(x_k^{MC} + 4p_k^{P_{50}} - 2) + s_k^{M_{100}}(2x_k^{MC} + 6p_k^{P_{100}} - 3)\right) \\ &\cdot \left(\frac{1}{2} + \frac{u_k^{WA} - 850}{550} + \frac{u_k^{SC}}{800}\right)^{\frac{1}{2}} \end{aligned} \tag{8.10}$$

$$x_{k+1}^{AP} = s_k^{PP} \tag{8.11}$$

Raw material in stock depends on the use of material represented by the slack variable for actual production and the purchase of new material. Wages and social expenses influence satisfaction and the machine capacity is determined by a slack variable:

$$x_{k+1}^{MS} = x_k^{MS} + u_k^{\Delta MS} - s_k^{PP}, \tag{8.12}$$

$$x_{k+1}^{SA} = \frac{1}{2} + \frac{u_k^{WA} - 850}{550} + \frac{u_k^{SC}}{800}, \tag{8.13}$$

$$x_{k+1}^{MC} = s_k^{MC}. \tag{8.14}$$

The equation for base capital,

$$x_{k+1}^{BC} = x_k^{CA} + s_k^{SS} \cdot u_k^{SP} - p_k^{PR} \cdot u_k^{\Delta MS} - 10000u_k^{\Delta M_{50}} - f^2(u_k^{CS})$$

$$
\begin{aligned}
&+ 8000\frac{x_k^{MC}}{p^{MM}}u_k^{\delta M_{50}} - 20000 u_k^{\Delta M_{100}} + 16000\frac{x_k^{MC}}{p^{MM}}u_k^{\delta M_{100}} \\
&- u_k^{AD} - u_k^{MA} - (x_k^{W_{50}} + u_k^{\Delta W_{50}} + x_k^{W_{100}} + u_k^{\Delta W_{100}}) \cdot (u_k^{WA} + u_k^{SC}) \\
&- 2s_k^{PP} - \frac{1}{2}(x_k^{MS} + u_k^{\Delta MS} - s_k^{PP}) - x_k^{ST} - 10000 \cdot u_k^{\Delta VA} \\
&+ (8000 - 100k) \cdot u_k^{\delta VA} - 500(x_k^{VA} + u_k^{\Delta VA} - u_k^{\delta VA}), \qquad (8.15)
\end{aligned}
$$

contains all income and expenses during a round added to the capital after interest from the previous round. The income consists of the amount of shirts sold times the shirt price $s_k^{SS} \cdot u_k^{SP}$, the sale of machines $8000\frac{x_k^{MC}}{p^{MM}}u_k^{\delta M_{50}}$ and $16000\frac{x_k^{MC}}{p^{MM}}u_k^{\delta M_{100}}$ (depending on the current machine capacity), and the sale of vans $(8000 - 100k) \cdot u_k^{\delta VA}$.

Money is spent for the raw material bought times the price of a raw material unit $-p_k^{PR} \cdot u_k^{\Delta MS}$, the purchase of machines $-10000\,u_k^{\Delta M_{50}}$ and $-20000\,u_k^{\Delta M_{100}}$, the purchase of vans $-10000\,u_k^{\Delta VA}$, advertisement and maintenance $-u_k^{AD} - u_k^{MA}$, and the number of workers times wages plus social expenses $-(x_k^{W_{50}} + u_k^{\Delta W_{50}} + x_k^{W_{100}} + u_k^{\Delta W_{100}}) \cdot (u_k^{WA} + u_k^{SC})$. Additionally, each unit of material in stock at the end of a round costs half a money unit (MU) $-\frac{1}{2}(x_k^{MS} + u_k^{\Delta MS} - s_k^{PP})$, the production of a shirt costs two MU $-2s_k^{PP}$, each shirt in stock costs one MU, and each van costs 500 MU per round $-500(x_k^{VA} + u_k^{\Delta VA} - u_k^{\delta VA})$. There is another amount of money to be paid, which depends on the site, $-f^2(u_k^{CS})$ (see Section 8.6.2).

From the base capital the capital after interest is computed by multiplying it with an interest rate factor $(1 + p^{IR})$. Overall balance, the objective function, besides capital after interest contains terms for material and shirts in stock, for machines, and for vans. However, machines are worth less in the overall balance than if they were sold:

$$
x_{k+1}^{CA} = x_{k+1}^{BC}(1 + p^{IR}) \qquad (8.16)
$$

$$
\begin{aligned}
x_{k+1}^{OB} = &\frac{x_k^{MC}}{p^{MM}}\left(8000(x_k^{M_{50}} + u_k^{\Delta M_{50}} - u_k^{\delta M_{50}}) + 16000(x_k^{M_{100}} + u_k^{\Delta M_{100}} - u_k^{\delta M_{100}})\right) \\
&+ (8000 - 100k) \cdot (x_k^{VA} + u_k^{\Delta VA} - u_k^{\delta VA}) \\
&+ 2(x_k^{MS} + u_k^{\Delta MS} - s_k^{PP}) + 20(x_k^{ST} + s_k^{PP} - s_k^{SS}) + x_k^{CA} \qquad (8.17)
\end{aligned}
$$

The feasible set of controls is defined by the following properties for all $k \in \{0, \ldots, 11\}$:

$$
\begin{aligned}
u_k^{AD} &\in [0, 10000], & u_k^{SP} &\in [10, 100], && (8.18) \\
u_k^{\Delta MS} &\in [0, 50000], & u_k^{MA} &\in [0.1, 100000], && (8.19) \\
u_k^{WA} &\in [850, 5000], & u_k^{SC} &\in [0, 10000], && (8.20) \\
u_k^{\Delta W_{50}} &\in \{-200, -199, \ldots, 200\}, & u_k^{\Delta W_{100}} &\in \{-200, -199, \ldots, 200\}, && (8.21) \\
u_k^{\Delta M_{50}} &\in \{0, 1, \ldots, 200\}, & u_k^{\Delta M_{100}} &\in \{0, 1, \ldots, 200\}, && (8.22) \\
u_k^{\delta M_{50}} &\in \{0, 1, \ldots, 200\}, & u_k^{\delta M_{100}} &\in \{0, 1, \ldots, 200\}, && (8.23)
\end{aligned}
$$

$$u_k^{CS} \in \{0,1,2\}. \tag{8.24}$$

Furthermore, for all $k \in \{0,\ldots,11\}$ the constraints

$$u_k^{\Delta W_{50}} \geq -x_k^{W_{50}}, \qquad u_k^{\Delta W_{100}} \geq -x_k^{W_{100}}, \tag{8.25}$$

$$u_k^{\delta M_{50}} \leq x_k^{M_{50}}, \qquad u_k^{\delta M_{100}} \leq x_k^{M_{100}} \tag{8.26}$$

need to hold. Slack variables are used to reformulate min expressions (see also 8.6.2) and the bounds on the slack variables read as

$$s_k^{PP} \leq x_k^{MS} + u_k^{\Delta MS}, \qquad s_k^{PP} \leq x_{k+1}^{PP}, \tag{8.27}$$

$$s_k^{MC} \leq p^{MM}, \qquad s_k^{MC} \leq 0.9 x_k^{MC} + 0.017 \frac{u_k^{MA}}{x_{k+1}^{M_{50}} + 10^{-8} x_{k+1}^{M_{100}} + 10^{-8}}, \tag{8.28}$$

$$s_k^{SS} \leq x_k^{ST} + x_{k+1}^{AP}, \qquad s_k^{SS} \leq \frac{5}{4}(\frac{x_k^{DE}}{2} + 280) \cdot 2.7181^{-\frac{u_k^{SP2}}{4250}}, \tag{8.29}$$

$$s_k^{M_{50}} \leq x_{k+1}^{W_{50}}, \qquad s_k^{M_{50}} \leq x_{k+1}^{M_{50}}, \tag{8.30}$$

$$s_k^{M_{100}} \leq x_{k+1}^{W_{100}}, \qquad s_k^{M_{100}} \leq x_{k+1}^{M_{100}}. \tag{8.31}$$

for all $k \in \{0,\ldots,11\}$. s_k^{PP} is used for the minimum of possible production and material in stock. With s_k^{MC}, the minimum of maximum machine capacity p^{MM} and the machine capacity determined by loss of capacity over time and the recovery by maintenance is described. Here, the first 10^{-8} in the denominator comes from a modeling bug, see 8.6.2. Finally, s_k^{SS} is used to reformulate the minimum of shirts available for sale $x_k^{ST} + x_{k+1}^{AP}$ and a nonlinear term depending on the demand and the shirt price. Note that 2.7181 has been used in the GW-Basic code instead of exp.

To sum up: every single optimization problem is of the general form (8.1), where the functions $G(\cdot)$ and $H(\cdot)$ are smooth, nonlinear functions of the unknown variables x, u and s. The nonlinearities are often bilinear, but sometimes also include denominators and exponentials.

8.3 Optimization and numerical results

We want to solve a series of optimization problems of the form (8.1) for different participant data that has been obtained experimentally. In Section 8.3.1 we describe the algorithms and software we used to achieve this goal. In Section 8.3.2 examples of optimal solutions are displayed and discussed for illustration. The important issues of integrality and non convexity that arise in our problems are discussed in Section 8.3.3. We close by discussing the use of Lagrange Multipliers of artificial constraints as a means to further investigate good and bad decisions of a participant in Section 8.3.4.

8.3.1 Implementation

To be able to analyze and visualize the data in a convenient way, to have a simulation environment for own studies, and to be able to automatize the optimization of all $2088 = 174 \cdot 12$ problems, we implemented the software framework *Tobago* [210]. It is publicly available under an open source license, includes a GUI, and may as well be used for experimental setups. In this study however we exclusively used the GW-Basic implementation for tests to have consistent data, and *Tobago* only for optimization and analysis.

We interface the data with optimization solvers via an automated call of *AMPL* [93] to be able to easily exchange optimization solvers that have an *AMPL* interface. In this study we compare three different optimization solvers: *Ipopt* [243], *Bonmin* [45], and *Couenne* [29]. The first one is a local nonlinear programming solver based on an interior point method. *Bonmin* is a solver for MINLPs whose continuous relaxation is convex (*convex MINLPs*) and uses *Ipopt* for the solution of relaxed problems. *Couenne* is a global solver for MINLPs whose continuous relaxation is nonconvex (*nonconvex MINLPs*). All three are available within the *COIN-OR* open source initiative. We used the currently latest stable version 0.2.2 of *Couenne*, and for better comparability the versions 1.1.1 of *Bonmin* and 3.6.1 of *Ipopt* it is interfaced with. For all solvers we used the default settings exclusively and the MA27 sparse solver for numerical linear algebra.

All computational times refer to a two core Intel CPU with 3GHz and 8GB RAM run under Ubuntu 9.10.

8.3.2 Optimal Solutions

In total, 2088 optimization problems have been solved. Depending on the value of n_s in (8.1), each consists of $13(N-n_\text{s})$ control, $16(N-n_\text{s})$ state, and $5(N-n_\text{s})$ slack variables. The total number of optimized variables for all 174 participants sums up to

$$n_\text{var} \quad = \quad 174 \sum_{n_\text{s}=0}^{N-1} 34(N-n_\text{s}) = 174 \cdot 2652 = 461448.$$

This many variables are obviously difficult to discuss and visualize comprehensively. From this large set of results we chose a few which illustrate how optimal solutions relate to the choices made by the participant, compared to solutions for different values of n_s, and compared to optimal solutions of other participants. These solutions have been obtained with the local optimization code *Ipopt* and an outer loop with random start values for the optimization. Hence it needs to be stressed that the interpretations are always under the assumption that the obtained results are close to the global optima.

In Figure 8.1 the shirt price control function u_k^{SP} of two participants is displayed. In addition to the values chosen by the participants, all optimal solutions are also depicted, giving an idea what the participants could have done to improve their performance. It is interesting to observe that the optimal solutions corresponding to the two participants show different behavior, depending on the start values $x^p_{n_s}$ in (8.1).

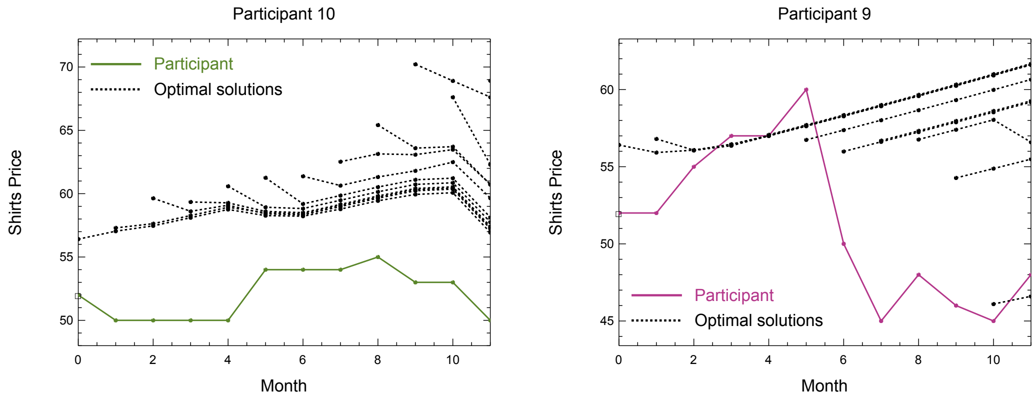


Figure 8.1: Shirt price u^{SP}_k of two different participants in solid lines. The dotted lines show optimal solutions of (8.1) starting at different months n_s. Both participants should have chosen higher prices most of the time. Depending on their other choices, the optimal solutions evolve differently over time. On the left hand side the participant's tailorshop is developing towards high demand and little stock of shirts, hence the optimal shirt price to maximize profit is increasing. On the right hand side the demand is declining and the stock of shirts increasing, hence the optimal price is falling with time.

The effect of the van modeling bug, see Section 8.6.2, is discussed in Figure 8.2.
In Figure 8.3 the state variable $x^{W_{100}}_k$ is depicted. It indicates how many workers for the 100 machines are employed at time k. In the left column of the visualization the modeling bug discussed in Section 8.6.2 plays a role. It leads to a denominator in (8.43) with a value of 10^{-8} whenever all machines are sold and allows hence to obtain the maximum machine capacity with a very small investment into u^{MA}_k. However, the optimal solution only exploits this in some cases, as can be clearly seen by comparing the left and the right column.
In Figure 8.4 the important state variable x^{OB}_k is depicted for one representative participant. As the value of this function at the end time $k = N$ is the objective function that is to be maximized, the function shows how much better the optimal solution performs in comparison to the participant. There are only minor deviations from a monotonic increase that result mainly from the investment into raw material which is not profitable within the overall balance, but as a resource for future profit.

8.3.3 Local maxima and integer solutions

The optimization problems (8.1) are nonconvex. Depending on initial values for the optimization variables different local maxima can be found. Hence one has to use a global optimization solver, such as *Couenne* or one of the solvers listed on [58]. As mentioned above, we used different solvers to obtain solutions. Table 8.2 shows an overview of average computational times and objective function values that have been obtained with *Ipopt* and *Bonmin*.

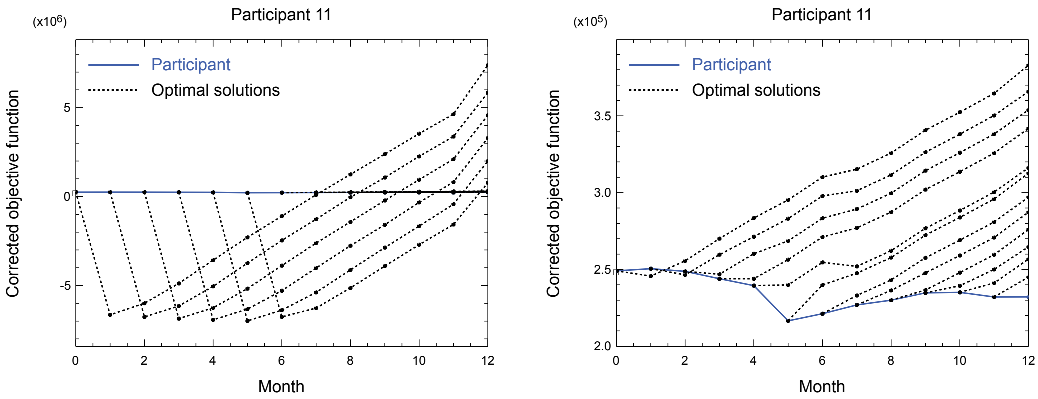


Figure 8.2: Development of the overall balance. The dotted lines show optimal solutions of (8.1) starting at different months n_s. A comparison of left and right shows the effect of the modeling bug discussed in Section 8.6.2: the number of vans increases the demand and without bounds on the variables the solution is unbounded. Left: In this case an upper bound of 200 for the purchase of machines per month is the limiting factor. Note that the effect can only be exploited for $n_s \leq 6$, as the investment in vans needs at least 6 months to pay off. The participant's trajectory as well as all solutions for $n_s > 6$ in the order of 10^5 seem to be zero because of the nonlogarithmic scale. Right: optimal solutions for fixed number of vans. This formulation was used for analysis.

We ran the global solvers *Couenne* and *BARON* only on single optimization instances, as the computational demand was too high. On typical instances, *Couenne* was able to solve (8.1) for $n_s = 11$ in approximately 3 seconds. For the next larger problem, $n_s = 10$, however, the Branch and Bound tree grew too fast. The solver terminated after processing 600000 nodes in 7 hours, because the computer ran out of memory. The stack comprised about 2 million open nodes at that time. The best solution at that time was 500497 with the upper bound of 506610 still leaving a certain gap. For comparison: the objective function values found by *Bonmin* and *Ipopt* are 490385 and 500779, respectively. When heuristic non-convexity options num_resolve_at_root and num_resolve_at_node are used with a value of 1 (or 2) for *Bonmin*, an integer solution with value 500188 (500438) is found after 142 (317) seconds, which is considerably higher than the 0.2 seconds with the standard settings. With tight bounds on all state, control, and slack variables (some of them even fixed) and the newer version *Couenne* 0.3.2 a solution could be obtained in 30 minutes, but even so $n_s = 9$ was not solvable on our machine.

A similar behavior occurred when we used *BARON* with our *GAMS* interface on the NEOS server. Although computational times are not comparable due to the different servers and the different preprocessing steps of *AMPL* and *GAMS*, the runtime for *BARON* also increased drastically when the number of variables doubled from $n_s = 11$ to $n_s = 10$. While instances for $n_s = 11$ could be solved within 3 seconds, the ones for $n_s = 10$ could only be solved in the time limit of 8 hours when bounds were tightened to small intervals. An exact investigation of the

	CPU [sec]		Objective function		
n_s	*Ipopt*	*Bonmin*	*Ipopt*	*Bonmin*	Gap
0	0.65	2289	397613	340163	4.5 %
1	0.62	1545	379243	319133	6.9 %
2	0.48	908	359958	296037	6.4 %
3	0.39	556	341110	282423	10.3 %
4	0.33	366	323728	274949	9.6 %
5	0.31	163	307665	263333	12.8 %
6	0.25	66	292389	254858	14.4 %
7	0.20	14	277730	251187	15.1 %
8	0.15	5.48	262800	235850	17.2 %
9	0.11	2.34	249186	233318	17.8 %
10	0.07	0.54	236290	220031	15.9 %
11	0.03	0.10	220717	210760	14.4 %

Table 8.2: Average computational times in seconds and average objective function values for the solutions of problems (8.1) per participant calculated from the 174 data sets. The rows show the start month n_s, the columns results for *Ipopt* for the relaxation of (8.1) and *Bonmin*, respecting the integrality conditions.

reasons for this drastic increase in computational demand is future work.

Obviously already for one participant data set the computational times are prohibitive for global approaches. For the analysis of all 174 participants we therefore solved 2088 NLP relaxations and MINLPs with the local optimizers *Ipopt* and *Bonmin*.

A crucial feature of our method is that the *How much is still possible*–function, see Section 8.4.1, decreases monotonically with n_s increasing. To take this into account, we exploit this knowledge in our a posteriori analysis. We define

$$(x^*, u^*, s^*) = (x^*_{n_s}, \dots, x^*_N, u^*_{n_s}, \dots, u^*_{N-1}, s^*_{n_s}, \dots, s^*_{N-1})$$

as a locally optimal solution obtained by solving problem (8.1) for month n_s. We initialize the variables for problem (8.1) for month $n_\mathrm{s} - 1$ according to

$$\begin{array}{rcll} x_{n_\mathrm{s}-1} &=& x^\mathrm{p}_{n_\mathrm{s}-1}, & \\ u_{n_\mathrm{s}-1} &=& u^\mathrm{p}_{n_\mathrm{s}-1}, & \\ x_k &=& x^*_k, & k = n_\mathrm{s} \dots N, \\ u_k &=& u^*_k, & k = n_\mathrm{s} \dots N-1 \\ s_k &=& s^*_k, & k = n_\mathrm{s} \dots N-1 \end{array} \tag{8.32}$$

and $s_{n_\mathrm{s}-1}$ according to equations (8.42–8.46). This is a feasible solution because of $x_{n_\mathrm{s}} = x^*_{n_\mathrm{s}} = x^\mathrm{p}_{n_\mathrm{s}} = G(x^\mathrm{p}_{n_\mathrm{s}-1}, u^\mathrm{p}_{n_\mathrm{s}-1}, s_{n_\mathrm{s}-1}, p)$ with objective function value $x^{*,OB}_N$. To avoid local maxima with a worse performance, we require that the inequality

$$x^{OB}_N \geq x^{*,OB}_N \tag{8.33}$$

holds. This inequality can either be added to (8.1) when relaxed problems are solved with local optimization algorithms, or be used as a *cutoff* value in a Branch-and-Bound setting to reduce the search tree. Computational experience shows that the primal-dual interior point solver we are using cannot exploit the initialization to its full extent and in many cases *Ipopt* converged to locally infeasible points although it started from a primally feasible one. Future studies should therefore include active set based solvers. For this study we iterated in an inner loop with random initializations until for all problems inequality (8.33) was fulfilled, i.e., *Ipopt* returned a feasible solution.

Within our analysis approach, local maxima can lead to a violation of the goal to have an objective measurement for participant performance. Whenever possible, global solvers with a guaranteed, deterministic global maximum should be used. If the size of the problem is still too large for current algorithms and computational platforms, we propose to use relaxations and include (8.33) as a heuristic compromise.

Several of the control variables are restricted to integer values, compare (8.18-8.24). A comparison of (locally) optimal relaxed and integer solutions shows that some of the variables show typical (qualitatively similar throughout all solutions, e.g., variables are at their upper bounds) behavior for most $x^p_{n_\mathrm{s}}$, such as the maintenance u^{MA}_k or the purchase of raw material $u^{\Delta MS}_k$. Others, in particular the numbers of machines and workers, the shirt price u^{SP}_k, and the choice of the site u^{CS}_k are more sensitive to local optima and/or the fixation of some of the variables to integer values. Figure 8.5 shows an example.

8.3.4 Analyzing Lagrange Multipliers

Using optimization as an analysis tool yields insight on several levels. A priori unknown structural properties of the problem, e.g., the unboundedness due to the van bug, can be detected. Also the performance of a participant can be compared to the optimal solution, and the *How* *much is still possible*–function to be discussed in Section 8.4 delivers a temporal resolution of this performance.

But even a more detailed analysis is possible. While an analysis of the *How much is still possible*–function indicates at what rounds the participant made particularly good or bad decisions, the question of what of the decisions contributed significantly to the success or failure remains and might be of importance in a given test scenario. A global approach[2] would be to fix exactly one entry of u_{n_s} to the value chosen by the participant and compare the result of the optimization to the one without this constraint. The difference between the two objective function values then indicates exactly how much impact this particular decision had. The obvious drawback is that the number of optimization problems that need to be solved increases by a factor of $N \cdot n_u$, where n_u is the number of controls per month.

As a compromise we propose to combine two concepts. First, the comparison of the participant's decisions at month n_s with the optimal solution, $u^\mathrm{p}_{n_\mathrm{s}} - u^*_{n_\mathrm{s}}$, gives a global indication of differences in the controls. However, it is unclear from this comparison how significant a single deviation is. Therefore we use, second, Lagrange Multipliers for the participant's decisions to measure the effect on the objective function. We augment problem (8.1) with the additional constraint

$$u_{n_\mathrm{s}} = u^\mathrm{p}_{n_\mathrm{s}} \tag{8.34}$$

to obtain the optimization problem

$$\begin{array}{llll} \max\limits_{x,u,s} & F(x_N) & & \\ \text{s.t.} & x_{k+1} = & G(x_k, u_k, s_k, p), & k = n_\mathrm{s} \dots N-1, \\ & 0 \leq & H(x_k, x_{k+1}, u_k, s_k, p), & k = n_\mathrm{s} \dots N-1, \\ & u_k \in & \Omega, & k = n_\mathrm{s}+1 \dots N-1, \\ & x_{n_\mathrm{s}} = & x^p_{n_\mathrm{s}}, & \\ & u_{n_\mathrm{s}} = & u^\mathrm{p}_{n_\mathrm{s}}. & \end{array} \tag{8.35}$$

Note that necessarily $x^*_{n_\mathrm{s}+1} = x^\mathrm{p}_{n_\mathrm{s}+1}$, hence problem (8.35) for month n_s has the same solution as problem (8.1) for $n_\mathrm{s}+1$. The replacement of (8.1) by (8.35) yields the same results for the series of all n_s and does not imply the need for additional optimization problems to be solved.

