Combinatorial Integral Decompositions for Mixed-Integer Optimal Control

Dissertation

zur Erlangung des akademischen Grades

doctor rerum naturalium

(Dr. rer. nat.)

von Clemens Ulrich Zeile, M.Sc. Mathematik

geb. am 30.03.1988 in Hamburg

genehmigt durch die Fakultät für Mathematik
der Otto–von–Guericke–Universität Magdeburg

Gutachter: Prof. Dr. Sebastian Sager
Prof. Dr. Christian Kirches
Prof. Dr. Sven Leyffer

eingereicht am: 03.02.2021

Verteidigung am: 30.04.2021

Clemens Ulrich Zeile
M.Sc. Mathematik

Institut für Mathematische Optimierung
Otto–von–Guericke–Universität Magdeburg
Gebäude 02
Universitätsplatz 2
39106 Magdeburg
Germany

clemens.zeile@ovgu.de

Mathematics Subject Classification (2010): 49K15; 34H05; 93C65; 90C59

Abstract

Many optimization problems in science, engineering, and medicine can be modeled by differential equations that can be steered via discrete control functions and are therefore called mixed-integer optimal control problems. The challenge of solving these problems lies in combining an infinite-dimensional optimization problem with discrete-valued optimization functions. After discretization, these problems become mixed-integer nonlinear programs for which the *combinatorial integral approximation* decomposition was proposed. The decomposition solves one nonlinear problem and one rounding problem formulated as a mixed-integer linear program.

This thesis generalizes and extends the combinatorial integral approximation decomposition algorithm. We define a framework that, through a sequence of several nonlinear optimization and rounding problems with increasing numbers of fixed integer variables, is designed to transfer feasibility from the obtained relaxed solution to the rounded solution. We derive several rounding problem versions based on different norms, the structure of the dynamical system, and the temporal ordering of the control approximation constraints. Based on the constructed integer control functions, we propose recombination strategies for generating promising candidate solutions with respect to the objective. For the combinatorial integral approximation decomposition in the unconstrained case, convergence to the optimal solution can be achieved by grid refinement. However, when refinement is not applicable or desirable, recombination methods are useful.

We provide an overview of a range of time-coupled combinatorial constraints that are common in practical applications. Specifically, we investigate decomposition algorithms for mixed-integer optimal control problems under minimum dwell time and bounded discrete total variation constraints. Typically, state constraints also arise in application-driven problems. We therefore propose methods for incorporating information from the nonlinear problem step into the rounding problem constraints to generate state constraint feasible integer control solutions. Independent of the additional constraints, we derive different, partly approximate algorithms to solve the mixed-integer linear rounding problem and perform a theoretical analysis by proving tight bounds in terms of the integral deviation gap.

Efficient software is indispensable for problem solving in practice. In this context, we present the package *pycombina*, which provides solution algorithms for the mixed-integer linear problem. Computational results from benchmark problems and real-world case studies of a hybrid electric vehicle and a heart assist device system highlight the relevance and applicability of the proposed algorithms.

Zusammenfassung

Viele Optimierungsprobleme aus den Natur- und Ingenieurwissenschaften sowie der Medizin können mit Differentialgleichungen modelliert werden, die über diskrete Steuerungsfunktionen geregelt werden können und daher als gemischt-ganzzahlige Optimalsteuerungsprobleme bezeichnet werden. Die Herausforderung bei der Lösung dieser Probleme liegt in der Kombination eines unendlich-dimensionalen Optimierungsproblems auf der einen Seite und diskretwertigen Optimierungsfunktionen auf der anderen Seite. Diese Probleme werden nach der Diskretisierung zu gemischt-ganzzahligen nichtlinearen Problemen. Für deren Lösung wurde wiederum die *combinatorial integral approximation* Dekomposition vorgeschlagen, welche aus dem Lösen eines nichtlinearen Optimierungsproblems und eines Rundungsproblems, das als ein gemischt-ganzzahliges lineares Problem formuliert werden kann, besteht.

Diese Arbeit verallgemeinert und erweitert den Dekompositions-Algorithmus in vielerlei Hinsicht. Wir definieren einen algorithmischen Rahmen, der durch eine Folge von mehreren nichtlinearen Optimierungs- und Rundungsproblemen mit zunehmender Anzahl von fixierten ganzzahligen Variablen die Zulässigkeit der konstruierten relaxierten zur gerundeten Lösung übertragen soll. Wir leiten mehrere Rundungsproblemversionen ab, die auf verschiedenen Normen, der Struktur des dynamischen Systems und der zeitlichen Anordnung der Nebenbedingungen der Steuerungs-Approximiation beruhen. Auf der Grundlage der konstruierten ganzzahligen Steuerungsfunktionen schlagen wir Rekombinationsstrategien vor, um vielversprechende Kandidatenlösungen im Hinblick auf die Zielfunktion zu generieren. Während im unbeschränkten Fall für den Dekompositionsalgorithmus eine Konvergenz zur optimalen Lösung durch Gitterverfeinerung bewiesen werden kann, sind Rekombinationsstrategien nützlich, wenn eine Verfeinerung nicht anwendbar oder erwünscht ist.

Wir geben einen Überblick über eine Reihe von zeitgekoppelten kombinatorischen Nebenbedingungen, die in vielen praktischen Anwendungen üblich sind. Insbesondere untersuchen wir Dekompositionsalgorithmen für gemischt-ganzzahlige optimale Steuerungsprobleme unter minimaler Verweilzeit und einer beschränkten Anzahl erlaubter Wechsel der aktiven diskreten Steuerungsfunktion. Typischerweise treten auch Beschränkungen der differentiellen Zustände bei anwendungsgetriebenen Problemen auf. Für die Konstruktion zulässiger ganzzahliger Steuerungslösungen schlagen wir Methoden vor, um Informationen aus dem nichtlinearen Problemschritt in das Rundungsproblem einzubeziehen. Unabhängig von den zusätzlichen Nebenbedingungen leiten wir verschiedene, teilweise approximative Algorithmen zur Lösung des gemischt-ganzzahligen linearen Rundungsproblems her und führen eine theoretische Analyse durch, indem wir scharfe Schranken in Bezug auf den Rundungsfehler beweisen.

Effiziente Software ist für die Problemlösung in der Praxis unumgänglich. In diesem Kontext stellen wir das Paket *pycombina* vor, das Lösungsalgorithmen für das gemischt-ganzzahlige lineare Problem bereitstellt und im Rahmen eines Gemeinschaftsprojekts mit anderen Forschern entwickelt wurde. Numerische Ergebnisse aus Benchmark-Problemen und realen Fallstudien aus einem Hybrid-Elektrofahrzeug und einem Herzunterstützungssystem heben die Relevanz und Anwendbarkeit der diskutierten Algorithmen hervor.

Contents

Chapter 1

Introduction

As a method for formulating and solving complex problems, mathematical optimization can serve as a comprehensive framework for providing decision support. The usual optimization procedure is to specify an objective function that is to be minimized and to identify constraints that limit the scope of action. An example of such an approach which is further investigated in this thesis is "How should we steer a given vehicle so that we use as little fuel as possible on a given route?". Physical limitations of the vehicle represent the optimization constraints.

This thesis follows a model-driven optimization approach: we assume that the processes underlying a problem can be mapped into mathematical relationships. In fact, numerous (dynamic) processes in various fields of science [262] and engineering [237] are well described by ordinary differential equations (ODEs) that quantify time-varying behaviors. In the above example, the vehicle dynamics can be characterized by a dynamical system based on ODEs. In the optimization context, it is natural to ask how these processes can be optimally controlled by external input functions, thus giving rise to optimal control theory and optimal control problems (OCPs).

A particular subclass of OCPs represent mixed-integer optimal control problems (MIOCPs) in which the dynamical process exhibits a discrete nature because the external control function only assumes a finite number of values. Typically, discrete control functions express on-off switching decisions or specific configurations of machines. The abruptly changing dynamics of MIOCPs can also result from switching events that occur when a specific differential state attains a threshold value. The switches may occur at any time point in the given time horizon. Discrete choice examples related to the vehicle driving optimization problem include "Should we propel the vehicle via the combustion engine or the electric motor?" and "Which gear should we choose?". Another mixed-integer optimal control (MIOC) application investigated in this thesis is the control of left ventricular assist devices (LVADs) where the dynamical system switches based on the opening and closing of the heart valves or by varying the piecewise constant rotational pump speed of the heart assist device.

Efficient and accurate solution methods are crucial for optimization problems in general and for MIOCPs specifically. The class of MIOCPs is challenging because it combines the difficulties of several optimization disciplines, namely integer and nonlinear programming as well as (continuous) optimal control theory. A useful approach to optimization problems is complexity reduction via decomposition, which refers to breaking up a complex problem into smaller ones and then solving the smaller problems separately. The advantage of decomposition stems from the fact that problem complexity grows more than linearly with size. Solving the smaller problems is therefore more efficient, albeit at the expense of a possible loss of optimality. Time-discretized MIOCPs result in mixed-integer nonlinear programs (MINLPs) [160], which are known to be generally *NP* hard [21], making decomposition relevant. Sager [218] proposed solving the partially outer convexified [72] MINLP, in which the integrality constraints are relaxed, making it an nonlinear program (NLP) problem. The obtained relaxed control values can be approximated with integral values via the sum-up rounding (SUR) [218] algorithm. For the second step of the decomposition, Sager, Jung, and Kirches [224] proposed solving the so-called combinatorial integral approximation (CIA) problem, which constitutes an mixed-integer linear program (MILP), instead of applying SUR. We call the solution method based on outer convexification, relaxation, and subsequent projection to integer controls the *CIA decomposition*. We write (CIA) for denoting the rounding subproblem of the decomposition.

The CIA decomposition has attracted attention for its numerical efficiency and for a convergence result: the constructed solution can be made arbitrarily close to the optimal solution by refining the discretization grid. This doctoral thesis investigates the algorithm from an application-oriented perspective, for which time-coupled combinatorial constraints are usually required. In the vehicle optimization context, these constraints include that the combustion engine must stay on for at least a few seconds after being switched on and that only certain gears can be selected given the current active gear. Another common feature of application-driven MIOCPs is that refinement of the discretization may be impractical for huge problem instances or time-critical solution requirements, as in model predictive control (MPC). We therefore propose algorithms that can create feasible and near-optimal solutions for a fixed discretization of a given MIOCP, and we quantify the resulting approximation errors.

1.1 Contributions

This thesis contributes to the theory and numerical methods of ODE constrained MIOCPs. We take the application-driven perspective in which combinatorial constraints on the integer controls play a major role. For this purpose, we propose several problem-specific and efficient algorithms as variants of the CIA decomposition. We describe the novel methods and results in the following.

A generalized CIA decomposition framework

Thus far, the CIA decomposition has mainly been considered an algorithm that involves two subsequent problems [218, 219, 145, 149]. In a few cases, three subproblems have been considered, where an NLP with fixed integer controls is the third step [135]. We generalize and extend the decomposition idea by defining a framework of multiple subsequent NLP and CIA problem steps. We derive different versions of the (CIA) problem, leading to multiple MILPs. The different (CIA) problem versions are based on different norms, the scaling information of the dynamical system, and the temporal ordering of the control approximation constraints. The solutions of the problem versions are themselves candidate solutions, and they can be recombined into new switching sequences. We propose several strategies for generating promising candidate solutions for the binary control functions in the original problem. These extensions are designed to construct a feasible solution with a near-optimal objective value for complex MIOCPs that entail multiple constraints. Sager [225] established that the quality of the solution obtained from the CIA decomposition can be improved by refining the discretization grid. The proposed generalization is particularly beneficial when refinement of the problem discretization is not applicable. Nevertheless, we prove that the established convergence result still holds for the extended CIA decomposition.

Mixed-integer optimal control under switching constraints

In this thesis, we discuss a broad range of switching restrictions on the integer control that arise in real-world application problems. For instance, tailored MIOC policies are usually required to avoid rapid successive changes of the active integer control. This requirement is expressed by minimum dwell time (MDT) constraints, which can be further identified as minimum up time and down time constraints. Another way to avoid unrealistic frequent switching is to constrain the number of switches between active integer controls. Such bounded discrete total variation (TV) constraints have been already included into the CIA problem [224, 137, 135]. Finally, we discuss further switching restrictions on the integer control, such as mode transition constraints.

Incorporation of path constraint information from the NLP into the (CIA) problem step

The solution obtained from the CIA decomposition may be infeasible with respect to path constraints on the differential states of the MIOCP. It has been established that the constraint violation can be made arbitrarily small by refining the discretization grid [225, 149]. To extend the CIA decomposition without grid refinement, we propose methods for incorporating the state constraint information from the nonlinear problem step into the rounding problem constraints to generate state constraint feasible integer control solutions. We propose forward integration of the differential states as part of a branch-and-bound (BnB) algorithm so that compliance with the path constraint can be checked directly. Moreover, we derive constraints for the (CIA) problem based on a first-order Taylor approximation of the path constraints.

Algorithms for solving (CIA) problems

The classic SUR [218] provides an approximate way to solve a (CIA) problem with the maximum norm and without combinatorial constraints on the integer controls. The (CIA) problem can also be formulated as an MILP; for its solution, Jung proposed and implemented an efficient BnB algorithm [224, 137]. We derive an extended formulation variant of this MILP that is based on the introduction of variables for tracking the switches of active controls. Furthermore, we introduce several rounding schemes for solving the (CIA) problem. Specifically, we propose dwell time sum-up rounding (DSUR) and dwell time next-forced rounding (DNFR) for MIOCPs under MDT constraints and adaptive maximum dwell rounding (AMDR) when the number of allowed switches is limited.

Integral deviation gap results and analysis of (CIA) problem solution algorithms

We investigate the integral deviation gap of the (CIA) problem based on the maximum norm. In this context, we prove tight upper bounds for the (CIA) problem without combinatorial constraints, with MDT, and with limited switching constraints. We also analyze the approximation error resulting from the different proposed rounding schemes. The derivation of the bounds is accompanied by results on the problem and run time complexity, for which we establish a link to scheduling theory.

Implementations for the solution of (CIA) problems

All algorithms for the solution of (CIA) problems that we introduce and discuss in this thesis, in particular BnB and rounding schemes, are implemented in the open-source software tool `pycombina`, which provides a comprehensive framework for formulating and solving (CIA) problems. Significant parts of the package and the user interface are written in Python, while the BnB algorithm relies on an efficient C++ implementation.

Moreover, we implemented the postprocessing heuristics and different (CIA) problem variants in AMPL [79] using the code `ampl_mintoc`, which is a modeling framework for handling OCPs.

Benchmarking the algorithms via case studies

The efficacy of the proposed algorithms and problem models is demonstrated via an adsorption cooling machine problem [48, 49], the Lotka-Volterra fishing problem, the Egerstedt standard problem, a three-tank flow system problem, and further MIOCPs from the benchmark library `https://mintOC.de` [221]. We illustrate our findings and discuss best practice usage of the CIA decomposition.

Multiphase mixed-integer optimal control of hybrid electric vehicles

In recent years, hybrid electric vehicles (HEVs) have become more common since they can reduce greenhouse gas emissions and fuel consumption while providing a high-quality ride. We consider the problem of computing a non-causal minimum-fuel energy management strategy for a given driving cycle of an HEV. When searching for the optimal gear choice, torque split, and engine on/off controls during off-line evaluations, the problem can be formulated as a multiphase MIOCP. We propose an efficient model by introducing vanishing constraints and a phase-specific right-hand side function that accounts for the different powertrain operating modes. The gearbox and drivability requirements are translated into combinatorial constraints that have not been included in previous research; however, they are part of the algorithmic framework for this investigation. Numerical experiments were performed to illustrate the proposed tailored CIA decomposition algorithm in terms of its solution quality and run time.

Mixed-integer optimal pump speed control of ventricular assist devices

A promising therapy for patients with congestive heart failure is the implementation of a left ventricular assist device that works as a mechanical pump. Modern devices work with a constant rotor speed and therefore, provide continuous blood flow; however, there have been recent attempts to generate pulsatile blood flow by oscillating the pump speed. We propose an MIOCP framework for constructing and evaluating optimal pump speed policies. We consider implicit switches enforced by system changes, such as valves opening and closing, and explicit system switches that stem from varying the constant pump speed. The developed algorithm can also be used to adapt the underlying model to patient-specific data. We suggest basing the *in silico* analysis on a model that captures the atrioventricular plane displacement, a physiological indicator of heart failure. As a proof-of-concept study, we personalize the model for a selected patient and present numerical results of the constructed optimal pump speed policies.

1.2 Contributions to publications

Major parts of this work are based on publications to which the author of this thesis made significant contributions. In the following, we list these articles and indicate their connection with the chapters of this thesis. Moreover, the contributions of the author of this thesis to the respective publications are summarized.

[282] C. Zeile, N. Robuschi, and S. Sager. Mixed-integer optimal control under minimum dwell time constraints. *Mathematical Programming*, pages 1–42, 2020. doi: https://doi.org/10.1007/s10107-020-01533-x

All chapters associated with MDT constraint methods are based on this publication, namely, Sections 6.5, 6.6, 7.2, 7.3, 7.4, and 9.3. As the main author, C. ZEILE developed the algorithmic ideas as well as the mathematical proofs and worked out the computational results and the first draft of the article. N. ROBUSCHI and S. SAGER contributed with discussions and reviewed the paper before submission.

[222] S. Sager and C. Zeile. On mixed-integer optimal control with constrained total variation of the integer control. *Computational Optimization and Applications*, 2020. doi: 10.1007/s10589-020-00244-5

Sections 6.7, 7.5, and 9.4 deal with models and approaches for limiting the number of switches allowed between active controls and are based on the above article. The study was designed and conducted by C. ZEILE as the main author. S. SAGER contributed to discussions and writing the final manuscript.

[280] C. Zeile, T. Weber, and S. Sager. Combinatorial integral approximation decompositions for mixed-integer optimal control. Technical report, 2018. (preprint available under `http://www.optimization-online.org/DB_HTML/2018/02/6472.html`)

This paper develops generalizations of the CIA decomposition to different MILP formulations and recombination strategies. Major parts of Chapters 4, 5, 6.1, and 9.1 are based on this study. C. ZEILE, the main author of this paper, conducted the implementations and the numerical study, and he was responsible for writing most of the article. T. WEBER contributed major algorithmic ideas and proofread the manuscript. S. SAGER contributed to the research design and discussions of the algorithmic ideas and wrote parts of the paper.

[48] A. Bürger, C. Zeile, Altmann-Dieses, S. A., Sager, and M. Diehl. An algorithm for mixed-integer optimal control of solar thermal climate systems with MPC-capable runtime. In *2018 European Control Conference (ECC)*, pages 1379–1385. IEEE, 2018

[49] A. Bürger, C. Zeile, Altmann-Dieses, S. A., Sager, and M. Diehl. Design, implementation and simulation of an MPC algorithm for switched nonlinear systems under combinatorial constraints. *Journal of Process Control*, 81:15–30, 2019

[50] A. Bürger, C. Zeile, M. Hahn, A. Altmann-Dieses, S. Sager, and M. Diehl. pycombina: An open-source tool for solving combinatorial approximation problems arising in mixed-integer optimal control. In *Proceedings of the IFAC World Congress*, 2020. accepted

These publications present the application of an MIOCP to solar thermal cooling in buildings and the software package `pycombina` for solving CIA problems. Parts of Section 6.1 and Chapter 8 describe ideas and methods from these three papers. In these publications, A. BÜRGER, the main author, designed the study and the mathematical models, and performed the numerical tests. C. ZEILE, the second author, contributed to algorithmic ideas and wrote minor parts of the articles. Furthermore, he was involved in general discussions of the papers and the implementations of `pycombina`. The other co-authors contributed to discussions and proofreading of the papers.

[211] N. Robuschi, C. Zeile, S. Sager, and F. Braghin. Multiphase mixed-integer nonlinear optimal control of hybrid electric vehicles. *Automatica*, 123:109325, 2021. doi: https://doi.org/10.1016/j.automatica.2020.109325

Section 10.1 is based on this article. This work resulted from the research visit of N. ROBUSCHI in Magdeburg in the summer of 2018. N. ROBUSCHI and C. ZEILE wrote the article together, and therefore share the first authorship. While N. ROBUSCHI was responsible for the engineering-related aspects, such as the conceptual aim and design of the study as well as the powertrain modeling, C. ZEILE's role was to advance the mathematical aspects. N. ROBUSCHI performed major parts of the numerical results study, to which C. ZEILE adapted the software package `pycombina`. S. SAGER and F. BRAGHIN contributed to discussions and reviewed the paper before submission.

[281] C. Zeile, T. Rauwolf, A. Schmeisser, J. Mizerski, R. C. Braun-Dullaeus, and S. Sager. A personalized switched systems approach for the optimal control of ventricular assist devices based on atrioventricular plane displacement. *IEEE Transactions on Biomedical Engineering*, 2020. submitted

Section 10.2 is based on this publication. C. ZEILE, as the main author, proposed the study design, performed the numerical computations, and wrote the manuscript. T. RAUWOLF and J. K. MIZERSKI contributed to interpreting results, the medical background, and general discussions on the study design. The clinical data for the experiments were provided by T. RAUWOLF and A. SCHMEISSER. All authors contributed to writing the final manuscript.

[114] M. Hahn, C. Kirches, P. Manns, S. Sager, and C. Zeile. Decomposition and approximation for PDE-constrained mixed-integer optimal control. In M. H. et al., editor, *SPP1962 Special Issue*. Birkhäuser, 2019. (accepted)

Since this publication reviews the CIA decomposition for partial differential equation (PDE)-constrained problems, it only overlaps slightly with this thesis, but Chapter 4 shares some of its ideas. The main contributions of C. ZEILE were Chapters 3.2 and 4 and discussions of the general study design. While P. MANNS contributed to Chapters 1, 2, and 3.1, M. HAHN performed major parts of the numerical experiments and wrote the corresponding chapter. C. KIRCHES and S. SAGER contributed to discussions and reviewed the paper before submission.