The advantage of formulation (8.35) is that an optimization code also calculates the dual variables or *Lagrange multipliers* λ_{n_s} for the constraints (8.34). It is well known that the Lagrange multipliers indicate the shadow prices, i.e., how much the objective function varies if the corresponding constraints were relaxed. However, it needs to be stressed that this information is a local one for the point $(x^\mathrm{p}_{n_\mathrm{s}}, \dots, x^\mathrm{p}_N, u^\mathrm{p}_{n_\mathrm{s}}, \dots, u^\mathrm{p}_{N-1})$ and assumes that the active set of inequality constraints does not change. As an example the Lagrange multiplier for the shirts price $\lambda^{SP}_{n_\mathrm{s}}$

[2] we assume that we solve all optimization problems to global optimality in this Section

denotes the deviation of the objective function for $u_{n_s}^{SP} + \varepsilon$. Table 8.3 shows an example. The control vector of a participant, the optimal choice of controls, and the Lagrange multipliers are listed.

	u_k^{AD}	u_k^{SP}	$u_k^{\Delta MS}$	u_k^{MA}	u_k^{WA}	u_k^{SC}	u_k^{CS}	
$u_{n_s}^{\mathrm{p}}$	3700	53	999	1400	1130	100	1	
$u_{n_s}^{*}$	4e-07	64.9	3e-07	34.3	1510	4e-07	0	
λ_{n_s}	-1.003	473	-1.2	-1.003	10.7	4.9	-752	

	$u_k^{\Delta W50}$	$u_k^{\Delta W100}$	$u_k^{\Delta M50}$	$u_k^{\Delta M100}$	$u_k^{\delta M50}$	$u_k^{\delta M100}$	$u_k^{\Delta VA}$	$u_k^{\delta VA}$
$u_{n_s}^{\mathrm{p}}$	0	0	0	0	0	0	1	0
$u_{n_s}^{*}$	-4.9	0	0	0	2.9	0	1	0
λ_{n_s}	-1233	3990	547.1	-4050	2552	40	-3726	619

Table 8.3: Lagrange Multipliers for the specific case of one participant and the final month $n_s = 11$. The columns show different entries of the control vector $u(\cdot)$, compare Table 8.1. The rows show three things: first, the decisions $u_{n_s}^{\mathrm{p}}$ taken by the participant. Second, the optimal (relaxed) solution $u_{n_s}^{*}$ calculated with *Ipopt*. Third, the Lagrange multipliers λ_{n_s} for the constraints (8.34).

The analysis of participant's decisions hence needs to take both into account: the global information of the difference $u_{n_s}^{\mathrm{p}} - u_{n_s}^{*}$ and the local quantification from the Lagrange multipliers λ_{n_s}. A good estimate can be obtained from the entries of the componentwise product $\lambda_{n_s} \cdot \left(u_{n_s}^{\mathrm{p}} - u_{n_s}^{*}\right)$.

8.4 A correct indicator function for Tailorshop

We propose to use the solutions of (8.1) for all n_s as an indicator function for the performance of a participant. The approach described in Section 8.4.1 is generic and should also be used for other test scenarios in complex problem solving in the future. In Section 8.4.2 we describe the results we obtained by using this indicator function for a psychological study.

8.4.1 How much is still possible

On an individual basis, the performance of every participant can be better understood by a comparison with optimal solutions as illustrated in Section 8.3. For an evaluation of large data sets that shall be related to characteristics of participants or experimental setup, an automatization and a reduction to an indicator function is necessary. Once the performance of all participants has been determined, an aggregation and further statistical analysis can be performed.

To measure performance within the *Tailorshop* scenario different indicator functions have been proposed in the literature. As discussed in the introduction, they have usually unknown reliability and validity.

We solve an optimization problem (8.1) for every round of the participant's data, starting with exactly the same conditions as the participant. We compare these optimal values that indicate *How much is still possible* if all future decisions were optimal. Thus, we can analyze at what rounds potential for a higher end time objective function value has not been used.

A comparison of the end time capital with the one of the optimal solution from start month $n_\mathrm{s} = 0$ (which is identical for all participants if we neglect the van purchase decision, compare Section 8.6.2) also gives an objective indicator. However, what we propose is a far more powerful analysis approach: we want to also say *when* (within the 12 rounds) significant performance deviations occurred, and we want to specify details on which decisions were particularly good or bad ones with respect to the overall outcome.

Note that a comparison with the *controls* of the optimal solution for starting month $n_\mathrm{s} = 0$ would not yield a good indicator function, as there might be multiple ways to perform well. E.g., if, due to his previous actions, a participant has many shirts on his stock, good decisions may differ drastically from the optimal solution for starting month $n_\mathrm{s} = 0$ in which always all shirts have been sold.

In a certain analogy to the cost-to-go-function in dynamic programming, the optimal objective function values for *all* rounds yield the monotonically decreasing *How much is still possible*–function. We look at the series of optimal objective function values $F^*(x_N; n_\mathrm{s})$ for $n_\mathrm{s} = 0, \ldots, N-1$. By comparing $F^*(x_N; n_\mathrm{s} = k)$ with $F^*(x_N; n_\mathrm{s} = k+1)$ we obtain the exact value of how much less the participant is still able to obtain, assumed he would take the best solutions from now on. In other words: whether the tailorshop is in a worse situation than it could be, after the participant's decisions. We define the non–positive (for global optima) *Use of Potential*–function

$$\Delta P_k =: F^*(x_N; n_\mathrm{s} = k+1) - F^*(x_N; n_\mathrm{s} = k). \tag{8.36}$$

Note that in general also a relative loss given as a percentage can be used, however this does not make sense when the function $F^*(\cdot)$ is not bounded as in our case.

As indicated in Figure 8.6 different ways to analyze the complex solving process may yield different results. Also the important issue of selling all shirts and material in the last round is only insufficiently captured by the previous indicator functions. For most of the participants' data the previous indicator and the new, optimization-based one coincide, compare Figure 8.7 (right). This is mainly due to the fact that two of the main effects to make non-intuitive investments into the future were almost never found by the participants: first, the purchase of a high number of vans to stimulate demand (compare Section 8.6.2) and second, the knowledge about the lowest price of the material in round 6.

We conclude that the newly proposed methodology is more reliable and generally applicable to test scenarios in complex problem solving. Non–optimization based indicator functions give good estimates as long as the aforementioned effects are not exploited, which is to be expected, e.g., in studies of learning behavior when participants would be tested several times.

8.4.2 Impact of Emotion Regulation

In the study 174 data sets have been used, every one from a different participant who had but one try. For 42 of them a *positive feedback* was used in the sense that in every round, regardless of the decisions the participant took, a sum of 20000 money units (MU) was added to the capital. For 42 participants a *negative feedback* in form of a reduction of 8000 MUs was implemented. These modifications are implemented in the model and readjusted in the a posteriori analysis, of course.

In a previous study [22] it was shown that participants who receive a negative feedback perform better than those who receive positive feedback. In our new study we additionally considered the ability to regulate emotion. The psychological results of this study are explained in [21] in which also details on the experimental setup can be found. As a main result, an interaction between feedback and emotion regulation could be shown: participants with a high ability of emotion regulation perform better when they get negative feedback, while those with a low ability to regulate their emotions perform bad for negative and good for positive feedback. This is illustrated in Figure 8.7 (left).

In a second study, films and music were used to induce happy, neutral, and sad affect. Additionally we measured emotion regulation. The study was based on data from 90 participants, 30 in each affect condition. Again, emotion regulation had a great impact on complex problem solving. A high ability to regulate emotion improved complex problem solving and reduced the amount of mistakes.

8.5 Summary

We presented a challenging problem class of nonconvex MINLPs. They originate from economic test scenarios that are used in the analysis of human complex problem solving. Starting from GW-Basic source code of the test scenario we developed a mathematical optimization model to optimize performance starting from pre-specified initial values. This model needed to be reformulated in several ways to avoid non-differentiabilities, division by zero, and unboundedness.

The *Tailorshop* test scenario was invented more than 25 years ago, without any intention to set it up suited for mathematical optimization. Our study revealed several shortcomings of the model. This insight can be used for defining better test scenarios in the future. All characteristics such as nondifferentiabilities, random values, or unbounded decision variables should be left out, as they do not really contribute to the difficulty of the scenario itself, but mainly to the difficulty of solving the problem mathematically to optimality.

We solved altogether 2088 optimization problems and discussed the role of integer variables and the nonconvexity by comparing different algorithms. The difficulties to do so for a large number of medium–scale nonconvex MINLPs are challenging. We formulated and used a structure exploiting lower bound to exclude certain unwanted local maxima. The optimization results were used in two ways. First, to gain additional insight into individual performance by comparing it to the optimal solution which is often non–obvious. Second, to use the results in an automated way as a new analysis tool for process-dependent evaluation of the performance.

This novel methodology yields a valuable (and accessible, [210]) analysis tool for psychologists to evaluate participants' performance. We discussed why there is no alternative to the *How much is still possible*–function, especially when participants have more insight, e.g., by repetition of tests. Furthermore, we proposed to add artificial constraints to the optimization problem and use the Lagrange multipliers of these constraints as an indication of what decisions contributed significantly to good or bad performance. By providing this mathematical technology to analyze participants' decisions in more detail, a whole set of interesting scenarios with a time– and decision–specific resolution can be included in future psychological investigations.

This work provides a reference for researchers in complex problem solving. But we also hope for a stimulating effect on optimization. Future studies should concentrate on restarts for the MINLPs, on a comparison with active-set based solvers, problem-specific cuts, tight bounds also for nonlinear subexpressions, and on more efficient techniques to find global optima.

8.6 Appendix

8.6.1 Details of the Optimization Model

We list several parameters and initial values that are of relevance for the optimization problem (8.1) in Tables 8.4 and 8.5. Figure 8.8 shows an extract of the original source code.

8.6.2 Derivation of the Optimization Model

We discuss some properties of (8.1) in more detail.

Integer Variables and Bounds

For a carefully specified optimization problem the definition of the feasible set of all control variables is crucial. Within the test scenario, for several decisions there are no bounds and it is unclear, whether variables are restricted to be from a finite set or not. Although the GW-Basic code does not specifically distinguish between integer and real variables, all participants restricted themselves to integer numbers for the choices they made. Hence we decided to define some of the variables, e.g., the number of workers to be hired, as integer variables.

The only clearly defined integer variable within the GW-Basic code is the choice of the showroom where the shirts are being sold. There are only three choices: city center, city, and suburbs,

State [unit]	x_k	$x_0 =$
machines 50 [machines]	$x_k^{M_{50}}$	10
shirts stock [shirts]	x_k^{ST}	80.7164
machines 100 [machines]	$x_k^{M_{100}}$	0
vans [vans]	x_k^{VA}	1
workers 50 [workers]	$x_k^{W_{50}}$	8
material stock [shirts]	x_k^{MS}	16.06787
workers 100 [workers]	$x_k^{W_{100}}$	0
machine capacity [shirts]	x_k^{MC}	47.04
demand [shirts]	x_k^{DE}	766.636
capital after interest [MU*]	x_k^{CA}	165774.66
Parameter [unit]	p	$p =$
max. demand [shirts]	p^{MD}	900
interest rate [—]	p^{IR}	0.0025
max. machine capacity [shirts]	p^{MM}	50
debt rate [—]	p^{DR}	0.0066
max. satisfaction [—]	p^{MS}	1.7

Table 8.4: Fixed initial values x_0 and parameters p. Note that some initial values are not needed, as they do not enter the right-hand-side function $G(\cdot)$. Note also that units are only implicitly given in the test scenario. * MU means money units.

which we identify with 2, 1, and 0, respectively. We define

$$f^1(u_k^{CS}) := \begin{cases} 1.2 & \text{if } u_k^{CS} = 2 \\ 1.1 & \text{if } u_k^{CS} = 1 \\ 1.0 & \text{if } u_k^{CS} = 0 \end{cases}, \qquad f^2(u_k^{CS}) := \begin{cases} 2000 & \text{if } u_k^{CS} = 2 \\ 1000 & \text{if } u_k^{CS} = 1 \\ 500 & \text{if } u_k^{CS} = 0 \end{cases}.$$

To be able to relax the feasible set of u_k^{CS}, we write these functions as

$$f^1(u_k^{CS}) := 1 + \frac{u_k^{CS}}{10}, \qquad f^2(u_k^{CS}) := 500 + 250\, u_k^{CS} + 250\, u_k^{CS} \cdot u_k^{CS}.$$

For optimization algorithms the existence of tight lower and upper bounds makes a huge difference in runtime. By a process of trial and error we found several bounds that were never violated by any optimal or participant control. We define the feasible set Ω of the control variables as given by the conditions (8.18–8.24).

k	p_k^{PR} [MU*]	p_k^{DE} [—]	$p_k^{P_{50}}$ [—]	$p_k^{P_{100}}$ [—]
0	4.00000	0.616192	0.583334	0.178080
1	4.09497	0.269502	0.080131	0.365665
2	8.26718	0.692422	0.599074	0.725099
3	4.87143	0.844487	0.177331	0.207369
4	4.85305	0.697927	0.075705	0.092567
5	5.90983	0.253290	0.669259	0.318009
6	5.18731	0.805071	0.587936	0.056364
7	7.09909	0.457335	0.107187	0.543777
8	6.77216	0.889342	0.788597	0.157994
9	7.61718	0.371173	0.370508	0.746488
10	8.02385	0.029353	0.908646	0.204585
11	2.68115	0.362480	0.166743	0.303585

Table 8.5: Fixed, but time-dependent parameters p. Note that only p_k^{PR} has an implicitly given unit. The other parameters are dimensionless. * MU means money units.

Reformulations

Although there are some shortcomings in the economic model and the mathematical representation including nondifferentiabilities and no tight bounds on the variables is everything but favorable for a fast and reliable solution, we had to postpone the formulation of test scenarios with better properties to future work, since most of the data of the 174 participants had already been evaluated when the interdisciplinary cooperation started. Hence the main issue was to reformulate the optimization problem to be able to solve it, under the constraint to keep it compatible with the available data.

Concerning non-differentiability we strived to formulate the problem as a smooth optimization problem to allow more solvers to be able to treat the problem instances, if possible without additional binary variables.

As a first example, consider the state progression equation for the machine capacity x_k^{MC}. A direct translation of the code would read as

$$x_{k+1}^{MC} \quad = \quad \min\left(p^{MM}, 0.9x_k^{MC} + 0.017\frac{u_k^{MA}}{x_{k+1}^{M_{50}} + 10^{-8}x_{k+1}^{M_{100}}}\right). \tag{8.37}$$

What was intended here was to include a safeguard to avoid division by zero by using $x_{k+1}^{M_{50}} + x_{k+1}^{M_{100}} + 10^{-8}$ as the denominator, but the GW-Basic implementation used for the evaluation includes the erroneous first version. In our model we add 10^{-8} to the denominator in equation (8.37) to avoid division by zero, but get comparable values for x_{k+1}^{MC}.

Intuitively the fact that we are dealing with a nonconvex model and that there are no bounds on the variables probably means that the problem is unbounded. Indeed, the analysis of optimization results confirmed that due to a combination of a modeling error and the unboundedness of the controls it is possible to drive the overall profit to infinity. In the equation that is describing the overall demand

$$x_{k+1}^{DE} = a + \left(\min\left(\frac{u_k^{AD}}{5}, p^{MD} \right) + 100 x_{k+1}^{VA} \right) \cdot b$$

there is an upper bound on the effect of the advertisement u_k^{AD} by means of a min expression, but not on the impact of vans x_k^{VA}. In other words, by buying more and more vans you can create an arbitrarily high demand. Demand itself enters into the number of sold shirts

$$x_{k+1}^{SS} = \min\left(x_k^{ST}, \frac{5}{4}\left(\frac{x_k^{DE}}{2} + 280 \right) \cdot 2.7181^{-\frac{u_k^{SP2}}{4250}} \right).$$

Therefore you can sell an arbitrary high number of shirts, if only you buy enough vans. However, none of the participants detected this error in the model — this only happened in a related study where participants got several repetitions. We discussed several ways to remove this unboundedness from the problem, e.g., setting a lower bound on the capital to avoid unrealistic infinite debts, possibly by fixing this lower bound to the lowest value over all data sets to keep things consistent. However, the effect of the vans was still too strong, compare Figure 8.2. Eventually we decided to fix the number of vans in the optimization problem to exactly that of the respective participant, and to focus on the other decisions that need to be taken.

The two expressions

$$x_{k+1}^{SA} = \min\left(p^{MS}, \frac{1}{2} + \frac{u_k^{WA} - 850}{550} + \frac{u_k^{SC}}{800} \right) \tag{8.38}$$

$$x_{k+1}^{DE} = \min\left(\frac{u_k^{AD}}{5}, p^{MD}\right) \tag{8.39}$$

can be directly replaced by

$$x_{k+1}^{SA} = \frac{1}{2} + \frac{u_k^{WA} - 850}{550} + \frac{u_k^{SC}}{800}, \qquad \frac{1}{2} + \frac{u_k^{WA} - 850}{550} + \frac{u_k^{SC}}{800} \le p^{MS}, \tag{8.40}$$

$$x_{k+1}^{DE} = \frac{u_k^{AD}}{5}, \qquad \frac{u_k^{AD}}{5} \le p^{MD}. \tag{8.41}$$

We replace the remaining min – max expressions by introducing

$$s_k^{PP} \approx \min(x_{k+1}^{PP}, x_k^{MS} + u_k^{\Delta MS}), \tag{8.42}$$

$$s_k^{MC} \approx \min\left(p^{MM}, 0.9 x_k^{MC} + 0.017 \frac{u_k^{MA}}{x_{k+1}^{M_{50}} + 10^{-8} x_{k+1}^{M_{100}} + 10^{-8}} \right), \tag{8.43}$$

$$\begin{aligned} s_k^{SS} &\approx \min(x_k^{ST} + x_{k+1}^{AP}, \frac{5}{4}(\frac{x_k^{DE}}{2} + 280) \cdot 2.7181^{-\frac{u_k^{SP^2}}{4250}}), && (8.44) \\ s_k^{M_{50}} &\approx \min(x_{k+1}^{W_{50}}, x_{k+1}^{M_{50}}), && (8.45) \\ s_k^{M_{100}} &\approx \min(x_{k+1}^{W_{100}}, x_{k+1}^{M_{100}}). && (8.46) \end{aligned}$$

and adding the corresponding constraints (8.27–8.31).

A constraint that states that new machines may only be bought when the machine capacity x_k^{MC} has at least the value of 35, or in other form

$$0 \leq u_k^{\Delta M_{100}} \leq \begin{cases} 0 & \text{if } x_k^{MC} < 35 \\ \infty & \text{if } x_k^{MC} \geq 35 \end{cases} \tag{8.47}$$

would be a little bit more tricky to reformulate in a way that is suited for a derivative-based optimization algorithm. Fortunately, due to the model bug in (8.37), x_k^{MC} is often at its upper bound p^{MM} in optimal solutions. The model error whenever a participant should have $x_k^{MC} < 35$ seems thus acceptable. Thus we simply ignore constraint (8.47).

Another issue are the interest rates, which have a constant value, but a different one for positive or negative capital x_{k+1}^{BC}. This non-differentiability in the right-hand side could be smoothened out easily by defining an appropriate function piecewise with the constant value p^{IR} for $x_{k+1}^{BC} \geq \delta$, the constant value p^{DR} for $x_{k+1}^{BC} \leq -\delta$ and a smoothing function for the interval $[-\delta, \delta]$, e.g., based on an arcus tangens. However, to facilitate implementation, we chose to use only the positive interest rate p^{IR}. Whenever the optimal solution does not require lending money (hence no $x_k^{BC} < 0$ for any month k), obviously without loss of generality this solution is also optimal for the case with the higher interest rate. This requires another post-processing that we needed to automatize.

The absolute value that occurs in the right-hand side of the state x_{k+1}^{PP} can be neglected because of the lower bound of 850 for the wages u_k^{WA}.

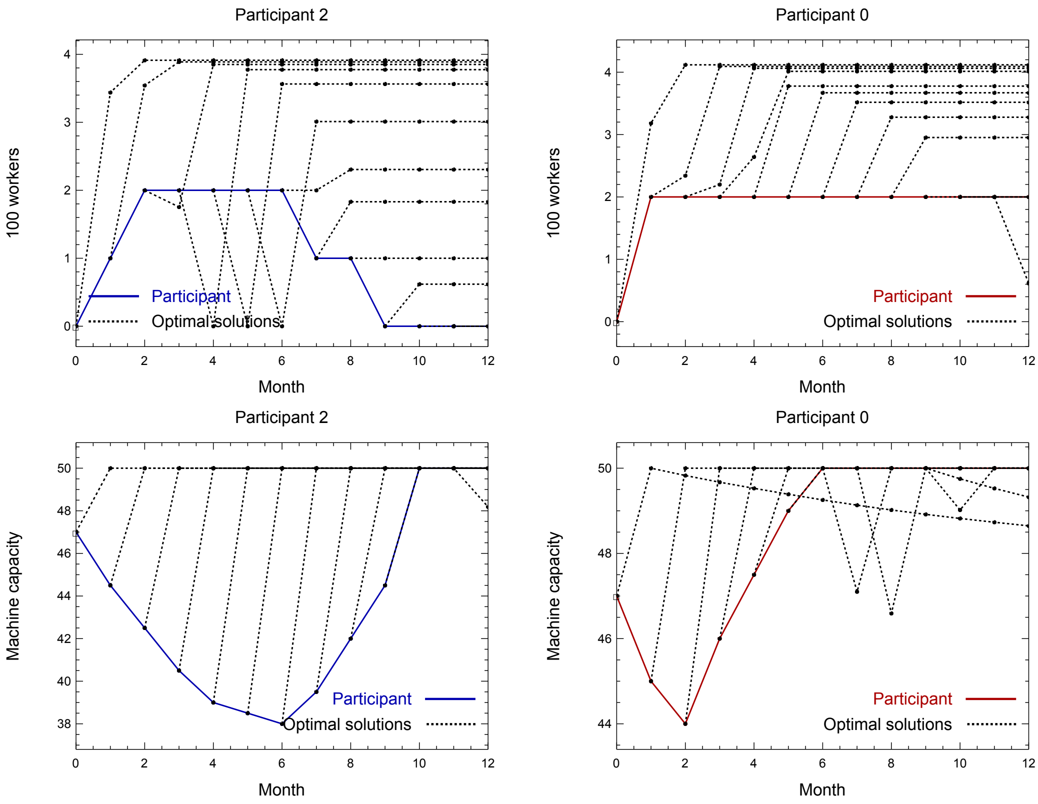


Figure 8.3: Top row: state variable $x_k^{W_{100}}$ that indicates how many workers for the 100 machines are employed. The left and right column show the results for two different participants. For both the optimal strategy is to have a fixed number of 0 to 4 workers which is decreasing as n_s increases. Note that the values are solutions of the relaxed problem where also non-integer values for the number of workers are possible. The main difference lies in the first decisions for $n_s = 3,4,5$ of the left participant to dismiss all workers for the next month. The reason can be understood by looking at the second row, which shows the machine capacity x_k^{MC}. The value on the left is so low that the optimal solution chooses to exploit the special form of the denominator in (8.43) to increase x_k^{MC} to its maximum value with low maintenance costs $u_k^{MA} = \varepsilon$. On the right hand side this effect does not dominate compared to the loss in production.

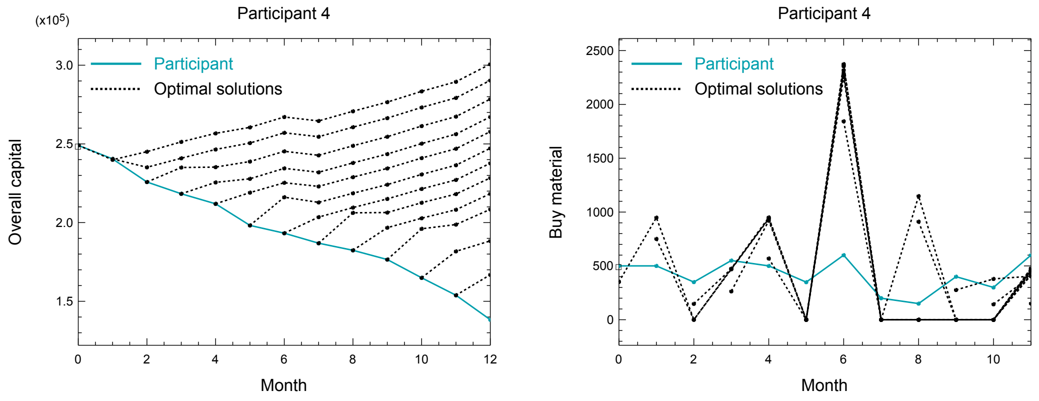


Figure 8.4: Left: state variable overall capital balance x_k^{OB}. The participant's trajectory in solid, the optimal solutions of (8.1) starting at different months n_{s} in dotted lines. The function is almost monotonically increasing, which is due to the number of vans being fixed to the participant's decision. The purchase of raw material is the main reason for the kink at month 6. Right: Purchase of raw material. Because of the comparatively low price in month 6, compare Table 8.5, a large part of the material that is needed for the months 7–12 is bought. Because the participant herself/himself did not do this, an additional kink at month 8 occurs for optimal solutions with $n_{\mathrm{s}} = 7, 8$. This is qualitatively similar for almost all data sets.