[278] C. Zeile, E. Scholz, and S. Sager. A simplified 2D heart model of the Wolff-Parkinson-White syndrome. In *Proceedings of the Foundations of Systems Biology in Engineering (FOSBE) Conference*, volume 49, pages 26–31. Magdeburg, Germany, Elsevier, 2016

[279] C. Zeile, T. Rauwolf, A. Schmeisser, T. Weber, and S. Sager. The influence of right ventricular afterload in cardiac resynchronization therapy: A circadapt study. In *Computing in Cardiology 2017 -PapersOnLine Proceedings*, 2017

These publications resulted from a project on modeling the human heart that the author of this thesis worked on during the course of his doctoral studies. However, they are not directly related to MIOCP and are therefore not included in this thesis.

1.3 Thesis outline

We relate the chapters of this work to the individual contributions described above and categorize the publications according to their thematic content. This subdivision facilitates a proper overview of certain aspects of the thesis, such as the algorithms or the theoretical analysis as a whole. This doctoral thesis is divided into three parts with eleven chapters in total and one appendix.

First, **Part I** introduces the problem class of MIOCPs and its background. We summarize relevant aspects and methods from optimal control theory in **Chapter 2**. We continue with an overview of the research field of MIOCPs in **Chapter 3**. To this end, we define a generic problem class in Section 3.1 and provide a survey of relevant numerical methods in Section 3.4.

Part II provides the algorithmic framework and a theoretical analysis of the CIA decomposition. **Chapter 4** addresses the different algorithm versions of the CIA decomposition. In particular, Sections 4.1 and 4.2 respectively define the partial outer convexification technique and the steps of the basic decomposition. We review the contributions that led to this algorithm in Section 4.3 and discuss generalizations and extensions in Sections 4.4 and 4.5.

Chapter 5 analyzes approximation properties of the solutions constructed by the CIA decomposition to the optimal solution. In particular, we discuss the inherited properties of the algorithmic extensions.

The principal focus of this thesis is the solution of (CIA) problems, which we address in **Chapter 6**. We present a problem size reduction heuristic based on singular arcs in Section 6.1. Complexity results and a novel link to scheduling theory reveal insights into the (CIA) problem's nature and are given in Section 6.2. Based on these results, we present a method for solving the MILP to optimality without additional combinatorial constraints via a simple algorithm. Section 6.3 describes a BnB method established in [224, 137] with time-dependent mode variables. We consider extended formulations of the (CIA) problem in Section 6.4, where we also elucidate their advantages. A widely used method for obtaining fast, robust (CIA) problem solutions is the SUR algorithm. In Section 6.5, we introduce this rounding family and discuss extensions to the MDT setting. Next-forced rounding is another approach that we generalize to the DNFR algorithm in Section 6.6. We propose solving (CIA) problems with MDT or limited switching constraints via AMDR, which is a fast heuristic algorithm that is presented in Section 6.7. Section 6.8 reviews other solution methods. Finally, we summarize this chapter, which contains abundant solution methods, in Section 6.9.

An appropriate way to provide a theoretical justification of the novel solution methods is to analyze their integral deviation gap, which we do in **Chapter 7**. After deriving auxiliary lemmata in Section 7.1, we derive bounds on this rounding error for the constructed solutions of the SUR and DNFR algorithms in Sections 7.2 and 7.3, respectively. The DNFR scheme gives rise to insights into bounds for the (CIA) problem itself and under MDT constraints, as highlighted in

Section 7.4. The situation with a limited number of allowed switches is highly complex, but we are able to deduce bounds using properties of the AMDR scheme in Section 7.5. We summarize the obtained results in Section 7.6.

In **Part III**, we consider practical implementation of the CIA decomposition. This first involves efficient software, which is the focus of **Chapter 8** where we introduce `pycombina`.

In **Chapter 9**, we present results from numerical benchmark computations. We first provide the results for different (CIA) problem variants and postprocessing heuristics in Section 9.1. Then, our algorithmic idea for the incorporation of path constraint information is presented in Section 9.2. Finally, we illustrate and discuss our findings for MIOCPs with MDT and bounded discrete total variation constraints in Sections 9.3 and 9.4.

This application-driven thesis is enriched by two realistic case studies, which are included in **Chapter 10**. In this context, Section 10.1 presents an MIOCP from the automotive field, where the minimum fuel policy of a hybrid electric vehicle is sought for a given driving cycle. We also provide a case study from cardiology in Section 10.2, where the problem of finding an optimal pump speed control for a heart assist device is formulated as an MIOCP.

The results and implications of this thesis are reviewed in **Chapter 11**. We also give an outlook for this work, where we discuss future work. We provide function space definitions in Appendix A.

Part I

Background and problem class

Chapter 2

Concepts and methods from optimal control theory

Since we base our investigations of mixed-integer optimal control problems (MIOCPs) on optimality concepts and methods from optimal control theory, this chapter introduces the fundamentals of optimal control problems (OCPs). To this end, we define the general problem class in Section 2.1, and we address optimality concepts and problem properties in Section 2.2. There are three main solution methods for OCPs, which are introduced in Section 2.3. An overview of the direct method, which is the method of choice for the combinatorial integral approximation (CIA) decomposition algorithm, is provided. For greater detail about the theory and methods of OCPs, we refer to [151, 218, 145]. The structure and content of this chapter follow those of [135].

2.1 Definition of the continuous optimal control problem

We consider a given, fixed time horizon $\mathscr{T} := [t_0, t_f] \subset \mathbb{R}$. With problem-specific dimensions for the *differential states* n_x, *control function* n_u, and *mixed control–state constraint* n_c, we utilize the following functions:

$$
\begin{aligned}
\Phi:&\quad \mathbb{R}^{n_\mathrm{x}} \;\to\; \mathbb{R}, && (2.1)\\
\Psi:&\quad \mathscr{T} \times \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}^{n_u} \;\to\; \mathbb{R}, && (2.2)\\
\boldsymbol{f}:&\quad \mathscr{T} \times \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}^{n_u} \;\to\; \mathbb{R}^{n_\mathrm{x}}, && (2.3)\\
\boldsymbol{c}:&\quad \mathscr{T} \times \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}^{n_u} \;\to\; \mathbb{R}^{n_c}. && (2.4)
\end{aligned}
$$

We assume that these functions are sufficiently smooth. Common regularity assumptions are $\Phi \in C^0(\mathbb{R}^{n_\mathrm{x}}, \mathbb{R})$, $\Psi \in C^0(\mathscr{T} \times \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}^{n_u}, \mathbb{R})$, $\boldsymbol{f} \in C^0(\mathscr{T} \times \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}^{n_u}, \mathbb{R}^{n_\mathrm{x}})$, and $\boldsymbol{c} \in C^1(\mathscr{T} \times \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}^{n_u}, \mathbb{R}^{n_c})$.

Definition 2.1 (Continuous optimal control problem)
Let the time horizon $\mathscr{T}$ and the functions $\Phi, \Psi, \boldsymbol{f}$, and $\boldsymbol{c}$ be given as above. The continuous optimal control problem (2.5) *is defined as*

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{u}} \quad & \mathscr{C}(\boldsymbol{x},\boldsymbol{u}) := \Phi(\boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} \Psi(t, \boldsymbol{x}(t), \boldsymbol{u}(t))\, \mathrm{d}t && (2.5a)\\
\text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \qquad \textit{for } t \in \mathscr{T}, && (2.5b)\\
& \mathbf{0}_{n_c} \le \boldsymbol{c}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \qquad \textit{for } t \in \mathscr{T}, && (2.5c)\\
& \boldsymbol{x}(t_0) = \boldsymbol{x}_0, && (2.5d)\\
& \boldsymbol{x}(t_f) = \boldsymbol{x}_f, && (2.5e)\\
& \boldsymbol{u} \in \mathscr{U}, && (2.5f)
\end{aligned}
$$

where $\boldsymbol{x}_0, \boldsymbol{x}_f \in \mathbb{R}^{n_x}$ denote the initial and terminal values for the dynamic process. We express the *feasible space of the control function $\boldsymbol{u}$ as $\mathscr{U} \subset L^{\infty}(\mathscr{T}, \mathbb{R}^{n_u})$. We minimize the cost functional $\mathscr{C}$ in* (2.5a) *over the control $\boldsymbol{u}$ and the differential state $\boldsymbol{x} \in W^{1,\infty}(\mathscr{T}, \mathbb{R}^{n_x})$. The latter is governed by a system of ordinary differential equations (ODEs) with right-hand side function $\boldsymbol{f}$ and affected by $\boldsymbol{u}$* (2.5b). *Mixed control–state constraints* (2.5c) *as well as boundary conditions* (2.5d)-(2.5e) *represent the restrictions of the problem.*

The objective functional $\mathscr{C}$ is of the so-called BOLZA type, consisting of a MAYER term Φ and LAGRANGE term Ψ. From Chapter 3 on, the cost function $\mathscr{C}$ and the LAGRANGE term Ψ will also depend on an integer control.

We defined the generic OCP with explicit dependency of the constraint and objective functions on the time t. Such problems are referred to as *non-autonomous*. We can equivalently reformulate the dynamic system to eliminate the explicit dependency on t by introducing an additional state, allowing for the consideration of *autonomous* problems.

It is typical to require $\boldsymbol{f}$ to be LIPSCHITZ continuous in order to ensure the existence and uniqueness of the solution of the dynamic system by the PICARD-LINDELÖF theorem [200, 169].

We use $\boldsymbol{u}$ to denote all control subfunctions $u_i \in L^{\infty}(\mathscr{T}, \mathbb{R})$, $i \in [n_u]$ and therefore write $\boldsymbol{u}(t) = (u_1(t), \dots, u_{n_u}(t))^T \in \mathbb{R}^{n_u}$. The mixed control-state constraints $\boldsymbol{c}$ may consist of pure state or control path constraints, and the OCP may appear without the boundary constraint (2.5d) or (2.5e).

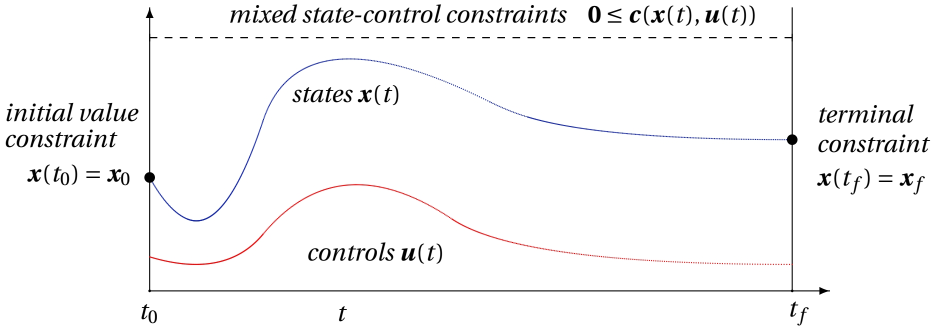

mixed state-control constraints $\boldsymbol{0} \le \boldsymbol{c}(\boldsymbol{x}(t), \boldsymbol{u}(t))$
states $\boldsymbol{x}(t)$
initial value constraint
$\boldsymbol{x}(t_0) = \boldsymbol{x}_0$
terminal constraint
$\boldsymbol{x}(t_f) = \boldsymbol{x}_f$
controls $\boldsymbol{u}(t)$
t_0
t
t_f

Figure 2.1: Example illustration of the state, control, and constraint trajectories of problem (2.5), adopted from [106, 218].

2.2 Problem concepts and properties

This section establishes definitions of concepts relevant to the OCP and provides results with respect to the optimal solutions.

Definition 2.2 (Feasibility of OCPs)
We define a state–control trajectory pair $(\boldsymbol{x}, \boldsymbol{u}) \in W^{1,\infty}(\mathscr{T}, \mathbb{R}^{n_x}) \times \mathscr{U}$ to be feasible for the continuous OCP (2.5) *if it satisfies constraints* (2.5b)-(2.5e). *A control trajectory $\boldsymbol{u} \in \mathscr{U}$ is called feasible if the corresponding state trajectory $\boldsymbol{x}$ is feasible, i.e., if $(\boldsymbol{x}, \boldsymbol{u})$ is feasible.*

We note that the term *feasible* is sometimes used interchangeably with *admissible*.

Definition 2.3 (Optimality of OCPs)
A feasible state–control trajectory pair $(\boldsymbol{x}^*, \boldsymbol{u}^*) \in W^{1,\infty}(\mathscr{T}, \mathbb{R}^{n_x}) \times \mathscr{U}$ *is defined to be globally optimal if*

$$\mathscr{C}(\boldsymbol{x}^*, \boldsymbol{u}^*) \leq \mathscr{C}(\boldsymbol{x}, \boldsymbol{u}) \tag{2.6}$$

holds for all feasible pairs $(\boldsymbol{x}, \boldsymbol{u})$. *We define* $(\boldsymbol{x}, \boldsymbol{u})$ *to be locally optimal if there exists* $\epsilon > 0$ *such that* (2.6) *is true for all feasible state–control trajectory pairs* $(\boldsymbol{x}, \boldsymbol{u})$ *with*

$$\|\boldsymbol{u} - \boldsymbol{u}^*\|_{L^\infty} \leq \epsilon.$$

To refer to a feasible or optimal *trajectory*, we also write feasible or optimal *solution*.

2.2.1 Pontryagin's maximum principle

PONTRYAGIN's maximum principle [202] formulates first-order necessary optimality conditions for OCPs. It is based on the HAMILTONIAN function. We consider a less restrictive variant of problem (2.5) to illuminate the essential points. We refer to [151, 182] for more details.

Definition 2.4 (HAMILTONIAN function)
Consider the autonomous OCP (2.5) *without constraints* (2.5c) *and* (2.5e). *The function*

$$\boldsymbol{\lambda} : \mathscr{T} \rightarrow \mathbb{R}^{n_x}$$

is called an adjoint or costate variable. We define the Hamiltonian *as*

$$\mathscr{H}(\boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{\lambda}(t)) := \Psi(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \boldsymbol{\lambda}(t)^T \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t) \qquad \textit{for } t \in \mathscr{T}. \tag{2.7}$$

We retain the term *maximum* in the following theorem for historical reasons, although this work deals with a minimization problem.

Theorem 2.1 (PONTRYAGIN's maximum principle)
Consider the autonomous OCP (2.5) *without constraints* (2.5c) *and* (2.5e), *and let* $(\boldsymbol{x}^*, \boldsymbol{u}^*)$ *be a local minimum of this problem. Then, there exists an adjoint function* $\boldsymbol{\lambda}^*$ *such that the following conditions hold:*

(i) *ODE model:* $\dot{\boldsymbol{x}}^*(t) = \dfrac{d\mathscr{H}}{d\boldsymbol{\lambda}}(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{\lambda}^*(t)) = \boldsymbol{f}(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t))$ *for* $t \in \mathscr{T}$,

(ii) *Initial values:* $\boldsymbol{x}^*(t_0) = \boldsymbol{x}_0$,

(iii) *Adjoint equations:* $\left(\dot{\boldsymbol{\lambda}}^*(t)\right)^\top = -\dfrac{d\mathscr{H}}{d\boldsymbol{x}}(\boldsymbol{x}^*(t), \boldsymbol{u}^*(t), \boldsymbol{\lambda}^*(t))$ *for* $t \in \mathscr{T}$,

(iv) *Final values of adjoints:* $\left(\boldsymbol{\lambda}^*(t_f)\right)^\top = -\dfrac{d\Phi}{d\boldsymbol{x}}(\boldsymbol{x}^*(t_f))$,

(v) *Minimum principle:* $\boldsymbol{u}^*(t) = \underset{\boldsymbol{u} \in \mathscr{U}}{\operatorname{argmin}} \mathscr{H}(\boldsymbol{x}^*(t), \boldsymbol{u}(t), \boldsymbol{\lambda}^*(t))$ *for* $t \in \mathscr{T}$.

Proof. A proof of this or similar versions of the maximum principle can be found in, e.g., [90], Theorem 3.4.4. □

This principle provides the basis for the *indirect* solution methods, as pointed out in Section 2.3.2. Since we build our algorithmic framework on *direct* methods, we do not rely heavily on these conditions. Nevertheless, they are useful for the specific rounding schemes in Section 6.5. Further intuitions about necessary conditions for OCPs can be found in [46, 120].

2.2.2 The Hamilton-Jacobi-Bellman equation

Broadly summarized, BELLMAN's principle of optimality [20] states that any subtrajectory of an optimal trajectory is optimal. This is expressed by the partial differential equation (PDE) system introduced in the following definition.

Definition 2.5 (HAMILTON-JACOBI-BELLMAN equation)
Consider the autonomous OCP (2.5)*. The function $J \in L^{\infty}(\mathcal{T} \times \mathbb{R}^{n_x}, \mathbb{R})$ is implicitly defined via the following PDE system and boundary condition:*

$$J(t_f, \boldsymbol{x}(t_f)) = \Phi(\boldsymbol{x}(t_f)), \tag{2.8}$$

$$\frac{\partial J}{\partial t}(t, \boldsymbol{x}(t)) = \min_{\boldsymbol{u} \in \mathcal{U}} \left(\Psi(\boldsymbol{x}(t), \boldsymbol{u}(t)) + \frac{dJ}{d\boldsymbol{x}}(t, \boldsymbol{x}(t)) \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \right), \qquad \textit{for } t \in \mathcal{T}. \tag{2.9}$$

This system is referred to as the Hamilton-Jacobi-Bellman *equation.*

Remark 2.1
The optimal control function $\boldsymbol{u}^*$ of the autonomous OCP (2.5) can be found by solving the HAMILTON-JACOBI-BELLMAN equation. here is a recognizable connection to the Hamiltonian and Theorem 2.1, where we can set the adjoint variables to be

$$\boldsymbol{\lambda}^T(t) = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{x}}(t, \boldsymbol{x}(t)) \qquad \text{for } t \in \mathcal{T}.$$

We refer to [218, 170] for further explanation of the link between the maximum principle and the HAMILTON-JACOBI-BELLMAN equation.

To structurally exploit BELLMAN's principle, we set up the cost-to-go function, yielding insight into the remaining costs of an optimal state-control trajectory beginning at a fixed time.

Definition 2.6 (Cost-to-go function)
Let $\tilde{t} \in \mathcal{T}$ and $\tilde{\boldsymbol{x}} := \boldsymbol{x}^*(\tilde{t})$, where $\boldsymbol{x}^*$ is the optimal state solution of Problem (2.5)*. The cost-to-go function J for $\tilde{t}$ and $\tilde{\boldsymbol{x}}$ is given by*

$$J(\tilde{t}, \tilde{\boldsymbol{x}}) := \min_{\boldsymbol{x}, \boldsymbol{u}} \Phi(\boldsymbol{x}(t_f)) + \int_{\tilde{t}}^{t_f} \Psi(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) \, \mathrm{d}t, \tag{2.10}$$

such that $\boldsymbol{x}(\tilde{t}) = \tilde{\boldsymbol{x}}$, and $\boldsymbol{x}, \boldsymbol{u}$ are feasible.

The optimal *cost-to-go* function is discretized and recursively minimized in the dynamic programming approach that we review in Section 2.3.1. Although this is not the solution approach of the CIA decomposition, we exploit this concept for specific variants of (CIA) rounding problems in Section 4.5.

2.2.3 Control solution structure

The structure of an optimal control solution $\boldsymbol{u}^*$ can be determined in specific cases, and it may be important for possible re-optimization stages. We thus take a closer look at the underlying relationships of the solution.

Definition 2.7 (Bang-bang and singular arcs)
Consider OCP (2.5) *without constraints* (2.5c) *and* (2.5e). *Furthermore, assume that* $\mathcal{U}$ *restricts* $\boldsymbol{u}$ *via box constraints in the (component-wise) sense of*

$$\boldsymbol{u}^{\mathrm{lb}} \leq \boldsymbol{u}(t) \leq \boldsymbol{u}^{\mathrm{ub}}, \qquad \textit{for } t \in \mathcal{T},$$

where $\boldsymbol{u}^{\mathrm{lb}}, \boldsymbol{u}^{\mathrm{ub}} \in \mathbb{R}^{n_u}$. *Let* $\boldsymbol{u}^*$ *denote the optimal control trajectory. We call the control function component* u_i^* *of* $\boldsymbol{u}^*$, $i \in [n_u]$, *on* $t \in \mathcal{T}_{\mathrm{arc}} \subset \mathcal{T}$ *singular if*

$$u_i^{\mathrm{lb}} < u_i^*(t) < u_i^{\mathrm{ub}} \qquad \textit{for } t \in \mathcal{T}_{\mathrm{arc}}.$$

Consequently, we call $\mathcal{T}_{\mathrm{arc}}$ *a singular arc if there is a singular control* u_i^* *on* $\mathcal{T}_{\mathrm{arc}}$. *In contrast, we define a control as* non-singular *or* bang-bang *if it attains its lower or upper bounds, i.e.,* $u_i^*(t) = u_i^{\mathrm{lb}}$ *or* $u_i^*(t) = u_i^{\mathrm{ub}}$ *for all* $i \in [n_u]$ *on the* bang-bang *arc* $\mathcal{T}_{\mathrm{arc}}$.

The definition of singular control functions used here differs from that used in other works in this research area, where singular controls and arcs are defined using the structure of the switching function, which is the derivative of the HAMILTONIAN with respect to $\boldsymbol{u}$ [106, 218]. When the switching function is not invertible with respect to $\boldsymbol{u}$, one speaks of a *singular arc* and when it is invertible, of a *bang-bang arc*.

Remark 2.2 (Bang-bang principle)
The situation described in Definition 2.7 can be further exploited in the case of control affine dynamics:

$$\boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t), \qquad \text{for } t \in \mathcal{T},$$

where $\boldsymbol{A}, \boldsymbol{B}$ are matrices of appropriate sizes. Here, the so-called bang-bang principle can be proven, asserting that the reachable set of differential state trajectories induced by all feasible controls is identical to the reachable set of bang-bang controls [123]. This implies that an optimal control solution with bang-bang structure can be found. For further details and proofs, we refer to [218, 182].