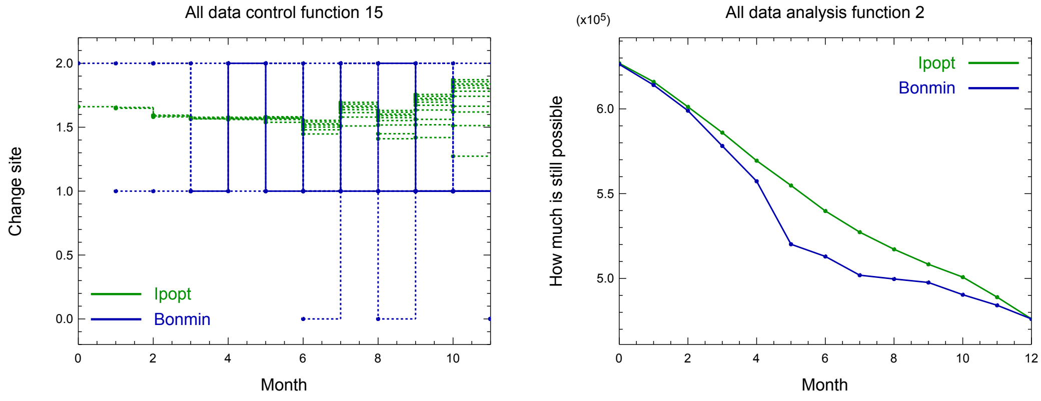


Figure 8.5: Left: optimal choices of site for one participant and all start months n_{s}, calculated with *Ipopt* (green, relaxed values between 1.1 and 1.9) and *Bonmin* (blue, integer values of 0, 1, and 2). Right: *How much is still possible*–function for one participant, calculated with *Ipopt* (green, upper curve) and *Bonmin* (blue, lower curve). As in this figure, the integer gap seems to be largest for intermediate values of n_{s} for most instances, compare also the average values in Table 8.2. However, this interpretation is subject to the fact that all solutions are only locally optimal.

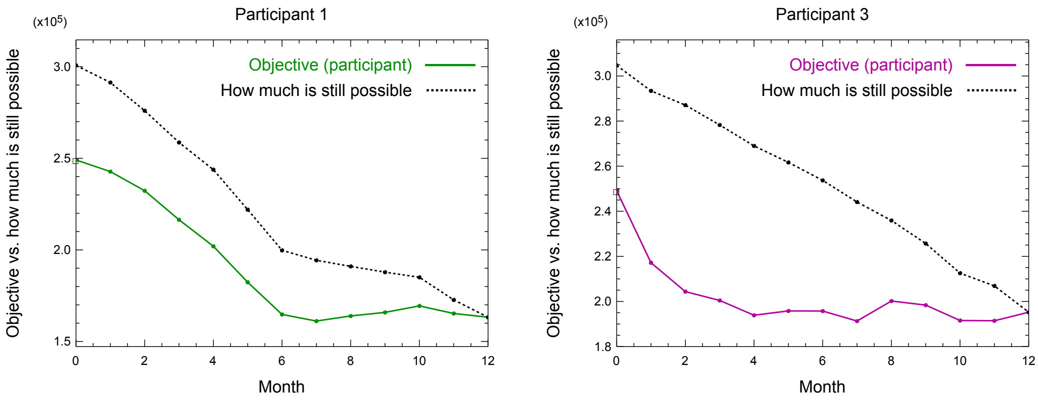


Figure 8.6: Different ways of determining good and bad participant–performance over time. The solid lines show the evolution of the objective function. The dotted lines show the *How much is still possible*–function which is composed of objective function values of separate optimization problems (8.1). The traditional way is to compare the changes in the objective function value. In our approach we compare the slopes of the *How much is still possible*-function. Left participant: the two variants would qualitatively coincide: not so good from 0–6, good performance from 7–10, not so good again from 11–12. In the right scenario the objective function values seem to correspond to alternations in the quality of the performance, which can not be verified by analyzing the *How much is still possible*–function which has an almost constant negative slope.

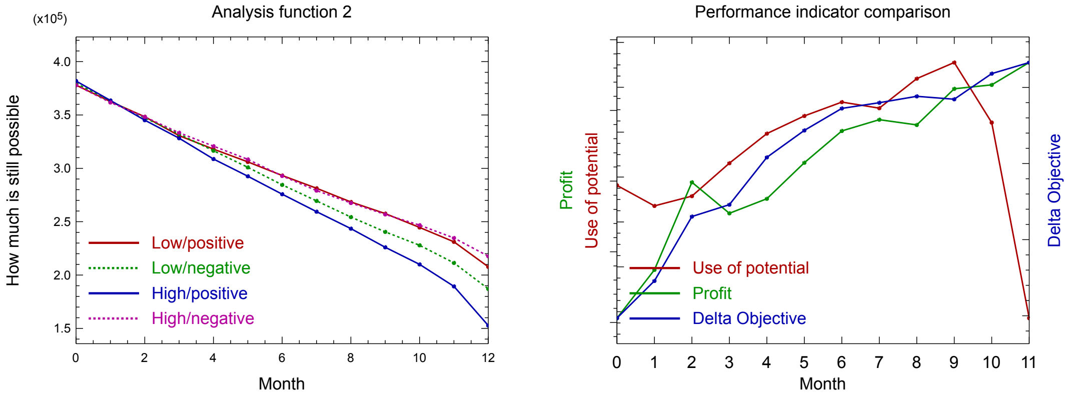


Figure 8.7: Left: average values of the *How much is still possible*–function over all participants with emotion regulation properties/feedback a) low/positive, b) low/negative, c) high/positive, d) high/negative. Participants with a low ability of emotion regulation performed better with positive feedback, those with high ability of emotion regulation performed better with negative feedback. Right: average values over all 174 participants for different indicator functions. The Use of Potential–function ΔP_k is given by (8.36). Profit indicates $\Delta x_k^{CA} = x_{k+1}^{CA} - x_k^{CA}$, Delta Objective indicates $\Delta x_k^{OB} = x_{k+1}^{OB} - x_k^{OB}$. The trajectories have been rescaled for better comparability. The potential in month $n_s = 11$ (selling all of the material on stock) has not been used by the majority of participants.

```
2650 ZA=.5+((LO-850)/550)+SM/800:IF ZA>ZM THEN:ZA=ZM
2660 SK=SM*(N1+N2):KA=KA-SK
2670 X=A1:IF N1<X THEN:X=N1
2680 Y=A2:IF N2<Y THEN:Y=N2
2690 PM=X*(MA+RND*4-2)+Y*(MA*2+RND*6-3):PM=PM*(ABS(ZA)^.5)
2700 X=PM:IF RL<X THEN:X=RL
2710 PA=X:HL=HL+PA:RL=RL-PA:KA=KA-(PA*1)-(RL*.5)
2720 NA=(NA/2+280)*1.25*2.7181^(-(PH^2)/4250):KA=KA-HL
2730 X=NA:IF HL<X THEN:X=HL
2740 VH=X:HL=HL-VH:KA=KA+VH*PH
2750 KA=KA-WE
2760 X1=WE/5:IF X1>NM THEN:X1=NM
2770 KA=KA-LW*500:X1=X1+LW*100
2780 KA=KA-GL*2000
2790 X=0:IF GL=.5 THEN:X=.1:ELSE IF GL=1 THEN:X=.2
2800 X1=X1+X1*X
2810 NA=X1+(RND*100-50)
2820 RP=2+(RND*6.5)
2830 MA=MA-.1*MA+(RS/(A1+A2*1E-08))*.017
2840 IF MA>MM THEN:MA=MM
2850 KA=KA-RS
2860 KA=KA-(N1+N2)*LO
2870 IF KA>0 THEN:KA=KA+KA*GZ:ELSE KA=KA+KA*SZ
```

Figure 8.8: Extract of the original GW-Basic code of the *Tailorshop* example which is the basis of the mathematical optimization problem. Special care is necessary to separate already updated variables x_{k+1} from the values x_k, compare the role of $x_k^{MS} \approx RL$ and $x_k^{PP} \approx PM$ in lines 2690 to 2710.

9 On Sampling Decisions in Optimum Experimental Design

The contents of this chapter are based on the paper

[206] S. Sager. Sampling Decisions in Optimum Experimental Design in the Light of Pontryagins Maximum Principle. *SIAM Journal on Control and Optimization*, submitted.

Chapter Summary. Optimum Experimental Design (OED) problems are optimization problems in which an experimental setting and decisions on when to measure – the so-called sampling design – are to be determined such that a follow-up parameter estimation yields accurate results for model parameters. We use the interpretation of OED as optimal control problems with a very particular structure for the analysis of optimal sampling decisions.

We introduce the information gain function, motivated by an analysis of necessary conditions of optimality. We highlight differences between problem formulations and propose to use a linear penalization of sampling decisions to overcome the intrinsic ill-conditioning of OED. The theoretic insight is illustrated by means of two numerical examples.

From a more abstract level, we shed additional light on an important subclass of mixed-integer optimal control problems.

9.1 Introduction

Modeling, simulation and optimization has become an indispensable tool in science, complementary to theory and experiment. It builds on detailed mathematical models that are able to represent real world behavior of complex processes. In addition to correct equations problem specific *model parameters*, such as masses, reaction velocities, or mortality rates, need to be estimated. The methodology optimum experimental design (OED) helps to design experiments that yield as much information on these model parameters as possible.

OED has a long tradition in statistics and practice, compare the textbook [196]. References to some algorithmic approaches are given, e.g., in [15, 216]. Algorithms for OED of nonlinear dynamic processes are usually based on the works of [25, 154, 155]. As investigated in [158], derivative based optimization strategies are the state-of-the-art. The methodology has been extended in [156] to cope with the need for robust designs. In [157] a reformulation is proposed that allows an application of Bock's direct multiple shooting method. An overview of model-based design of experiments can be found in [94]. Applications of OED to process engineering are given in [20, 219].

OED of dynamic processes is a non-standard optimal control problem in the sense that the objective function is a function of either the Fisher information matrix, or of its inverse, the variance-covariance matrix. The Fisher matrix can be formulated as the time integral over derivative information. This results in an optimal control problem with a very specific structure. We analyze this structure to shed light on the question under which circumstances it is optimal to measure.

Notation. When analyzing OED problems with the maximum principle, one encounters one notational challenge. We have an objective function that is a function of a matrix, however the necessary conditions are usually formulated for vector-valued variables. We have two options: either we redefine matrix operations as the inverse, trace or determinant for vectors, or we need to interpret matrices as vectors and define a scalar product for matrix-valued variables that allows to multiply them with Lagrange multipliers and obtain a map to the real numbers. We decided to use the second option. In the interest of better readability we use bold symbols for all matrices. Inequalities and equalities are always meant to hold componentwise, also for matrices.

Definition 9.1.1. (Scalar Product of Matrices)
The map $\langle\cdot,\cdot\rangle : (\lambda, A) \mapsto \ \langle\lambda, A\rangle\ \in \mathbb{R}$ with two matrices λ and $A \in \mathbb{R}^{m\times n}$ is defined as

$$\langle\lambda, A\rangle \ = \ \sum_{i=1}^{m}\sum_{j=1}^{n} \lambda_{i,j}\, A_{i,j}.$$

Partial derivatives are often written as subscripts, e.g. $\mathscr{H}_\lambda = \frac{\partial\mathscr{H}}{\partial\lambda}$. In our analysis we encounter the necessity to calculate directional derivatives of matrix functions with respect to matrices. In order to conveniently write them, we define a map analogously to the case in $\mathbb{R}^n$.

Definition 9.1.2. (Matrix-valued Directional Derivatives)
Let a differentiable map $\Phi : \mathbb{R}^{n\times n} \mapsto \mathbb{R}^{n\times n}$ be given, and let $A, \Delta A \in \mathbb{R}^{n\times n}$. Then the directional derivative is denoted by

$$\left(\frac{\partial\Phi(A)}{\partial A}\cdot\Delta A\right)_{k,l} := \sum_{i=1}^{m}\sum_{j=1}^{n} \frac{\partial\Phi(A)_{k,l}}{\partial A_{i,j}}\,\Delta A_{i,j} = \lim_{h\to 0}\frac{\Phi(A+h\Delta A)_{k,l}-\Phi(A)_{k,l}}{h}$$

for $1 \le k,l \le n$, hence $\frac{\partial\Phi(A)}{\partial A}\cdot\Delta A \in \mathbb{R}^{n\times n}$.
Let a differentiable map $\phi : \mathbb{R}^{n\times n} \mapsto \mathbb{R}$ be given, and let $A, \Delta A \in \mathbb{R}^{n\times n}$. Then the directional derivative $\lim_{h\to 0}\frac{\phi(A+h\Delta A)-\phi(A)}{h}$ is denoted by

$$\left\langle\frac{\partial\phi(A)}{\partial A}, \Delta A\right\rangle \ = \ \frac{\partial\phi(A)}{\partial A}\cdot\Delta A \ := \ \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{\partial\phi(A)}{\partial A_{i,j}}\,\Delta A_{i,j},$$

hence $\left\langle \frac{\partial \phi(A)}{\partial A}, \Delta A \right\rangle = \frac{\partial \phi(A)}{\partial A} \cdot \Delta A \in \mathbb{R}$.

In the following we use the map $\Phi(\cdot)$ for the inverse operation, and the map $\phi(\cdot)$ for either trace, determinant, or maximum eigenvalue function.

Outline. This chapter is organized as follows. In Section 9.2 we revise results from optimal control theory. In Section 9.3 we formulate the OED problem as an optimal control problem. In Section 9.4 we apply the integer gap theorem to show that there is always an ε-optimal solution with integer measurements, if the measurement grid is fine enough. We apply the maximum principle to OED in Section 9.5, and derive conclusions from our analysis. Two numerical examples are presented in Section 9.6, before we summarize in Section 9.7. Useful lemmata are provided for convenience in the Appendix.

9.2 Indirect approach to optimal control

In this section we will generalize the maximum principle from Section 2.3 to better suit our needs.

The basic idea of indirect approaches is *first optimize, then discretize*. In other words, first necessary conditions for optimality are applied to the optimization problem in function space, and in a second step the resulting boundary value problem is solved by an adequate discretization, such as multiple shooting. The necessary conditions for optimality are given by the famous maximum principle of Pontryagin. Assume we want to solve the optimal control problem of Bolza type

$$\begin{array}{llll} \min\limits_{y,u} & \multicolumn{3}{l}{\Phi(y(t_\mathrm{f})) + \int_{\mathscr{T}} L(y(\tau),u(\tau))\, \mathrm{d}\tau} \\ \multicolumn{4}{l}{\text{subject to}} \\ & \dot{y}(t) = & f(y(t),u(t),p), & t \in \mathscr{T}, \\ & u(t) \in & \mathscr{U}, & t \in \mathscr{T}, \\ & 0 \leq & c(y(t_\mathrm{f})), & \\ & y(0) = & y_0, & \end{array} \tag{9.1}$$

on a fixed time horizon $\mathscr{T} = [0,t_\mathrm{f}]$ with differential states $y : \mathscr{T} \mapsto \mathbb{R}^{n_y}$, fixed model parameters $p \in \mathbb{R}^{n_p}$, a bounded feasible set $\mathscr{U} \in \mathbb{R}^{n_u}$ for the control functions $u : \mathscr{T} \mapsto \mathbb{R}^{n_u}$ and sufficiently smooth functions $\Phi(\cdot), L(\cdot), f(\cdot), c(\cdot)$. To state the maximum principle we need the concept of the Hamiltonian.

Definition 9.2.1. (Hamiltonian, Adjoint states, End-point Lagrangian)
The Hamiltonian of optimal control problem (9.1) is given by

$$\mathscr{H}(y(t),u(t),\lambda(t),p) \;:=\; -L(x(t),u(t)) + \lambda(t)^T f(y(t),u(t),p) \tag{9.2}$$

with variables $\lambda : \mathscr{T} \mapsto \mathbb{R}^{n_y}$ *called adjoint variables. The end–point Lagrangian function* ψ *is* *defined as*

$$\psi(y(t_f),\mu) := \Phi(y(t_f)) - \mu^T c(y(t_f)) \tag{9.3}$$

with non-negative Lagrange multipliers $\mu \in \mathbb{R}^{n_c}_+$.

The *maximum principle* in its basic form, also sometimes referred to as *minimum principle*, goes back to the early fifties and the works of Hestenes, Boltyanskii, Gamkrelidze, and of course Pontryagin. Although we refer to it as maximum principle for historic reasons, we chose to use a formulation with a minimization term which is more standard in the optimization community. Precursors of the maximum principle as well as of the Bellman equation can already be found in Carathéodory's book of 1935, compare [189] for details.

The maximum principle states the existence of adjoint variables $\lambda^*(\cdot)$ that satisfy adjoint differential equations and transversality conditions. The optimal control $u^*(\cdot)$ is characterized as an implicit function of the states and the adjoint variables — a minimizer $u^*(\cdot)$ of problem (9.1) also minimizes the Hamiltonian subject to additional constraints.

Theorem 9.2.2. (Maximum principle)
Let problem (9.1) have a feasible optimal solution $(y^*,u^*)(\cdot)$. Then there exist adjoint variables $\lambda^*(\cdot)$ and Lagrange multipliers $\mu^* \in \mathbb{R}^{n_c}_+$ such that

$$\begin{aligned}
\dot{y}^*(t) &= \mathscr{H}_\lambda(y^*(t),u^*(t),\lambda^*(t),p) = f(y^*(t),u^*(t),p), && (9.4a)\\
\dot{\lambda}^{*T}(t) &= -\mathscr{H}_y(y^*(t),u^*(t),\lambda^*(t),p), && (9.4b)\\
y^*(0) &= y_0, && (9.4c)\\
\lambda^{*T}(t_f) &= -\psi_y(y^*(t_f),\mu^*), && (9.4d)\\
u^*(t) &= \arg\min_{u\in\mathscr{U}} \mathscr{H}(y^*(t),u,\lambda^*(t),p), && (9.4e)\\
0 &\leq c(y(t_f)), && (9.4f)\\
0 &\leq \mu^*, && (9.4g)\\
0 &= \mu^{*T} c(y(t_f)). && (9.4h)
\end{aligned}$$

for $t \in \mathscr{T}$ almost everywhere.

For a proof of the maximum principle and further references see, e.g., [51, 192]. Although formulation (9.1) is not the most general formulation of an optimal control problem, it covers the experimental design optimization task as we formulate it in the next section. However, one may also be interested in the case where measurements are not performed continuously over time, but rather at discrete points in time. To include such discrete events on a given time grid, we need to extend (9.1) to

$$\begin{array}{lll} \min\limits_{y,u,w} & \multicolumn{2}{l}{\Phi(y(t_{\mathrm{f}})) + \int_{\mathscr{T}} L(y(\tau),u(\tau))\,\mathrm{d}\tau + \sum_{k=1}^{n_m} L^{\mathrm{tr}}(w_k)} \\ \text{subject to} & & \\ \quad \dot{y}(t) &= f(y(t),u(t),p), & t \in \mathscr{T}^k, \\ \quad y(t_k^+) &= f^{\mathrm{tr}}(y(t_k^-),w_k,p), & k = 1 \dots n_m, \\ \quad u(t) &\in \mathscr{U}, & t \in \mathscr{T}, \\ \quad w_k &\in \mathscr{W}, & k = 1,\dots,n_m, \\ \quad 0 &\leq c(y(t_{\mathrm{f}})), & \\ \quad y(0) &= y_0, & \end{array} \tag{9.5}$$

on fixed time horizons $\mathscr{T}^k = [t_k, t_{k+1}]$, $k = 0,\dots,n_m - 1$ with $t_0 = 0$ and $t_{n_m} = t_{\mathrm{f}}$. In addition to (9.1) we have variables $w = (w_1,\dots,w_{n_m})$ with $w_k \in \mathscr{W} \subset \mathbb{R}$, a second smooth Lagrange term function $L^{\mathrm{tr}}(\cdot)$ and a smooth transition function $f^{\mathrm{tr}}(\cdot)$ that causes jumps in some of the differential states.

The boundary value problem (9.4) needs to be modified by additional jumps in the adjoint variables, e.g., for $k = 1 \dots n_m$

$$\lambda^{*T}(t_k^+) = \lambda^{*T}(t_k^-) - \mathscr{H}_y^{\mathrm{tr}}(y^*(t_k^-), w_k^*, p, \lambda^*(t_k^+)) \tag{9.6}$$

$$w_k^* = \arg\min_{w_k \in \mathscr{W}} \mathscr{H}^{\mathrm{tr}}(y(t_k^-), w_k, p, \lambda^*(t_k^+)), \tag{9.7}$$

with the discrete time Hamiltonian

$$\mathscr{H}^{\mathrm{tr}}(y(t_k^-), w_k, p, \lambda^*(t_k^+)) := -L^{\mathrm{tr}}(w_k) + \lambda^T(t_k^+)\, f^{\mathrm{tr}}(y(t_k^-), w_k, p). \tag{9.8}$$

A derivation and examples for the discrete time maximum principle can be found, e.g., in [222].

One interesting aspect about the global maximum principle (9.4) is that the constraint $u \in \mathscr{U}$ has been transferred towards the inner minimization problem (9.4e). This is done on purpose, so no assumptions need to be made on the feasible control domain $\mathscr{U}$. The maximum principle also applies to nonconvex and disjoint sets $\mathscr{U}$, such as, e.g., $\mathscr{U} = \{0,1\}$ in mixed-integer optimal control. For a disjoint set $\mathscr{U}$ of moderate size the pointwise minimization of (9.4e) can be performed by enumeration between the different choices, implemented as switching functions that determine changes in the minimum. This approach, the *Competing Hamiltonians* approach, has to our knowledge first been successfully applied to the optimization of operation of subway trains with discrete acceleration stages in New York by [43].

In this study we are not interested in applying the maximum principle directly to the disjoint set $\mathscr{U}$, but rather to its convex hull. We are interested in the question when the solutions of the two problems coincide, and which exact problem formulations are favorable in this sense. Having analyzed problem structures with the help of the maximum principle, we switch to direct, *first-discretize then-optimize* approaches to actually solve optimum experimental design problems. Using the convex hull simplifies the usage of modern gradient-based optimization strategies.

9.3 Optimum experimental design problems

In this section we formulate the problem classes of experimental design problems we are interested in.

9.3.1 Problem Formulation: Discrete Time

We are interested in optimal parameter values for a model-measurements fit. Assuming an experimental setup is given by means of control functions $u^i(\cdot)$ and sampling decisions w^i that indicate whether a measurement is performed or not for $n_{\exp}$ experiments, we formulate this parameter estimation problem as

$$\begin{array}{lrcl} \min\limits_{x,p} & \multicolumn{3}{l}{\frac{1}{2}\sum\limits_{i=1}^{n_{\exp}}\sum\limits_{k=1}^{n_h^i}\sum\limits_{j=1}^{n_t^i} w_{k,j}^i \frac{(\eta_{k,j}^i - h_k^i(x^i(t_j^i)))^2}{\sigma_{k,j}^{i\;2}}} \\ \text{s.t.} & \dot{x}^i(t) & = & f(x^i(t), u^i(t), p), \quad t \in \mathcal{T}, \\ & x^i(0) & = & x_0^i. \end{array} \tag{9.9}$$

Here $n_{\exp}, n_h^i, n_t^i$ indicate the number of independent experiments, number of different measurement functions per experiment, and number of time points for possible measurements per experiment, respectively. The $n_{\exp} \cdot n_{\mathrm{x}}$ dimensional differential state vector $(x^i)_{(i=1,\ldots,n_{\exp})}$ with $x^i : \mathcal{T} \mapsto \mathbb{R}^{n_x}$ is evaluated on a finite time grid $\{t_j^i\}$. The states $x^i(\cdot)$ of experiment i enter the model response functions $h_k^i : \mathbb{R}^{n_x} \mapsto \mathbb{R}^{n_{h_k^i}}$. The variances are denoted by $\sigma_{k,j}^i \in \mathbb{R}$, the sampling decisions $w_{k,j}^i \in \Omega$ denote how many measurements are taken at time t_j^i. If only one measurement is possible then $\Omega = \{0,1\}$. We are also interested in the possibility of multiple measurements, then we have $\Omega = \{0,1,\ldots,w^{\max}\}$. The measurement errors leading to the measurement values $\eta_{k,j}^i$ are assumed to be random variables free of systematic errors, independent from one another, attributed with constant variances, distributed around a mean of zero, and distributed according to a common probability density function. All these assumptions lead to this special form of least squares minimization.

In the interest of a clearer presentation we neglect time-independent control values, such as initial values, consider only an unconstrained parameter estimation problem, assume we only do have one single measurement function per experiment, $n_h = n_h^i = 1$, and define all variances to be one, $\sigma_{k,j}^i = 1$. We need the following definitions.

Definition 9.3.1. (Solution of Variational Differential Equations)
The matrix-valued maps $G^i(\cdot) = \frac{\mathrm{d}x^i}{\mathrm{d}p}(\cdot) : \mathcal{T} \mapsto \mathbb{R}^{n_x \times n_p}$ *are defined as the solutions of the Varia-*

tional Differential Equations

$$\dot{G}^i(t) \quad = \quad f_x(x^i(t), u^i(t), p)G^i(t) + f_p(x^i(t), u^i(t), p), \quad G^i(0) = 0 \tag{9.10}$$

obtained from differentiating $x^i(t) = x_0^i + \int_{\mathscr{T}} f(x^i(\tau), u^i(\tau), p)\ \mathrm{d}\tau$ *with respect to time and parameters* $p \in \mathbb{R}^{n_p}$*. As they denote the dependency of differential states upon parameters, we also refer to* $G^i(\cdot)$ *as* sensitivities. *Note that throughout this chapter the ordinary differential equations are meant to hold componentwise for the matrices on both sides of the equation.*

Definition 9.3.2. (Fisher Information Matrix)
The matrix $F = F(t_f) \in \mathbb{R}^{n_p \times n_p}$ *defined by*

$$F(t_f) = \sum_{i=1}^{n_{\exp}} \sum_{j=1}^{n_t^i} w_j^i \left(h_x^i(x^i(t_j^i)) G^i(t_j^i) \right)^T h_x^i(x^i(t_j^i)) G^i(t_j^i)$$

is called (discrete) Fisher information matrix.

Definition 9.3.3. (Covariance Matrix)
The matrix $C = C(t_f) \in \mathbb{R}^{n_p \times n_p}$ *defined by*

$$C(t_f) = F^{-1}(t_f)$$

is called (discrete) covariance matrix of the unconstrained parameter estimation problem (9.9).

We assume that we have $n_{\exp}$ experiments for which we can determine control functions $u^i(\cdot)$ and sampling decisions w^i in the interest to optimize a performance index, which is related to information gain with respect to the parameter estimation problem (9.9). As formulated in the groundbreaking work of [154], the optimum experimental design task is then to optimize over $u(\cdot)$ and w. The performance index is a function $\phi(\cdot)$ of either the Fisher information matrix $F(t_\mathrm{f})$ or of it's inverse, the covariance matrix $C(t_\mathrm{f})$.

Definition 9.3.4. (Objective OED Functions)
We call

- $\phi_A^F(F(t_f)) := -\frac{1}{n_p}\ \mathrm{trace}\ (F(t_f))$ *the Fisher A-criterion,*
- $\phi_D^F(F(t_f)) := -(\det\ (F(t_f)))^{\frac{1}{n_p}}$ *the Fisher D-criterion,*
- $\phi_E^F(F(t_f)) := -\min\{\lambda : \lambda \textit{ is eigenval of } F(t_f)\}$ *the Fisher E-criterion,*
- $\phi_A^C(F(t_f)) := \frac{1}{n_p}\ \mathrm{trace}\ (F^{-1}(t_f))$ *the A-criterion,*
- $\phi_D^C(F(t_f)) := (\det\ (F^{-1}(t_f)))^{\frac{1}{n_p}}$ *the D-criterion,*

- $\phi_E^C(F(t_f)) := \max\{\lambda : \lambda \textit{ is eigenval of } F(t_f)\}$ *the E-criterion,*

and write $\phi(F(t_f))$ *for any one of them in the following. If* $\phi \in \{\phi_A^F, \phi_D^F, \phi_E^F\}$ *we speak of a* Fisher objective function*, otherwise if* $\phi \in \{\phi_A^C, \phi_D^C, \phi_E^C\}$ *of a* Covariance objective function.

Note that maximizing a function (which we want to do for the Fisher information matrix) is equivalent to minimizing its negative. Additionally there are typically constraints on state and control functions, plus restrictions on the sampling decisions, such as a maximum number of measurements per experiment.

We follow the alternative formulation of [157], in which the sensitivities $G^i(\cdot)$ and the Fisher information matrix function $F(\cdot)$ are included as states in one structured optimal control problem. The performance index $\phi(\cdot)$ then has the form of a standard Mayer type functional. The optimal control problem reads

$$
\begin{array}{lll}
\multicolumn{3}{l}{\min\limits_{x^i,G^i,F,z^i,u^i,w^i} \quad \phi(F(t_{\mathrm{f}}))} \\
\multicolumn{3}{l}{\text{subject to}} \\
\dot{x}^i(t) & = & f(x^i(t),u^i(t),p), \\
\dot{G}^i(t) & = & f_x(x^i(t),u^i(t),p)G^i(t) + f_p(x^i(t),u^i(t),p), \\
F(t^i_j) & = & F(t^i_{j-1}) + \sum\limits_{i=1}^{n_{\exp}} w^i_j \left(h^i_x(x^i(t^i_j))G^i(t^i_j)\right)^T \left(h^i_x(x^i(t^i_j))G^i(t^i_j)\right), \\
z^i(t^i_j) & = & z^i(t^i_{j-1}) + w^i_j, \\
x^i(0) & = & x_0, \\
G^i(0) & = & 0, \\
F(0) & = & 0, \\
z^i(0) & = & 0, \\
u^i(t) & \in & \mathscr{U}, \\
w^i_j & \in & \mathscr{W}, \\
0 & \leq & M^i - z^i(t_{\mathrm{f}})
\end{array}
\tag{9.11}
$$

for experiment number $i = 1 \dots n_{\exp}$, time index $j = 1 \dots n^i_t$, and $t \in \mathscr{T}$ almost everywhere. Note that the Fisher information matrix $F(t_{\mathrm{f}})$ is calculated as a discrete time state, just as the measurement counters $z^i(\cdot)$. The values $M^i \in \mathbb{R}$ give an upper bound on the possible number of measurements per experiment. Of course also other problem formulations, e.g., a penalization of measurements via costs in the objective function, are possible. In our study we exemplarily treat the case of an explicitly given upper bound.

The set $\mathscr{W}$ is either $\mathscr{W} = \Omega$ or its convex hull $\mathscr{W} = \operatorname{conv} \Omega$, i.e., either $\mathscr{W} = \{0,\dots,w^{\max}\}$ or $\mathscr{W} = [0, w^{\max}]$. In the first setting we refer to (9.11) as a mixed-integer optimal control problem

(MIOCP). In the second case we use the term *relaxed* optimal control problem. It is our main aim to shed more light on the question under which circumstances the optimal solution of the relaxed problem (which is the outcome of most numerical approaches) is identical to the one of the MIOCP.

9.3.2 Problem Formulation: Continuous Measurements

It is interesting to also look at the case in which measurements are not performed at a single point in time, but over a whole interval. The continuous data flow would result in a slightly modified parameter estimation problem

$$\begin{array}{lrcl} \min\limits_{x,p} & \multicolumn{3}{l}{\frac{1}{2}\sum\limits_{i=1}^{n_{\text{exp}}}\int\limits_{0}^{t_{\text{f}}} w^i(t)\cdot\frac{(\eta^i(t)-h^i(x^i(t)))^2}{\sigma^i(t)^2}\,\mathrm{d}t} \\ \text{s.t.} & \dot{x}^i(t) & = & f(x^i(t),u^i(t),p), \quad t\in\mathscr{T}, \\ & x^i(0) & = & x_0^i. \end{array} \tag{9.12}$$

This results in a modified definition of the Fisher information matrix.

Definition 9.3.5. (Fisher Information Matrix)
The matrix $F = F(t_f) \in \mathbb{R}^{n_p\times n_p}$ defined by

$$F(t_f) = \sum_{i=1}^{n_{\text{exp}}}\int\limits_0^{t_f} w^i(t)\left(h_x^i(x^i(t))G^i(t)\right)^T h_x^i(x^i(t))G^i(t)\,\mathrm{d}t$$

is called (continuous) Fisher information matrix.

All other definitions from Section 9.3.1 are identical. This allows us to formulate the optimum experimental design problem as

$$\begin{array}{lcl} \min\limits_{x^i,G^i,F,z^i,u^i,w^i} & & \phi(F(t_{\text{f}})) \\ \text{subject to} & & \\ \dot{x}^i(t) & = & f(x^i(t),u^i(t),p), \\ \dot{G}^i(t) & = & f_x(x^i(t),u^i(t),p)G^i(t)+f_p(x^i(t),u^i(t),p), \\ \dot{F}(t) & = & \sum\limits_{i=1}^{n_{\text{exp}}} w^i(t)\left(h_x^i(x^i(t))G^i(t)\right)^T\left(h_x^i(x^i(t))G^i(t)\right), \\ \dot{z}^i(t) & = & w^i(t), \\ x^i(0) & = & x_0, \quad G^i(0)=0, \quad F(0)=0, \quad z^i(0)=0, \\ u^i(t) & \in & \mathscr{U}, \quad w^i(t)\in\mathscr{W}, \\ 0 & \leq & M^i - z^i(t_{\text{f}}). \end{array} \tag{9.13}$$

Comparing (9.13) to the formulation (9.11) with measurements on the discrete time grid, one observes that now the states $F(\cdot)$ and $z^i(\cdot)$ are specified by means of ordinary differential equations instead of difference equations, and the finite-dimensional control vector w now is a time-dependent integer control function $w(\cdot)$.

The two formulations have the advantage that they are separable, and hence accessible for the direct multiple shooting method, [157]. In addition, they fall into the general optimal control formulations (9.5) and (9.1), respectively, and allow for an application of the maximum principle.

9.4 Applying the integer gap lemma to OED

A first immediate advantage of the formulation (9.13) as a continuous optimal control problem is that we can apply the integer gap lemma proposed in Chapter 3. In the interest of an easier presentation let us assume $w^{\max} = 1$.

Corollary 9.4.1. (Integer Gap)
Let $(x^{i*}, G^{i^*}, F^*, z^{i*}, u^{i*}, \alpha^{i*})(\cdot)$ *be a feasible trajectory of the relaxed problem* (9.13) *with the measurable functions* $\alpha^{i*} : [0,t_f] \to [0,1]$ *replacing* $w^i(\cdot)$ *in problem* (9.13)*, with* $i = 1 \dots n_{\text{exp}}$*.*
Consider the trajectory $(x^{i*}, G^{i^*}, F^{SUR}, z^{i,SUR}, u^{i*}, \omega^{i,SUR})(\cdot)$ *which consists of controls* $\omega^{i,SUR}(\cdot)$ *determined via Sum Up Rounding (3.7-3.8) on a given time grid from* $\alpha^{i*}(\cdot)$ *and differential states* $(F^{SUR}, z^{i,SUR})(\cdot)$ *that are obtained by solving the initial value problems in* (9.13) *for the fixed differential states* $(x^{i*}, G^{i^*})(\cdot)$ *and* $\omega^{i,SUR}(\cdot)$*.*
Then for any $\delta > 0$ *there exists a grid size* Δt *such that*

$$|z^{i,SUR}(t_f) - z^{i*}(t_f)| \quad \leq \quad \delta, \quad i = 1, \dots, n_{\text{exp}}. \tag{9.14}$$

Assume in addition that constants $C, M \in \mathbb{R}^+$ *exist such that the functions*

$$\hat{f}^i(x^{i*}, G^{i^*}) \quad := \quad \left(h_x^i(x^i(t))G^i(t)\right)^T \left(h_x^i(x^i(t))G^i(t)\right)$$

are differentiable with respect to time and it holds

$$\left\| \frac{\mathrm{d}}{\mathrm{d}t} \hat{f}^i(x^{i*}, G^{i^*}) \right\| \quad \leq \quad C$$

for all $i = 1 \dots n_{\text{exp}}$*,* $t \in [0, t_f]$ *almost everywhere and* $\hat{f}^i(x^{i*}, G^{i^*})$ *are essentially bounded by* M*.*
Then for any $\delta > 0$ *there exists a grid size* Δt *such that*

$$|\phi(F^{SUR}(t_f)) - \phi(F^*(t_f))| \quad \leq \quad \delta. \tag{9.15}$$

Proof. Follows from Corollary 3.5.3 on page 44 and the fact that all assumptions on the right hand side function are fulfilled. Note that the condition on the Lipschitz constant is automatically fulfilled, because $z(\cdot)$ and $F(\cdot)$ do not enter in the right hand side of the differential equations.□

Corollary 9.4.1 implies that the exact lower bound of the OED problem (9.13) can be obtained by solving the relaxed problem in which $w^i(t) \in \text{conv}\ \Omega$ instead of $w^i(t) \in \Omega$. In other words, anything that can be done with fractional sampling can also be done with an integer number of measurements. However, the price might be a so-called chattering behavior, i.e., frequent switching between yes and no.

9.5 Analyzing relaxed sampling decisions

An observation in practice is that the relaxed samplings $w^i(t) \in \text{conv}\ \Omega$ are almost always $w^i(t) \in \Omega$. To get a better understanding of what is going on, we apply the maximum principle from Theorem (9.2.2). We proceed with the continuous case of the control problem (9.13). The vector of differential states of the general problem (9.1) is then given by

$$y(\cdot) = \begin{pmatrix} x^i(\cdot) \\ G^i(\cdot) \\ F(\cdot) \\ z^i(\cdot) \end{pmatrix}_{(i=1\dots n_{\exp})}$$

with $i = 1 \dots n_{\exp}$. Hence $y(\cdot)$ is a map $y : \mathscr{T} \mapsto \mathbb{R}^{n_y}$ with dimension

$$n_y = n_{\exp} n_x + n_{\exp} n_x n_p + n_p n_p + n_{\exp}.$$

Note that some components of this vector are matrices that need to be "flattened" in order to write y as a vector. We define the right hand side function

$$\tilde{f} : \mathbb{R}^{n_y \times n_{\exp} n_u \times n_{\exp} \times n_p} \mapsto \mathbb{R}^{n_y}$$

as

$$\tilde{f}(y(t), u(t), w(t), p) := \begin{pmatrix} f(x^i(t), u^i(t), p) \\ f_x(x^i(t), u^i(t), p) G^i(t) + f_p(x^i(t), u^i(t), p) \\ \sum\limits_{i=1}^{n_{\exp}} w^i(t) \left(h_x^i(x^i(t)) G^i(t)\right)^T \left(h_x^i(x^i(t)) G^i(t)\right) \\ w^i(t) \end{pmatrix} \tag{9.16}$$

again with multiple entries for all $i = 1 \dots n_{\exp}$. We define $\lambda_{x^i}, \lambda_{G^i}, \lambda_F, \lambda_{z^i}$ to be corresponding adjoint variables with dimensions n_x, $n_x \times n_p$, $n_p \times n_p$, and 1, respectively, and λ as the compound of these variables. Note that λ_{G^i} and λ_F are treated as matrices, just like their associated states G^i and F. The Hamiltonian is then given as

$$\begin{aligned} \mathscr{H}(y(t),u(t),w(t),\lambda(t),p) = {} & \left\langle \lambda(t), \tilde{f}(y(t),u(t),w(t),p) \right\rangle \\ = {} & \sum_{i=1}^{n_{\text{exp}}} \lambda_{x^i}^T f^i(\cdot) + \sum_{i=1}^{n_{\text{exp}}} \left\langle \lambda_{G^i}, f_x^i(\cdot) G^i + f_p^i(\cdot) \right\rangle \\ & + \left\langle \lambda_F, \sum_{i=1}^{n_{\text{exp}}} w^i \left(h_x^i(\cdot) G^i \right)^T \left(h_x^i(\cdot) G^i \right) \right\rangle + \sum_{i=1}^{n_{\text{exp}}} \lambda_{z^i} w^i, \end{aligned} \tag{9.17}$$

where we are leaving away the time arguments (t) and argument lists of f and h. Note that Definition 9.1.1 of the scalar product allows to use the matrices $\lambda_{G^i} \in \mathbb{R}^{n_x \times n_p}$ and $\lambda_F \in \mathbb{R}^{n_p \times n_p}$ in a straight-forward way.

Corollary 9.5.1. (Maximum principle for OED problems)
Let problem (9.13) *have a feasible optimal solution* (y^*, u^*, w^*). *Then there exist adjoint variables* $\lambda^*(\cdot)$ *and Lagrange multipliers* $\mu^* \in \mathbb{R}^{n_{\text{exp}}}$ *such that for* $t \in \mathscr{T}$ *it holds almost everywhere*

$$\dot{y}^*(t) = \tilde{f}(y^*(t), u^*(t), w^*(t), p), \tag{9.18a}$$

$$\begin{aligned} \dot{\lambda}_{x^i}^{*T}(t) = {} & \lambda_{x^i}^T f_x^i(\cdot) + \frac{\partial}{\partial x^i} \left(\left\langle \lambda_{G^i}, f_x^i(\cdot) G^i + f_p^i(\cdot) \right\rangle \right)^T \\ & + \frac{\partial}{\partial x^i} \left(\left\langle \lambda_F, w^i \left(h_x^i(\cdot) G^i \right)^T \left(h_x^i(\cdot) G^i \right) \right\rangle \right)^T, \end{aligned} \tag{9.18b}$$

$$\dot{\lambda}_{G^i}^{*T}(t) = \left\langle \lambda_{G^i}, f_x^i(\cdot) \right\rangle + \frac{\partial}{\partial G^i} \left(w^i \left\langle \lambda_F, \left(h_x^i(\cdot) G^i \right)^T \left(h_x^i(\cdot) G^i \right) \right\rangle \right)^T, \tag{9.18c}$$

$$\dot{\lambda}_F^{*T}(t) = 0, \tag{9.18d}$$

$$\dot{\lambda}_{z^i}^{*T}(t) = 0, \tag{9.18e}$$

$$y^*(0) = y_0, \tag{9.18f}$$

$$\lambda_{x^i}^{*T}(t_f) = 0, \tag{9.18g}$$

$$\lambda_{G^i}^{*T}(t_f) = 0, \tag{9.18h}$$

$$\lambda_F^{*T}(t_f) = -\frac{\partial \phi(F(t_f))}{\partial F}, \tag{9.18i}$$

$$\lambda_{z^i}^{*T}(t_f) = -\frac{-\partial \mu_i^*(M^i - z^i(t_f))}{\partial z} = -\mu_i^*, \tag{9.18j}$$

$$(u^*, w^*)(t) = \arg \min_{u \in \mathscr{U}^{n_{\text{exp}}}, w \in \mathscr{W}^{n_{\text{exp}}}} \mathscr{H}(y^*(t), u, w, \lambda^*(t), p), \tag{9.18k}$$

$$0 \leq M - z(t_f), \tag{9.18l}$$

$$0 \leq \mu^*, \tag{9.18m}$$

$$0 = \mu^{*T}(M - z(t_f)) \tag{9.18n}$$

with $i = 1 \ldots n_{\text{exp}}$ *and* $y, \lambda, \tilde{f}$ *defined as above.*

Proof. Follows directly from applying the maximum principle (9.4) to the control problem (9.13) and taking the partial derivatives of the Hamiltonian (9.17) and the objective function $\phi(\cdot)$ of the OED control problem with respect to the state variables $x^i(\cdot), G^i(\cdot), F(\cdot)$ and $z^i(\cdot)$. □

This corollary serves as a basis for further analysis. A closer look at (9.18k) and the Hamiltonian reveals structure.

Corollary 9.5.2. *The Hamiltonian $\mathscr{H}$ decouples with respect to $u^i(\cdot)$ and $w^i(\cdot)$ for all experiments $i = 1 \dots n_{\text{exp}}$. Hence the optimal controls $u^{i*}(\cdot)$ and $w^{i*}(\cdot)$ can be determined independently from one another for given states $y^*(\cdot)$, adjoints $\lambda^*(\cdot)$ and parameters p.*

Proof. Follows directly from Equation (9.17) and the fact that $f^i(\cdot)$ and the partial derivatives $f^i_x(\cdot)$ and $f^i_p(\cdot)$ do not depend on the sampling functions $w^i(\cdot)$.
Let $\tilde{w}^T = (w^{1,*^T}(t), \dots, w^{i-1,*^T}(t), w^{i^T}, w^{i+1,*^T}(t), \dots, w^{n_{\text{exp}},*^T})(t)$, then

$$\begin{aligned} w^{i*}(t) &= \arg\min\nolimits_{w^i \in \mathscr{W}} \mathscr{H}(y^*(t), u^*(t), \tilde{w}, \lambda^*(t), p) \\ &= \arg\min\nolimits_{w^i \in \mathscr{W}} \left\langle {\lambda_F}^*, w^i \left(h^i_x(\cdot) G^{i*}\right)^T \left(h^i_x(\cdot) G^{i*}\right) \right\rangle + \lambda^*_{z^i} w^i. \end{aligned} \tag{9.19}$$

Likewise, the experimental controls $u^{i*}(\cdot)$ are given as

$$\begin{aligned} u^{i*}(t) &= \arg\min\nolimits_{u^i \in \mathscr{U}} \mathscr{H}(y^*(t), \tilde{u}, w^*(t), \lambda^*(t), p) \\ &= \arg\min\nolimits_{u^i \in \mathscr{U}} \lambda^{*T}_{x^i} f^i(\cdot) + \left\langle {\lambda_{G^i}}^*, f^i_x(\cdot) G^{i*} + f^i_p(\cdot) \right\rangle \end{aligned} \tag{9.20}$$

because the measurement function $h(\cdot)$ and its partial derivative do not depend explicitly on $u(\cdot)$. □

We would like to stress that the decoupling of the control functions holds only in the sense of necessary conditions of optimality, and for given optimal states and adjoints. Clearly they may influence one another indirectly. We come back to this issue in Section 9.5.1.
A closer look at equation (9.19) reveals that the sampling control function $w(\cdot)$ enters linearly into the Hamiltonian. This implies that the sign of the switching function determines whether $w(\cdot) \in [0, w^{\max}]$ is at its lower or upper bound, which corresponds in our case to integer feasibility, $w(\cdot) \in \{0, w^{\max}\}$.

Definition 9.5.3. (Local and Global Information Gain)
The matrix $P^i(t) \in \mathbb{R}^{n_p \times n_p}$

$$P^i(t) := P(x^i(t), G^i(t)) := \left(h^i_x(x^i(t)) G^i(t)\right)^T \left(h^i_x(x^i(t)) G^i(t)\right)$$

is called local information gain *matrix of experiment i. Note that $P^i(t)$ is positive semi-definite, and positive definite if the matrix $h^i_x(x^i(t)) G^i(t)$ has full rank n_p.*

If ${F^*}^{-1}(t_f)$ *exists, we call*

$$\Pi^i(t) := {F^*}^{-1}(t_f) P^i(t) {F^*}^{-1}(t_f) \in \mathbb{R}^{n_p \times n_p}$$

the global information gain *matrix.*

We use Corollary 9.5.2 as a justification to concentrate our analysis on the case of a single experiment. Hence we leave the superscript i away for notational convenience, assuming $n_{\text{exp}} = 1$, and come back to the multi experiment case in Section 9.5.2.

Definition 9.5.4. (Switching function)
The derivative of the Hamiltonian (9.17)

$$\mathscr{H}_w(t) := \frac{\partial \mathscr{H}(\cdot)}{\partial w} \;=\; \langle \lambda_F(t), P(t) \rangle + \lambda_z(t)$$

is called switching function with respect to $w(\cdot)$*. The derivative*

$$\mathscr{H}_u(t) := \frac{\partial \mathscr{H}(\cdot)}{\partial u} = \frac{\partial}{\partial u} \left(\lambda_x^{*T} f(\cdot) + \left\langle {\lambda_G}^*, f_x(\cdot) G^{i^*} + f_p(\cdot) \right\rangle \right)$$

is called switching function with respect to $u(\cdot)$*.*

We are now set to investigate the conditions for either measuring or not at a time t for different objective functions. From now on we assume that $(y^*, u^*, w^*, \lambda^*, \mu^*)(\cdot)$ is an optimal trajectory of the relaxed optimal control problem (9.13) with $n_{\text{exp}} = 1$ and $\mathscr{W} = [0, w^{\max}]$, and hence a solution of the boundary value problem (9.18).