In this subsection, we neglect path constrained problems via Constraint (2.5c). In path constrained problems, one may analyze the problem structure with respect to constraint-seeking arcs as in Definition 2.7, see [218].

The concept of singular arcs is useful for our investigations as part of the outer convexification technique in Section 4.1. Solving the relaxed MIOCP can already result in integer control values by means of bang-bang arcs. The presence of singular arcs, conversely, gives rise to (CIA) rounding problems. Specifically, we exploit knowledge about singular arcs in the recombination strategies in Section 4.5 and to reduce the size of the (CIA) rounding problem in Section 6.1.

2.3 Solution methods

We note that OCP (2.5) is an infinite-dimensional optimization problem, as the trajectories of $\boldsymbol{x}$ and $\boldsymbol{u}$ are sought in function space. Therefore, this problem class is generally challenging, and various solution methods have been proposed. This section reviews the three main approaches.

2.3.1 Dynamic programming

Dynamic programming is based on the principle of optimality that led to the HAMILTON-JACOBI-BELLMAN equation in Definition 2.5 and the cost-to-go function given in Definition 2.6. The idea is to first discretize the problem in time, e.g., via an explicit EULER method, so that a state-space tabulation can be achieved. Second, the cost-to-go function is applied recursively to each discretization interval, and for each subproblem, the discretized control values that result in state values with the minimal objective value are sought. The advantage of this method is that the obtained solution is globally optimal with respect to the chosen discretization because all feasible solutions are evaluated. The main disadvantage stems from BELLMAN's "curse of dimensionality", which states that dynamic programming suffers severely from an increasing number of differential states or controls as the computational burden increases (exponentially in this case). The approach is thus restricted to small state dimensions. We refer to [26] for more information on the usage of dynamic programming for OCPs.

2.3.2 Indirect methods

The indirect approach follows the idea to *first optimize, then discretize*. By *first optimize*, we indicate that the necessary optimality conditions from PONTRYAGIN's maximum principle as stated in Theorem 2.1 are considered, thereby yielding a multipoint boundary value problem after deriving an analytical solution for the control $\boldsymbol{u}$ in the associated equations. The boundary value problem is then discretized with respect to time and solved via a method such as *indirect multiple shooting* [188, 32] or *indirect collocation* [15, 18]. This approach is advantageous because the obtained state and control trajectories are highly accurate since the infinite-dimensional problem has been solved. However, the approach also includes pitfalls: it is necessary to compute problem-specific derivatives of the HAMILTONIAN function to obtain the first-order optimality conditions. This can be challenging for large systems, and the boundary problem is often ill-conditioned such that nontrivial analytical considerations are necessary to solve it. This is particularly true for path-constrained problems, which result in even more complex optimality conditions. Related discussion of aspects of the indirect methods can be found in [120, 106].

2.3.3 Direct methods

In contrast to the *indirect* approach, *direct* methods follow the idea to *first discretize, then optimize.* This method parameterizes the infinite-dimensional optimization problem finitely by means of decision variables, notably the states $\boldsymbol{x}$ and the controls $\boldsymbol{u}$, such that the original problem is approximated by a finite-dimensional one, which is generally a nonlinear program (NLP). This is what is referred to by "first-discretize". The NLPs can be addressed in the second step, for which tailored and structure-exploiting numerical solution methods exist [264, 53]. That is, "then-optimize" refers to solving the resulting finite-dimensional optimization problem numerically to optimality.

Direct methods allow a general problem class, including inequality constraints such as path constraints (2.5c), to be handled with high flexibility and robustness, making direct methods suitable for the optimal control of large systems of practical relevance. This is due to the existence of well-developed NLP methods that can appropriately treat structural changes in the active constraints during the optimization procedure.

Because of these advantages over dynamic programming and indirect methods, the CIA decomposition is built on direct methods. We therefore introduce the approach of direct methods in greater detail. More extensive overviews of direct methods for continuous OCPs can be found in, e.g., [31, 91].

In the following, we review the relevant discretization steps of controls, states, constraints, and objective functions as well as the subsequent NLP solution methods. Throughout this thesis, we will apply the following, not necessarily equidistant, grid given in Definition 2.8.

Definition 2.8 (Discretization grid $\mathscr{G}_N, \Delta_i, \bar{\Delta}, \underline{\Delta}$)
Consider a given time horizon $\mathscr{T}$. Let $N \in \mathbb{N}$ denote the number of discretization intervals. We define the (discretization) grid with $N+1$ grid points as the ordered set

$$\mathscr{G}_N := \{t_0 < t_1 < \ldots < t_N = t_f\}.$$

Further, let the grid length quantities Δ be given by

$$\Delta_j := t_j - t_{j-1}, \text{ with } t_j, t_{j-1} \in \mathscr{G}_N,\ j \in [N], \qquad \bar{\Delta} := \max_{j\in[N]} \Delta_j, \qquad \underline{\Delta} := \min_{j\in[N]} \Delta_j.$$

Note that this grid is employed for the control discretization and that different (superset) grids may be used for the differential states. In later chapters, for the sake of simplicity, we refer only to grid $\mathscr{G}_N$, which is not restrictive since the proposed methods can be directly applied to finer state grids.

Control discretization

The approximation of the control function $\boldsymbol{u}$, which consists of input controls $\boldsymbol{u} = (u_1, \ldots, u_{n_u})^T$, can be achieved by different types of base functions such as *B-Splines* [246]. Even though high order parametrized base functions yield more accurate problem solutions, the most widespread parameterizations are piecewise constant controls because they require far lower computational effort [247]. To this end, consider the piecewise constant control functions u_i, $i \in [n_u]$:

$$u_i(t) := u_{ij}, \quad \text{for } t \in [t_{j-1}, t_j), \text{ with } t_j, t_{j-1} \in \mathscr{G}_N,\ j \in [N], \tag{2.11}$$

where u_{ij} are the variables to be optimized. Hence, we allow the controls to change value solely on grid points,[1] and the discretized control function $\boldsymbol{u}(\cdot)$ can be uniquely represented by the matrix $(u_{ij})_{i\in[n_u], j\in[N]}$.

State discretization

The approximation of differential states is typically performed by applying a *sequential* (single shooting) or *simultaneous* (multiple shooting or collocation) approach.

1. *Direct Single Shooting.* This method is attributable to HICKS and RAY [127] as well as Sargent and Sullivan [231]. It is based on the idea of regarding the states $\boldsymbol{x}$ as dependent variables that are constructed by forward integration of the dynamic system, starting at

[1] Note that $\boldsymbol{u}$ is unspecified on t_N according to the definition on half-open intervals. Since it is defined as L^∞ representatives of an equivalence class in $\mathscr{L}^\infty$, it is justified for $\boldsymbol{u}$ to be unspecified on sets of measure zero, such as grid points of $\mathscr{G}_N$.

$\boldsymbol{x}_0$ and using the discretized controls (u_{ij}). For this purpose, a numerical integration scheme, also called an *integrator*, is used to obtain the state trajectory as the result of an initial value problem (IVP). A common example of an integrator is the family of Runge-Kutta methods [156, 216]. Single shooting is relatively easy to implement and leads to a small problem size. Nevertheless, using this method it is impossible to use a priori information of the state trajectory as an initial guess. Furthermore, convergence issues often occur with stiff or highly nonlinear systems due to numerical error propagation.

2. *Direct Multiple Shooting*. This method was introduced by Bock and Plitt [34]. It proposes to solve the ODE separately on each discretization interval $[t_{j-1}, t_j]$, $j \in [N]$. For this, artificial initial values s_j are introduced as additional *shooting variables* of the discretized problem, and the corresponding IVPs are solved:

$$\begin{aligned} \dot{\boldsymbol{x}}(t; s_j, \boldsymbol{u}_j) &= \boldsymbol{f}(t, \boldsymbol{x}(t; s_j, \boldsymbol{u}_j), \boldsymbol{u}_j), \qquad t \in [t_{j-1}, t_j], \\ \boldsymbol{x}(t_j; s_j, \boldsymbol{u}_j) &= s_j, \end{aligned}$$

   where $\boldsymbol{x}(t; s_j, \boldsymbol{u}_j)$ indicates the dependency of $\boldsymbol{x}(t)$ on the initial value s_j and control value $\boldsymbol{u}_j$. As in the single shooting method, a numerical integrator is used to forward calculate the trajectory and to obtain $\boldsymbol{x}(t_{j+1}; s_j, \boldsymbol{u}_j)$. Because the piecewise solution is generally not continuous at the shooting nodes, a continuity condition is required to ensure equality between the constructed state value at the end of the previous interval and the next shooting variable:

$$\boldsymbol{x}(t_{j+1}; s_j, \boldsymbol{u}_j) = s_{j+1}, \qquad \text{for } j \in [N-1]_0.$$

   The main advantages of direct multiple shooting are that initial guesses of the state trajectory can be applied and that it leads to superior local convergence properties, particularly for unstable or highly nonlinear systems [106]. Compared with single shooting, the NLP problem size increases, but tailored solution methods such as *condensing* [106] exist.

3. *Direct Collocation* [253, 18, 30]. In this approach, both the controls and states must be discretized on the same (relatively fine) grid $\mathscr{G}_N$. Rather than using a numerical integration scheme as in single or multiple shooting, the state trajectory is approximated on each (collocation) interval $[t_{j-1}, t_j]$ by a polynomial $P_j^c(t, \boldsymbol{p}_j^c)$ with a coefficient vector $\boldsymbol{p}_j^c$ that needs to be optimized. Consequently, each collocation interval corresponds to an integrator step. Exemplary variants are the RADAU [31] and LEGENDRE [14] collocation. For each interval, there are m collocation points on which the time derivative of the polynomial must be equal to the evaluated right-hand side $\boldsymbol{f}$ of the ODE. Moreover, initial value variables s_j are introduced, and continuity of the state solution across interval boundaries is enforced, as in the multiple shooting method. Collocation methods share similar advantages and disadvantages with the multiple shooting approach, particularly with respect to stability and state initialization.

We stress that our algorithmic ideas regarding the CIA decomposition can be applied independently of the discretization method that is used, though we typically employ direct multiple shooting or direct collocation because of their favorable numerical performance.

Constraint and objective discretization

A common approach is for the constraint (2.5c) to only be enforced at the discrete grid points and to rely on the approximated state and control values. Hence, the same grid is chosen as for the controls and states, but it is possible to check the constraints for a finer sampling, such as the collocation points, or on intermediate integrator steps in the case of multiple shooting.

The Bolza objective value can be computed numerically through the chosen integrator for both the single and multiple shooting methods. In the case of collocation, the integral term of the objective can be approximated by a quadrature formula [205] using the collocation points.

Solution of NLPs

The obtained NLP is typically solved via NEWTON-type optimization methods [106]. Such methods find a (local) optimum iteratively from a given starting point, which is the initial guess. The choice of the initial guess is crucial since it affects not only the solution time but also the quality of the solution. These methods exploit (numerically approximated) first- and second-order derivatives and can be divided into two widely used approaches: *sequential quadratic programming*-type [195, 36] and *nonlinear interior point* [181, 259, 264] methods. While the latter is based on solving a sequence of linear problems with penalized constraint violations, the former transforms the NLP into quadratic problems that are solved iteratively. We will rely mostly on the interior point solver `IPOPT` [264] in the computational experiments due to its robustness in achieving an optimal solution.

Chapter 3

Mixed-integer optimal control

MIOCPs can be seen as a generalization of OCPs in which the right-hand side function $\boldsymbol{f}$ of the ODE depends not only on continuous controls but also on discrete-valued control or switching functions. These discrete-valued control functions can be interpreted as different operation modes of the ODE. Depending on the type of control function – continuous or discrete – we distinguish different types of constraints. In particular, discrete control functions may have to obey combinatorial constraints that would not make sense in the continuous setting. This chapter first defines a generic MIOCP class and introduces *explicit* and *implicit* switches in Section 3.1. We then classify possible combinatorial constraints in Section 3.2. Finally, we conduct a literature survey and review solution approaches in Sections 3.3 and 3.4, respectively.

3.1 Problem statement

We begin this section with a definition of integer and binary control functions.

Definition 3.1 (Integer and binary control function $\boldsymbol{v}$)
We define an integer *control $\boldsymbol{v}$ to be a function whose image space is a finite discrete set with $n_\omega \in \mathbb{N}$ different realizations $\boldsymbol{v}^i \in \mathbb{R}^{n_v}$, $i \in [n_\omega]$:*

$$\boldsymbol{v} \in L^\infty(\mathcal{T}, V), \qquad \textit{with} \qquad V := \{\boldsymbol{v}^1, \boldsymbol{v}^2, \dots, \boldsymbol{v}^{n_\omega}\}.$$

We call $\boldsymbol{v}$ a binary *control function if $\boldsymbol{v}^i \in \{0,1\}^{n_v}$ for all $i \in [n_\omega]$.*

The discrete nature of $\boldsymbol{v}$ is manifested in its n_ω different realizations, which implies that there exists $\epsilon > 0$ such that for all $i \neq j,\ i, j \in [n_\omega]$, we have $\|\boldsymbol{v}^i - \boldsymbol{v}^j\| > \epsilon$, where $\|\cdot\|$ represents an arbitrary norm. The term integer control can be misleading since the discrete values $\boldsymbol{v}^i$ do not need to be integers by definition. Nevertheless, we use this term because the discrete values can be translated into different modes of the dynamic system, as explained in Remark 3.2, and therefore the discrete-valued control $\boldsymbol{v}$ has an integer nature. We use this type of control function in the following definition of a general MIOCP.

Definition 3.2 (Mixed-integer optimal control problem (MIOCP))
For a given time horizon $\mathcal{T}$, consider the integer control function $\boldsymbol{v}$ from Definition 3.1. We denote the space of feasible integer controls by $\mathcal{V} \subseteq L^\infty(\mathcal{T}, V)$, which represents combinatorial

constraints. We refer to the following control problem (3.1) *as* ***(MIOCP)****.*

$$\begin{aligned}
&\inf_{\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}} \; \mathscr{C}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{v}) := \Phi(\boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} \Psi(t,\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{v}(t))\,\mathrm{d}t && \text{(3.1a)}\\
&\text{s.t.} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t,\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{v}(t)), \qquad \textit{for a.a. } t \in \mathscr{T}, && \text{(3.1b)}\\
&\qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, && \text{(3.1c)}\\
&\qquad \boldsymbol{x}(t_f) = \boldsymbol{x}_f, && \text{(3.1d)}\\
&\qquad \boldsymbol{0}_{n_c} \le \boldsymbol{c}(t,\boldsymbol{x}(t),\boldsymbol{u}(t)), \qquad \textit{for a.a. } t \in \mathscr{T}, && \text{(3.1e)}\\
&\qquad \boldsymbol{0}_{n_d} \le \boldsymbol{d}(t,\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{v}(t)), \qquad \textit{for a.a. } t \in \mathscr{T}, && \text{(3.1f)}\\
&\qquad \boldsymbol{u} \in \mathscr{U}, \quad \boldsymbol{v} \in \mathscr{V}, && \text{(3.1g)}
\end{aligned}$$

where $\boldsymbol{x}_0, \boldsymbol{x}_f \in \mathbb{R}^{n_x}$ *respectively denote the initial and final values of the dynamic process. We reuse the definitions of the differential states* $\boldsymbol{x}$*, the path constraint function* $\boldsymbol{c}$*, and the feasible space* $\mathscr{U}$ *of the continuous control from Definition 2.1 in Chapter 2. Here, the cost function* $\mathscr{C}$ *and Lagrange term* Ψ *also depend on the integer control, in contrast to Definition 2.1. The aim of the problem* (3.1) *is to find continuous and integer controls* $\boldsymbol{u}$ *and* $\boldsymbol{v}$*, respectively, and the differential state* $\boldsymbol{x}$ *that minimize the performance index* $\mathscr{C}$ *such that* $\boldsymbol{x}$ *satisfies the dynamic process constraint* (3.1b)*, which is an ODE governed by the right-hand side function* $\boldsymbol{f}: \mathscr{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times V \to \mathbb{R}^{n_x}$*. The dynamic process is further restricted by the initial and terminal value constraints* (3.1c) *and* (3.1d)*, respectively. Additionally, the mode–independent and mode–dependent mixed control–state constraints,* (3.1e) *and* (3.1f)*, must be fulfilled by constraint functions* $\boldsymbol{c}$ *and* $\boldsymbol{d}: \mathscr{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times V \to \mathbb{R}^{n_d}$*, respectively.*

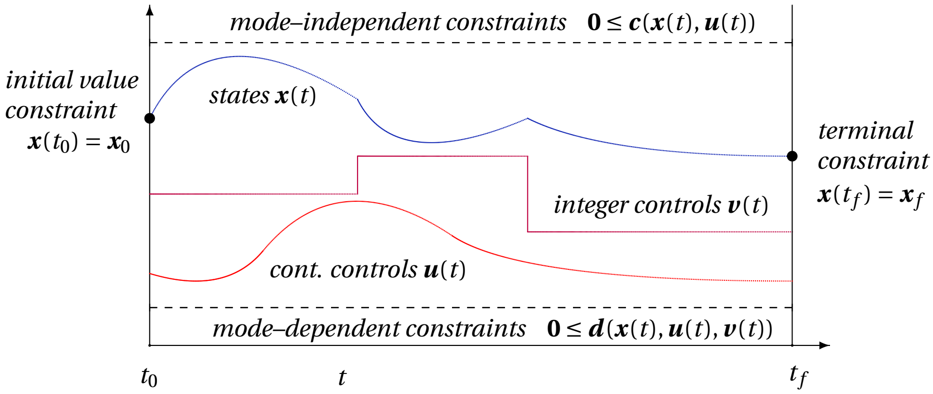

mode–independent constraints $\boldsymbol{0} \le \boldsymbol{c}(\boldsymbol{x}(t),\boldsymbol{u}(t))$
initial value constraint $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$
states $\boldsymbol{x}(t)$
terminal constraint $\boldsymbol{x}(t_f) = \boldsymbol{x}_f$
integer controls $\boldsymbol{v}(t)$
cont. controls $\boldsymbol{u}(t)$
mode–dependent constraints $\boldsymbol{0} \le \boldsymbol{d}(\boldsymbol{x}(t),\boldsymbol{u}(t),\boldsymbol{v}(t))$
t_0
t
t_f

Figure 3.1: Example illustration of the state, control, and constraint trajectories of problem (MIOCP).

Remark 3.1 (Use of *min* instead of *inf*)
Following the convention in the optimization community, we write “min” instead of “inf” in the following problem definitions even though the existence of an optimal solution of (MIOCP) is not guaranteed.

The problem class (MIOCP) is of central importance since this dissertation deals mainly with solution algorithms for this problem. We pay special attention to combinatorial constraints on the integer control, which are expressed by the function space $\mathcal{V}$; these constraints are defined in Section 3.2. We notice that $\boldsymbol{v}$ induces discontinuities into $\boldsymbol{f}$ because it only assumes discrete values leading to jumps in $\boldsymbol{f}$ each time $\boldsymbol{v}$ changes value. This behavior is characterized as a *switch* in Definition 3.4. First, we examine another possible source of discontinuities in $\boldsymbol{f}$: a so-called *switching* function.

Definition 3.3 (Implicit switching function $\boldsymbol{s}$)
Consider the set

$$S := \{\boldsymbol{s}^1, \boldsymbol{s}^2, \dots, \boldsymbol{s}^{n_\iota}\}, \quad \boldsymbol{s}^i \in \mathbb{R}^{n_s}, \textit{ for all } i \in [n_\iota],$$

where $n_s, n_\iota \in \mathbb{N}$ with $n_s \leq n_\iota$. We define an (implicit) switching function $\boldsymbol{s}$ *via its domain and codomain:*

$$\boldsymbol{s} : \mathcal{T} \times \mathbb{R}^{n_x} \to S.$$

We assume $\boldsymbol{s}$ to be sufficiently smooth in its state component. Note that the above definition does not specify the particular outcome of the switching function. Typically, it expresses state-dependent conditions $\boldsymbol{\iota}(\boldsymbol{x}(t)) \lesseqgtr 0$, where $\boldsymbol{\iota}$ denotes a sufficiently smooth vector-valued function, and $\boldsymbol{s}$ assumes a different value as soon as a component of $\boldsymbol{\iota}$ crosses zero. Based on this notion, the switching function can also be defined by the sign structure of the function $\boldsymbol{\iota}$, as in many works [35, 150]. The term “implicit” derives from the condition that changes in $\boldsymbol{s}$, unlike those in the integer control $\boldsymbol{v}$, are not explicitly controllable but rather implicitly depend on the differential states or time. The following definition concretizes the two different types of switches.

Definition 3.4 (Explicit and implicit switch)
Let $\boldsymbol{v}$ be an integer control and $\boldsymbol{s}$ be a switching function, as respectively introduced in Definitions 3.1 and 3.3. We call a discontinuity in at least one component of $\boldsymbol{v}$ at time $t_1 \in (t_0, t_f)$ an explicit switch *of $\boldsymbol{v}$ at time t_1. Analogously, we refer to an* implicit switch *of $\boldsymbol{s}$ at time t_1 if there is a discontinuity in at least one component of $\boldsymbol{s}(t_1)$. We also say that $\boldsymbol{f}$ switches at time t_1 if it depends on $\boldsymbol{v}$ or $\boldsymbol{s}$ and that an explicit or implicit switch occurs at time t_1. The time point t_1 is referred to as switching time.*

Other names of switch types, such as *externally forced* or *controllable* instead of *explicit* and *internally forced* or *autonomous* instead of *implicit* [285], are common in the literature. As implicit switches are not the main focus of this thesis, we omit a thorough specification and further details on useful theoretical concepts such as the consistency and transversality of implicit switches here but refer to [35, 182]. Nevertheless, we address an MIOCP with implicit switches as part of the applications in Chapter 10 and thus introduce the corresponding problem class.