Lemma 9.5.5. (Maximize trace of Fisher matrix)
Let $\phi(F(t_f)) = \phi_A^F(F(t_f)) = -\text{trace}(F(t_f))$ *be the objective function of the OED problem* (9.13)*, and let* $w^*(\cdot)$ *be an optimal control function. If*

$$\text{trace}\,(P(t)) \;\;>\;\; \mu^*$$

for $t \in (0, t_f)$*, then there exists a* $\delta > 0$ *such that* $w^*(t) = w^{max}$ *almost everywhere on* $[t - \delta, t + \delta]$*.*

Proof. As $w^*(t)$ is the pointwise minimizer of the Hamiltonian and according to Corollary 9.5.2 it decouples from the other control functions, and as it enters linearly, it is at its upper bound of $w^{\max}$ whenever the sign of the switching function is positive. The switching function is given by

$$\mathscr{H}_w(t) \;\;=\;\; \langle \lambda_F^*(t), P(t) \rangle + \lambda_z^*(t).$$

With Corollary 9.5.1 we have

$$\begin{aligned}\mathscr{H}_w(t) &= \left\langle -\frac{\partial \phi(F(t_\mathrm{f}))}{\partial F}, P(t) \right\rangle - \mu^* \\ &= \left\langle -\frac{\partial - \mathrm{trace}(F(t_\mathrm{f}))}{\partial F}, P(t) \right\rangle - \mu^*.\end{aligned}$$

Applying Lemma 9.8.2 from the Appendix we obtain

$$\mathscr{H}_w(t) = \mathrm{trace}\,(P(t)) - \mu^*.$$

As trace $(P(t))$ is differentiable with respect to time, there exists a time interval of positive measure around t where this expression is also positive, which concludes the proof. □

Lemma 9.5.6. (Minimize trace of Covariance matrix)
For the assumptions of Lemma 9.5.5, but the objective function

$$\phi(F(t_f)) = \phi_C^F(F(t_f)) = \mathrm{trace}(C(t_f)),$$

the sufficient condition for $w^*(t) = w^{max}$ in an optimal solution is that

$$\mathrm{trace}\,(\Pi(t)) > \mu^*$$

holds.

Proof. The argument is similar to the one in Lemma 9.5.5. We have

$$\begin{aligned}\mathscr{H}_w(t) &= -\left\langle \frac{\partial \mathrm{trace}(F^{*-1}(t_\mathrm{f}))}{\partial F}, P(t) \right\rangle - \mu^* \\ &= -\left\langle \frac{\partial \mathrm{trace}(F^{*-1}(t_\mathrm{f}))}{\partial F^{-1}}, \frac{\partial F^{*-1}(t_\mathrm{f})}{\partial F} P(t) \right\rangle - \mu^*\end{aligned}$$

Note here that the expression $\frac{\partial F^{*-1}(t_\mathrm{f})}{\partial F} P(t)$ is a matrix in $\mathbb{R}^{n_p \times n_p}$ by virtue of Definition 9.1.2. Applying Lemma 9.8.2 from the Appendix we obtain

$$\mathscr{H}_w(t) = -\mathrm{trace}\left(\frac{\partial F^{*-1}(t_\mathrm{f})}{\partial F} P(t) \right) - \mu^*$$

To evaluate the directional derivative of the inverse operation we apply Lemma 9.8.3 and obtain

$$\mathscr{H}_w(t) = \mathrm{trace}\left(F^{*-1}(t_\mathrm{f}) P(t) F^{*-1}(t_\mathrm{f}) \right) - \mu^*$$

which concludes the proof, as $\Pi(t) = F^{*-1}(t_\mathrm{f}) P(t) F^{*-1}(t_\mathrm{f})$. □

Lemma 9.5.7. (Minimization of max eigenvalue of Covariance matrix)
For the assumptions of Lemma 9.5.5, but the objective function

$$\phi(F(t_f)) \quad = \quad \phi_E^C(F(t_f)) = \max\{\lambda : \lambda \textit{ is eigenvalue of } C(t_f))\},$$

the sufficient condition for $w^*(t) = w^{max}$ in an optimal solution is that, if $\lambda_{\max}$ is a single eigenvalue,

$$v^T \Pi(t) v > \mu^*$$

holds, where $v \in \mathbb{R}^{n_p}$ is an eigenvector of $C(t_f)$ to $\lambda_{\max}$ with norm 1.

Lemma 9.5.8. (Minimization of determinant of Covariance matrix)
For the assumptions of Lemma 9.5.5, but the objective function

$$\phi(F(t_f)) \quad = \quad \phi_D^C(F(t_f)) = \det(C(t_f)),$$

the sufficient condition for $w^*(t) = w^{max}$ in an optimal solution is that

$$\det(C^*(t_f)) \sum_{i,j=1}^{n_p} (F^*(t_f))_{i,j} \, (\Pi(t))_{i,j} > \mu^*$$

holds.

The proofs of Lemmata 9.5.7 and 9.5.8 and for other objective functions are similar to the one in Lemma 9.5.6, making use of the Appendix Lemmata 9.8.4 and 9.8.5.
The local information gain matrix $P(t)$ is positive definite, whenever the measurement function is sensitive with respect to the parameters. This attribute carries over to the matrix state $F(\cdot)$ in which $P(t)$ is integrated, to the covariance matrix function (as the inverse of a positive definite matrix is also positive definite), and to the product of positive definite matrices. The considered functions of $P(t)$ and $\Pi(t)$ are hence all positive values, compare, e.g., Lemma 9.8.1.
This implies for non-existent constraints on the number of measurements with $\mu^* = 0$ the trivial conclusion that measuring all the time with $w(t) \equiv w^{\max}$ is optimal.
In the more interesting case when the constraint $c(z^*(t_\mathrm{f})) = M - z^*(t_\mathrm{f}) \geq 0$ is active, the Lagrange multiplier μ^* indicates the threshold. The Lagrange multipliers are also called shadow prices, as they indicate how much one gains from increasing a resource. In this particular case relaxing the measurement bound M would yield the information gain μ^* in the objective function $\phi(\cdot)$.
The main difference between using the Fisher information matrix $F(\cdot)$ and the covariance matrix $C(\cdot) = F^{-1}(\cdot)$, e.g., in Lemmata 9.5.5 and 9.5.6, lies in the local $P(t)$ and global $\Pi(t) = F^{-1}(t_\mathrm{f})P(t)F^{-1}(t_\mathrm{f})$ information gain matrices that yield a sufficient criterion, respectively. The fact that the sufficient criterion for a maximization of the Fisher information matrix does not depend on the value of $F^{-1}(t_\mathrm{f})$ has an important consequence. Modifying the value of $w(t)$, e.g., by rounding, does not have any recoupling effect on the criterion itself. Therefore, whenever

$w(t) \not\in \{0, w^{\max}\}$ on different time intervals, one can round these values up and down (making sure that $\int_{\mathscr{T}} w(\tau)\,\mathrm{d}\tau$ keeps the value of M) to obtain a feasible integer solution with the same objective function value. This is *not* the case when we have a Covariance objective function, as measurable modifications of $w(t)$ have an impact on $F(t_\mathrm{f})$ and hence also on $F^{-1}(t_\mathrm{f})$ and the sufficient criterion.

The procedure for the case with finitely many measurements that enter as noncontinuous jumps in finite difference equations (9.11) is very similar to the one above, only some definitions would need to be modified. The main results are identical and we have the same criteria to validate whether the control values w_j^i are on their upper bound of $w^{\max}$ or not. The main difference is that measurements in the continuous setting average the information gain on a time interval, whereas point measurements are on the exact location of the maxima of the global information gain function.

9.5.1 Singular Arcs

As we saw above, the sampling controls $w(t)$ enter linearly into the control problem. If for control problems with linear controls the switching function is zero on an interval of positive measure, one usually proceeds by taking higher order time derivatives of the switching function to determine an explicit representation of this singular control, which may occur if at all in even degree time derivatives as shown by [142]. This approach is not successful for sampling functions in experimental design problems.

Lemma 9.5.9. (Infinite order of singular arcs)
Let $n_u = 0$. For all values $j \in \mathbb{N}$ the time derivatives $S^j := \frac{\mathrm{d}^j}{\mathrm{d}t^j}\mathscr{H}_w(t)$ never depend explicitly on $w(\cdot)$.

Proof. The switching functions above are functions of either $P(t)$ or in the case of a Covariance objective function of $F^{*-1}(t_\mathrm{f})P(t)F^{*-1}(t_\mathrm{f})$. Taking the time derivative only affects $P(t)$. We see that in

$$\begin{aligned}
\frac{\mathrm{d}P(t)}{\mathrm{d}t} &= \frac{\mathrm{d}\,(h_x(x(t))G(t))^T\,(h_x(x(t))G(t))}{\mathrm{d}t} \\
&= 2\,(h_x(x(t))G(t))^T\,\frac{\mathrm{d}\,(h_x(x(t))G(t))}{\mathrm{d}t} \\
&= 2\,(h_x(x(t))G(t))^T\,\big(h_{xx}(x(t))\dot{x}(t)G(t) + h_x(x(t))\dot{G}(t)\big) \\
&= 2\,(h_x(x(t))G(t))^T\,(h_{xx}(x(t))f(x(t),u(t),p)G(t) \\
&\quad + h_x(x(t))(f_x(x(t),u(t),p)G(t) + f_p(x(t),u(t),p)))
\end{aligned}$$

only time derivatives of $x(\cdot)$ and $G(\cdot)$ appear. Also in higher order derivatives $F(\cdot)$ and $z(\cdot)$ never enter, and as $n_u = 0$ no expressions from a singular control $u^*(\cdot)$ may appear, hence also $w(\cdot)$ never enters in any derivative. □

The assumption that $n_u = 0$ is rather strong though. It is an open and interesting question, whether one can construct non-trivial instances of OED control problems for which the joint control vector $(u,w)(\cdot)$ is a singular control. This would imply that the interplay of the singular controls results in a constant value of the global information gain matrix $\Pi(t)$ on a measurable time interval.

9.5.2 L^1 Penalization and Sparse Controls

We are interested in how changes in the formulation of the optimization problem influence the role of the global information gain functions. We first consider a L_1 penalty term in the objective function. We are going back to the multi-experiment case and use the upperscript $i = 1 \dots n_{\text{exp}}$.

Corollary 9.5.10. (Switching function for L_1 penalty)
Let $\mathscr{H}_{w^i}^{\text{old}}(\cdot)$ denote the switching function for problem (9.13) *and $\mathscr{H}_{w^i}^{\text{penalty}}(\cdot)$ the switching function with respect to $w^i(\cdot)$ for problem* (9.13) *with an objective function that is augmented by a Lagrange term,*

$$\min_{x^i, G^i, F, z^i, u^i, w^i} \phi(F(t_f)) + \int_{\mathscr{T}} \sum_{i=1}^{n_{\text{exp}}} \varepsilon^i w^i(\tau)\, \mathrm{d}\tau.$$

Then it holds

$$\mathscr{H}_{w^i}^{\text{penalty}}(t) = \mathscr{H}_{w^i}^{\text{old}}(t) - \varepsilon^i$$

Proof. By definition (9.2) of the Hamiltonian we have

$$\mathscr{H}^{\text{penalty}}(t) = \mathscr{H}^{\text{old}}(t) - \sum_{i=1}^{n_{\text{exp}}} \varepsilon^i w^i(t)$$

which already concludes the proof. □

Corollary 9.5.10 allows a direct connection between the penalization parameter ε and the information gain function. For the minimization of the trace of the covariance matrix, compare Lemma 9.5.6, this implies that a sufficient condition for $w^{i*}(t) = w^{\max}$ is

$$\operatorname{trace}\left(\Pi^i(t)\right) > \varepsilon^i + \mu^{i*}.$$

As a consequence, an optimal sampling design never performs measurements when the value of the trace of the information gain function is below the penalization parameter ε^i.
The case is similar for the time discrete OED problem (9.11). Assume we extend the objective with a penalization term

$$\sum_{i=1}^{n_{\text{exp}}} \sum_{j=1}^{n_t^i} L^{\text{tr}}(w_j^i) = \sum_{i=1}^{n_{\text{exp}}} \sum_{j=1}^{n_t^i} \varepsilon^i\, w_j^i,$$

then the derivative of the discrete Hamiltonian (9.8) with respect to the control w^i_j is again augmented by $-\varepsilon^i$.

9.5.3 L^2 Penalization and Singular Arcs

An alternative penalization is a L_2 penalization of the objective function with a Lagrange term

$$\int_{\mathscr{T}} \sum_{i=1}^{n_{\text{exp}}} \varepsilon^i \, w^i(\tau)^2 \, \mathrm{d}\tau.$$

This formulation has direct consequences. As the controls $u^i(\cdot)$ and $w^i(\cdot)$ decouple, compare Corollary 9.5.2, the optimal sampling design may be on the boundary of its domain, or can be determined via the necessary condition that the derivative of the Hamiltonian with respect to $w^i(\cdot)$ is zero, i.e.,

$$w^i(t) = \frac{1}{\varepsilon^i} \text{trace}\left(\Pi^i(t)\right) - \mu^{i*}$$

for the case of the minimization of the trace of the covariance matrix. This implies that $w^i(\cdot)$ may be a singular control with fractional values $w(t) \in (0, w^{\max})$. Hence, we discourage to use this formulation.

9.6 Numerical examples

In this section we illustrate several effects with numerical examples. Our analysis so far has been based on the so-called *first optimize, then discretize* approach. Now we solve the numerical OED problems with direct or *first discretize, then optimize* methods. In particular, we use the code MS MINTOC that has been developed for generic mixed-integer optimal control problems by the author. It is based on Bock's direct multiple shooting method, adaptive control discretizations, and switching time optimization. A comprehensive survey of how this algorithm works can be found in [214]. Note however that there are many specific structures that can, should or even have to be exploited to take into account the special structure of the OED control problems in an efficient implementation. It is beyond the scope of this chapter to go into details, instead we refer to [134, 157] for a more detailed discussion.

Having obtained an optimal solution, it is possible to evaluate the functions $\Pi^i(t)$ for an a posteriori analysis. This is what we do in the following. As we have derived an explicit formula for the switching functions $\Pi^i(t)$ in terms of primal state variables, we do not even have to use discrete approximations of the adjoint variables.

Although the algorithm has also been applied to higher-dimensional problems, such as the bimolecular catalysis benchmark problem of [154], we focus here on two small-scale academic benchmark problems, that allow us to illustrate many of the interesting features of optimal sampling designs.

9.6.1 One-dimensional academic example

We are interested in estimating the parameter $p \in \mathbb{R}$ of the initial value problem

$$\dot{x}(t) = p\,x(t),\ t \in [0, t_\mathrm{f}], \quad x(0) = x_0.$$

We assume x_0 and t_f to be fixed and are only interested in when to measure, with an upper bound M on the measuring time. We can measure the state directly, $h(x(t)) = x(t)$. The experimental design problem (9.13) then simplifies to

$$\begin{array}{lcl} \min\limits_{x,G,F,z,w} & \frac{1}{F(t_\mathrm{f})} & \\ \text{subject to} & & \\ \dot{x}(t) & = & p\,x(t), \\ \dot{G}(t) & = & p\,G(t) + x(t), \\ \dot{F}(t) & = & w(t)\,G(t)^2, \\ \dot{z}(t) & = & w(t), \\ x(0) & = & x_0,\ G(0) = F(0) = z(0) = 0, \\ w(t) & \in & \mathscr{W}, \\ 0 & \leq & M - z(t_\mathrm{f}) \end{array} \tag{9.21}$$

with $t_\mathrm{f} = 1, M = 0.2 w^{\max}$.

Although problem (9.21) is as easy as an optimum experimental design problem can be, it allows already to investigate certain phenomena that may occur. First, assume that $x_0 = 0$. This implies $\dot{x}(t) = \dot{G}(t) = 0$ for all $t \in \mathscr{T}$, and hence the degenerated case in which $G(\cdot) \equiv 0$ and the inverse of the Fisher information matrix does not even exist. If we were to maximize a function of the Fisher information matrix, the sampling design would be a singular decision, as there is no sensitivity with respect to the parameter throughout.

If we choose an initial value of $x_0 \neq 0$, this degenerated case does not occur: obviously a $0 < \tau < t_\mathrm{f}$ exists such that $\int_0^\tau x(t)\,\mathrm{d}t \neq 0$ and hence also $G(\tau) \neq 0$ and therefore $F(t_\mathrm{f}) > 0$. The global information function for (9.21) is given by

$$\Pi(t) = \frac{G(t)^2}{F(t_\mathrm{f})^2}.$$

As the matrix is one-dimensional, all considered criteria carry directly over to this expression. The switching function for (9.21) is given by $\mathscr{H}_w = \frac{G^2(t)}{F^2(t_\mathrm{f})} - \mu$. Hence it is clear that a singular arc with $\mathscr{H}_w = 0$ can only occur on an interval $[\tau_\mathrm{s}, \tau_\mathrm{e}]$ when $\dot{G}(\tau) = 0$ for $\tau \in [\tau_\mathrm{s}, \tau_\mathrm{e}]$ almost everywhere. With $\dot{G}(\tau) = pG(\tau) + x(\tau)$ this would imply that also $x(\cdot)$ is constant on $[\tau_\mathrm{s}, \tau_\mathrm{e}]$, which is impossible for $x_0 \neq 0$. Therefore problem (9.21) with $x_0 \neq 0$ always has a bang-bang

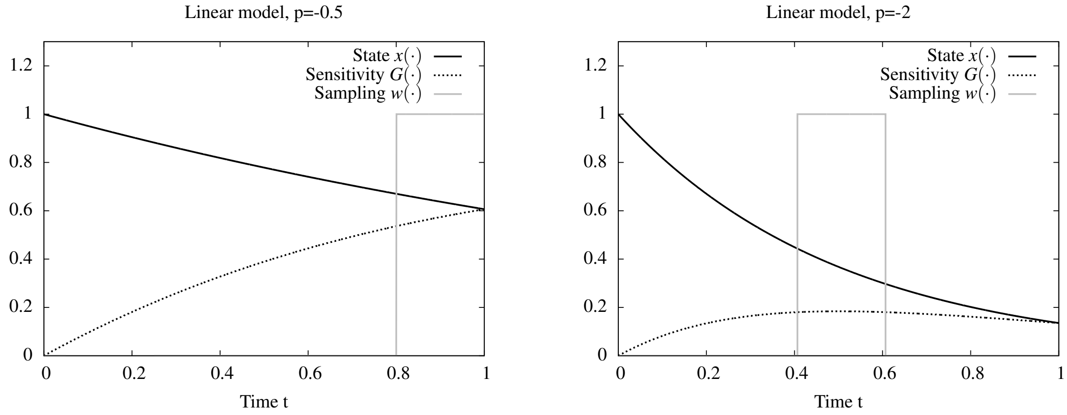


Figure 9.1: Linear optimum experimental design problem (9.21) with one state and one sampling function for different values of p. Left: $p = -0.5$, right: $p = -2$.

solution with respect to $w(\cdot)$.

We choose $x_0 = 1$ in the following. If $G(\cdot)$ happens to be a positive, monotonically increasing function on $\mathscr{T}$, then we can deduce that the optimal sampling $w(\cdot)$ is given by a $0-1$ arc, where the switching point is determined by the value of M. Such a scenario is obtained for the expected optimal parameter value of $p = -0.5$, compare Figure 9.1 left.

The switching structure depends not only on functions and initial values, but may also depend on the very value of p itself. An example with an optimal $0-1-0$ solution is depicted in Figure 9.1 right for the value of $p = -2$. Here the optimal sampling is

$$w(t) = \begin{cases} 0 & t \in [0,\tau] \cup [\tau+0.2,1] \\ w^{\max} & t \in [\tau,\tau+0.2] \end{cases} \tag{9.22}$$

Figure 9.1 also illustrates the connection between the discrete-time measurements in Section 9.3.1 and the measurements on intervals as in Section 9.3.2. If the interval width is reduced, the solutions eventually converge to a single point ($\arg\max_{t\in\mathscr{T}} \Pi(t)$) and coincide with the optimal solution of (9.11).

One interesting feature of one-dimensional problems is that the effect of additional measurements is a pure scaling of $\Pi(t)$, but not a qualitative change that would result in measurements at different times. In other words: it is always optimal to measure as much as possible at the point / interval in time where $\Pi(t)$ has its maximum value. The measurement reduces the value of $\Pi(t)$, but its maximum remains in the same time point. This is visualized in Figure 9.6 left, where the optimal sampling (9.22) for different values of $w^{\max}$ results in differently scaled $\Pi(t)$. We see in the next section that this is not necessarily the case for higher-dimensional OED problems.

9.6.2 Lotka Volterra

We are interested in estimating the parameters $p_2, p_4 \in \mathbb{R}$ of the Lotka-Volterra type predator-prey fish initial value problem

$$\begin{aligned} \dot{x_1}(t) &= p_1\, x_1(t) - p_2 x_1(t) x_2(t) - p_5 u(t) x_1(t),\ t \in [0, t_\mathrm{f}], \quad x_1(0) = 0.5, \\ \dot{x_2}(t) &= -p_3\, x_2(t) + p_4 x_1(t) x_2(t) - p_6 u(t) x_2(t),\ t \in [0, t_\mathrm{f}], \quad x_2(0) = 0.7, \end{aligned}$$

where $u(\cdot)$ is a fishing control that may or may not be fixed. The other parameters, the initial values and $t_\mathrm{f} = 12$ are fixed, in consistency with a benchmark problem in mixed-integer optimal control, [209]. We are interested in how to fish and when to measure, again with an upper bound M on the measuring time. We can measure the states directly, $h^1(x(t)) = x_1(t)$ and $h^2(x(t)) = x_2(t)$. We use two different sampling functions, $w^1(\cdot)$ and $w^2(\cdot)$ in the same experimental setting. This can be seen either as a two-dimensional measurement function $h(x(t))$, or as a special case of a multiple experiment, in which $u(\cdot)$, $x(\cdot)$, and $G(\cdot)$ are identical. The experimental design problem (9.13) then reads

$$\begin{aligned}
&\min_{x,G,F,z^1,z^2,u,w^1,w^2} \quad \text{trace}\ \left(F^{-1}(t_\mathrm{f})\right) \\
&\text{subject to} \\
&\begin{array}{lcl}
\dot{x_1}(t) &=& p_1\, x_1(t) - p_2 x_1(t) x_2(t) - p_5 u(t) x_1(t), \\
\dot{x_2}(t) &=& -p_3\, x_2(t) + p_4 x_1(t) x_2(t) - p_6 u(t) x_2(t), \\
\dot{G_{11}}(t) &=& f_{x11}(\cdot)\, G_{11}(t) + f_{x12}(\cdot)\, G_{21}(t) + f_{p12}(\cdot), \\
\dot{G_{12}}(t) &=& f_{x11}(\cdot)\, G_{12}(t) + f_{x12}(\cdot)\, G_{22}(t), \\
\dot{G_{21}}(t) &=& f_{x21}(\cdot)\, G_{11}(t) + f_{x22}(\cdot)\, G_{21}(t), \\
\dot{G_{22}}(t) &=& f_{x21}(\cdot)\, G_{12}(t) + f_{x22}(\cdot)\, G_{22}(t) + f_{p24}(\cdot), \\
\dot{F_{11}}(t) &=& w^1(t) G_{11}(t)^2 + w^2(t) G_{12}(t)^2, \\
\dot{F_{12}}(t) &=& w^1(t) G_{11}(t) G_{12}(t) + w^2(t) G_{12}(t) G_{22}(t), \\
\dot{F_{22}}(t) &=& w^1(t) G_{21}(t)^2 + w^2(t) G_{22}(t)^2, \\
\dot{z^1}(t) &=& w^1(t), \\
\dot{z^2}(t) &=& w^2(t), \\
 & & \\
x(0) &=& (0.5, 0.7), \\
G(0) &=& F(0) = 0, \\
z^1(0) &=& z^2(0) = 0, \\
 & & \\
u(t) &\in& \mathscr{U},\ w^1(t) \in \mathscr{W},\ w^2(t) \in \mathscr{W}, \\
0 &\leq& M - z(t_\mathrm{f})
\end{array}
\end{aligned} \tag{9.23}$$

with $t_\mathrm{f} = 12$, $p_1 = p_2 = p_3 = p_4 = 1$, and $p_5 = 0.4$, $p_6 = 0.2$ and $f_{x11}(\cdot) = \partial f_1(\cdot)/\partial x_1 = p_1 - p_2x_2(t) - p_5u(t)$, $f_{x12}(\cdot) = -p_2x_1(t)$, $f_{x21}(\cdot) = p_4x_2(t)$, $f_{x22}(\cdot) = -p_3 + p_4x_1(t) - p_6u(t)$, and $f_{p12}(\cdot) = \partial f_1(\cdot)/\partial p_2 = -x_1(t)x_2(t)$, $f_{p24}(\cdot) = \partial f_2(\cdot)/\partial p_4 = x_1(t)x_2(t)$.
Note that the state $F_{21}(\cdot) = F_{12}(\cdot)$ has been left out for reasons of symmetry. We start by looking at the case where the control function $u(\cdot)$ is fixed to zero. In this case the states and the sensitivities are given as the solution of the initial value problem, independent of the sampling functions $w^1(\cdot)$ and $w^2(\cdot)$. Figure 9.2 shows the trajectories of $x(\cdot)$ and $G(\cdot)$.

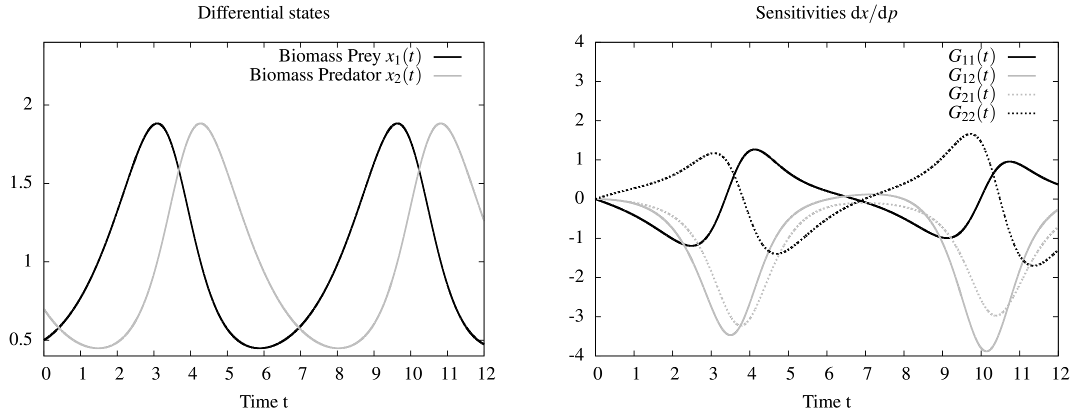


Figure 9.2: States and sensitivities of problem (9.23) for $u(\cdot) \equiv 0$ and $p_2 = p_4 = 1$.

We set $\mathscr{W} = [0,1]$ and $M = (4,4)$. The optimal solution for this control problem is plotted in Figure 9.3. It shows the sampling functions $w^1(\cdot)$ and $w^2(\cdot)$ and the trace of the global information gain matrices

$$\Pi^1(t) \quad = \quad F^{-1}(t_\mathrm{f}) \begin{pmatrix} G_{11}(t)^2 & G_{11}(t)G_{12}(t) \\ G_{11}(t)G_{12}(t) & G_{21}(t)^2 \end{pmatrix} F^{-1}(t_\mathrm{f}) \tag{9.24a}$$

$$\Pi^2(t) \quad = \quad F^{-1}(t_\mathrm{f}) \begin{pmatrix} G_{12}(t)^2 & G_{12}(t)G_{22}(t) \\ G_{12}(t)G_{22}(t) & G_{22}(t)^2 \end{pmatrix} F^{-1}(t_\mathrm{f}) \tag{9.24b}$$

with $F^{-1}(t_\mathrm{f}) = \begin{pmatrix} F_{11}(t_\mathrm{f}) & F_{12}(t_\mathrm{f}) \\ F_{12}(t_\mathrm{f}) & F_{22}(t_\mathrm{f}) \end{pmatrix}^{-1}$.
Comparing this solution that measures at the time intervals when the interval over the trace of $\Pi(t)$ is maximal to a simulated one with all measurements at the first four time intervals, the main effect of the measurements seems to be a homogeneous downscaling over time, comparable to the one-dimensional case in the last example. The value of what could be gained by additional measurements is reduced by a factor of ≈ 10. These values for both measurement functions are, as we have seen in the last section, identical to the Lagrange multipliers μ_i^*. The numerical result for these Lagrange multipliers are also plotted as horizontal lines in Figure 9.3. As one expects

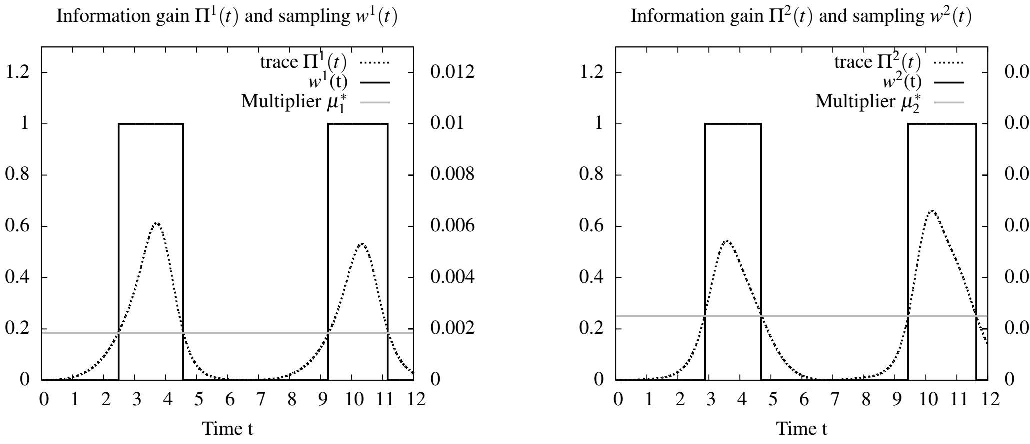


Figure 9.3: Optimal solution of problem (9.23) for $u(\cdot) \equiv 0$ and $p_2 = p_4 = 1$. Left: measurement of prey state $h^1(x(t)) = x_1(t)$. Right: measurement of predator state $h^2(x(t)) = x_2(t)$. The dotted lines show the traces of the functions (9.24) over time, their scale is given at the right borders of the plots. One clearly sees the connection between the timing of the optimal sampling, the evolution of the global information gain matrix, and the Lagrange multipliers of the total measurement constraint.

they are identical to the maximal values of the trace of $\Pi(t)$ outside of the time intervals in which measurements take place.

The same is true for the optimal solution for problem (9.23), again with $u(\cdot) \equiv 0$ and $M = (4,4)$, but now $p_4 = 4$. The difference in parameters results in stronger oscillations and differences between the two differential states. The optimal sampling hence needs to take the heavy oscillations into account and do measurements on multiple intervals in time, see Figure 9.4. As one can observe, the optimal solution is a sampling design such that the values of the traces of $\Pi(t)$ at the border points of the $w^i \equiv 1$ arcs are identical to the values of the corresponding Lagrange multipliers. Hence, performing a measurement does have an inhomogeneous (over time) effect on the scaling of $\Pi(t)$. The coupling between measurements at different points in time, and also between different experiments, takes place via the transversality conditions of the adjoint variables.

The inhomogeneous scaling can also be observed in Figure 9.5, where a sampling design for $w^{\max} = 20$ is plotted. One sees that fewer measurement intervals are chosen and that the shape of the local information gain function $\Pi^1(t)$ is different from the one in Figure 9.4.

The same effect – an inhomogeneous scaling of the information gain function – is the reason why fractional values $w(\cdot) \notin \{0,1\}$ may be obtained as optimal values when fixed time grids are used with piecewise constant controls. We use the same scenario as above, hence $u(\cdot) \equiv 0$, $M = (4,4)$, and $p_4 = 4$. Additionally we fix $w^2(\cdot) \equiv 0$ and consider a piecewise constant control discretization on the grid $t_i = i$ with $i = 0 \ldots 12$. We consider the trajectories for $w^1(t) = w_i$ when

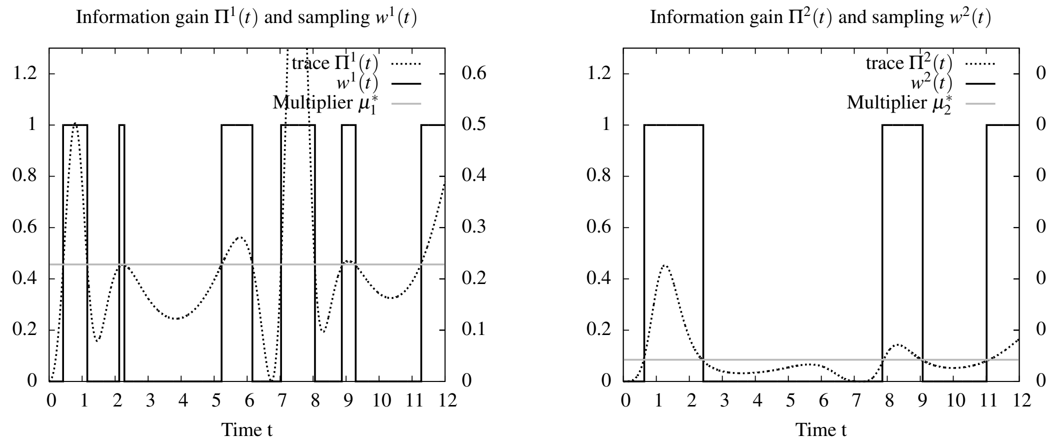


Figure 9.4: Optimal solution of problem (9.23) for $u(\cdot) \equiv 0$ and $p_2 = 1, p_4 = 4$. The traces of the information gain functions have more local maxima, hence the sampling is distributed in time. Note that the Lagrange multipliers indicate entry and exit of the functions into the intervals of measurement.

$t \in [t_i, t_{i+1}]$, $i = 0 \dots 11$ with

$$w_0 = w_7 = w_{11} = 1, \quad w_1 = w_3 = w_4 = w_6 = w_7 = w_8 = w_{10} = 0 \tag{9.25}$$

and the three cases

$$w_2 = 0.3413, w_5 = 0.6587, \tag{9.26a}$$

$$w_2 = 0.6413, w_5 = 0.3587, \tag{9.26b}$$

$$w_2 = 0.9413, w_5 = 0.0587, \tag{9.26c}$$

where the trajectory corresponding to (9.26b) is the optimal one, and the two others have been slightly modified to visualize the effect of scaling the information gain matrix by modifying the sampling design. See Figure 9.6 right for the corresponding information gain functions. One sees clearly the inhomogeneous scaling. The optimal solution (9.26b) on this coarse grid is the solution which scales the information gain function in a way such that the integrated values on $[2,3]$ and $[5,6]$ are identical. To get an integer feasible solution with $w(\cdot) \in \{0,1\}$ we therefore recommend to refine the measurement grid rather than rounding.

Next, we shed some light on the case where we have additional degrees of freedom. We choose $\mathscr{U} = [0,1]$ and allow for additional fishing, again for the case $p_2 = p_4 = 1$. In Figure 9.7 left one sees the optimal control $u^*(\cdot)$, which is also of bang-bang type. The effect of this control is an increase in amplitude of the states' oscillations, which leads to an increase in sensitivity information, see Figure 9.7 right. The corresponding optimal sampling design is plotted in Figure 9.8. The timing is comparable to the one in Figure 9.3. However, the combination of control

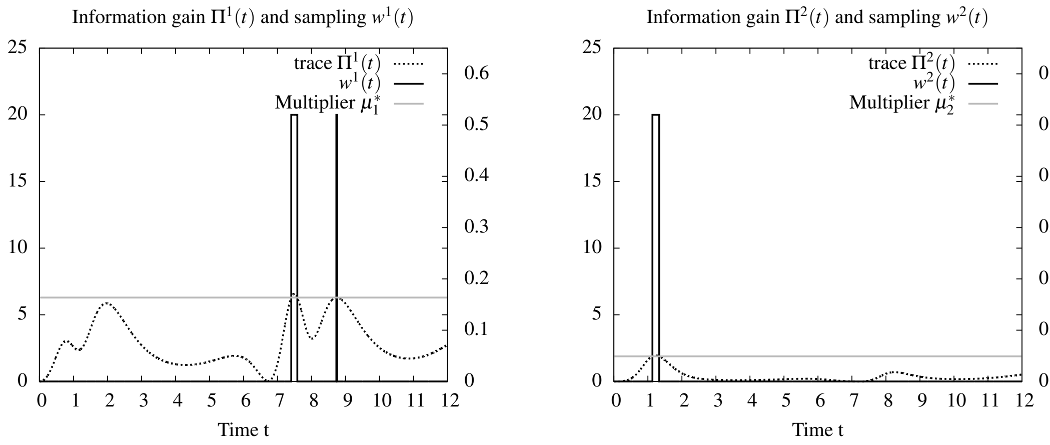


Figure 9.5: Optimal solution of problem (9.23) as in Figure 9.4, but now with $w^{\max} = 20$. Comparing trace $\Pi^1(t)$ to the one in Figure 9.4, one observes a modification and hence a change in the number of arcs with $w^1(t) \equiv 1$. The objective function value is reduced, which is reflected in the fact that the values of the optimal Lagrange multipliers μ_i^* are smaller than in Figure 9.4.

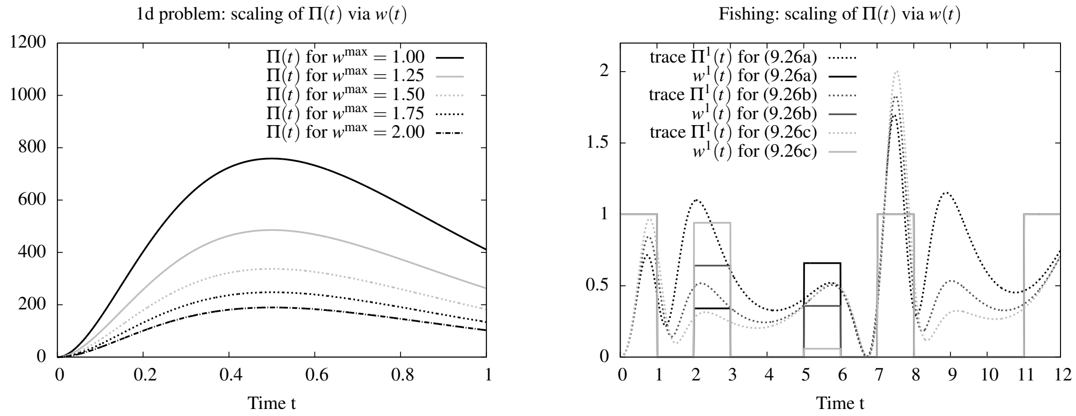


Figure 9.6: Left: Global information gain function for one-dimensional OED problem (9.21) and controls $w(\cdot)$ obtained from (9.22) for different values of $w^{\max}$. Note that the information gain matrix is scaled uniformly over the whole time horizon. Right: Global information gain functions for OED problem (9.23) and controls $w(\cdot)$ obtained from (9.25) and either one from (9.26b-9.26c). One sees that the information gain matrix $\Pi^1(t)$ is scaled differently, depending on the values of w_2 and w_5. The optimal solution (9.26b) on this coarse grid is the solution which scales the information gain function in a way such that the integrated values on $[2,3]$ and $[5,6]$ are identical.

function $u^*(\cdot)$ and the sampling design leads to a concentration of information in the time intervals in which measurements are being done. This is best seen by comparing the values of the Lagrange multipliers in Figure 9.3 of $\mu^* \approx (1.8, 2.6)10^{-3}$ versus the ones of Figure 9.8 with $\mu^* \approx (3, 3.6)10^{-4}$ which are one order of magnitude smaller.

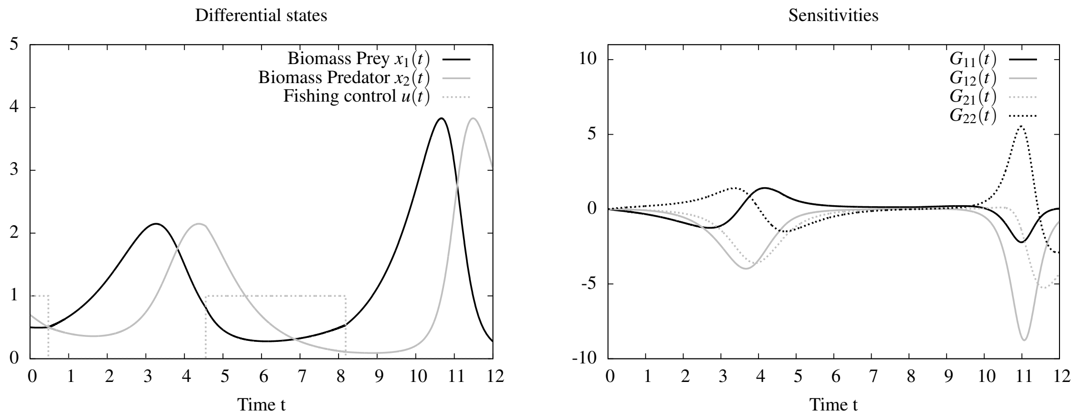


Figure 9.7: States and sensitivities of problem (9.23) for $u(\cdot) \in \mathscr{U} = [0,1]$ and $p_2 = p_4 = 1$. See the increased variation in amplitude compared to Figure 9.2.

As a last illustrating case study we consider an additional L^1 penalty of the sampling design in the objective function as discussed in Section 9.5.2. We consider problem (9.23) for $u(\cdot) \equiv 0$ and $p_2 = p_4 = 1$ and $M = \infty$. The objective function now reads

$$\min_{x,G,F,z^1,z^2,u,w^1,w^2} \text{trace } \left(F^{-1}(t_\mathrm{f})\right) + \int_{\mathscr{T}} \varepsilon(w^1(\tau) + w^2(\tau))\, \mathrm{d}\tau \tag{9.27}$$

with $\varepsilon = 1$.

As can be seen in Figure 9.9, the L_1 penalization has the effect that the optimal sampling functions are given by

$$w^i(t) = \begin{cases} w^{\max} & \text{trace } \Pi^i(t) \geq \varepsilon \\ 0 & \text{else} \end{cases} \tag{9.28}$$

This implies that the value of ε in the problem formulation can be used to directly influence the optimal sampling design. Especially for ill-posed problems with small values in the information gain matrix $\Pi(t)$ this penalization is beneficial from a numerical point of view, as it avoids flat regions in the objective function landscape that might lead to an increased number of iterations. Also it allows a direct economic interpretation by coupling the costs of a single measurement to the information gain. To give an idea on the impact on the number of iterations until convergence we consider an instance with both measurement functions, $u(\cdot) \in [0,1]$ and $M = (6,6)$. Dependent on the penalization value ε in (9.27) we get the following number of SQP iterations (with default settings) with the optimal control code MUSCOD-II:

ε	0	10^{-3}	10^{-2}	10^{-1}	1	10
SQP iterations	312	275	286	255	116	15

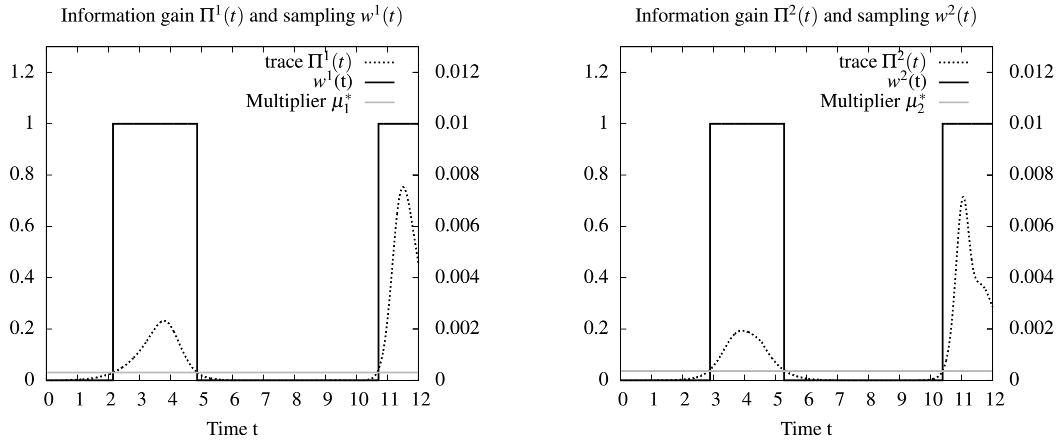


Figure 9.8: Optimal sampling corresponding to Figure 9.7. Note the reduction of the Lagrange multiplier by one order of magnitude compared to Figure 9.3 due to the amplification of states and sensitivities.

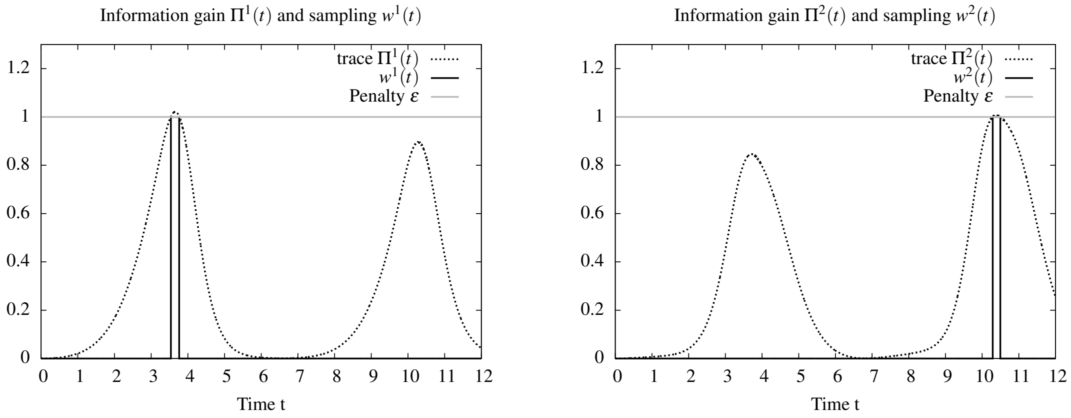


Figure 9.9: Optimal sampling for problem (9.23) with objective function augmented by linear penalty term $\int_{\mathcal{T}} \varepsilon(w^1(\tau)+w^2(\tau))\,\mathrm{d}\tau$. The sampling functions $w^i(t)$ are at their upper bounds of 1 if and only if trace $\Pi^i(t) \geq \varepsilon = 1$.

The optimal solutions are of course different, hence a comparison is somewhat arbitrary. However, it at least gives an indication of the potential.

We discourage to use a L_2 penalization as discussed in Section 9.5.3. It often results in sensitivity seeking arcs with values in the interior of $\mathcal{W}$, and there is no useful economic interpretation.

9.7 Summary

We have applied the integer gap theorem and the maximum principle to an optimal control formulation of a generic optimum experimental design problem. Thus we were able to analyze the role of sampling functions that determine when measurements should be performed to maximize the information gain with respect to unknown model parameters. We showed the similarity between a continuous time formulation with measurements on intervals of time, and a formulation with measurements at single points in time. We defined the *information gain* functions that apply to both formulations as the result of a theoretical analysis of the necessary conditions of optimality. Based on information gain functions we were able to shed light on several aspects, both theoretical as by means of two numerical examples.

Differences between Fisher and Covariance Objective Function. We showed that the information gain matrix for a Fisher objective function has a local character, whereas the one for a covariance objective function includes terms that depend on differential states at the end of the time horizon. This implies that measurements effect the information gain function in the covariance objective case, but not in the Fisher objective case. This noncorrelation for a maximization of a function of the Fisher information matrix has direct consequences: integral-neutral rounding of fractional solutions does not have any influence on the objective function. It also means that other experiments do not influence the choice of the measurements. Third, providing a feedback law in the context of first optimize then discretize methods is possible. All this is usually not true for Covariance Objective Functions.

Scaling of Global Information Gain Function by Measuring. Taking measurements changes the global information matrix $\Pi(t)$. The impact may be in form of a uniform downscaling, but also as a nonhomogeneous over time modification. In the latter case it is not optimal to take as many measurements as possible in one single point of time, as is the case for a Fisher objective function or one-dimensional problems, if one allows more than one measurement per time point / interval. The coupling between the information function and the measurement functions takes place via the transversality conditions, thus the impact also carries over to other experiments and measurement functions.

Role of Lagrange multipliers. We showed that the Lagrange multipliers of constraints that limit the total number of measurements on the time horizon give a threshold for the information gain function. Whenever the function value is higher, measurements are performed, otherwise the value of w is 0.

Role of additional control functions. We used a numerical example to exemplarily demonstrate the effect of additional control functions on the shape of the information gain function.

Role of fixed grids and piecewise constant approximations. For the practically interesting case that optimizations are performed on a given measurement grid we showed that fractional solutions may be optimal. We recommend to further refine the measurement grid instead of rounding.

Penalizations and ill-posed problems. By its very nature, optimal solutions result in small values of the global information gain function. This explains why OED problems are often ill-posed if the upper bounds on the total amount of measurements are chosen too high: additional measurements only yield small contributions to the objective function once the other measurements have been placed in an optimal way. As a remedy to overcome this intrinsic problem of OED we propose to use L_1 penalizations of the measurement functions. We showed that the penalization parameter can be directly interpreted in terms of the information gain functions. Therefore such a formulation would couple the costs of a measurement to a minimum amount of information it has to yield, which makes sense from a practical point of view. Of course, the value of ε can also be decreased in a homotopy.

9.8 Appendix: useful lemmata

In this Appendix we list useful lemmata we use in this chapter.

Lemma 9.8.1. (Positive trace)
If $A \in \mathbb{R}^{n\times n}$ is positive definite, then $\operatorname{trace}(A) > 0$.