Definition 3.5 (MIOCP with implicit switches (MIOCPi) and multiphase MIOCP (MMIOCP))
Consider the setting of Definition 3.2, and let $\boldsymbol{s}$ be a given implicit switching function. We refer to the following control problem (3.2) *as* ***(MIOCPi)****:*

$$\min_{\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}} \quad \mathcal{C}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{v}) \tag{3.2a}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t), \boldsymbol{s}(t)), \qquad \textit{for a.a. } t \in \mathcal{T}, \tag{3.2b}$$

$$(3.1\text{e}) - (3.1\text{g}),$$

where the ODE is determined by the right-hand side function $\boldsymbol{f} : \mathcal{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times V \times S \to \mathbb{R}^{n_x}$. (MIOCPi) is the same as (MIOCP) from Definition 3.2 except that $\boldsymbol{f}$ also depends on the switching *function* $\boldsymbol{s}$*. If* $\boldsymbol{s}$ *is independent of the differential states* $\boldsymbol{x}$ *but depends on* t*, we call* ***(MIOCPi)*** *a* multiphase mixed-integer optimal control problem*, in short* ***(MMIOCP)****.*

We note that in (MMIOCP) the function $\boldsymbol{f}$ switches every time a specific switching time $t_1 \in \mathcal{T}$ that marks the beginning of a new model phase is reached; this explains the term *multiphase*. This problem class is included in our algorithmic and theoretical investigations in Chapters 4 and 5 before the presentation of an application in Chapter 10.

Of course, it is also possible to define an MIOCP based only on implicit switches, i.e., independently of any explicit switches. Let $\mathcal{I}(P)$ denote the set of all problem instances of an optimization problem class (P). Then, by Definition 3.5 and by considering generalizations of the defined problems, we get the inclusion

$$\mathcal{I}(\text{MIOCP}) \subset \mathcal{I}(\text{MMIOCP}) \subset \mathcal{I}(\text{MIOCPi}).$$

In view of the occurrence of switches, we can interpret the different realizations of $\boldsymbol{v}$ and $\boldsymbol{s}$ as modes of operation of the model function $\boldsymbol{f}$. The following remark establishes this perspective, which is commonly applied in the research area of MIOCPs, also referred to as *optimal control of switched systems.*

Remark 3.2 (Switched systems and switching signal)
The IVP composed of the ODE (3.1b) or (3.2b) together with the initial value constraint (3.1c) is called a *switched system.* The discrete event dynamics of a switched system can also be written in terms of modes $i(t) \in [n_\omega \cdot n_\iota]$, $t \in \mathcal{T}$, for $\boldsymbol{f}$:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_{i(t)}(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \qquad \text{for a.a. } t \in \mathcal{T},$$

where each realization of the integer control function $\boldsymbol{v}$ and the implicit switching function $\boldsymbol{s}$ corresponds to a mode-specific function $\boldsymbol{f}_i$. Then, solving MIOCPs can also be described as finding the optimal *switching signal* Ξ, which is a timed sequence of active modes i combined with its switching time instants, i.e.,

$$\Xi = \{(t_0, i_0), (t_1, i_1), \ldots, (t_K, i_K)\},$$

where $K \in \mathbb{N}_0$, $t_j \in \mathcal{T}$ and $i_j \in [n_\omega \cdot n_\iota]$ for $j \in [K]_0$.

3.2 Classification of combinatorial constraints

Combinatorial constraints on the integer control are significant in many applications, as illustrated in Chapter 10. This section discusses different variants of these restrictions. We first introduce so-called vanishing constraints that are defined pointwise in time before we elaborate constraints that couple over time, typically rendering them very challenging to handle.

We now assume that $\boldsymbol{v}$ is a binary control function, i.e., $V = \{0,1\}^{n_v}$. This is not a restriction since it is possible to transform the model function $\boldsymbol{f}$ such that binary controls replace the integer controls. For the transformation, assume the integer control $\tilde{v}_i, i \in [n_v]$, enters $\boldsymbol{f}$ with codomain $\tilde{V} \subset \mathbb{N}$. Then, the control can be represented with binary control functions v_j ($n_\omega \in \mathbb{N}$ sufficiently large):

$$\tilde{v}_i(t) := 1 + \sum_{j=0}^{\lceil \log_2 n_\omega \rceil} 2^j v_j(t).$$

We say that the ith component of $\boldsymbol{v}$ is *activated* or *active* at time $t \in \mathcal{T}$, if and only if $v_i(t) = 1$. Similarly, saying that v_i is *deactivated* at time t indicates that $v_i(t) = 0$. Next, we focus on the switching properties of $\boldsymbol{v}$ in order to later define the time-coupled constraints.

Definition 3.6 (Switching variation $SV(\boldsymbol{v},\tilde{\mathcal{T}})$)
Consider an integer control $\boldsymbol{v} \in \mathcal{V}$ with finitely many switches. Further, let $\tilde{\mathcal{T}} \subseteq \mathcal{T}$. The switching variation *of $\boldsymbol{v}$ on the interval $\tilde{\mathcal{T}}$ is defined as*

$$SV(\boldsymbol{v},\tilde{\mathcal{T}}) := \left|\{t_j \in \tilde{\mathcal{T}} \mid t_j \textit{ is switching time for } \boldsymbol{v}\}\right|, \tag{3.3}$$

where we apply Definition 3.4, and $|\cdot|$ denotes the cardinality of the set. By $SV(v_i,\tilde{\mathcal{T}})$, we denote the switching variation of the ith component of $\boldsymbol{v}$, which is defined analogously.

Remark 3.3 (Total variation $TV(\boldsymbol{v})$)
We introduced the concept of switching variation, because the total variation generally differs from the number of switches. The *total variation $TV(v_i)$* of the ith component of an integer control $\boldsymbol{v}$ is defined to be the quantity

$$TV(v_i) := \sup_{\boldsymbol{P}\in\mathcal{P}} \left\{ \sum_{j\in[n_P]} |v_i(t_j) - v_i(t_{j-1})| \right\}, \tag{3.4}$$

where $\boldsymbol{P} = (t_0, \ldots, t_{n_P})$ is a partition from the set of all partitions $\mathcal{P}$ of the interval $\mathcal{T}$, and n_P denotes the partition-specific number of time points. If $\boldsymbol{v}$ is a binary control whose switches lead to an exclusive change of values in two components, we can define its *total variation $TV(\boldsymbol{v})$* as follows:

$$TV(\boldsymbol{v}) := \sup_{\boldsymbol{P}\in\mathcal{P}} \left\{ \frac{1}{2} \sum_{i\in[n_v]} \sum_{j\in[n_\mathcal{P}]} |v_i(t_j) - v_i(t_{j-1})| \right\}. \tag{3.5}$$

In this case, we have $TV(\boldsymbol{v}) = SV(\boldsymbol{v},\mathcal{T})$, so we often refer to the more common concept of *total variation* instead of *switching variation*.

Remark 3.4 (Alternative definition of total variation)
For the sake of completeness, we mention that the total variation of the ith component of a binary control $\boldsymbol{v}$ can also be defined as

$$TV(v_i) := \sup\left\{ \int_{\mathcal{T}} v_i(t)\varphi'(t)\,\mathrm{d}t \mid \varphi \in C_c^1(\mathcal{T}),\ \|\varphi\|_{L^\infty} \leq 1 \right\},$$

where $C_c^1(\mathcal{T})$ is the set of continuously differentiable functions with compact support. This definition is standard in the field of measure theory [214] and yields the same quantity as the one in Remark 3.3.

Some of the definitions of combinatorial constraints rely on the one-sided limit of $t_s \in \mathcal{T}$ from below, which we define as $\boldsymbol{v}(t_s^-) := \lim_{t \to t_s^-} \boldsymbol{v}(t)$.

3.2.1 Vanishing constraints

The constraint (3.1f) depends on $\boldsymbol{v}$ and can therefore be interpreted as a combinatorial constraint. It is possible to reformulate these constraints via outer convexification into *vanishing* *constraints*, as we show in Section 4.1. Vanishing constraints are inequalities of the type

$$0 \leq v_i(t) d_i(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \qquad i \in [n_v],\ t \in \mathcal{T}, \tag{3.6}$$

where the constraint function d_i depends on the mode $i \in [n_v]$. This constraint sets up a conditional requirement that $0 \leq d_i(t, \cdot)$ holds true if $v_i(t)$ is activated. In the context of mixed-integer optimal control (MIOC), vanishing constraints have been investigated in various studies [136, 138, 149, 179, 194]. The discretized problem generally falls into the class of *Mathematical Programs with Vanishing constraints*, which are particularly studied in [2, 128].

3.2.2 Limited switching constraints

Limited switching constraints are designed to limit the number of switches used on the defined time horizon. Let $\sigma_{i,\max} \in \mathbb{N}$ denote the maximum number of allowed switches for the ith component of $\boldsymbol{v}$. This limitation is imposed by

$$SV(v_i, \mathcal{T}) \leq \sigma_{i,\max}, \qquad i \in [n_v]. \tag{3.7}$$

Rather than limiting the number of mode-specific switches, it is also possible to bound the total number of switches $\sigma_{\max} \in \mathbb{N}$ via

$$SV(\boldsymbol{v}, \mathcal{T}) \leq \sigma_{\max}. \tag{3.8}$$

As mentioned in Remark 3.3, for specific binary controls, this constraint may be equivalently imposed via the concept of total variation. Thus, we also call the above restriction a *bounded total variation constraint* [222]. This constraint has been investigated in various publications, particularly related to the CIA decomposition [218, 145, 224, 207]. These works were extended in [222], which constitutes significant parts of this thesis.

3.2.3 Minimum dwell time constraints

Minimum dwell time constraints freeze the current value of a mode i after a switch occurs. We distinguish between *minimum up (MU) time* spans $C_{i,U} \geq 0$, in which a mode i must remain active once it has been switched on, and *minimum down (MD) time* spans $C_{i,D} \geq 0$, in which a mode i must stay inactive after it has been deactivated. For this, let $\mathcal{S}^v$ denote the set of switching time points for $\boldsymbol{v} \in \mathcal{V}$, which is assumed to be finite. For $t_s \in \mathcal{S}^v$, we define $C_{i,U}^{t_f} := \min\{C_{i,U}, t_f - t_s\}$ and similarly $C_{i,D}^{t_f} := \min\{C_{i,D}, t_f - t_s\}$. The constraints read

$$v_i(t_s) - v_i(t_s^-) \leq v_i(t_s + t), \qquad \text{for } i \in [n_v],\ t_s \in \mathcal{S}^v, \text{ and } t \in [0, C_{i,U}^{t_f}), \tag{3.9}$$

$$v_i(t_s^-) - v_i(t_s) \leq 1 - v_i(t_s + t), \quad \text{for } i \in [n_v],\ t_s \in \mathcal{S}^v, \text{ and } t \in [0, C_{i,D}^{t_f}). \tag{3.10}$$

MU time constraints are expressed in (3.9), while (3.10) enforces MD time restrictions. The restrictions can also be formulated in terms of the switching variation:

$$SV(v_i, (t, t + C_{i,U})) + v_i(t + C_{i,U}) - v_i(t) \leq 2, \quad \text{for } i \in [n_v],\ t \in (t_0, t_f - C_{i,U}). \tag{3.11}$$

We note that the above formulation imposes both an MU time and an MD time of $C_{i,U}$ for mode $i \in [n_v]$. Recent case studies of MIOCPs with minimum dwell time (MDT) considerations can be found in the literature, e.g., for pesticide scheduling in agriculture [4], electric transmission lines [103], solar thermal climate systems [49], and hybrid electric vehicles [211]. MDT constraints have also attracted substantial attention as part of mixed-integer linear programs (MILPs), see [203] for a study of unit-commitment problems and [161] for a corresponding polytope investigation. For a recent work on model predictive control (MPC) under MDT constraints see [56].

As part of the direct method and *variable time transformation* approaches [275, 16, 89], MDT have been addressed; see [4, 193]. Another approach that includes MDT constraints into MIOCPs is the application of dynamic programming, which is, however, computationally expensive; see [51]. Other recent approaches have used the framework of approximate dynamic programming [126] or command governors [80]. Due to its relevance for practical applications and the increased attention it has received in the literature, the CIA decomposition was recently investigated in relation to MDT constraints in [282], which provides a basis for major parts of this thesis.

3.2.4 Maximum dwell time constraints

In contrast to MDT conditions, we can also limit the activation length from above, for which we introduce the term *maximum dwell time.* If $C_{i,M} \geq 0$ is such a time span for mode $i \in [n_v]$, then the condition can be formulated as:

$$SV(v_i, (t - C_{i,M}, t)) + v_i(t) - v_i(t - C_{i,M}, t) \geq 2, \quad \text{for all } t \in (C_{i,M}, t_f). \tag{3.12}$$

3.2.5 Total maximum up time constraints

In some cases, it makes sense to limit the total activation time over the entire time horizon of specific modes. Let $C_{i,\max} \geq 0$ denote such a time span for mode $i \in [n_v]$. Then, this constraint is imposed by

$$\int_{\mathcal{T}} v_i(t)\,\mathrm{d}t \leq C_{i,\max}, \qquad i \in [n_v]. \tag{3.13}$$

3.2.6 Mode transition constraints

It is sometimes necessary to exclude the activation of specific modes if a certain other mode is currently active. We therefore define the modes that can be followed or preceded by another mode by introducing the set $\mathscr{I}_i^A$, which denotes the modes that are allowed to be activated directly after mode i has been active. Further, we reuse the set of switching time points $\mathscr{S}^v$ and assume that exactly one mode i is active for all time points. The mode transition requirements then read

$$v_i(t_s^-) + \sum_{j \notin \mathscr{I}_i^A} v_j(t_s) \leq 1, \qquad \text{for } i \in [n_v] \text{ and all } t_s \in \mathscr{S}^v. \tag{3.14}$$

These constraints take effect when, e.g., modeling a sequence of gear shifts, see [211]. They are discussed in Section 10.1.

3.2.7 Other constraints

There is no limit to the variety of combinatorial constraints, and indeed all the conditions of combinatorial optimization [270, 154] can conceivably be included in problem (3.1). For example, another simple but practically relevant requirement is to restrict the allowed modes for defined time periods $\tilde{\mathcal{T}} \subset \mathcal{T}$. This is realized for mode $i \in [n_v]$ by trivially setting

$$v_i(t) = 0, \quad \text{for } t \in \tilde{\mathcal{T}}. \tag{3.15}$$

3.3 Literature survey

The study of MIOCPs has received substantial attention over the last decades. This is because many different application problems can be modeled as switched systems, making the problem class a powerful tool. Along with their practical relevance, the interesting theoretical nature of MIOCPs has prompted researchers from different communities in mathematics, science, and engineering to investigate them. In addition to *mixed-integer optimal control* and *optimal control of switched systems*, alternate names for the same or similar problem classes have been established, such as *mixed-logic dynamic optimization* [191], *mixed logical dynamical systems* [22], *mixed-integer programming for control* [206], and *hybrid optimal control* or *optimal control of hybrid systems* [10, 235, 249]. In fact, *hybrid systems* can be described as general heterogeneous dynamical systems that involve both continuous models that classify the physical part and discrete event models that define the logical behavior [285]. This combination explains the term “hybrid” and classifies switched systems as a particular kind of hybrid system. Hybrid systems are suitable for modeling state jumps, i.e., discontinuities in $\boldsymbol{x}$ (see e.g. [150]), which are usually not covered by switched systems and are therefore excluded from our considerations.

Due to the abundance of literature (with specific topical journals such as *Nonlinear Analysis: Hybrid Systems*), different research communities, terms, and problem variants, no approach or methodology for MIOC has emerged as the most established. For these reasons, there have even been parallel developments of similar algorithms with different names. As an example that is relevant to this thesis, in the mid-2000s Sager [218] and Bengea together with DeCarlo [24] independently developed a convexification idea for MIOCPs on which the CIA decomposition is based. In the control engineering community, the term *embedding transformation* was coined to describe this idea, leading to various subsequent publications [255, 260, 204, 184]. Nevertheless, in these works and the survey [285], there is no reference to Sager’s works or those based on them. In the mathematics community, this connection seems to similarly have gone unnoticed, although here, individual cross-references to the embedding transformation method have recently begun to appear [149, 35, 212]. We provide details on the (partial outer) convexification idea and identify the differences between the CIA decomposition and the embedding transformation in Chapter 4. The literature concerning MIOC can be classified with regard to the following:

1. **Underlying dynamics**: ODEs, differential-algebraic equations (DAEs), or PDEs;

2. **Type of switches**: explicit or implicit;

3. **Problem structure**: linear, linear-quadratic, or nonlinear;

4. **Further restrictions**: state, mixed control-state, combinatorial, or no constraints;

5. **Open-loop** (feedforward) and **closed-loop** (feedback) control;
6. **Aim of study**: theoretical oriented or solution method oriented;
7. **Algorithmic approach**: dynamic programming, direct, indirect, or other methods; and
8. **Usage of learning**: mixed data-model or pure model-driven.

Next, we briefly review some of these aspects and also address recent applications, dissertations, surveys, and software packages. As part of the next chapter, we comment on solution methods for the problem (MIOCP), which is the ODE-constrained, explicit switches, generally nonlinear and restricted, open-loop, and pure model-driven variant of the above aspects.

DAE- and PDE-constrained MIOC

GERDTS and SAGER investigate the necessary optimality conditions for a switched DAE system of index 1 and derive lower bounds [92]. For further studies, we refer to the references therein and to [252].

MIOC under PDE constraints, also known as *MIPDECO*, has emerged in recent years as an intensively investigated, highly salient research field. HANTE and SAGER extended the partial outer convexification approach from [218] to semilinear evolution equations [117]. HANTE also contributed several further works to other variants of MIPDECOs, e.g., [118, 119]. MANNS and KIRCHES proposed methods based on relaxation and rounding approximations for general MIPDECO problem classes by applying, e.g., space-filling curves [177, 178]. LEYFFER has also contributed to the development of this field via reviewing approaches and introducing benchmark problems [164]. Recently, HAHN proposed a set optimization framework [115] that can be applied to both ODE- and PDE-constrained MIOCPs and that seems to provide auspicious results. Other recent methods have address a KOOPMAN operator-based model reduction [198] and penalty algorithms [87]. For a more detailed overview, we refer to the excellent synopsis in MANNS' dissertation [176] and to [114].

Problems involving implicit switches

In the context of implicit switches, the literature largely focuses on piecewise affine problems, where it is possible to partition the state space into polyhedral regions [243, 285]. These systems can be appropriately handled by the mixed logical dynamic framework, which consists of a collection of linear (boolean) difference equations [22]. In terms of solution methods, mixed-integer programming [213] and dynamic programming variants [59] have been the main proposals. MEYER et al. proposed an approach to solving problems that involve both implicit and explicit switches [35]. The approach uses partial outer convexification [218] and translates implicit switches into variables, reducing the problem to a mathematical program with vanishing constraints. For further literature on implicit switches, see [35] and the references therein.

Mixed-integer (nonlinear) model predictive control

Works concerning online MIOCPs are abundant due to their application orientation [40]. In principle, all solution methods for offline MIOCPs can be used. In this sense, KIRCHES has successfully investigated the CIA decomposition with sum-up rounding (SUR) as a rounding step in the MPC context. More recent forms of the CIA decomposition applied to MPC can be found in [49] and [56]. In this context, the already mentioned mixed logical dynamic framework [81] and a direct discretization of linearized problems have already been used; the latter leads to solving MILPs [142]. Recent studies have also investigated warm-starting or updating strategies [180, 125, 174]. The MIOC *turnpike property* can be interpreted as such [78].

System-theoretical investigations

Apart from tailor-made optimal control method solutions, the theoretical investigation of the problem structure of switched systems has received notable attention in the literature. This includes the *stability* [165], *stabilizability* [168], *controllability* [248], *average dwell-time stability* [124], and *Lyapunov function* [65] of switched systems.

Learning supported MIOC

Artificial intelligence frameworks such as machine learning and deep learning have become increasingly popular in recent years and can therefore be considered a *hot topic*. This development does not stop at MIOC, where the first beginnings of the use of artificial intelligence have appeared. Methods of reinforcement learning [71, 42, 55] and machine learning [276] have been used, for example, to assist in solving MIOCPs by applying pre-calculated solutions and neural networks.

Literature reviews and Ph.D. theses

Recent detailed overviews of solution methods for MIOC were given by ZHU and ANTSAKLIS [285], who survey approaches for both explicit and implicit switches, and in the Ph.D. thesis of PASSENBERG [196]. A review of the theoretical aspects of switched systems can be found in [95]. Recent Ph.D. theses in the area of solution methods for MIOC include SCHORI [233] for indirect methods, STELLATO [244] and PALAGACHEV [193] for two-level approaches as part of direct methods; JUNG [135], LENDERS [162], RIECK [207] and MEYER [182] for CIA decomposition related approaches as part of direct methods, MANNS [176], and RÜFFLER [215] for PDE constraints; and BÜRGER [47], ROBUSCHI [210], SIRVENT [239], and NAIK [186] for application-oriented results.

Applications

The problem class (MIOCP) is ubiquitous in various application areas as it can be a powerful modeling tool. Apart from the applications presented in Chapter 10, other application areas include water, gas, traffic, and supply chain networks [52, 70, 153, 107, 82, 102, 101, 100]; distributed autonomous systems [1]; processes in chemical engineering that involve valves [140, 242]; the choice of gears in automotive control [89, 146]; thermodynamics [104]; systems biology [159]; and chemotherapy [271]. Further real-world problem applications can be found in the Ph.D. theses listed in the previous paragraph.