Proof. As A is positive definite, it holds $x^T Ax > 0$ for all $x \in \mathbb{R}^n$, in particular for all unit vectors. Hence it follows $a_{ii} > 0$ for all $i = 1 \dots n$ and thus trivially $\operatorname{trace}(A) = \sum_{i=1}^n a_{ii} > 0$. □

Lemma 9.8.2. (Derivative of trace function)
Let A be a quadratic $n \times n$ matrix. Then

$$\left\langle \frac{\partial \operatorname{trace}(A)}{\partial A}, \Delta A \right\rangle = \operatorname{trace}(\Delta A). \tag{9.29}$$

Proof.

$$\left\langle \frac{\partial \operatorname{trace}(A)}{\partial A}, \Delta A \right\rangle = \lim_{h\to 0} \frac{\operatorname{trace}(A + h\Delta A) - \operatorname{trace}(A)}{h} = \lim_{h\to 0} \frac{h \operatorname{trace}(\Delta A)}{h} = \operatorname{trace}(\Delta A).$$

Lemma 9.8.3. (Derivative of inverse operation)
Let $A \in \mathrm{GL}_n(\mathbb{R})$ be an invertible $n \times n$ matrix. Then

$$\frac{\partial A^{-1}}{\partial A} \cdot \Delta A = -A^{-1} \Delta A A^{-1}. \tag{9.30}$$

Lemma 9.8.4. (Derivative of eigenvalue operation)
Let $\lambda(A)$ be a single eigenvalue of the symmetric matrix $A \in \mathbb{R}^{n\times n}$. Let $z \in \mathbb{R}^n$ be an eigenvector of A to $\lambda(A)$ with norm 1. Then it holds

$$\left\langle \frac{\partial \lambda(A)}{\partial A}, \Delta A \right\rangle = z^T \Delta A z. \tag{9.31}$$

Lemma 9.8.5. (Derivative of determinant operation)
Let $A \in \mathbb{R}^{n\times n}$ be a symmetric, positive definite matrix. Then it holds

$$\left\langle \frac{\partial \det(A)}{\partial A}, \Delta A \right\rangle \quad = \quad \det(A) \sum_{i,j=1}^{n} A_{i,j}^{-1}\, \Delta A_{i,j}. \tag{9.32}$$

Proofs for the Lemmata 9.8.3, 9.8.4, and 9.8.5 can be found in [154].

10 Outlook

As summarized in the introduction, it was the goal of this thesis to advance the state-of-the-art of mixed-integer nonlinear optimal control. With the presented methods it is possible to solve MIOCPs constrained by ordinary differential equations or differential algebraic equations, whenever the continuous relaxations of the outer convexifications are solvable. This is a huge advantage compared to combinatorial algorithms like Branch&Bound that add an outer level of complexity and increase the number of optimization problems that need to be solved drastically.

Also, the methodology allows to incorporate additional characteristics, such as multiple objectives [170], control delays (Chapter 6), uncertainties that can be formulated by means of scenario trees (Chapter 6), or that are additionally constrained by combinatorial constraints (Chapter 5). Also, the results form the basis for efficient nonlinear model predictive control algorithms that solve mixed-integer nonlinear optimal control problems in real time, compare [143]. Several benchmark problems have been addressed, including whole problem classes as optimum experimental design in Chapter 9 or novel application areas for optimization, as *complex problem solving* in Chapter 8.

Yet, there are still many open questions and challenges remaining, such as

- whether the theory carries over to the case of time–dependent partial differential equations and whether the algorithms can be adapted in a straightforward way. This is certainly possible, whenever a “Method of Lines” approach is taken, however there might be pitfalls whenever the spatial discretization becomes an issue.
- what to do with spatially distributed integer controls. Apparently the successful Sum Up Rounding strategy makes use of the “ordering” of the time–axis. In higher dimensions this can only be done, if the process is directed. An analogous theory for “checker-boarding” is still missing, although there are many interesting ideas around in the area of topology optimization.
- how stochastic differential equations can be treated in this deterministic setting.
- the development of even more efficient numerical algorithms, especially in combination with the increasingly dominating multi-core architectures.
- the issue of global optimization and a coherent integration with simultaneous optimal control methods.
- structure–exploitation for specific applications as gas, water or commodity networks and for whole problem classes, such as optimum experimental design.

Bibliography

[1] K. Abhishek, S. Leyffer, and J.T. Linderoth. Filmint: An outer-approximation-based solver for nonlinear mixed integer programs. Preprint ANL/MCS-P1374-0906, Argonne National Laboratory, Mathematics and Computer Science Division, September 2006.

[2] P. Abichandani, H.Y. Benson, and M. Kam. Multi-vehicle path coordination under communication constraints. In *American Control Conference*, pages 650–656, 2008.

[3] W. Achtziger and C. Kanzow. Mathematical programs with vanishing constraints: optimality conditions and constraint qualifications. *Mathematical Programming Series A*, 114:69–99, 2008.

[4] European Space Agency. GTOP database: Global optimisation trajectory problems and solutions. http://www.esa.int/gsp/ACT/inf/op/globopt.htm.

[5] M. Alamir and S. A. Attia. On solving optimal control problems for switched hybrid nonlinear systems by strong variations algorithms. In *6th IFAC Symposium, NOLCOS, Stuttgart, Germany, 2004*, 2004.

[6] J. Albersmeyer, D. Beigel, C. Kirches, L. Wirsching, H.G. Bock, and J.P. Schlöder. Fast nonlinear model predictive control with an application in automotive engineering. In L. Magni, D.M. Raimondo, and F. Allgöwer, editors, *Proceedings of the international workshop on assessment and future directions of nonlinear model predictive control*, 2008.

[7] J. Albersmeyer and H.G. Bock. Sensitivity Generation in an Adaptive BDF-Method. In Hans Georg Bock, E. Kostina, X.H. Phu, and R. Rannacher, editors, *Modeling, Simulation and Optimization of Complex Processes: Proceedings of the International Conference on High Performance Scientific Computing, March 6–10, 2006, Hanoi, Vietnam*, pages 15–24. Springer Verlag Berlin Heidelberg New York, 2008.

[8] J. Albersmeyer and M. Diehl. The Lifted Newton method and its application in optimization. *SIAM Journal on Optimization*, 20(3):1655–1684, 2010.

[9] W. Amaldoss and S. Jain. Conspicuous consumption and sophisticated thinking. *Management Science*, 51:1449–1466, 2005.

[10] W. Amaldoss and S. Jain. Pricing of conspicuous goods: a competitive analysis of social effects. *Journal of Marketing Research*, 42:30–42, 2005.

[11] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 3rd edition, 1999. ISBN 0-89871-447-8 (paperback).

[12] P. Antsaklis and X. Koutsoukos. On hybrid control of complex systems: A survey. In 3rd International Conference ADMP'98, Automation of Mixed Processes: Dynamic Hybrid Systems, pages 1–8, Reims, France, March 1998., 1998.

[13] D. Applegate, R.E. Bixby, V. Chvátal, W. Cook, D. Espinoza, M. Goycoolea, and K. Helsgaun. Certification of an optimal TSP tour through 85, 900 cities. *Oper. Res. Lett.*, 37(1):11–15, 2009.

[14] P.K. Asea and P.J. Zak. Time–to–build and cycles. *Journal of Economic Dynamics & Control*, 23:1155–1175, 1999.

[15] A.C. Atkinson, A.N. Donev, and R.D. Tobias. *Optimum experimental designs, with SAS*. Oxford University Press, 2007.

[16] AT&T Bell Laboratories, University of Tennessee, and Oak Ridge National Laboratory. Netlib linear programming library. http://www.netlib.org/lp/.

[17] S.A. Attia, M. Alamir, and C. Canudas de Wit. Sub optimal control of switched nonlinear systems under location and switching constraints. In *IFAC World Congress*, 2005.

[18] M. Bambi. Endogenous growth and time to build: the AK case. *Journal of Economic Dynamics and Control*, 32:1015–1040, 2008.

[19] M. Bambi, G. Fabbri, and F. Gozzi. Optimal policy and consumption smoothing effects in the time-to-build AK model. MPRA Paper No. 17128, Munich Personal RePEc Archive, 2009.

[20] A. Bardow, W. Marquardt, V. Göke, H. J. Koss, and K. Lucas. Model-based measurement of diffusion using raman spectroscopy. *AIChE journal*, 49(2):323–334, 2003.

[21] C.M. Barth. *The Impact of Emotions on Complex Problem Solving Performance and Ways of Measuring this Performance*. PhD thesis, Ruprecht–Karls–Universität Heidelberg, 2010.

[22] C.M. Barth and J. Funke. Negative affective environments improve complex solving performance. *Cognition and Emotion*, 24:1259–1268, 2010.

[23] R.A. Bartlett and L.T. Biegler. QPSchur: A dual, active set, Schur complement method for large-scale and structured convex quadratic programming algorithm. *Optimization and Engineering*, 7:5–32, 2006.

[24] R.A. Bartlett, A. Wächter, and L.T. Biegler. Active set vs. interior point strategies for model predicitve control. In *Proceedings of the American Control Conference*, pages 4229–4233, Chicago, IL, 2000.

[25] I. Bauer, H.G. Bock, S. Körkel, and J.P. Schlöder. Numerical methods for optimum experimental design in DAE systems. *J. Comput. Appl. Math.*, 120(1-2):1–15, 2000.

[26] B.T. Baumrucker and L.T. Biegler. MPEC strategies for optimization of a class of hybrid dynamic systems. *Journal of Process Control*, 19(8):1248 – 1256, 2009. Special Section on Hybrid Systems: Modeling, Simulation and Optimization.

[27] B.T. Baumrucker, J.G. Renfro, and L.T. Biegler. MPEC problem formulations and solution strategies with chemical engineering applications. *Computers and Chemical Engineering*, 32:2903–2913, 2008.

[28] R.E. Bellman. *Dynamic Programming*. University Press, Princeton, N.J., 6th edition, 1957. ISBN 0-486-42809-5 (paperback).

[29] P. Belotti. Couenne: a user's manual. Technical report, Lehigh University, 2009.

[30] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Waechter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24:597–634, 2009.

[31] A. Ben-Tal, S. Boyd, and A. Nemirovski. Extending scope of robust optimization: Comprehensive robust counterparts of uncertain problems. *Mathematical Programming*, 107(1–2):63–89, June 2006.

[32] D.P. Bertsekas. *Dynamic programming and optimal control, Volume 1*. Athena Scientific, Belmont, Mass., 3. ed. edition, 2005.

[33] J.T. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming*. SIAM, Philadelphia, 2001.

[34] L.T. Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers and Chemical Engineering*, 8:243–248, 1984.

[35] L.T. Biegler. An overview of simultaneous strategies for dynamic optimization. *Chemical Engineering and Processing*, 46:1043–1053, 2007.

[36] L.T. Biegler. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. Series on Optimization. SIAM, 2010.

[37] T. Binder, L. Blank, H.G. Bock, R. Bulirsch, W. Dahmen, M. Diehl, T. Kronseder, W. Marquardt, J.P. Schlöder, and O.v. Stryk. Introduction to model based optimization of chemical processes on moving horizons. In M. Grötschel, S.O. Krumke, and J. Rambau, editors, *Online Optimization of Large Scale Systems: State of the Art*, pages 295–340. Springer, 2001.

[38] Robert E. Bixby, Mary Fenelon, Zonghao Gu, Edward Rothberg, and Roland Wunderling. Mixed-integer programming: A progress report. In *The Sharpest Cut: The Impact of Manfred Padberg and His Work*. SIAM, 2004.

[39] H.G. Bock. Numerical treatment of inverse problems in chemical reaction kinetics. In K.H. Ebert, P. Deuflhard, and W. Jäger, editors, *Modelling of Chemical Reaction Systems*, volume 18 of *Springer Series in Chemical Physics*, pages 102–125. Springer, Heidelberg, 1981.

[40] H.G. Bock. Recent advances in parameter identification techniques for ODE. In P. Deuflhard and E. Hairer, editors, *Numerical Treatment of Inverse Problems in Differential and Integral Equations*, pages 95–121. Birkhäuser, Boston, 1983.

[41] H.G. Bock, E. Kostina, and J.P. Schlöder. Numerical methods for parameter estimation in nonlinear differential algebraic equations. *GAMM Mitteilungen*, 30/2:376–408, 2007.

[42] H.G. Bock and R.W. Longman. Optimal control of velocity profiles for minimization of energy consumption in the new york subway system. In *Proceedings of the Second IFAC Workshop on Control Applications of Nonlinear Programming and Optimization*, pages 34–43. International Federation of Automatic Control, 1980.

[43] H.G. Bock and R.W. Longman. Computation of optimal controls on disjoint control sets for minimum energy subway operation. In *Proceedings of the American Astronomical Society. Symposium on Engineering Science and Mechanics*, Taiwan, 1982.

[44] H.G. Bock and K.J. Plitt. A Multiple Shooting algorithm for direct solution of optimal control problems. In *Proceedings of the 9th IFAC World Congress*, pages 242–247, Budapest, 1984. Pergamon Press. Available at http://www.iwr.uni-heidelberg.de/groups/agbock/FILES/Bock1984.pdf.

[45] P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2009.

[46] P. Bonami, G. Cornuejols, A. Lodi, and F. Margot. Feasibility pump for mixed integer nonlinear programs. *Mathematical Programming*, 199:331–352, 2009.

[47] P. Bonami, M. Kilinc, and J. Linderoth. Algorithms and software for convex mixed integer nonlinear programs. Technical Report 1664, University of Wisconsin, October 2009.

[48] R. Boucekkine, O. Licandro, L. Puch, and F. del Rio. Vintage capital and the dynamics of the AK model. *Journal of Economic Theory*, 120:39–72, 2005.

[49] U. Brandt-Pollmann, R. Winkler, S. Sager, U. Moslener, and J.P. Schlöder. Numerical solution of optimal control problems with constant control delays. *Computational Economics*, 31(2):181–206, 2008.

[50] B. Brehmer. *Feedback delays in dynamic decision making*, pages 103–130. Complex problem solving: The European Perspective. Lawrence Erlbaum Associates, 1995.

[51] A.E. Bryson and Y.-C. Ho. *Applied Optimal Control*. Wiley, New York, 1975.

[52] A. Buchner. Auf Dynamischer Programmierung basierende nichtlineare modellprädiktive Regelung für LKW. Diploma thesis, Ruprecht–Karls–Universität Heidelberg, January 2010.

[53] H. Burgdörfer. Strukturausnutzende Algorithmen der Quadratischen Programmierung für gemischt-ganzzahlige Optimalsteuerung. Master's thesis, Universität Heidelberg, 2011.

[54] J. Burgschweiger, B. Gnädig, and M.C. Steinbach. Optimization models for operative planning in drinking water networks. *Optimization and Engineering*, 10(1):43–73, 2008. Online first.

[55] J. Burgschweiger, B. Gnädig, and M.C. Steinbach. Nonlinear programming techniques for operative planning in large drinking water networks. *The Open Applied Mathematics Journal*, 3:1–16, 2009.

[56] M. Buss, M. Glocker, M. Hardt, O. v. Stryk, R. Bulirsch, and G. Schmidt. *Nonlinear Hybrid Dynamical Systems: Modelling, Optimal Control, and Applications*, volume 279. Springer-Verlag, Berlin, Heidelberg, 2002.

[57] Michael R. Bussieck, Arne Stolbjerg Drud, and Alexander Meeraus. Minlplib–a collection of test models for mixed-integer nonlinear programming. *INFORMS J. on Computing*, 15(1):114–119, 2003.

[58] M.R. Bussieck. Gams performance world. http://www.gamsworld.org/performance.

[59] C. Castro, F. Palacios, and E. Zuazua. An alternating descent method for the optimal control of the inviscid Burgers equation in the presence of shocks. *Mathematical Models and Methods in Applied Sciences*, 3(18):369–416, 2008.

[60] J.P. Caulkins, G. Feichtinger, D. Grass, R.F. Hartl, P.M. Kort, and A. Seidl. Two-stage conspicuous consumption model. Working paper, 2010.

[61] J.P. Caulkins, G. Feichtinger, D. Grass, R.F. Hartl, P.M. Kort, and A. Seidl. Optimal pricing of a conspicuous product during a recession that freezes capital markets. *Journal of Economic Dynamics and Control*, 35(1):163–174, 2011. doi:10.1016/j.jedc.2010.09.001.

[62] J.P. Caulkins, R.F. Hartl, and P.M. Kort. Delay equivalence in capital accumulation models. *Journal of Mathematical Economics*, 46(6):1243–1246, 2010. doi:10.1016/j.jmateco.2010.08.021.

[63] B. Chachuat, A.B. Singer, and P.I. Barton. Global methods for dynamic optimization and mixed-integer dynamic optimization. *Industrial and Engineering Chemistry Research*, 45(25):8573–8392, 2006.

[64] T. Christof and G. Reinelt. Combinatorial optimization and small polytopes. *TOP*, 4(1):1 – 53, June 1996.

[65] Thomas Christof and Andreas Löbel. PORTA – POlyhedron Representation Transformation Algorithm. http://www.zib.de/Optimization/Software/Porta/. PORTA Homepage.

[66] CMU-IBM. Cyber-infrastructure for MINLP collaborative site. http://minlp.org.

[67] F. Collard, O. Licandro, and L. Puch. The short-run dynamics of optimal growth models with delays. *Annales d'Économie et de Statistique*, 90:127–143, 2008.

[68] F. Colonius and W. Kliemann. *The dynamics of control*. Birkhäuser, Boston, 2000.

[69] GAMS Development Corporation. GAMS homepage. http://www.gams.com/.

[70] D. Danner, D. Hagemann, A. Schankin, M. Hager, and J. Funke. Beyond iq. a latent state-trait analysis of general intelligence, dynamic decision making, and implicit learning. *Intelligence*, 2011. (in press).

[71] W. D. Dechert and K. Nishimura. A complete characterization of optimal growth paths in an aggregated model with a non-concave production function. *Journal of Economic Theory*, 31:332–354, 1983.

[72] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*. PhD thesis, Universität Heidelberg, 2001.

[73] M. Diehl, H.G. Bock, and E. Kostina. An approximation technique for robust nonlinear optimization. *Mathematical Programming*, 107:213–230, 2006.

[74] M. Diehl, R. Findeisen, S. Schwarzkopf, I. Uslu, F. Allgöwer, H.G. Bock, E.D. Gilles, and J.P. Schlöder. An efficient algorithm for nonlinear model predictive control of large-scale systems. Part I: Description of the method. *Automatisierungstechnik*, 50(12):557–567, 2002.

[75] M. Diehl and A. Walther. A test problem for periodic optimal control algorithms. Technical report, ESAT/SISTA, K.U. Leuven, 2006.

[76] A.V. Dmitruk and A.M. Kaganovich. The hybrid maximum principle is a consequence of Pontryagin maximum principle. *Systems and Control Letters*, 57(11):964–970, 2008.

[77] T. Donchev. Approximation of lower semicontinuous differential inclusions. *Numerical Functional Analysis and Optimization*, 22(1):55–67, 2001.

[78] D. Dörner. On the difficulties people have in dealing with complexity. *Simulation and Games*, 11:87–106, 1980.

[79] I.S. Duff. MA57 — a code for the solution of sparse symmetric definite and indefinite systems. *ACM Transactions on Mathematical Software*, 30(2):118–144, 2004.

[80] I.S. Duff and J.K. Reid. The multifrontal solution of indefinite sparse symmetric linear equations. *ACM Transactions on Mathematical Software*, 9(3):302–325, 1983.

[81] The Economist. Riding the rollercoaster: Six firms in cyclical industries and battle excess debt. The Economist, December 11 2008.

[82] M. Egerstedt, Y. Wardi, and H. Axelsson. Transition-time optimization for switched-mode dynamical systems. *IEEE Transactions on Automatic Control*, 51:110–115, 2006.

[83] M.A. El-Hodiri, E. Loehman, and A. Whinston. An optimal growth model with time lags. *Econometrica*, 40:1137–1146, 1972.

[84] S. Engell and A. Toumi. Optimisation and control of chromatography. *Computers and Chemical Engineering*, 29:1243–1252, 2005.

[85] W.R. Esposito and C.A. Floudas. Deterministic global optimization in optimal control problems. *Journal of Global Optimization*, 17:97–126, 2000.

[86] P. H. Ewert and J. F. Lambert. Part II: The effect of verbal instructions upon the formation of a concept. *Journal of General Psychology*, 6:400–411, 1932.

[87] Brian C. Fabien. dsoa: Dynamic system optimization. http://abs-5.me.washington.edu/noc/dsoa.html.

[88] H.J. Ferreau. An online active set strategy for fast solution of parametric quadratic programs with applications to predictive engine control. Diploma thesis, Ruprecht–Karls–Universität Heidelberg, 2006.

[89] H.J. Ferreau, H.G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[90] A.F. Filippov. Differential equations with discontinuous right hand side. *AMS Transl.*, 42:199–231, 1964.

[91] R. Fletcher. Resolving degeneracy in quadratic programming. Numerical Analysis Report NA/135, University of Dundee, Dundee, Scotland, 1991.

[92] C.A. Floudas, I.G. Akrotirianakis, S. Caratzoulas, C.A. Meyer, and J. Kallrath. Global optimization in the 21st century: Advances and challenges. *Computers and Chemical Engineering*, 29(6):1185–1202, 2005.

[93] R. Fourer, D.M. Gay, and B.W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, 2002.

[94] G. Franceschini and S. Macchietto. Model-based design of experiments for parameter precision: State of the art. *Chemical Engineering Science*, 63:4846–4872, 2008.

[95] P. A. Frensch and J. Funke, editors. *Complex problem solving: The European perspective*. Lawrence Erlbaum Associates, 1995.

[96] A. Fügenschuh, M. Herty, A. Klar, and A. Martin. Combinatorial and continuous models for the optimization of traffic flows on networks. *SIAM Journal on Optimization*, 16(4):1155–1176, 2006.

[97] A.T. Fuller. Study of an optimum nonlinear control system. *Journal of Electronics and Control*, 15:63–71, 1963.

[98] J. Funke. Einige Bemerkungen zu Problemen der Problemlöseforschung oder: Ist Testintelligenz doch ein Prädiktor? *Diagnostica*, 29:283–302, 1983.

[99] J. Funke. Using simulation to study complex problem solving: A review of studies in the frg. *Simulation & Games*, 19:277–303, 1988.

[100] J. Funke. *Problemlösendes Denken*. Kohlhammer, 2003.

[101] J. Funke. Complex problem solving: A case for complex cognition? *Cognitive Processing*, 11:133–142, 2010.

[102] J. Funke and P.A. Frensch. *Complex problem solving: The European perspective – 10 years after*, pages 25–47. Learning to solve complex scientific problems. Lawrence Erlbaum, 2007.

[103] M. Garavello and B. Piccoli. Hybrid necessary principle. *SIAM Journal on Control and Optimization*, 43(5):1867–1887, 2005.

[104] W.L. Garrard and J.M. Jordan. Design of nonlinear automatic control systems. *Automatica*, 13:497–505, 1977.

[105] M. Gerdts. Solving mixed-integer optimal control problems by Branch&Bound: A case study from automobile test-driving with gear shift. *Optimal Control Applications and Methods*, 26:1–18, 2005.

[106] M. Gerdts. A variable time transformation method for mixed-integer optimal control problems. *Optimal Control Applications and Methods*, 27(3):169–182, 2006.

[107] M. Gerdts and S. Sager. *Control and Optimization with Differential-Algebraic Constraints*, chapter Mixed-Integer DAE Optimal Control Problems: Necessary conditions and bounds. SIAM, 2012. (accepted).

[108] Matthias Gerdts. *Optimal Control of Ordinary Differential Equations and Differential-Algebraic Equations.* Habilitation, University of Bayreuth, 2006.

[109] E.M. Gertz and S.J. Wright. Object-oriented software for quadratic programming. *ACM Transactions on Mathematical Software*, 29:58–81, 2003.

[110] P.E. Gill, G.H. Golub, W. Murray, and M. A. Saunders. Methods for modifying matrix factorizations. *Mathematics of Computation*, 28(126):505–535, 1974.

[111] P.E. Gill, W. Murray, and M.A. Saunders. *User's Guide For QPOPT 1.0: A Fortran Package For Quadratic Programming*, 1995.

[112] C. Gonzalez. Learning to make decisions in dynamic environments: Effects of time constraints and cognitive abilities. *Human Factors*, 46(3):449–460, 2004.

[113] C. Gonzalez, P. Vanyukov, and M.K. Martin. The use of microworlds to study dynamic decision making. *Computers in Human Behavior*, 21(2):273–286, 2005.

[114] S. Göttlich, M. Herty, C. Kirchner, and A. Klar. Optimal control for continuous supply network models. *Networks and Heterogenous Media*, 1(4):675–688, 2007.

[115] N.I.M. Gould, D. Orban, and P.L. Toint. CUTEr testing environment for optimization and linear algebra solvers. http://cuter.rl.ac.uk/cuter-www/, 2002.

[116] N.I.M. Gould and P.L. Toint. A quadratic programming bibliography. Technical Report 01/02, Rutherford Appleton Laboratory, Computational Science and Engineering Department, June 2003.

[117] G. Grammel. Towards fully discretized differential inclusions. *Set-Valued Analysis*, 11(1):1–8, 2003.

[118] D. Grass, J.P. Caulkins, G. Feichtinger, G. Tragler, and D.A. Behrens. *Optimal Control of Nonlinear Processes: With Applications in Drugs, Corruption, and Terror*. Springer, Berlin, 2008.

[119] A. Größler, F.H. Maier, and P.M. Milling. Enhancing learning capabilities by providing transparency in business simulators. *Simulation & Gaming*, 31(2):257–278, 2000.

[120] I.E. Grossmann. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering*, 3:227–252, 2002.

[121] I.E. Grossmann, P.A. Aguirre, and M. Barttfeld. Optimal synthesis of complex distillation columns using rigorous models. *Computers and Chemical Engineering*, 29:1203–1215, 2005.

[122] M. Gugat, M. Herty, A. Klar, and G. Leugering. Optimal control for traffic flow networks. *Journal of Optimization Theory and Applications*, 126(3):589–616, 2005.

[123] G. Häckl. *Reachable sets, control sets and their computation*, volume 7 of *Augsburger Mathematisch-Naturwissenschaftliche Schriften*. Dr. Bernd Wißner, Augsburg, ISBN: 3-89639-019-8 1996. Dissertation, Universität Augsburg, Augsburg, 1995.

[124] S.P. Han. Superlinearly convergent variable-metric algorithms for general nonlinear programming problems. *Mathematical Programming*, 11:263–282, 1976.

[125] E. Hellström, M. Ivarsson, J. Aslund, and L. Nielsen. Look-ahead control for heavy trucks to minimize trip time and fuel consumption. *Control Engineering Practice*, 17:245–254, 2009.

[126] H.J. Hörmann and M. Thomas. Zum Zusammenhang zwischen Intelligenz und komplexem Problemlösen. *Sprache & Kognition*, 8:23–31, 1989.

[127] B. Houska, H.J. Ferreau, and M. Diehl. An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica*, 2011. (submitted).

[128] T. Huschto, G. Feichtinger, P.M. Kort, R.F. Hartl, S. Sager, and A. Seidl. Numerical solution of a conspicuous consumption model with constant control delay. *Automatica*, 47:1868–1877, 2011.

[129] W. Hussy. Komplexes Problemlösen – Eine Sackgasse? *Zeitschrift für Experimentelle und Angewandte Psychologie*, 32:55–77, 1985.

[130] W. Hussy. Komplexes Problemlösen und Verarbeitungskapazität. *Sprache & Kognition*, 10:208–220, 1991.

[131] H. Huynh. *A Large-Scale Quadratic Programming Solver Based On Block-LU Updates of the KKT System*. PhD thesis, Stanford University, 2008.

[132] Tomlab Optimization Inc. Propt - matlab optimal control software (dae, ode). http://tomdyn.com/.

[133] A.F. Izmailov and M.V. Solodov. Mathematical programs with vanishing constraints: Optimality conditions, sensitivity, and a relaxation method. *Journal of Optimization Theory and Applications*, 142:501–532, 2009.

[134] D. Janka. Optimum experimental design and multiple shooting. Master's thesis, Universität Heidelberg, 2010.

[135] M. Kalecki. A macroeconomic theory of business cycles. *Econometrica*, 3:327–344, 1935.

[136] S. Kameswaran and L.T. Biegler. Simultaneous dynamic optimization strategies: Recent advances and challenges. *Computers and Chemical Engineering*, 30:1560–1575, 2006.

[137] Y. Kawajiri and L.T. Biegler. A nonlinear programming superstructure for optimal dynamic operations of simulated moving bed processes. *I&EC Research*, 45(25):8503–8513, 2006.

[138] Y. Kawajiri and L.T. Biegler. Optimization strategies for Simulated Moving Bed and PowerFeed processes. *AIChE Journal*, 52(4):1343–1350, 2006.

[139] C.Y. Kaya and J.L. Noakes. Computations and time-optimal controls. *Optimal Control Applications and Methods*, 17:171–185, 1996.

[140] C.Y. Kaya and J.L. Noakes. A computational method for time-optimal control. *Journal of Optimization Theory and Applications*, 117:69–92, 2003.

[141] F. Kehrle, Frasch J.V., C. Kirches, and S. Sager. Optimal control of formula 1 race cars in a vdrift based virtual environment. In *IFAC World Congress Milan*, 2011.

[142] H.J. Kelley, R.E. Kopp, and H.G. Moyer. Singular extremals. In G. Leitmann, editor, *Topics in Optimization*, pages 63–101. Academic Press, 1967.

[143] C. Kirches. *Fast numerical methods for mixed–integer nonlinear model–predictive control*. PhD thesis, Ruprecht–Karls–Universität Heidelberg, July 2010. Available at http://www.ub.uni-heidelberg.de/archiv/11636/.

[144] C. Kirches, H.G. Bock, J.P. Schlöder, and S. Sager. Complementary condensing for the direct multiple shooting method. In H.G. Bock, E. Kostina, H.X. Phu, and R. Rannacher, editors, *Proceedings of the Fourth International Conference on High Performance Scientific Computing: Modeling, Simulation, and Optimization of Complex Processes, Hanoi, Vietnam, March 2–6, 2009*, Berlin Heidelberg New York, 2010. Springer Verlag. (accepted).

[145] C. Kirches, H.G. Bock, J.P. Schlöder, and S. Sager. Block structured quadratic programming for the direct multiple shooting method for optimal control. *Optimization Methods and Software*, 26(2):239–257, April 2011.

[146] C. Kirches, H.G. Bock, J.P. Schlöder, and S. Sager. A factorization with update procedures for a KKT matrix arising in direct optimal control. *Mathematical Programming Computation*, 3(4), 2011. (accepted).

[147] C. Kirches, S. Sager, H.G. Bock, and J.P. Schlöder. Time-optimal control of automobile test drives with gear shifts. *Optimal Control Applications and Methods*, 31(2):137–153, March/April 2010.

[148] C. Kirches, L. Wirsching, S. Sager, and H.G. Bock. Efficient numerics for nonlinear model predictive control. In M. Diehl, F. Glineur, E. Jarlebring, and W. Michiels, editors, *Recent Advances in Optimization and its Applications in Engineering*, pages 339–359. Springer, 2010. ISBN 978-3-6421-2597-3.

[149] M. Kleinmann and B. Strauß. Validity and applications of computer simulated scenarios in personal assessment. *International Journal of Seclection and Assessment*, 6(2):97–106, 1998.

[150] R. H. Kluwe. *Knowledge and performance in complex problem solving*, pages 401–423. The cognitive psychology of knowledge. Elsevier Science Publishers, 1993.

[151] Z.H. Kluwe, C. Misiak, and H. Haider. *Systems and Performance in Intelligence Tests*, pages 227–244. Intelligence: Reconceptualization and Measurement. Erlbaum, 1991.

[152] S. Kolb, F. Petzing, and S. Stumpf. Komplexes Problemlösen: Bestimmung der Problemlösegüte von Probanden mittels Verfahren des Operations Research – ein interdisziplinärer Ansatz. *Sprache & Kognition*, 11:115–128, 1992.

[153] V. Kolmanovskii and A. Myshkis. *Applied theory of functional differential equations*. Dordrecht academic publishers, 1992.

[154] S. Körkel. *Numerische Methoden für Optimale Versuchsplanungsprobleme bei nichtlinearen DAE-Modellen*. PhD thesis, Universität Heidelberg, Heidelberg, 2002.

[155] S. Körkel and E. Kostina. Numerical methods for nonlinear experimental design. In Hans Georg Bock, E. Kostina, H. X. Phu, and R. Rannacher, editors, *Modelling, Simulation and Optimization of Complex Processes, Proceedings of the International Conference on High Performance Scientific Computing*, pages 255–272, Hanoi, Vietnam, 2004. Springer.

[156] S. Körkel, E. Kostina, H.G. Bock, and J.P. Schlöder. Numerical methods for optimal control problems in design of robust optimal experiments for nonlinear dynamic processes. *Optimization Methods and Software*, 19:327–338, 2004.

[157] S. Körkel, A. Potschka, H.G. Bock, and S. Sager. A multiple shooting formulation for optimum experimental design. *Mathematical Programming*, 2012. (submitted revisions).

[158] S. Körkel, H. Qu, G. Rücker, and S. Sager. Derivative based vs. derivative free optimization methods for nonlinear optimum experimental design. In *Proceedings of HPCA2004 Conference, August 8-10, 2004*, pages 339–345, Shanghai, 2005. Springer.

[159] P.M. Kort, J.P. Caulkins, R.F. Hartl, and G. Feichtinger. Brand image and brand dilution in the fashion industry. *Automatica*, 42:1363–1370, 2006.

[160] P. Krämer-Eis. *Ein Mehrzielverfahren zur numerischen Berechnung optimaler Feedback–Steuerungen bei beschränkten nichtlinearen Steuerungsproblemen*, volume 166 of *Bonner Mathematische Schriften*. Universität Bonn, Bonn, 1985.

[161] P.M. Krämer-Eis, H.G. Bock, R.W. Longman, and J.P. Schlöder. Numerical determination of optimal feedback control in nonlinear problems with state/control constraints. *Advances in the Astronautical Sciences*, 105:53–71, 2000.

[162] L.F.S. Larsen, R. Izadi-Zamanabadi, R. Wisniewski, and C. Sonntag. Supermarket refrigeration systems – a benchmark for the optimal control of hybrid systems. Technical report, Technical report for the HYCON NoE., 2007. http://www.bci.tu-dortmund.de/ast/hycon4b/index.php.

[163] D. Lebiedz, S. Sager, H.G. Bock, and P. Lebiedz. Annihilation of limit cycle oscillations by identification of critical phase resetting stimuli via mixed-integer optimal control methods. *Physical Review Letters*, 95:108303, 2005.

[164] H.W.J. Lee, K.L. Teo, L.S. Jennings, and V. Rehbock. Control parametrization enhancing technique for optimal discrete-valued control problems. *Automatica*, 35(8):1401–1407, 1999.

[165] H.W.J. Lee, K.L. Teo, V. Rehbock, and L.S. Jennings. Control parametrization enhancing technique for time-optimal control problems. *Dynamic Systems and Applications*, 6:243–262, 1997.

[166] J. Lee, J. Leung, and F. Margot. Min-up/min-down polytopes. *Discrete Optimization*, 1:77–85, 2004.

[167] D.B. Leineweber. *Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models*, volume 613 of *Fortschritt-Berichte VDI Reihe 3, Verfahrenstechnik*. VDI Verlag, Düsseldorf, 1999.

[168] D.B. Leineweber, I. Bauer, H.G. Bock, and J.P. Schlöder. An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part I: Theoretical aspects. *Computers and Chemical Engineering*, 27:157–166, 2003.

[169] D.B. Leineweber, I. Bauer, A.A.S. Schäfer, H.G. Bock, and J.P. Schlöder. An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization (Parts I and II). *Computers and Chemical Engineering*, 27:157–174, 2003.

[170] F. Logist, S. Sager, C. Kirches, and J.F. van Impe. Efficient multiple objective optimal control of dynamic systems with integer controls. *Journal of Process Control*, 20(7):810–822, August 2010.

[171] J. London. *John Barleycorn*. The Century Company, New York, 1st edition, 1913.

[172] M. Margaliot. A counterexample to a conjecture of Gurvits on switched systems. *IEEE Transactions on Automatic Control*, 52(6):1123–1126, 2007.

[173] A. Martin, T. Achterberg, T. Koch, and G. Gamrath. Miplib - mixed integer problem library. http://miplib.zib.de/.

[174] A. Martin, M. Möller, and S. Moritz. Mixed integer models for the stationary case of gas network optimization. *Mathematical Programming*, 105:563–582, 2006.

[175] J. Mattingley and S. Boyd. Automatic code generation for real-time convex optimization. In Y. Eldar and D.P. Palomar, editors, *Convex Optimization in Signal Processing and Communications*. Cambridge University Press, 2010.

[176] H. Maurer, C. Büskens, J.H.R. Kim, and Y. Kaya. Optimization methods for the verification of second-order sufficient conditions for bang-bang controls. *Optimal Control Methods and Applications*, 26:129–156, 2005.

[177] H. Maurer and N.P. Osmolovskii. Second order sufficient conditions for time-optimal bang-bang control. *SIAM Journal on Control and Optimization*, 42:2239–2263, 2004.

[178] G.P. McCormick. Computability of global solutions to factorable nonconvex programs. part I. Convex underestimating problems. *Mathematical Programming*, 10:147–175, 1976.

[179] B. Meyer and W. Scholl. Complex problem solving after unstructured discussion. Effects of information distribution and experiece. *Group Process and Intergroup Relations*, 12:495–515, 2009.

[180] J.A. Nelder and R. Mead. A simplex method for function minimization. *Computer Journal*, 7:308–313, 1965.

[181] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Verlag, Berlin Heidelberg New York, 2nd edition, 2006. ISBN 0-387-30303-0 (hardcover).

[182] European Network of Excellence Hybrid Control. Website. http://www.ist-hycon.org/.

[183] J. Oldenburg. *Logic–based modeling and optimization of discrete–continuous dynamic systems*, volume 830 of *Fortschritt-Berichte VDI Reihe 3, Verfahrenstechnik*. VDI Verlag, Düsseldorf, 2005.

[184] J. Oldenburg and W. Marquardt. Disjunctive modeling for optimal control of hybrid systems. *Computers and Chemical Engineering*, 32(10):2346–2364, 2008.

[185] J. Oldenburg, W. Marquardt, D. Heinz, and D.B. Leineweber. Mixed logic dynamic optimization applied to batch distillation process design. *AIChE Journal*, 49(11):2900–2917, 2003.

[186] M. Osman. Observation can be as effective as action in problem solving. *Cognitive Science: A Multidisciplinary Journal*, 32(1):162–183, 2008.

[187] J.H. Otto and Lantermann E.-D. Wahrgenommene Beeinflussbarkeit von negativen Emotionen, Stimmung und komplexes Problemlösen. *Zeitschrift für Differentielle und Diagnostische Psychologie*, 25:31–46, 2004.

[188] I. Papamichail and C.S. Adjiman. Global optimization of dynamic systems. *Computers and Chemical Engineering*, 28:403–415, 2004.

[189] H.J. Pesch and R. Bulirsch. The maximum principle, Bellman's equation and Caratheodory's work. *Journal of Optimization Theory and Applications*, 80(2):203–229, 1994.

[190] A. Pietrus and V. M. Veliov. On the discretization of switched linear systems. *Systems & Control Letters*, 58:395–399, 2009.

[191] K.J. Plitt. Ein superlinear konvergentes Mehrzielverfahren zur direkten Berechnung beschränkter optimaler Steuerungen. Diploma thesis, Rheinische Friedrich–Wilhelms–Universität Bonn, 1981.

[192] L.S. Pontryagin, V.G. Boltyanski, R.V. Gamkrelidze, and E.F. Miscenko. *The Mathematical Theory of Optimal Processes*. Wiley, Chichester, 1962.

[193] A. Potschka, H.G. Bock, and J.P. Schlöder. A minima tracking variant of semi-infinite programming for the treatment of path constraints within direct solution of optimal control problems. *Optimization Methods and Software*, 24(2):237–252, 2009.

[194] M.J.D. Powell. Algorithms for nonlinear constraints that use Lagrangian functions. *Mathematical Programming*, 14(3):224–248, 1978.

[195] Adrian Prata, Jan Oldenburg, Andreas Kroll, and Wolfgang Marquardt. Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor. *Computers and Chemical Engineering*, 32:463–476, 2008.

[196] F. Pukelsheim. *Optimal Design of Experiments*. Classics in Applied Mathematics 50. SIAM, 2006. ISBN 978-0-898716-04-7.

[197] W. Putz-Osterloh. Über die Beziehung zwischen Testintelligenz und Problemlöseerfolg. *Zeitschrift für Psychologie*, 189:79–100, 1981.

[198] W. Putz-Osterloh, B. Bott, and K. Köster. Models of learning in problem solving – are they transferable to tutorial systems? *Computers in Human Behavior*, 6:83–96, 1990.

[199] V. Rehbock and L. Caccetta. Two defence applications involving discrete valued optimal control. *ANZIAM Journal*, 44(E):E33–E54, 2002.

[200] T.W. Robbins, E.J. Anderson, D. R. Barker, A.C. Bradley, C. Fearnyhough, R. Henson, S.R. Hudson, and A. Baddeley. Working memory in chess. *Memory & Cognition*, 24(1):83–93, 1996.

[201] R.T. Rockafellar. *Tutorials in operations research: OR tools and applications : glimpses of future technologies*, chapter Coherent Approaches to Risk in Optimization Under Uncertainty, pages 38–61. INFORMS, 2007.

[202] S. Sager. MIOCP benchmark site. http://mintoc.de.

[203] S. Sager. *Numerical methods for mixed–integer optimal control problems*. Der andere Verlag, Tönning, Lübeck, Marburg, 2005. ISBN 3-89959-416-9.

[204] S. Sager. Reformulations and algorithms for the optimization of switching decisions in nonlinear optimal control. *Journal of Process Control*, 19(8):1238–1247, 2009.

[205] S. Sager. A benchmark library of mixed-integer optimal control problems. In *Proceedings MINLP09*, 2011. (accepted).

[206] S. Sager. Sampling decisions in optimum experimental design in the light of Pontryagin's maximum principle. *SIAM Journal on Control and Optimization*, 2012. (submitted).

[207] S. Sager, C. Barth, H. Diedam, M. Engelhart, and J. Funke. Optimization as an analysis tool for human complex problem solving. *SIAM Journal on Optimization*, 2011. (accepted).

[208] S. Sager, H.G. Bock, and M. Diehl. The integer approximation error in mixed-integer optimal control. *Mathematical Programming A*, 2011. DOI 10.1007/s10107-010-0405-3.

[209] S. Sager, H.G. Bock, M. Diehl, G. Reinelt, and J.P. Schlöder. Numerical methods for optimal control with binary control functions applied to a Lotka-Volterra type fishing problem. In A. Seeger, editor, *Recent Advances in Optimization*, volume 563 of *Lectures Notes in Economics and Mathematical Systems*, pages 269–289, Heidelberg, 2009. Springer. ISBN 978-3-5402-8257-0.

[210] S. Sager, H. Diedam, and M. Engelhart. Tailorshop: Optimization Based Analysis and data Generation tOol. TOBAGO web site https://sourceforge.net/projects/tobago.

[211] S. Sager, M. Diehl, G. Singh, A. Küpper, and S. Engell. Determining SMB superstructures by mixed-integer control. In K.-H. Waldmann and U.M. Stocker, editors, *Proceedings OR2006*, pages 37–44, Karlsruhe, 2007. Springer.

[212] S. Sager, M. Jung, and C. Kirches. Combinatorial integral approximation. *Mathematical Methods of Operations Research*, 73(3):363–380, 2011.

[213] S. Sager, C. Kirches, and H.G. Bock. Fast solution of periodic optimal control problems in automobile test-driving with gear shifts. In *Proceedings of the 47th IEEE Conference on Decision and Control (CDC 2008), Cancun, Mexico*, pages 1563–1568, 2008. ISBN: 978-1-4244-3124-3.

[214] S. Sager, G. Reinelt, and H.G. Bock. Direct methods with maximal lower bound for mixed-integer optimal control problems. *Mathematical Programming*, 118(1):109–149, 2009.

[215] K. Schittkowski. Test problems for nonlinear programming - user's guide. Technical report, Department of Mathematics, University of Bayreuth, 2002.

[216] K. Schittkowski. Experimental design tools for ordinary and algebraic differential equations. *Mathematics and Computers in Simulation*, 79(3):521–538, 2007.

[217] M. Schlegel and W. Marquardt. Detection and exploitation of the control switching structure in the solution of dynamic optimization problems. *Journal of Process Control*, 16:275–290, 2006.

[218] C. Schmid and L.T. Biegler. Quadratic programming methods for tailored reduced Hessian SQP. *Computers & Chemical Engineering*, 18(9):817–832, September 1994.

[219] J. Schöneberger, H. Arellano-Garcia, H. Thielert, S. Körkel, and G. Wozny. Optimal experimental design of a catalytic fixed bed reactor. In B. Braunschweig and X. Joulia, editors, *Proceedings of 18th European Symposium on Computer Aided Process Engineering - ESCAPE 18*, 2008.

[220] S.P. Sethi. Nearest feasible paths in optimal control problems: Theory, examples, and counterexamples. *Journal of Optimization Theory and Applications*, 23:563–579, 1977.

[221] S.P. Sethi. Optimal advertising policy with the contagion model. *Journal of Optimization Theory and Applications*, 29:615–627, 1979.

[222] S.P. Sethi and G.L. Thompson. *Optimal Control Theory: Applications to Management Science and Economics*. Springer, 2nd edition edition, 2005. ISBN-13: 978-0387280929.

[223] M.S. Shaikh. *Optimal Control of Hybrid Systems: Theory and Algorithms*. PhD thesis, Department of Electrical and Computer Engineering, McGill University, Montreal, Canada, 2004.

[224] M.S. Shaikh and P.E. Caines. On the hybrid optimal control problem: Theory and algorithms. *IEEE Transactions on Automatic Control*, 52:1587–1603, 2007.

[225] Y. Sharon and M. Margaliot. Third-order nilpotency, finite switchings and asymptotic stability. *Journal of Differential Equations*, 233:135–150, 2007.

[226] A. K. Skiba. Optimal growth with a convex-concave production function. *Econometrica*, 46:527–539, 1978.

[227] C. Sonntag, O. Stursberg, and S. Engell. Dynamic optimization of an industrial evaporator using graph search with embedded nonlinear programming. In *Proc. 2nd IFAC Conf. on Analysis and Design of Hybrid Systems (ADHS)*, pages 211–216, 2006.

[228] B. Srinivasan, S. Palanki, and D. Bonvin. Dynamic Optimization of Batch Processes: I. Characterization of the nominal solution. *Computers and Chemical Engineering*, 27:1–26, 2003.

[229] M.C. Steinbach. *Fast recursive SQP methods for large-scale optimal control problems*. PhD thesis, Ruprecht–Karls–Universität Heidelberg, 1995.

[230] S. Strohschneider and D. Güss. The fate of the moros: A cross-cultural exploration of strategies in complex and dynamic decision making. *International Journal of Psychology*, 34:235–252, 1999.

[231] H.-M. Süß, K. Oberauer, and M. Kersting. Intellektuelle Fähigkeiten und die Steuerung komplexer Systeme. *Sprache & Kognition*, 12:83–97, 1993.

[232] H.J. Sussmann. A maximum principle for hybrid optimal control problems. In *Conference proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, 1999.

[233] M. Szymkat and A. Korytowski. The method of monotone structural evolution for dynamic optimization of switched systems. In *IEEE CDC08 Proceedings*, 2008.

[234] N. Tauchnitz. *Das Pontrjaginsche Maximumprinzip für eine Klasse hybrider Steuerungsprobleme mit Zustandsbeschränkungen und seine Anwendung*. PhD thesis, BTU Cottbus, 2010.

[235] M. Tawarmalani and N. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Kluwer Academic Publishers, 2002.

[236] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.

[237] S. Terwen, M. Back, and V. Krebs. Predictive powertrain control for heavy duty trucks. In *Proceedings of IFAC Symposium in Advances in Automotive Control*, pages 451–457, Salerno, Italy, 2004.

[238] J. Till, S. Engell, S. Panek, and O. Stursberg. Applied hybrid system optimization: An empirical investigation of complexity. *Control Engineering Practice*, 12:1291–1303, 2004.

[239] The New York Times. Dim days for luxury hotels. The New York Times, October 28 2008. By J. Sharkey.

[240] R.J. Vanderbei. LOQO: An interior point code for quadratic programming. *Optimization Methods and Software*, 11(1–4):451–484, 1999.

[241] V. Veliov. Relaxation of Euler-type discrete-time control system. ORCOS 273, TU-Wien, 2003.

[242] V.M. Veliov. On the time discretization of control systems. *SIAM Journal of Control and Optimization*, 35(5):1470–1486, 1997.

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

[244] D. Wagener. Personalauswahl und -Entwicklung mit komplexen Szenarios. *Wirtschaftspsychologie*, 3:69–76, 2001.

[245] D. Wenke and P. A. Frensch. *Is success or failure at solving complex problems related to intellectual ability?*, pages 87–126. The psychology of problem solving. Cambridge University Press, 2003.

[246] R.C. Whaley and A. Petitet. Minimizing development and maintenance costs in supporting persistently optimized BLAS. *Software: Practice and Experience*, 35(2):101–121, February 2005.

[247] R. Winkler. Time-lagged accumulation of stock pollutants. Discussion paper No. 408, Alfred Weber-Institute of Economics, University of Heidelberg, 2004.

[248] R. Winkler, U. Brandt-Pollmann, U. Moslener, and J.P. Schlöder. Time-lags in capital accumulation. In D. Ahr, R. Fahrion, M. Oswald, and G. Reinelt, editors, *Operations Research Proceedings*, pages 451–458, 2003.

[249] R. Winkler, U. Brandt-Pollmann, U. Moslener, and J.P. Schlöder. On the transition from instantaneous to time-lagged capital accumulation: The case of leontief type production functions. ZEW Discussion Papers 05-30, 2005.

[250] L. Wirsching, J. Albersmeyer, P. Kühl, M. Diehl, and H.G. Bock. An adjoint-based numerical method for fast nonlinear model predictive control. In M.J. Chung and P. Misra, editors, *Proceedings of the 17th IFAC World Congress, Seoul, Korea, July 6–11, 2008*, volume 17, pages 1934–1939. IFAC-PapersOnLine, July 2008.

[251] W. W. Wittmann and K. Hattrup. The relationship between performance in dynamic systems and intelligence. *Systems Research and Behavioral Science*, 21:393–409, 2004.

[252] M.I. Zelikin and V.F. Borisov. *Theory of chattering control with applications to astronautics, robotics, economics and engineering*. Birkhäuser, Basel Boston Berlin, 1994.