Software packages

Due to the heterogeneity of solution methods for MIOCPs, there are various software packages for specific approaches. An online benchmark collection of MIOCPs and their modeling with a range of software tools is presented in [220][1]. There are `MATLAB` packages for general OCPs that can handle switched systems, such as the `Multi-Parametric Toolbox` [157], the `Convex Dynamic Programming` tool [122], and the `FALCON` toolbox [207]. An open source `SCIP` plug-in, which uses direct method relaxations based on piecewise linearization, is available at [261]. Further modeling frameworks with the capability to handle MIOCPs are `GEKKO` [19] and `APMonitor` [121]. An `AMPL` extension, which interfaces with the MIOC extension `MS-MINTOC` [219] of `MUSCOD-II` [69], is described in [147]. In this dissertation, we describe the software package `pycombina` [50] for solving CIA problems in Chapter 8.

3.4 Algorithms for mixed-integer optimal control

We give an overview of solution methods for (MIOCP). The dynamic programming technique is applicable to MIOCPs and is advantageous since global optimality is achieved. It has been used in many studies [122, 39] and is constantly being investigated [171, 283]. However, the *curse of dimensionality* remains a key issue and prevents the application of dynamic programming to large, general problems.

In the early 1980s, BOCK and LONGMAN had already applied indirect methods to MIOC [33]. Their approach was based on PONTRYAGIN's maximum principle with disjoint integer control sets and is referred to as the *Competing Hamiltonian* approach. Hybrid variants of the maximum principle were established in several further works (see, e.g. [249, 235, 251, 92]) and were then used to derive algorithmic implications. As pointed out in Section 2.3.2, indirect methods have multipoint boundary problem-related drawbacks when compared to direct methods, and it is not clear how combinatorial constraints could be incorporated in them. Nevertheless, the indirect approach is beneficial for analyzing necessary optimality conditions and is thus still observed in many current studies [266, 192].

A study on global optimization of an MIOC gas network problem was recently performed by HABECK et al. through spatial branching with convex under- and concave over-estimators [112]. Alternative works aiming for global optimality are related to particle swarm optimization [139] or genetic algorithms [228]. Global MIOC can also be investigated via moment relaxations [226, 60] or semidefinite programming [66, 284].

Discretizing (MIOCP)in the spirit of first-discretize-then-optimize methods generally results in an mixed-integer nonlinear program (MINLP). The already mentioned *mixed logical dynamics* approach can then be used to approximate the MINLP via piecewise system linearizations by an MILP or mixed-integer quadratic program (MIQP) [213]. Another option is to apply solution methods from MINLP, such as branch-and-bound (BnB), to provide global optimality [54, 88]. While these two approaches are useful for specific applications, they often lead to an excessive computational burden. To this end, it is preferable to consider reformulations, relaxations, and heuristics of the discretized (MIOCP), giving rise to the CIA decomposition. In the following, we summarize three methods that transform (MIOCP) into simpler subproblems to which concepts from both direct and indirect methods are applicable.

[1] See also `https://mintOC.de`.

3.4.1 Bilevel optimization

A common approach for addressing MIOCPs is to consider two separate stages of optimization and solve them at different levels [275, 23]. The optimal switching sequence is searched for at the upper level, whereas at the lower level, the cost function is optimized over the space of switching time instants under a fixed switching sequence. Hence, the method can be summarized with by the following steps:

1. Start with an initial guess for the switching signal.
2. Repeat until the termination criterion is reached:
   a) Update the mode sequence, and possibly insert new modes.
   b) Perform switching time optimization.

In practice, step 2.a) can be solved by evaluating a *mode insertion gradient* [99, 98]. The upper-level problem can generally be approached with dynamic programming [274], indirect methods [16], or direct methods [193]. In step 2.b), the algorithm aims to minimize the cost functional with respect to the switching times and continuous control input $\boldsymbol{u}$, if available. This problem is referred to as *switching time optimization* and is discussed in the following subsection. Despite successful case studies [4, 193], the bilevel optimization strategy is limited by the restriction to a fixed mode sequence at the upper level.

3.4.2 Switching time optimization

Although we list *switching time optimization* as the second step of the bilevel algorithm, we explain this method separately since it can also serve as a stand-alone approach. In fact, if the number of active modes in the fixed switching sequence is large enough, this approach yields a bijective transformation of (MIOCP), as stated in [92].

The idea of switching time optimization relies on the time transformation $t = (t_2 - t_1)\tau$, which exploits the fact that $\mathrm{d}t = (t_2 - t_1)\mathrm{d}\tau$ holds. Then,

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t), \boldsymbol{u}(t), \boldsymbol{v}(t)), \qquad \text{for a.a. } t \in \mathcal{T},$$

is equivalent to

$$\dot{\boldsymbol{x}}(\tau) = (t_f - t_0)\boldsymbol{f}(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau), \boldsymbol{v}(\tau)), \qquad \text{for a.a. } \tau \in [0, 1].$$

This can be applied to several subintervals of $\mathcal{T}$ with different model functions $\boldsymbol{f}_1$, $\boldsymbol{f}_2$ that correspond to different control realizations of $\boldsymbol{v}$. Hence,

$$\dot{\boldsymbol{x}}(t) = \begin{cases} \boldsymbol{f}_1(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), & \text{if } t \in [t_0, t_1], \\ \boldsymbol{f}_2(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), & \text{if } t \in [t_1, t_f], \end{cases}$$

is transformed into

$$\dot{\boldsymbol{x}}(\tau) = \begin{cases} (t_1 - t_0)\boldsymbol{f}_1(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau)), & \text{if } \tau \in [0, 1], \\ (t_f - t_1)\boldsymbol{f}_2(\tau, \boldsymbol{x}(\tau), \boldsymbol{u}(\tau)), & \text{if } \tau \in [1, 2]. \end{cases}$$

In this way, the switching times t_i enter the problem as continuous variables, transforming the original discrete optimization problem into a continuous one. This justifies common names for this approach such as *transition-time optimization* [75, 74] and *variable time transformation method* [89]. It has been extended to various settings, such as switching costs [67], vanishing constraints [194], and structure-exploiting linearization [245]. Its advantage lies in eliminating the discrete variables. Nevertheless, numerical convergence issues may arise, and the mode sequence may impact the computational performance, as discussed in [208]. We employ switching time optimization in the heart assist device application in Section 10.2.

3.4.3 Embedding transformation

The embedding transformation technique [24] is equivalent to the partial outer convexification method [218], which we introduce in Section 4.2. It is based on reformulating the model equation (3.1b) into

$$\dot{\boldsymbol{x}}(t) = \sum_{i=1}^{n_\omega} v_i(t)\boldsymbol{f}_i(t, \boldsymbol{x}(t), \boldsymbol{u}(t)), \quad v_i(t) \in [0,1], \quad \sum_{i=1}^{n_\omega} v_i(t) = 1, \qquad \text{for a.a. } t \in \mathcal{T}, \tag{3.16}$$

where $\boldsymbol{f}_i$ denotes the model function with the fixed ith mode represented by the binary controls v_i. The main difference from the CIA decomposition lies in its originally theoretical view. Rather than suggesting an algorithm for constructing binary controls from relaxed ones, BENGEA and DECARLO used indirect methods to prove that the binary solutions are dense in the relaxed ones and argued that singular arcs are uncommon [24]. The embedding transformation technique has been extended and applied in various subsequent publications. MEYER et al. compared and differentiated the approach from bilevel optimization, multi-parametric programming [157], and hybrid maximum principle-based methods in [183]. To create binary controls from relaxed ones, VASUDEVAN et al. proposed a projection method in which the MIOCP objective function is accounted for in contrast to the CIA rounding method. The embedding transformation was successfully applied for MIOCPs in automotive control [184] and cancer chemotherapy [271]. Recently, WU has extended the approach with penalty functions and a time transformation to avoid the rounding step [272].

Part II

Algorithms and theory

Chapter 4

Combinatorial integral approximation decompositions

This chapter defines the CIA decomposition algorithm. Section 4.1 summarizes the *partial outer convexification* reformulation method, which is equivalent to the *embedding transformation* from Section 3.4.3. Based on this method, we describe the CIA decomposition in basic form in Section 4.2. The ideas and methods behind the algorithm have been developed since the mid-2000's, and we provide a literature review of these results in Section 4.3. Section 4.4 deals with approaches for constructing binary control functions that are feasible in terms of different constraints, such as combinatorial or path constraints. Finally, in Section 4.5, we discuss generalizations of the CIA decomposition. These include different MILP variants, a sequence of several rounding and NLP steps, and recombinations of binary control functions.

4.1 The partial outer convexification reformulation

We now consider a convex reformulation of (MIOCP) introduced by SAGER in his dissertation in 2006. The idea is to replace the integer controls $\boldsymbol{v}$ that enter the model function $\boldsymbol{f}$ with binary controls $\boldsymbol{\omega}$ that the dynamics linearly, by introducing a binary control function for each possible mode, respectively control realization, of $\boldsymbol{v}$. The term *partial* refers to the fact that only the input of the discrete-valued controls is convexified. However, the dynamics can still be non-convex due to the function $\boldsymbol{f}$. The benefit of this reformulation is the suitable subsequent relaxation of the problem using continuous control functions $\boldsymbol{\alpha}$ that assume values on the interval $[0,1]$. We open this section by introducing the spaces of the control functions before we define the convexified problem (BOCP) and its relaxation (ROCP). Throughout this thesis, we assume a problem involving $n_\omega \geq 2$ modes of the integer control, i.e., $\boldsymbol{\omega}$ can assume n_ω different binary control realizations.

Definition 4.1 (Binary ω and relaxed control functions α)
Let the vector of binary controls $\boldsymbol{\omega}$ on the simplex and its corresponding vector of relaxed controls $\boldsymbol{\alpha}$ be defined by their function space domains

$$\Omega := \left\{ \boldsymbol{\omega} \in L^\infty(\mathcal{T}, \{0,1\}^{n_\omega}) \mid \sum_{i \in [n_\omega]} \omega_i(t) = 1, \ \textit{for a.a. } t \in \mathcal{T} \right\},$$

$$\mathcal{A} := \left\{ \boldsymbol{\alpha} \in L^\infty(\mathcal{T}, [0,1]^{n_\omega}) \mid \sum_{i \in [n_\omega]} \alpha_i(t) = 1, \ \textit{for a.a. } t \in \mathcal{T} \right\}.$$

In the following, we sometimes write in short *control i* to abbreviate a control realization, respectively control mode, $\omega_i(\cdot), i \in [n]$, of a control function $\omega(\cdot) = (\omega_1(\cdot), \ldots, \omega_n(\cdot))^\mathsf{T}$.

Definition 4.2 ((BOCP),(ROCP))
Consider the time horizon $\mathcal{T}$, the initial and terminal values $\boldsymbol{x}_0, \boldsymbol{x}_f$, the objective function $\mathcal{C}$, the differential states $\boldsymbol{x}$, the path constraint function $\boldsymbol{c}$, and the feasible space $\mathcal{U}$ for the continuous

control function $\boldsymbol{u}$ from Definition 3.1 for the problem (MIOCP). Let $\Psi_i, i \in [n_\omega]$, denote the function Ψ with the ith control realization of $\boldsymbol{v}$ fixed. We denote the space of feasible binary control functions by $\Omega_{comb} \subseteq \Omega$, which represents combinatorial constraints. We refer to the following control problem (4.1) *as **(BOCP)**:*

$$\min_{\boldsymbol{x},\boldsymbol{u},\boldsymbol{\omega}} \quad \mathscr{C}(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\omega}) := \Phi(\boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} \sum_{i\in[n_\omega]} \omega_i(t)\, \Psi_i(t,\boldsymbol{x}(t),\boldsymbol{u}(t))\,\mathrm{d}t \tag{4.1a}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}_0(t,\boldsymbol{x}(t),\boldsymbol{u}(t)) + \sum_{i\in[n_\omega]} \omega_i(t)\, \boldsymbol{f}_i(t,\boldsymbol{x}(t),\boldsymbol{u}(t)), \qquad \textit{for a.a. } t\in\mathscr{T}, \tag{4.1b}$$

$$\boldsymbol{x}(t_0) = \boldsymbol{x}_0, \tag{4.1c}$$

$$\boldsymbol{x}(t_f) = \boldsymbol{x}_f, \tag{4.1d}$$

$$\boldsymbol{0}_{n_c} \leq \boldsymbol{c}(t,\boldsymbol{x}(t),\boldsymbol{u}(t)), \qquad \textit{for a.a. } t\in\mathscr{T}, \tag{4.1e}$$

$$\boldsymbol{0}_{n_d} \leq \omega_i(t)\, \boldsymbol{d}_i(t,\boldsymbol{x}(t),\boldsymbol{u}(t)), \qquad \textit{for a.a. } t\in\mathscr{T},\ i\in[n_\omega], \tag{4.1f}$$

$$\boldsymbol{u} \in \mathscr{U}, \quad \boldsymbol{\omega} \in \Omega_{comb}. \tag{4.1g}$$

Constraint (4.1b) *expresses the dynamical system as a switched system in partial outer convexified form, i.e., as a sum of a drift term $\boldsymbol{f}_0$ and control-specific functions $\boldsymbol{f}_i$, both with the domain and codomain: $\mathscr{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$. The mode–dependent mixed control–state constraint* (4.1f) *must be fulfilled for the functions $\boldsymbol{d}_i : \mathscr{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_d}$. We define (**ROCP**) as the canonical relaxation of problem (BOCP), where we optimize over $\boldsymbol{\alpha} \in \mathscr{A}$ instead of $\boldsymbol{\omega} \in \Omega_{comb}$.*

We recognize that the partial outer convexification addresses not only the dynamics in (4.1b) but also the Lagrange term and the mode–dependent mixed control–state constraint (4.1f), which are in fact vanishing constraints, as introduced in Section 3.2.1. In his dissertation [135], JUNG investigated other reformulations, such as *inner convexification*, and showed that outer convexification is favorable because it yields a tight relaxation.

Remark 4.1 (Assumption on the existence of a solution for (ROCP))
We assume that there exists an optimal solution $\boldsymbol{x}^*$ for the problem (ROCP). Thus, we may assume that a uniform Lipschitz estimate on $\boldsymbol{f}$ exists so that the theorem by Picard–Lindelöf [200] is applicable. This assumption is essential for the applicability of the CIA decomposition because it is based on the optimal solution of the relaxed problem.

Proposition 4.1 (Equivalence of (MIOCP) and (BOCP))
The problems (MIOCP) and (BOCP) are equivalent in the sense that there is a bijection between any feasible, respectively optimal, solution for (MIOCP) to a feasible, respectively optimal, solution for (BOCP).

Proof. The mapping

$$\Omega \to L^\infty(\mathscr{T},V), \quad \boldsymbol{\omega}(t) \mapsto \boldsymbol{v}(t) := \sum_{i\in[n_\omega]} \omega_i(t)\boldsymbol{v}^i, \ \text{ for a.a. } t\in\mathscr{T},\ \boldsymbol{v}^i \in V$$

is the desired bijection and preserves both feasibility and objective function value. Furthermore, we can identify $\boldsymbol{f}_i(\cdot,\cdot,\cdot)$ with $\boldsymbol{f}(\cdot,\cdot,\cdot,\boldsymbol{v}^i)$ for $i \in [n_\omega]$ and analogously identify $\boldsymbol{d}_i(\cdot,\cdot,\cdot)$ with $\boldsymbol{d}(\cdot,\cdot,\cdot,\boldsymbol{v}^i)$. □

Remark 4.2 (Large number of possible modes n_ω due to several integer controls)
Enumerating all possible modes as combinations of several integer control functions $\boldsymbol{v}$ may result in exponentially many control functions ω_i. However, one may reduce the number n_ω of feasible combinations by exploiting separability properties of $\boldsymbol{f}_i$, i.e., by decoupling and considering the integer control functions independently, see [135]. Still, it remains an open research question whether it is beneficial to directly relax integer controls without partial outer convexification if they enter the dynamics and constraints in an already linear fashion.

4.2 The basic combinatorial integral approximation decomposition

This section is dedicated to recapitulating the decomposition approach that was proposed in similar form in [218, 224, 135].

4.2.1 Problem discretization of (BOCP) and (ROCP)

The CIA decomposition relies on the use of direct methods (the *first discretize, then optimize* approach) introduced in Section 2.3.3. Thus, we explain and define the temporal discretization of the problem and the subproblems that constitute the decomposition algorithm. We reuse Definition 2.8, which formalizes the discretization grid. If not declared otherwise, we assume $N \in \mathbb{N}$ discretization intervals and n_ω to be the number of binary controls throughout this thesis. Next, we define the matrix sets of the discretized binary and relaxed control functions Ω_N, $\mathscr{A}_N$.

Definition 4.3 (Convex combination constraint (Conv), $\boldsymbol{\Omega_N}$, $\boldsymbol{\mathscr{A}_N}$)
Let $N \in \mathbb{N}$. We express the requirement that the columns of a matrix $(m_{i,j}) \in [0,1]^{n_\omega \times N}$ sum up to one by

$$\sum_{i \in [n_\omega]} m_{i,j} = 1, \qquad \textit{for } j \in [N], \tag{Conv}$$

and call this the convex combination constraint (Conv) in the remainder. Based on this constraint, we define

$$\Omega_N := \left\{ \boldsymbol{w} \in \{0,1\}^{n_\omega \times N} \mid \boldsymbol{w} \textit{ satisfies } (Conv) \right\}, \quad \mathscr{A}_N := \left\{ \boldsymbol{a} \in [0,1]^{n_\omega \times N} \mid \boldsymbol{a} \textit{ satisfies } (Conv) \right\}.$$

We note the geometric nature of Ω_N and $\mathscr{A}_N$: they are the vertices, respectively the set of faces, of the N-fold iterated standard simplex without the origin and spanned by the n_ω unit vectors.

We define the discretizations of (BOCP) and (ROCP) below.

Definition 4.4 ((MINLP),(NLP$_{\text{rel}}$), (NLP$_{\text{bin}}$))
Consider the problems (BOCP) and (ROCP) with the following modifications:

- *We discretize* (4.1b) *and the differential states $\boldsymbol{x}$ with $\mathscr{G}_N$ and by using direct collocation or direct multiple shooting together with an appropriate integrator function (e.g., Runge-Kutta methods [216, 156]).*
- *The control functions $\boldsymbol{u}$ and $\boldsymbol{\omega}$, respectively $\boldsymbol{\alpha}$, are assumed to be piecewise constant on $\mathscr{G}_N$ as defined in* (2.11) *in Section 2.3.3.*

*We denote the resulting discretized optimization problem with binary control functions $\boldsymbol{w} \in \Omega_N$ and relaxed binary control functions $\boldsymbol{a} \in \mathscr{A}_N$ by (**MINLP**) and (**NLP**$_{rel}$), respectively. We refer to (**NLP**$_{bin}$) when we consider (MINLP) with given and fixed binary control functions.*

In the case of absent continuous controls, the problem (NLP$_{\text{bin}}$) reduces to a forward integration of the differential states and, hence, to an evaluation of the objective function.

Remark 4.3 ($\boldsymbol{w}$ and $\boldsymbol{a}$ interpreted as control functions)
Although $\boldsymbol{w}$ and $\boldsymbol{a}$ were introduced as matrices, Definition 4.4 establishes a canonical way to consider their values as well-defined control functions:

$$\boldsymbol{\omega}(t) = \sum_{j=1}^{N-1} \boldsymbol{w}_{\cdot,j}\chi_{[t_{j-1},t_j)}(t) + \boldsymbol{w}_{\cdot,N}\chi_{[t_{N-1},t_N]}(t).$$

$\boldsymbol{\alpha}$ can be defined analogously. Hence, when we speak in the following of control functions $\boldsymbol{w}$ or $\boldsymbol{a}$, we are referring to its corresponding piecewise constant function.

4.2.2 Definition of the rounding problem (CIA)

The optimal objective value of (ROCP) represents a lower bound for that of (BOCP). Let $\boldsymbol{\alpha}^*$ denote the optimal relaxed binary control for (ROCP). To construct a binary control function $\boldsymbol{\omega}$ that is close to the optimum, we aim to minimize the so-called integral deviation gap:

$$\min_{\boldsymbol{\omega}\in\Omega} \max_{t\in\mathcal{T}} \left\| \int_{t_0}^{t} \boldsymbol{\alpha}^*(s) - \boldsymbol{\omega}(s) \, \mathrm{d}s \right\|, \tag{4.2}$$

where $\|\cdot\|$ is an unspecified norm, and the integral applies component-wise to the difference of $\boldsymbol{\alpha}^*(s)$ and $\boldsymbol{\omega}(s)$. For the moment, we neglect combinatorial constraints and consider the space Ω instead of Ω_{comb}, but we return to their formulation in the discretized setting in Section 4.4. In Chapter 5, we make use of the integral deviation gap to prove convergence properties of the CIA decomposition. We formalize the integral deviation gap for the discretized setting with the commonly applied maximum norm, as this is relevant for the practical solution process.

Definition 4.5 (Discretized integral deviation gap $\boldsymbol{\theta(w)}$)
Consider $\boldsymbol{a}^* \in \mathcal{A}_N$. The integral deviation gap $\theta(\boldsymbol{w})$ for $\boldsymbol{w} \in \Omega_N$ is defined as

$$\theta(\boldsymbol{w}) := \max_{i\in[n_\omega],\, j\in[N]} \left| \sum_{l\in[j]} (a^*_{i,l} - w_{i,l})\Delta_l \right|.$$

Expressed in words, the *integral deviation gap* refers to the accumulated control deviation between $\boldsymbol{a}^*$ and $\boldsymbol{w}$. We note that the term *integrality gap* has been used for the same mathematical term [149, 282]. However, the term *integrality gap* is already used in combinatorial optimization for the ratio of integer to relaxed optimal solutions of an MILP, so we deliberately avoid this term in this thesis. The integral deviation gap can be formulated as an MILP, which gives rise to the following definition.

Definition 4.6 (CIA)
*Let $\boldsymbol{a}^* \in \mathcal{A}_N$ be given. Then, we define the problem (**CIA**) to be*

$$\min_{\boldsymbol{w}\in\Omega_N,\theta\geq 0} \quad \theta \tag{4.3}$$

$$\text{s.t.} \quad \theta \geq \pm \sum_{l\in[j]} (a^*_{i,l} - w_{i,l})\Delta_l, \qquad \textit{for } i \in [n_\omega],\ j \in [N]. \tag{4.4}$$

The following property will be useful for proving near-optimality for (BOCP) of the binary control functions constructed by the CIA decomposition in Chapter 5.

Definition 4.7 (Rounding gap consistency property)
Consider any $\boldsymbol{a}^* \in \mathscr{A}_N$ *and any grid* $\mathscr{G}_N$*. We say an algorithm has the* rounding gap consistency property *if it produces a binary control* $\boldsymbol{w} \in \Omega_N$ *with*

$$\theta(\boldsymbol{w}) \leq C(n_\omega)\bar{\Delta}, \tag{4.5}$$

where $C(n_\omega) \in \mathbb{R}^+$ *is a positive constant that depends on* n_ω*.*

A heuristic way to solve the (CIA) problem is to run SUR, which is widely used due to its simplicity and short run time.

Definition 4.8 (Sum-up rounding (SUR) [218])
For given $\boldsymbol{a}^* \in \mathscr{A}_N$ *and* $j = 1, \ldots, N$*, the sum-up rounding (SUR) scheme computes*

$$w_{i,j} := \begin{cases} 1, & \textit{if } i = \underset{k=1,\ldots,n_\omega}{\operatorname{argmax}} \left\{ \sum\limits_{l=1}^{j} a^*_{k,l}\Delta_l - \sum\limits_{l=1}^{j-1} w_{k,l}\Delta_l \right\} \quad \textit{(break ties arbitrarily)}, \\ 0, & \textit{else}, \end{cases} \qquad \textit{for } i = 1, \ldots, n_\omega.$$

It has been proven that SUR fulfills the *rounding gap consistency* property [225]. This has a direct consequence for (CIA), as the following proposition reveals.

Proposition 4.2 ((CIA) has rounding gap consistency property)
(CIA) fulfills the rounding gap consistency property in the sense that its optimal solution $\boldsymbol{w}^*$ *satisfies*

$$\theta(\boldsymbol{w}^*) \leq C(n_\omega)\bar{\Delta}.$$

Proof. SUR constructs a feasible solution $\boldsymbol{w}^{\text{SUR}}$ for (CIA), but the solution is not necessarily optimal, i.e., $\theta(\boldsymbol{w}^{\text{SUR}}) \geq \theta(\boldsymbol{w}^*)$. Since SUR fulfills the *rounding gap consistency* property, we conclude this property is inherited by (CIA). □

We prove specific values of $C(n_\omega)$ for optimal solutions of (CIA) in Chapter 7. In Chapter 6, we discuss algorithms that solve (CIA) or that provide heuristic solutions with near-optimal objective values.

4.2.3 Definition of the algorithm

With the subproblem definitions from the previous subsections, we can summarize the CIA decomposition in Algorithm 4.1. We first solve the relaxed problem (NLP_{rel}) and approximate the resulting relaxed binary controls with binary values in the (CIA) problem. The last step consists of solving (NLP_{bin}) with a fixed binary control function $\boldsymbol{w}^*$ in order to obtain the objective value of (MINLP).

Rather than solving (NLP_{bin}), it is possible to fix the continuous controls $\boldsymbol{u}$ so that the third step only consists of an evaluation of the objective function in (MINLP). This happens regardless if (BOCP) involves no continuous controls. We stress that the above algorithm solves three problems, all of which are easier to solve than the original (MINLP). The CIA decomposition is now well-established in the literature due to the following advantages:

Algorithm 4.1: CIA decomposition algorithm for error-controlled solution of (BOCP)

```
Input  : (MINLP) instance with time grid $\mathscr{G}_N$ as discretization of (BOCP).
Solve (NLP$_{\text{rel}}$) $\rightarrow$ $\boldsymbol{x}^*$, $\boldsymbol{u}^*$, $\boldsymbol{a}^*$, $\mathscr{C}^*$;
Solve (CIA) for $\boldsymbol{a}^*$ $\rightarrow$ $\boldsymbol{w}^*$;
Solve (NLP$_{\text{bin}}$) with $\boldsymbol{w} = \boldsymbol{w}^*$ fixed $\rightarrow$ $\boldsymbol{x}^{**}$, $\boldsymbol{u}^{**}$, $\mathscr{C}^{**}$;
return: $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}, \mathscr{C}) = (\boldsymbol{x}^{**}, \boldsymbol{u}^{**}, \boldsymbol{w}^*, \mathscr{C}^{**})$;
```

1. MINLPs generally fall into the class of *NP-hard* problems; using approaches that bypass the direct solution of such problems is therefore computationally favorable.

2. Convergence results have been proven for (BOCP) without combinatorial constraints in the sense that under mild assumptions, the obtained solution has been shown to be arbitrarily close to the optimal solution with grid length $\bar{\Delta}$ going to zero [218, 225, 177]; we elaborate on this in Chapter 5. Moreover, the solution of (NLP$_{\text{rel}}$) represents a useful a priori lower bound on the objective if solved to global optimality.[1] Hence, one may loop over the discretization grid and iteratively apply Algorithm 4.1 with a refined grid until the obtained solution is satisfactorily close to the computed objective lower bound value, respectively the constraint violation is as small as desired, as proposed in the algorithm *MS MINTOC* in [223].

3. An MILP enables the intuitive inclusion of a large variety of combinatorial constraints. Numerical case studies have shown that the resulting feasible solution is close to the relaxed solution as long as the applied combinatorial constraints are not too restrictive [48, 49].

We note that any rounding gap consistency algorithm, particularly SUR, can be applied to achieve the convergence result in Chapter 5, though (CIA) is favorable in terms of the inclusion of combinatorial constraints. Figure 4.1 illustrates the relationship of the subproblems introduced in this chapter and to the CIA decomposition as a whole.

4.3 A brief survey of the combinatorial integral approximation decomposition

The main contributions and developments that led to the CIA decomposition are listed in Table 4.1 (not intended to be a complete list). In his dissertation, SAGER introduced the idea of reformulating (MIOCP) into (BOCP) (respectively a similar problem version of thereof) by using outer convexification [218] and proved that under mild conditions, there is no integer objective value gap between (BOCP) and (ROCP). For the practical construction of binary controls, he proposed the SUR scheme. KIRCHES investigated the outer convexification of constraints as well as applied SUR to nonlinear MPC in his dissertation [145].

[1] Solving an NLP to global optimality is generally computationally expensive. We are therefore content with a local solution constructed by a solver such as `IPOPT` [264], as elaborated in the numerical results chapter.

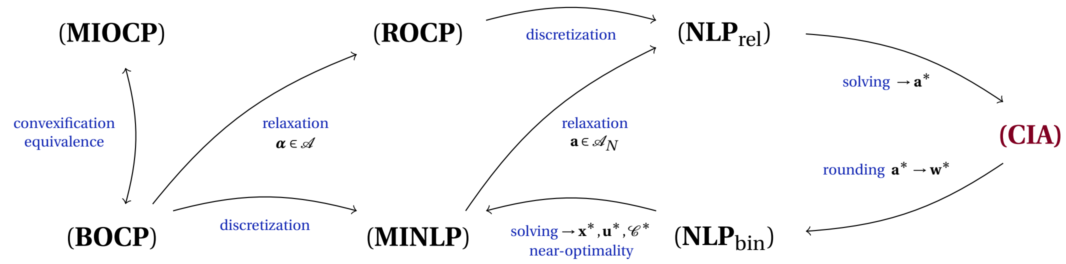

(MIOCP)
(ROCP)
discretization
(NLP$_{\text{rel}}$)
convexification
equivalence
relaxation
$\boldsymbol{\alpha} \in \mathscr{A}$
relaxation
$\mathbf{a} \in \mathscr{A}_N$
solving $\to \mathbf{a}^*$
(CIA)
rounding $\mathbf{a}^* \to \mathbf{w}^*$
(BOCP)
discretization
(MINLP)
solving $\to \mathbf{x}^*, \mathbf{u}^*, \mathscr{C}^*$
near-optimality
(NLP$_{\text{bin}}$)

Figure 4.1: Schematic representation of the CIA decomposition: (MIOCP) can be equivalently reformulated into its partially outer convexified counterpart problem (BOCP), which is then transformed into (MINLP) via a temporal discretization. Next, allowing a convex combination $\boldsymbol{a}$ in (Conv) yields the relaxed problem (NLP$_{\text{rel}}$). After solving this problem, we obtain $\boldsymbol{a}^*$, which is then approximated with binary values $\boldsymbol{w}^*$ in the rounding problem (CIA). Finally, we use $\boldsymbol{w}^*$ as fixed variables to solve (NLP$_{\text{bin}}$) as a continuous variable problem. The constructed solution is (almost) feasible and (sub)optimal for (MINLP).

Year	Author & Study	Contribution
2006	Sager [218]	Partial outer convexification reformulation, SUR, proof: no integer gap
2009	Sager [219]	Overview article
2010	Kirches [145]	Extension to MPC and path constraints
2011	Sager, Jung, Kirches [224]	(CIA) problem, BnB algorithm
2012	Gerdts, Sager [92] Sager, Bock, Diehl [225]	DAE extension Exact error estimates
2013	Jung [135] Hante, Sager [117]	Next-forced rounding, relaxations Time-dependent PDE extension
2015	Jung, Reinelt, Sager [135] Sager, Clayes, Messine [226]	Lagrangian relaxation for (CIA) Comparison with other approaches
2017	Hante [118]	Hyperbolic PDE extension
2018	Zeile, Weber, Sager [280] Bock et. al [35] Hahn, Sager [113] Bürger et. al [48, 49]	Generalizations of CIA decomposition Extension to implicit switches (CIA) for PDE case MPC extension of (CIA)
2019	Kirches, Lenders, Manns [149, 179, 162] Manns, Kirches [177, 176] Göttlich et. al [102] Hahn et. al [114]	Inclusion of vanishing constraints General PDE cases ADMM for combinatorial constraints Overview article with PDE focus
2020	Bürger et. al [50] Zeile, Robuschi, Sager [282, 222] Bestehorn et. al [27, 28]	Software `pycombina` Rounding gap results, rounding algorithms Switching costs, shortest-path algorithm

Table 4.1: Timeline of contributions to and developments of the CIA decomposition. ADMM refers to the alternating direction method of multipliers.

Several refinements were provided by SAGER, DIEHL, and BOCK in their article [225] in which they proved exact error estimates between the relaxed and binary control constructed problem solutions. They used SUR as a constructive element in the proof and showed a linear dependence on the grid length $\bar{\Delta}$. The rounding step was generalized to solve (CIA), i.e., to solve an MILP, by SAGER, JUNG, and KIRCHES in [224]. They proposed an efficient BnB algorithm and argued for the intuitive inclusion of combinatorial constraints in (CIA) [137]. In his dissertation [135], JUNG examined further relaxation approaches and proposed next-forced rounding (NFR) to construct feasible binary controls.

Definition 4.9 (Next-forced rounding (NFR))
Consider a given $\boldsymbol{a}^* \in \mathcal{A}_N$ on a grid $\mathcal{G}_N$. For all $i = 1, \dots, n_\omega$ and iteratively for $j = 1, \dots, N$, we define the quantity

$$\mathcal{N}_j(i) := \begin{cases} \underset{k=j,\dots,N}{\operatorname{argmin}} \left\{ \sum_{l=1}^{k} a^*_{i,l}\Delta_l - \sum_{l=1}^{j-1} w_{i,l}\Delta_l > \bar{\Delta} \right\}, & \textit{if} \quad \sum_{l=1}^{N} a^*_{i,l}\Delta_l - \sum_{l=1}^{j-1} w_{i,l}\Delta_l > \bar{\Delta}, \\ \infty, & \textit{else}. \end{cases} \tag{4.6}$$

A control with index $i^\star \in [n_\omega]$ on interval j is defined to be next-forced, if and only if

$$\mathcal{N}_j(i^\star) = \min_{i \in [n_\omega]} \mathcal{N}_j(i) \quad \textit{and} \quad \mathcal{N}_j(i^\star) < \infty. \tag{4.7}$$

Then, the NFR algorithm iteratively sets the next forced control i equal to one (break ties arbitrarily), i.e., $w_{i,j} = 1$, for $j = 1, \dots, N$. If there is no such control, the active control is chosen according to the SUR scheme.

KIRCHES, LENDERS, and MANNS proved tight bounds on the integral deviation gap for SUR-constructed solutions in their article [149]. They also extended the rounding scheme to the vanishing constrained case and proved δ-feasibility of the produced solution [162, 179]. More recent results focus on problems with greater real-world applicability and can be divided into the following areas:

- Solving problems with PDE constraints [117, 118, 177],
- Algorithms for (nonlinear) MPC [48, 49, 56],
- The efficient inclusion of time-coupled combinatorial constraints [282, 222, 28, 102], and
- Generalizations of the CIA decomposition and algorithms for other problem settings [280, 35, 277].

As of 2020, the research field of MIOCPs is still being actively investigated.

4.4 Incorporation of constraints

In this chapter, we present different ways of considering the combinatorial constraints from Section 3.2, path constraints of type (4.1e), and multiphase dynamics from (MMIOCP) in the CIA decomposition.

4.4.1 Combinatorial constraints

There are essentially two different ways for the CIA decomposition to construct binary controls that satisfy combinatorial constraints:

1. Impose the constraints in the rounding problem (CIA) or apply rounding algorithms that consider these constraints.

2. Add the constraints or auxiliary variants for $\boldsymbol{a} \in \mathscr{A}_N$ into (NLP$_{\text{rel}}$).

Combinations of the above options are possible.

Incorporation into (CIA)

The possibility of including combinatorial constraints into (CIA) arises naturally since MILPs provide the necessary modeling capability. Here, we consider discretized versions of the limited switching and MDT constraints from Section 3.2. We also call the requirement to limit the number of switches used the discrete total variation (TV) constraint.

Definition 4.10 (Discrete bounded total variation (TV) constraint)
Let a maximum number of switches $\sigma_{\max} \in \mathbb{N}$ be given together with the grid $\mathscr{G}_N$. The TV *constraint for $\boldsymbol{w} \in \Omega_N$ is defined as*

$$\sigma_{\max} \geq \frac{1}{2} \sum_{i \in [n_\omega]} \sum_{j \in [N-1]} |w_{i,j+1} - w_{i,j}|. \tag{4.8}$$

In terms of the TV concept introduced in Section 3.2, the constraint (4.8) indicates $\sigma_{\max} \geq TV(\boldsymbol{\omega})$, where $\boldsymbol{\omega}$ is discretized with $\boldsymbol{w}$. In fact, we count the switches in (4.8) twice since we sum the control that has just been deactivated with the one that has just been activated, explaining the factor of one-half in (4.8). We need the following differentiable reformulation of the TV constraint in order to solve the upcoming (CIA) subproblems efficiently.

Remark 4.4 (Reformulation of (4.8)**)**
Let $\mathscr{G}_N$ and $\sigma_{\max} \in \mathbb{N}$ be given. We use the auxiliary variables $\sigma_{i,j} \in \mathbb{N}$ and obtain a reformulation of the TV constraint (4.8) without the absolute value term by

$$\sigma_{\max} \geq \frac{1}{2} \sum_{i \in [n_\omega]} \sum_{j \in [N-1]} \sigma_{i,j}, \tag{4.9}$$

$$\sigma_{i,j} \geq \pm(w_{i,j+1} - w_{i,j}), \qquad i \in [n_\omega],\ j \in [N-1]. \tag{4.10}$$

Here we do not assume any mode-specific switching limits, which could be imposed by splitting up (4.9) into n_ω inequalities and dropping the first sum.

Remark 4.5 (Alternative TV reformulations)
Alternatives to (4.9)–(4.10) for reformulating the TV constraint (4.8) include, e.g., facet defining inequalities, as proposed in [224]. KIRCHES [145] suggested introducing further convex multipliers $\beta_{i,j} \in [0,1]$ and replacing (4.10) by setting

$$\sigma_{i,j} = (2\beta_{i,j+1} - 1)(w_{i,j+1} + w_{i,j} - 1) + 1 \quad \text{for } i \in [n_\omega], j \in [N-1].$$

A similar reformulation was introduced by Rieck [207]:

$$\sigma_{i,j} = w_{i,j-1} + w_{i,j+1} + 2w_{i,j}(1 - w_{i,j-1} - w_{i,j+1}) \quad \text{for } i \in [n_\omega], j \in \{2, \ldots, N-1\}.$$

These reformulations can be beneficial for the solution process with MILP solvers.

The following definition addresses the formulation of MDT constraints in the discretized setting.

Definition 4.11 (Discretized minimum dwell time constraints)
Let a grid $\mathscr{G}_N$ be given together with an MU time $C_U \geq 0$ and an MD time $C_D \geq 0$. For $\boldsymbol{w} \in \Omega_N$, we refer to the following constraints as MDT constraints:

$$w_{i,l} \geq w_{i,k+1} - w_{i,k}, \quad \textit{for } i \in [n_\omega],\, k \in [N-1],\, l \in \mathscr{J}_{k+1}(C_U), \tag{4.11}$$

$$1 - w_{i,l} \geq w_{i,k} - w_{i,k+1}, \quad \textit{for } i \in [n_\omega],\, k \in [N-1],\, l \in \mathscr{J}_{k+1}(C_D), \tag{4.12}$$

where we denote the intervals affected by the MDT $C_1 = C_U, C_D$ from interval $k \in [N]$ on with the set

$$\mathscr{J}_k(C_1) := \{k\} \cup \{j \mid t_{j-1} \in \mathscr{G}_N \cap [t_{k-1}, t_{k-1} + C_1)\}.$$

If a binary control is active after a switch on t_j, it must remain active for a time period of at least C_U, as required by the MU constraint (4.11), whereas the MD constraint (4.12) enforces the analogous case for the deactivation of a control. We remark that we assume no mode-specific MU times $C_{i,U}$ or MD times $C_{i,D}$, which may by included by setting $C_U = \max_{i \in [n_\omega]} C_{i,U}$ and accordingly $C_D = \max_{i \in [n_\omega]} C_{i,D}$, even though this simplification may result in suboptimal solutions.

Remark 4.6 (MDT reformulations)
Concerning the constraints (4.11) and (4.12), we stress that there are other, often computationally more efficient, formulations of MDT constraints, e.g., in the spirit of extended formulations [161]. Extended formulations may lead to relaxations that are less likely to deliver fractional solutions [161] and may therefore facilitate including the constraints (4.11) and (4.12) into the NLP solving procedure. We propose an extended formulation of (CIA) that allows us to deal with such reformulations in Chapter 6.

Due to their importance in applications, we will elaborate on tailored algorithms for (CIA) with TV and MDT constraints (Chapter 6) and investigate the resulting integral deviation gap (Chapter 7). To this end, we introduce specific (CIA) problem variants here.

Definition 4.12 (Problems (CIA-U), (CIA-D), (CIA-UD), and (CIA-TV))
Consider (CIA) from Definition 4.6. We define the (CIA) problem with the added MU time constraint (4.11) *from Definition 4.11 as (**CIA-U**). We call (CIA) with an added MD time constraint* (4.12) *(**CIA-D**), and we call (CIA) with both* (4.11) *and* (4.12) *(**CIA-UD**). Finally, (CIA) with added TV constraints* (4.9)-(4.10) *is hereafter referred to as (**CIA-TV**).*

Other combinatorial constraints, as defined in Section 3.2, can be discretized and included in (CIA) in a straightforward manner. Moreover, the idea of the *SUR-VC* algorithm [149], which is to prefix controls $w_{i,j} = 0$ if $a_{i,j} = 0$, can be directly transferred to (CIA) so that the constructed controls inherit the properties of *SUR-VC* with respect to vanishing constraints.

Incorporation into the NLP step

Combinatorial constraints cannot be modeled by (NLP_{rel}) in a natural way since it does not include binary variables. Nevertheless, there are auxiliary ways of accommodating these restrictions in this problem step. First, the constraints from the Remark 4.4 and Definition 4.11 can be applied to $\boldsymbol{a}$ and added to (NLP_{rel}). There is no guarantee that the optimal $\boldsymbol{a}^*$ is binary, but it may appear that it structurally almost obeys the constraints. Numerical studies indicate, however, no apparent effect of adding the constraints on $\boldsymbol{a}$ [282]; moreover (NLP_{rel}) becomes more challenging to solve because of the increased number of constraints.

Another possibility, at least for MDT constraints, are *blocking constraints* [56]. In this approach, the intervals $j \in [N]$ are sorted into $K \in \mathbb{N}$ subsequent subsets $\mathscr{I}_k,\ k \in [K]$, representing the MDT periods. The values $\boldsymbol{a}_{\cdot,j}$ are then set to be equal on these subsets, i.e., for all $k \in [K]$, we impose

$$a_{i,j} = a_{i,l}, \qquad \text{for } j, l \in \mathscr{I}_k,\ \ i \in [n_\omega].$$

As an advantage, the SUR scheme can be applied to these blocks of intervals, and the constructed solution satisfies MDT constraints. Nevertheless, the number of degrees of freedom in (NLP_{rel}) is already greatly reduced, which can lead to highly suboptimal solutions.

To avoid fractional values and a large number of switches in the relaxed solution $\boldsymbol{a}^*$, it has been proposed [218] to add penalty terms Φ_{pen} of the following form to the (NLP_{rel}) objective:

$$\Phi_{\text{pen}} = \beta \sum_{i \in [n_\omega]} \sum_{j \in [N]} a_{i,j}(1 - a_{i,j}),$$

where $\beta \geq 0$ refers to a penalty factor that can be iteratively scaled down. The issue with this approach is that the relationship between the optimal solutions of the penalized and original problems remains unclear [218], and the penalties may attract solutions in which switches appear more frequently [145].

Considering the aspects discussed above, it seems advisable to add combinatorial constraints to (CIA) as a standard case and to only consider modifications of (NLP_{rel}) in specific cases.

4.4.2 Path constraints

Even if (NLP_{rel}) is feasible and the optimal solution $\boldsymbol{a}^*$ is used to construct a binary control function $\boldsymbol{w}^*$ in (CIA), the resulting problem (NLP_{bin}) with fixed $\boldsymbol{w}^*$ is not necessarily feasible due to the path constraint (4.1e). As we show in Chapter 5, the possible infeasibility disappears as the problem discretization becomes finer. However, in many applications the grid is fixed or refinement is undesirable, so a workaround is needed to construct binary controls $\boldsymbol{w}^*$ that lead to a feasible problem (NLP_{bin}).

Forward integration of differential states in (CIA)

Consider given $\boldsymbol{u}^*_{\cdot,j}$ from (NLP_{rel}) with $j \in [N]$, a chosen integrator function, and initial state values $\boldsymbol{x}_0$. Then, the validation of the discretized path constraint (4.1e) is possible as soon as $\boldsymbol{x}_{\cdot,j}$ is known, which is through the dynamic system (4.1b)-(4.1c) and given $\boldsymbol{u}^*_{\cdot,j}$ and $\boldsymbol{x}_0$ a dependent variable of $\boldsymbol{w}$. We can thus reinterpret the path constraint as a nonlinear constraint on $\boldsymbol{w}$

evaluated on the intervals $j \in [N]$:

$$\boldsymbol{g}_j(\boldsymbol{w}) := \boldsymbol{c}(t_j, \boldsymbol{x}_{\cdot,j}(\boldsymbol{w}), \boldsymbol{u}^*_{\cdot,j}) \geq \boldsymbol{0}_{n_c}, \tag{4.13}$$

where the constraint function $\boldsymbol{g}_j$ depends on the outcome of $\boldsymbol{x}_{\cdot,j}$, i.e., the ODE constraint and the integration scheme. The constraint (4.13) may be added to (CIA) to ensure feasibility with respect to (4.1e). However, we note that this constraint induces a nonlinear structure into the MILP since the discretized ODE is generally nonlinear. Moreover, (4.13) can only be evaluated on the interval $j \in [N]$ if $\boldsymbol{x}$, and thus $\boldsymbol{w}$ is known for all previous intervals $k \leq j$. This time-dependent structure induces a huge number of possible outcomes for $\boldsymbol{g}_j$ and therefore a huge number of constraints. In Sections 6.3 and 6.4, we discuss time-exploiting BnB algorithms that branch forward in time. These algorithms can be extended to save the state values and thus check the feasibility of the path constraints. This algorithmic idea was tested in [166] with the result of guaranteed feasibility at the expense of decreased run time performance.

A heuristic first-order Taylor approximation constraint

After solving (NLP_{rel}), the intervals on which the optimal solution fulfills (4.1e) with (close to) equality can be identified. On these intervals, the variables $\boldsymbol{w}$ may be critical in terms of path constraint violations if they are “rounded in the wrong direction” such that (4.1e) can no longer be satisfied in (NLP_{bin}). To this end, we approximate the path constraint function value of the state trajectory $\boldsymbol{x}^{\boldsymbol{w}}(t)$ that is based on the binary controls $\boldsymbol{w}$ with a first-order Taylor polynomial. The latter consists of the path constraint function value of the state trajectory $\boldsymbol{x}^{\boldsymbol{a}}(t)$, which is based on the relaxed controls $\boldsymbol{a}^*$, and of a product term of the difference of the state trajectory of the relaxed and binary controls multiplied by the path constraint function derivative with respect to $\boldsymbol{x}$. The idea is to impose a constraint in (CIA) that restricts the first-order Taylor term on the intervals that could be critical with respect to path constraint violations. We first introduce the evaluated model function terms $\tilde{f}_{i,j,k}$ and then express this constraint idea in Definition 4.14.

Definition 4.13 (Evaluated model function $\tilde{f}_{i,j,k}$)
Let $\boldsymbol{x}^*$, $\boldsymbol{u}^*$ be the (discretized) optimal solution of (NLP_{rel}). We define the evaluated right-hand side function terms from (4.1b) *as*

$$\tilde{f}_{i,j,k} := \frac{1}{\Delta_j} \int_{t_{j-1}}^{t_j} f_{i,k}(t, \boldsymbol{x}^*(t), \boldsymbol{u}^*(t)) \, \mathrm{d}t, \qquad \textit{for } i \in [n_\omega],\ k \in [n_x]. \tag{4.14}$$

Definition 4.14 (Path constraint Taylor approximation)
Consider a grid $\mathscr{G}_N$. Let $\boldsymbol{x}^*$, $\boldsymbol{u}^*$, and $\tilde{\boldsymbol{f}}$ be given after solving (NLP_{rel}). Further, let $\mathscr{J}_{\text{path}} \subseteq [N]$ denote the intervals on which the optimal solution of (NLP_{rel}) is constrained by (4.1e). *We abbreviate the partial derivative of $\boldsymbol{c}$ with respect to $\boldsymbol{x}$ by $\boldsymbol{c_x} := \frac{d\boldsymbol{c}}{d\boldsymbol{x}}$. Then, we define the* path constraint Taylor approximation constraint *as*

$$\boldsymbol{0}_{n_c} \leq \boldsymbol{c_x}(t_j, \boldsymbol{x}^*_{\cdot,j}, \boldsymbol{u}^*_{\cdot,j}) \sum_{l \in [j]} \sum_{i \in [n_\omega]} (w_{i,l} - a^*_{i,l}) \Delta_l \tilde{\boldsymbol{f}}_{i,l,\cdot}, \qquad \textit{for } j \in \mathscr{J}_{\text{path}}. \tag{4.15}$$

The above constraint is a heuristic approach to preventing path constraint infeasibilities, and it can be added to (CIA) without changing the linear problem nature. We motivate the constraint approximation in Section 5.5, where we compute the (vanishing) distance of the two state trajectories based on relaxed and binary controls.

4.4.3 Multiphase dynamics

We recall the multiphase problem (MMIOCP), where the model function $\boldsymbol{f}$ depends on the switching function $s(t)$ that maps each time point $t \in \mathcal{T}$ to its associated phase $p \in [n_p]$, $n_p \in \mathbb{N}$:

$$s: \ \mathcal{T} \to [n_p].$$

Next, we define the corresponding partial outer convexification variant of (MMIOCP).

Definition 4.15 (Multiphase binary optimal control problem (MBOCP))
Consider the function space for the multiphase binary controls defined by

$$\Omega^p := \left\{ \boldsymbol{\omega} \in L^\infty\left(\mathcal{T}, \{0,1\}^{n_\omega \times n_p}\right) \mid \sum_{i\in[n_\omega]} \sum_{p\in[n_p]} \omega_{i,p}(t) = 1, \ \textit{for a.a. } t \in \mathcal{T} \right\}.$$

We define (MBOCP) as the problem (BOCP), in which the ODE constraint (4.1b) *is replaced by*

$$\dot{\boldsymbol{x}}(t) \ = \ \boldsymbol{f}_0(t, \boldsymbol{x}(t), \boldsymbol{u}(t)) + \sum_{i\in[n_\omega]} \sum_{p\in[n_p]} \omega_{i,p}(t) \ \boldsymbol{f}_i(t, \boldsymbol{x}(t), \boldsymbol{u}(t), s(t)), \quad \textit{for a.a. } t \in \mathcal{T}, \tag{4.16}$$

and the vanishing constraint (4.1f) *holds for all* $p \in [n_p]$ *with* $\omega_{i,p}$ *applied instead of* ω_i. *Moreover, the Lagrange objective term is altered by an additional sum over all phases.*

Proposition 4.3 (Equivalence of (MBOCP), (MMIOCP), and (BOCP))
The problems (MBOCP), (MMIOCP), and (BOCP) are equivalent in the sense that there is a bijection between any feasible, respectively optimal, solution for any of these problems to a feasible, respectively optimal, solution for any of the other problems.

Proof. The equivalence of (MBOCP) and (MMIOCP) can be analogously shown as in the proof of Proposition 4.1. For the equivalence of (MBOCP) and (BOCP), we first observe that every solution of the latter problem corresponds to a solution of the former with only one phase, i.e., $n_p = 1$. For the other direction, we argue that if we set $\boldsymbol{f}_i(\cdot, s(t))$ to zero outside the corresponding phase p for all $i \in [n_\omega]$, we obtain an equivalent (BOCP) with $n_p \cdot n_\omega$ modes. In this way, each phase in (MBOCP) induces n_ω additional modes in (BOCP). □

With the above proposition at hand, the CIA decomposition is generally applicable to the multiphase setting. We briefly outline the practical outcome of the algorithm. First, the discretized relaxed and binary control functions are also indexed by the phases $p \in [n_p]$, i.e., $a_{i,j,p} \in [0,1]$ and $w_{i,j,p} \in \{0,1\}$. We need to fix the controls outside their corresponding phases because otherwise, we might accumulate controls that do not belong to the same phase and thus the same model function $\boldsymbol{f}_{i,p}$:

$$w_{i,j,p} = 0, \ \text{for } i \in [n_\omega], \ j \in [N] \text{ if } \ s(t_{j-1}) \neq p. \tag{4.17}$$

We note that the above constraint corresponds to simple variable fixing since $s(t_{j-1})$ is known beforehand. Finally, we introduce the multiphase CIA problem.

Definition 4.16 (MCIA)
Let $\boldsymbol{a}^* \in [0,1]^{n_\omega \times N \times n_p}$ be given. We define the multiphase CIA **(MCIA)** *problem as*

$$\begin{aligned}
\min_{w_{i,j,p} \in \{0,1\},\ \theta \geq 0} \quad & \theta \\
\text{s.t.} \quad & 1 = \sum_{i \in [n_\omega]} \sum_{p \in [n_p]} w_{i,j,p}, && j \in [N], \\
& \theta \geq \pm \sum_{l \in [j]} (a^*_{i,l,p} - w_{i,l,p}) \Delta_l, && i \in [n_\omega],\ j \in [N],\ p \in [n_p], \\
& \textit{Phase fixing constraint (4.17).}
\end{aligned}$$

We apply this multiphase CIA decomposition in Section 10.1. We remark that for each phase (MCIA) may be further decomposed into (CIA) subproblems for complexity reduction; however, this comes at the expense of possible suboptimal solutions when combinatorial constraints are included.

4.5 Generalized combinatorial integral approximation decompositions

This section is largely based on [280] and [212]. The solution constructed by the CIA decomposition for (BOCP) can be improved in terms of the objective function value and constraint violations by applying the algorithm on a finer discretization grid [225]. Nevertheless, in some cases, it is not possible or desirable to apply grid refinement:

1. In time-critical settings, such as MPC, another round of problem solving with an increased problem size may not be applicable [49].

2. The problem size of some instances of discretized MIOCP may already huge, as in PDE constrained problems [114], limiting the possibility for refinement.

3. There are applications in which the discretization is fixed or confined to narrow limits.

We therefore propose generalizations of the decomposition approach that work without grid refinement. One of these approaches is to apply different rounding problems, instead of (CIA). For instance, one may construct a second-order Taylor approximation modification of the binary controls in the optimal solution of (NLP$_{\text{rel}}$). This results in an MIQP based on a Gauss-Newton-type linear-quadratic expansion that is currently being investigated.

4.5.1 Different MILP formulations

We introduced (CIA) as the problem of minimizing the discretized integral deviation gap from Definition 4.5. We now consider control approximation problems that rely on different scalings. The following definition provides a notion of these problems, in which the vector norm $\|\cdot\|$ remains unspecified.

Definition 4.17 ($\tilde{\boldsymbol{\lambda}}_{j,k}, \boldsymbol{\theta}^*_{\text{CIA}}, \boldsymbol{\theta}^*_{\text{SCIA}}, \boldsymbol{\theta}^*_{\lambda\text{CIA}}$)
Let $\boldsymbol{a}^* \in \mathscr{A}_N$ be the given optimal solution of (NLP$_{\text{rel}}$), and let the evaluated model function values $\tilde{\boldsymbol{f}}$ be given as introduced in Definition 4.13. We denote by $\tilde{\lambda}_{j,k} \in \mathbb{R},\ j \in [N],\ k \in [n_x]$, the

discretized and evaluated dual variables of the ODE constraint (4.1b) *in (NLP*$_{\text{rel}}$*). Consider a vector norm* $\|\cdot\|$*. We introduce the following optimization problems:*

$$\theta^*_{CIA} := \min_{\boldsymbol{w}\in\Omega_N} \max_{j\in[N]} \left\| \sum_{l\in[j]} (\boldsymbol{a}^*_{\cdot,l} - \boldsymbol{w}_{\cdot,l})\Delta_l \right\|, \tag{4.18}$$

$$\theta^*_{SCIA} := \min_{\boldsymbol{w}\in\Omega_N} \max_{j\in[N]} \left\| \sum_{l\in[j]} \sum_{i\in[n_\omega]} (a^*_{i,l} - w_{i,l})\Delta_l \tilde{\boldsymbol{f}}_{i,l} \right\|, \tag{4.19}$$

$$\theta^*_{\lambda CIA} := \min_{\boldsymbol{w}\in\Omega_N} \max_{j\in[N]} \left| \sum_{k\in[n_x]} \tilde{\lambda}_{j,k} \sum_{l\in[j]} \sum_{i\in[n_\omega]} (a^*_{i,l} - w_{i,l})\Delta_l \tilde{f}_{i,l,k} \right|. \tag{4.20}$$

Norm dependent MILP formulation

Thus far, we have considered the maximum norm in (CIA). We now introduce the 1-norm MILP analogue with auxiliary variables $\zeta_{i,j} \geq 0,\ i \in [n_\omega], j \in [N]$, and thereby specify the norm choices for Definition 4.17.

Definition 4.18 (CIA1)
Let $\boldsymbol{a}^* \in \mathscr{A}_N$ *and a grid* $\mathscr{G}_N$ *be given. We define* ***(CIA1)*** *as the problem*

$$\min_{\boldsymbol{w}\in\Omega_N,\theta,\zeta_{i,j}\geq 0} \quad \theta \tag{4.21}$$

$$\text{s.t.} \quad \theta \geq \sum_{i\in[n_\omega]} \zeta_{i,j}, \qquad \textit{for } j \in [N], \tag{4.22}$$

$$\zeta_{i,j} \geq \pm \sum_{l\in[j]} (a^*_{i,l} - w_{i,l})\Delta_l, \qquad \textit{for } i \in [n_\omega],\ j \in [N]. \tag{4.23}$$

The next definition is dedicated to θ^*_{SCIA} and both the maximum norm and 1-norm.

Definition 4.19 ((SCIAmax),(SCIA1))
Let $\boldsymbol{a}^* \in \mathscr{A}_N$*, the evaluated model functions* $\tilde{\boldsymbol{f}} \in \mathbb{R}^{n_\omega \times N \times n_x}$*, and a grid* $\mathscr{G}_N$ *be given. We define* ***(SCIAmax)*** *as*

$$\min_{\boldsymbol{w}\in\Omega_N,\theta\geq 0} \quad \theta \tag{4.24}$$

$$\text{s.t.} \quad \theta \geq \pm \sum_{l\in[j]} \sum_{i\in[n_\omega]} (a^*_{i,l} - w_{i,l})\Delta_l \tilde{f}_{i,l,k}, \qquad \textit{for } j \in [N],\ k \in [n_x]. \tag{4.25}$$

We introduce the auxiliary variables $\zeta_{j,k} \geq 0,\ j \in [N],\ k \in [n_x]$*, and define* ***(SCIA1)*** *as*

$$\min_{\boldsymbol{w}\in\Omega_N,\theta,\zeta_{j,k}\geq 0} \quad \theta \tag{4.26}$$

$$\text{s.t.} \quad \theta \geq \sum_{k\in[n_x]} \zeta_{j,k}, \qquad \textit{for } j \in [N], \tag{4.27}$$

$$\zeta_{j,k} \geq \pm \sum_{l\in[j]} \sum_{i\in[n_\omega]} (a^*_{i,l} - w_{i,l})\Delta_l \tilde{f}_{i,l,k}, \qquad \textit{for } j \in [N],\ k \in [n_x]. \tag{4.28}$$

In the context of different MILP formulations, we refer to the problem (CIA) as **(CIAmax)** to emphasize the application of the maximum norm. The following definition presents an MILP variant that takes the dual variables into account.

Definition 4.20 ($\boldsymbol{\lambda}$CIA1)
*Let $\boldsymbol{a}^* \in \mathscr{A}_N$, the evaluated model functions $\tilde{\boldsymbol{f}} \in \mathbb{R}^{n_\omega \times N \times n_x}$, the discretized and evaluated dual variables $\tilde{\lambda}_{j,k} \in \mathbb{R}$, $j \in [N]$, $k \in [n_x]$, and a grid $\mathscr{G}_N$ be given. We define **(λCIA1)** as*

$$\min_{\boldsymbol{w} \in \Omega_N, \theta, \zeta_{j,k} \geq 0} \quad \theta \tag{4.29}$$

$$\text{s.t.} \quad \theta \geq \sum_{k \in [n_x]} \zeta_{j,k}, \qquad \text{for } j \in [N], \tag{4.30}$$

$$\zeta_{j,k} \geq \pm \tilde{\lambda}_{j,k} \sum_{l \in [j]} \sum_{i \in [n_\omega]} (a^*_{i,l} - w_{i,l}) \Delta_l \tilde{f}_{i,l,k}, \qquad \text{for } j \in [N],\ k \in [n_x]. \tag{4.31}$$

Of course, other norms, such as the Euclidean norm, can be used. We do not consider them here since they would impose nonlinear constraints on the MILPs.

Chronologically ordered constraints

A further possibility for modifying the (CIA) problems is to alter the chronological order in the constraints for the accumulated difference $\| \sum_{l \in [j]} (\boldsymbol{a}^*_{\cdot,l} - \boldsymbol{w}_{\cdot,l}) \Delta_l \|$ for $j \in [N]$. Instead of starting from the first interval $j = 1$, we may use an arbitrary ordering of time intervals. We consider backward accumulation, starting from the interval with index $j = N$, i.e., $[t_{N-1}, t_N]$:

$$\theta \geq \pm \sum_{l=j}^{N} (a^*_{i,l} - w_{i,l}) \Delta_l, \qquad \text{for } i \in [n_\omega],\ j \in [N]. \tag{4.32}$$

We denote the problem in which (4.32) replaces (4.4) in (CIAmax) by **(CIAmaxB)**. The other defined MILPs can be modified analogously with backward time accumulation and are named accordingly; e.g., **(SCIA1B)** refers to (SCIA1) with backward accumulation.

4.5.2 Recombination as postprocessing

We group the following CIA-type MILP formulations from Section 4.5.1 into the set S^{CIA}.

Definition 4.21 ($\boldsymbol{S^{\text{CIA}}, S^{\text{REC}}}$)
We define the set of CIA problems S^{CIA} from the previous section via

$$S^{CIA} := \{(\text{CIAmax}), (\text{CIA1}), (\text{CIAmaxB}), (\text{CIA1B}), (\lambda\text{CIA1}), (\lambda \mathit{CIA1B}),$$
$$(\text{SCIAmax}), (\text{SCIA1}), (\text{SCIAmaxB}), (\text{SCIA1B})\}.$$

For a subset $\tilde{S}^{CIA} \subseteq S^{CIA}$, let $n_{CIA} := |\tilde{S}^{CIA}|$ denote the number of different CIA problem formulations. Let the elements of $\tilde{S}^{CIA}$ be numbered by $1, \ldots, n_{CIA}$. We define the set S^{REC} of recombination mappings $F^{rec} \in S^{REC}$ via

$$F^{rec}: \bigtimes_{k \in [n_{CIA}]} \Omega_N \to \Omega_N, \qquad F^{rec}\left(\boldsymbol{w}^1, \ldots, \boldsymbol{w}^{n_{CIA}}\right) \mapsto \boldsymbol{w}^{rec}, \tag{4.33}$$

where $\boldsymbol{w}^k$ denotes the optimal solution of the problem $(milp)^k \in \tilde{S}^{CIA}$.

In Algorithm 4.2, we present a generalized CIA decomposition based on the solution of different (CIA) problems and recombination heuristics. We solve different MILPs to approximate the

Algorithm 4.2: Generalized CIA decomposition algorithm with several MILPs and recombination for error-controlled solution of (BOCP)

```
Input : (MINLP) instance with time grid $\mathscr{G}_N$ as discretization of (BOCP), algorithmic
        choices of sets $\tilde{S}^{\text{CIA}} \subseteq S^{\text{CIA}}$ and $\tilde{S}^{\text{REC}} \subseteq S^{\text{REC}}$.
1 Solve $(\text{NLP}_{\text{rel}}) \rightarrow \boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{a}^*, \mathscr{C}^*$;
2 for $m \in \tilde{S}^{CIA}$ do
3     Solve $m$ for $\boldsymbol{a}^* \rightarrow \boldsymbol{w}^m$;
4     Solve $(\text{NLP}_{\text{bin}})$ with $\boldsymbol{w} = \boldsymbol{w}^m$ fixed $\rightarrow \boldsymbol{x}^m, \boldsymbol{u}^m, \mathscr{C}^m$;
5 for $F^{rec} \in \tilde{S}^{REC}$ do
6     Create $\boldsymbol{w}^{\text{rec}}$ using $F^{\text{rec}}(\boldsymbol{w}^{\text{m}})$, $\mathscr{C}^{\text{m}}$ from all m $\in \tilde{S}^{\text{CIA}}$.
7     Solve $(\text{NLP}_{\text{bin}})$ with $\boldsymbol{w} = \boldsymbol{w}^{\text{rec}}$ fixed $\rightarrow \boldsymbol{x}^{\text{rec}}, \boldsymbol{u}^{\text{rec}}, \mathscr{C}^{\text{rec}}$;
8 Find opt $\in \tilde{S}^{\text{CIA}} \cup \tilde{S}^{\text{REC}}$ with $\mathscr{C}^{\text{opt}} = \min\left\{ \min_{m \in \tilde{S}^{\text{CIA}}} \mathscr{C}^m, \min_{F^{\text{rec}} \in \tilde{S}^{\text{REC}}} \mathscr{C}^{\text{rec}} \right\}$;
9 return: $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}, \mathscr{C}) = (\boldsymbol{x}^{\text{opt}}, \boldsymbol{u}^{\text{opt}}, \boldsymbol{w}^{\text{opt}}, \mathscr{C}^{\text{opt}})$;
```

relaxed controls with binary ones (lines 2–3). Their performance is evaluated (line 4) by solving $(\text{NLP}_{\text{bin}})$ and thus, calculating their corresponding state trajectories and objective values. We use the binary controls in several recombination heuristics to create new candidate binary controls (lines 5–6), which we also evaluate (line 7). Finally, the best performing binary control is selected as a solution (line 8). We note that the objective $\mathscr{C}$ may include a constraint violation term $\mathscr{C}^{\text{pen}}$ and a tracking term $\mathscr{C}^{\text{track}}$ that minimizes the approximation error between differential states of the relaxed and binary controls. Algorithm 4.2 is a generalization of the basic CIA Algorithm 4.1 in the sense that for the latter, $\tilde{S}^{\text{REC}}$ is empty and $\tilde{S}^{\text{CIA}}$ contains only one CIA problem formulation. In the case of a highly constrained (BOCP), Algorithm 4.2 can be altered such that the recombination heuristics aim to create binary controls that result in a feasible problem $(\text{NLP}_{\text{bin}})$. The algorithms can be adapted to check the feasibility of the constructed solution for (MINLP), but we argue that the algorithm may construct a feasible solution by using $\mathscr{C}^{\text{pen}}$ and a tracking term $\mathscr{C}^{\text{track}}$ in the objective. The general framework is open to the application of different heuristics, such as genetic algorithms [97]. In the following, we provide examples of the recombination heuristics in S^{REC}.

GreedyTime recombination

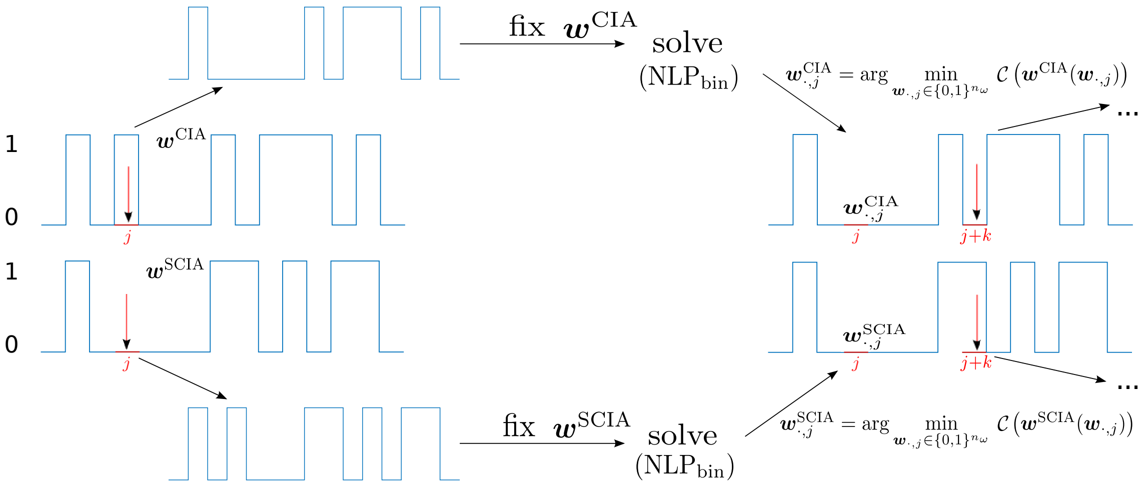
Algorithm 4.3 establishes a routine for using the MILP solutions in a greedy approach with the aim of constructing solutions $\boldsymbol{w}$ with an improved objective value $\mathscr{C}(\boldsymbol{w})$, where we write $\mathscr{C}(\boldsymbol{w})$ to indicate the (indirect) dependency of the objective on the specific binary control function $\boldsymbol{w}$.

GreedyTime iterates over all intervals $j \in [N]$ in chronological order (line 1). In line 2, on every interval we check if there are MILP pairs (m_1, m_2) that differ in their binary control vectors. For each of these pairs, we recombine the m_1 solution with the binary control vector from m_2 at interval j to create a temporary solution $\tilde{\boldsymbol{w}}^{m_1}$ (line 3). Then, we evaluate the objective of this new solution in line 4 and overwrite the binary control $\boldsymbol{w}^{m_1}$ with the recombined solution $\tilde{\boldsymbol{w}}^{m_1}$ if that latter results in a better objective value (lines 5–6). In the same way, we proceed with the second solution m_2 when the (same) pair (m_2, m_1) appears in the inner loop.

Algorithm 4.3: *GreedyTime* heuristic for recombining binary controls $\boldsymbol{w}^m$, $\mathrm{m} \in S^{\mathrm{CIA}}$.

```
Input : Grid $\mathscr{G}_N$, binary controls $\boldsymbol{w}^{\mathrm{m}}$ as optimal solutions of $\mathrm{m} \in S^{\mathrm{CIA}}$, corresponding
        objective values $\mathscr{C}(\boldsymbol{w}^{\mathrm{m}})$.
1 for $j \in [N]$ do
2     for $(m_1, m_2) \in S^{CIA} \times S^{CIA}$, $m_1 \neq m_2$, $\boldsymbol{w}^{m_1}_{\cdot,j} \neq \boldsymbol{w}^{m_2}_{\cdot,j}$ do
3         Set $\tilde{\boldsymbol{w}}^{m_1}_{\cdot,k} = \boldsymbol{w}^{m_1}_{\cdot,k}, k \neq j, k \in [N]$ and $\tilde{\boldsymbol{w}}^{m_1}_{\cdot,j} = \boldsymbol{w}^{m_2}_{\cdot,j}$;
4         Solve (NLP$_{\mathrm{bin}}$) with $\boldsymbol{w} = \tilde{\boldsymbol{w}}^{m_1}$ fixed $\rightarrow \mathscr{C}(\tilde{\boldsymbol{w}}^{m_1})$;
5         if $\mathscr{C}(\tilde{\boldsymbol{w}}^{m_1}) \leq \mathscr{C}(\boldsymbol{w}^{m_1})$ then
6             Set $\boldsymbol{w}^{m_1}_{\cdot,j} = \tilde{\boldsymbol{w}}^{m_1}_{\cdot,j}$ and $\mathscr{C}(\boldsymbol{w}^{m_1}) = \mathscr{C}(\tilde{\boldsymbol{w}}^{m_1})$;
7 return: $\mathscr{C}(\boldsymbol{w}^{\mathrm{rec}}) := \min_{\mathrm{m} \in S^{\mathrm{CIA}}} \mathscr{C}(\boldsymbol{w}^m)$;
```

Note that with a large number of calculated MILPs, there may also be a large number of pairs with unequal solutions. Instead of swapping and testing each variation for every $\boldsymbol{w}^{m_1}$, it is advisable to only swap the $\boldsymbol{w}^{m_2}$ solution with the currently smallest objective value. When there are no continuous controls $\boldsymbol{u}$, it is straightforward to evaluate (NLP$_{\mathrm{bin}}$) in line 4 with the previously found and fixed $\boldsymbol{x}$ until grid interval j. Since (NLP$_{\mathrm{bin}}$) needs to be solved iteratively, this speeds up the process in problems with fine grids and numerous MILP solutions. Moreover, if an MILP solution m_1 differs from two MILPs m_2, m_3 with identical binary control vectors $\boldsymbol{w}^{m_2}_{\cdot,j} = \boldsymbol{w}^{m_3}_{\cdot,j}$, it is sufficient to test recombination with only one of the two. Fig. 4.2 illustrates an example recombination step for the pairs (CIA,SCIA) and (SCIA,CIA).



fix $\boldsymbol{w}^{\mathrm{CIA}}$
solve (NLP$_{\mathrm{bin}}$)
$\boldsymbol{w}^{\mathrm{CIA}}_{\cdot,j} = \arg \min_{\boldsymbol{w}_{\cdot,j} \in \{0,1\}^{n_w}} \mathcal{C}\left(\boldsymbol{w}^{\mathrm{CIA}}(\boldsymbol{w}_{\cdot,j})\right)$
...
1
0
$\boldsymbol{w}^{\mathrm{CIA}}$
j
$\boldsymbol{w}^{\mathrm{CIA}}_{\cdot,j}$
j
$j{+}k$
1
0
$\boldsymbol{w}^{\mathrm{SCIA}}$
j
$\boldsymbol{w}^{\mathrm{SCIA}}_{\cdot,j}$
j
$j{+}k$
...
fix $\boldsymbol{w}^{\mathrm{SCIA}}$
solve (NLP$_{\mathrm{bin}}$)
$\boldsymbol{w}^{\mathrm{SCIA}}_{\cdot,j} = \arg \min_{\boldsymbol{w}_{\cdot,j} \in \{0,1\}^{n_w}} \mathcal{C}\left(\boldsymbol{w}^{\mathrm{SCIA}}(\boldsymbol{w}_{\cdot,j})\right)$

Figure 4.2: Visualization of the *GreedyTime* algorithm. Two candidate controls, here from (CIA) and (SCIA), are used to construct new candidates. An enumeration between 0 and 1 is performed at all intervals j when the input vectors differ. The two candidate controls $\boldsymbol{w}$ are fixed and (NLP$_{\mathrm{bin}}$) is solved for both vectors. The resulting objective function values are compared with their previous values, and the vectors $\boldsymbol{w}_{\cdot,j}$ with the smaller objective values are fixed in the candidate solutions. This procedure is repeated on the subsequent intervals with unequal candidate solutions.

Remark 4.7 (*GreedyTime* modifications)

1. The outer loop in Algorithm 4.3 can also be applied backward in time. We name the backward version *GreedyTimeBackward*.

2. Instead of looping over all intervals, we may consider only singular arcs since the constructed binary controls are likely to be equal on bang-bang-arcs. Following Definition 2.7, here by singular arcs, we refer to intervals on which $\epsilon < a_{i,j} < 1-\epsilon$ holds, for a certain threshold $\epsilon > 0$.

3. *Greedy-cost-to-go*: Assume we have calculated the dual variables $\tilde{\lambda}_{j,k},\ j \in [N], k \in [n_\mathrm{x}]$, as introduced in Definition 4.17. Then, re-sort the intervals $[N]$ in descending order according to $\sum_{k\in[n_\mathrm{x}]} |\tilde{\lambda}_{j,k}|,\ j \in [N]$. This results in a new ordered grid $\mathscr{G}_N^{\tilde{\lambda}}$ to be applied in Algorithm 4.3.

We test *GreedyTime* and its modifications in Section 9.1.

Singular arc recombination

Usually, when $\boldsymbol{a}^*$ is (almost) binary on certain intervals, $\boldsymbol{w}$ should attain these binary values as well as an optimal solution of a CIA rounding problem – regardless of the MILP choice. To this end, we formalize singular arcs of $\boldsymbol{a}^*$ as sets of consecutive intervals on which the relaxed control takes values smaller than ϵ or larger than $1-\epsilon$, where $\epsilon > 0$ is a chosen tolerance.

Definition 4.22 (Singular arc interval sets $\mathscr{I}_l^{\mathrm{sing}}$, number of singular arcs n_{sing})
Consider $\boldsymbol{a}^* \in \mathscr{A}_N$ and a small chosen tolerance $\epsilon > 0$. Let $k_0^{\mathrm{end}} := 0$. We define the following singular arc interval index sets iteratively for $l \geq 1$:

$$\begin{aligned} k_l^{\mathrm{start}} &:= \min\left\{ j \in [N] \mid j > k_{l-1}^{\mathrm{end}} \wedge \exists i \in [n_\omega] : a_{i,j}^* \in [\epsilon, 1-\epsilon] \right\}, \\ k_l^{\mathrm{end}} &:= \max\left\{ j \in [N] \mid \forall r = k_l^{\mathrm{start}}, \dots, j \;\; \exists i \in [n_\omega] : a_{i,r}^* \in [\epsilon, 1-\epsilon] \right\}, \\ \mathscr{I}_l^{\mathrm{sing}} &= \left\{ k_l^{\mathrm{start}}, \dots, k_l^{\mathrm{end}} \right\}. \end{aligned}$$

Moreover, we introduce the number of singular arcs n_{sing} as

$$n_{\mathrm{sing}} := \arg\max_{l\in\mathbb{N}} \left\{ k_l^{\mathrm{end}} \right\}.$$

Algorithm 4.4 aims to recombine singular arc realizations of the different MILP solutions from S^{CIA}.

The algorithm initializes the set of visited binary controls as empty and sets the so-far best objective value $\mathscr{C}^{\mathrm{rec}}$ to infinity (line 1). We set the temporary binary control $\boldsymbol{w}^{\mathrm{tmp}}$ on the bang-bang arcs as equal to the rounded relaxed control (line 2). Then, we test every possible variation (line 3) of the different MILP solutions on the singular arcs to fill up the singular arcs of the temporary binary control $\boldsymbol{w}^{\mathrm{tmp}}$ (line 4). We check the constructed control $\boldsymbol{w}^{\mathrm{tmp}}$ if it has already been visited (line 5), and if so, the algorithm jumps to the next iteration (line 6). Otherwise, $\boldsymbol{w}^{\mathrm{tmp}}$ is included in the set of visited controls (line 8), and we evaluate its objective value (line 9). When a recombination has a lower objective value than the so-far best control, it will be saved as the so-far best control (lines 10–12). Figure 4.3 illustrates the algorithm.

Algorithm 4.4: *Singular arc block* heuristic for recombining binary controls $\boldsymbol{w}^m$, m $\in S^{\mathrm{CIA}}$

```
Input : Grid $\mathcal{G}_N$, small singular arc tolerance $\epsilon > 0$, singular arcs interval sets $\mathcal{I}_l^{\mathrm{sing}}$,
        relaxed control $\boldsymbol{a}^* \in \mathcal{A}_N$, binary controls $\boldsymbol{w}^{\mathrm{m}}$, m $\in S^{\mathrm{CIA}}$, corresponding objective
        values $\mathcal{C}(\boldsymbol{w}^{\mathrm{m}})$.
Set $S^{\mathrm{vis}} = \emptyset$ and $\mathcal{C}^{\mathrm{rec}} = \infty$;
Set $w_{i,j}^{\mathrm{tmp}} = \lfloor a_{i,j}^* + \epsilon \rfloor$, for $i \in [n_\omega]$, $j \in [N] \setminus \left\{ \mathcal{I}_l^{\mathrm{sing}} \right\}_{l \in [n_{\mathrm{sing}}]}$;
for $(m_1, \dots, m_{n_{\mathrm{sing}}}) \in \bigtimes_{l \in [n_{\mathrm{sing}}]} S^{CIA}$ do
    Set $\boldsymbol{w}_{\cdot,j}^{\mathrm{tmp}} = \boldsymbol{w}_{\cdot,j}^{m_l}$, for $j \in \mathcal{I}_l^{\mathrm{sing}}$, $l \in [n_{\mathrm{sing}}]$;
    if $\boldsymbol{w}^{\mathrm{tmp}} \in S^{\mathrm{vis}}$ then
        continue;
    else
        Set $S^{\mathrm{vis}} = S^{\mathrm{vis}} \cup \{\boldsymbol{w}^{\mathrm{tmp}}\}$;
        Solve (NLP$_{\mathrm{bin}}$) with $\boldsymbol{w} = \boldsymbol{w}^{\mathrm{tmp}}$ fixed → $\mathcal{C}\left(\boldsymbol{w}^{\mathrm{tmp}}\right)$;
        if $\mathcal{C}\left(\boldsymbol{w}^{\mathrm{tmp}}\right) < \mathcal{C}^{\mathrm{rec}}$ then
            Set $\mathcal{C}^{\mathrm{rec}} = \mathcal{C}\left(\boldsymbol{w}^{\mathrm{tmp}}\right)$;
            Set $\boldsymbol{w}^{\mathrm{rec}} = \boldsymbol{w}^{\mathrm{tmp}}$;
return: $\boldsymbol{w}^{\mathrm{rec}}$ together with $\mathcal{C}^{\mathrm{rec}}$.
```

To avoid a combinatorial explosion, one has to take care of the number of possible variations of singular blocks and MILP solutions $|S^{\mathrm{CIA}}|^{n_{\mathrm{arc}}}$. It is therefore advisable to choose $\tilde{S}^{\mathrm{CIA}}$ in Algorithm 4.2 with a small number of MILPs. Usually, only a few singular arcs result after solving (NLP$_{\mathrm{rel}}$). For more than four singular arcs, Algorithm 4.4 may be modified to be greedy, i.e., to apply the idea of *GreedyTime* on arcs instead of on single intervals.

The singular arc recombination yields an objective value that is at least as good as those previously constructed via the MILPs. However, no framework for quantifying these possible improvements in terms of new rounding errors of the objective currently exists.

4.5.3 Several rounding and NLP steps

This section is based on [212]. In contrast to the previous section where the primary aim was to improve the objective value $\mathcal{C}$, here we deal with a generalization of the CIA decomposition that addresses feasibility issues in (NLP$_{\mathrm{bin}}$) for a fixed binary control. The large number of binary variables in the “all-at-once” rounding in (CIA) and other MILPs from S^{CIA} can lead to an infeasible (NLP$_{\mathrm{bin}}$) related to terminal (4.1d), path (4.1e), or vanishing (4.1f) constraints that cannot be met. Our idea is therefore to apply more than one rounding step to create greater freedom for achieving a feasible solution. To this end, we propose solving a sequence of alternating NLP and (CIA) problems, where the number of fixed binary variables is gradually increased. The following definition formalizes this idea.

Definition 4.23 (Binary subset (CIA)-(NLP) sequence: n_{dec}, $\mathcal{S}_j$, (CIA($\mathcal{S}_j$)), (NLP($\mathcal{S}_j$)))
Let a number of decompositions $2 \le n_{\mathrm{dec}} \le n_\omega$ be given. Let $\mathcal{S}_1 := [n_\omega]$ be the index set of all binary control variables. We denote by $\mathcal{S}_j$, $j = 2, \dots, n_{\mathrm{dec}}$, a chosen sequence of binary control index

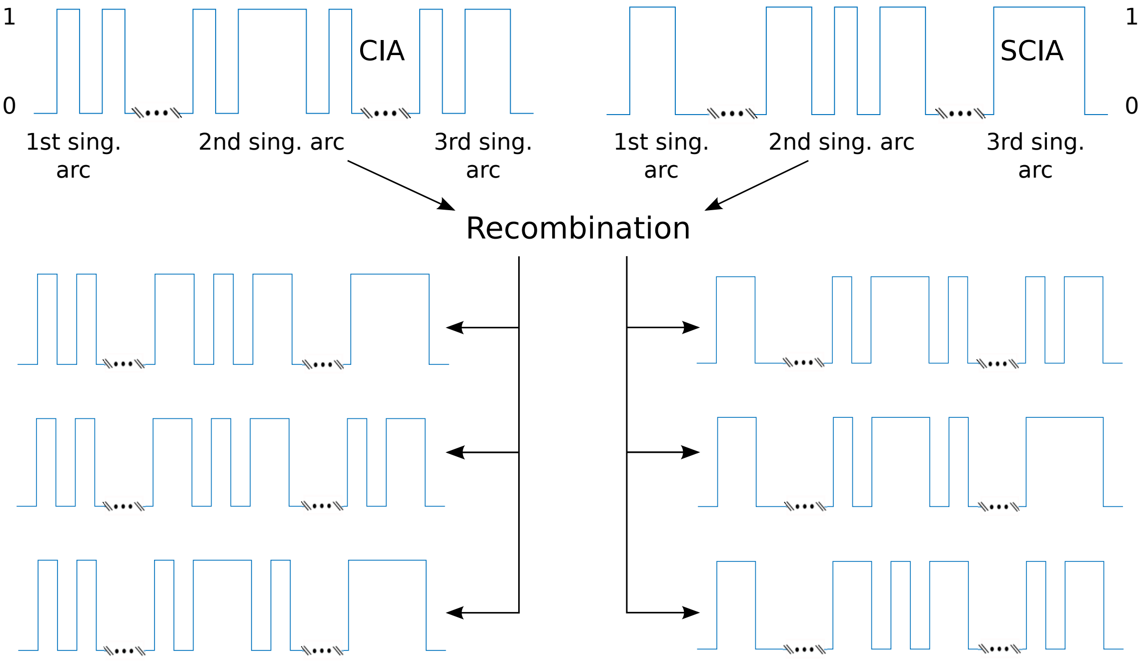


Figure 4.3: Exemplary visualization of the singular arc block recombination heuristic for two MILP control vectors with three singular arcs. We generate every possible variation from the singular arcs and candidate controls and solve (NLP_{rel}) for each constructed variation. The minimal objective value of all the variations represents the heuristic's output.

subsets $\mathscr{S}_{n_{\text{dec}}} \subset \ldots \subset \mathscr{S}_2 \subset \mathscr{S}_1$, where we set $\mathscr{S}_{n_{\text{dec}}} := \emptyset$. We define ***(CIA($\mathscr{S}_j$))****, $j = 1, \ldots, n_{\text{dec}} - 1$, as the problem (CIA), in which the binary variables with control indices out of $\mathscr{S}_j$ are optimized for a given, corresponding relaxed control $\boldsymbol{a}^*$, and all other variables are fixed. Analogously,* ***(NLP($\mathscr{S}_j$))*** *refers to (NLP_{rel}), where we relax all $\boldsymbol{w}_{i,\cdot}$, $i \in \mathscr{S}_j$, and all $\boldsymbol{w}_{i,\cdot}$, $i \in [n_\omega] \setminus \mathscr{S}_j$, are considered to be fixed with values from (CIA($\mathscr{S}_{j-1}$)).*

In Algorithm 4.5, we present a tailored version of the CIA decomposition that consists of solving n_{dec} NLPs and $(n_{\text{dec}} - 1)$ (CIA) problems with a gradually decreasing number of free binary controls, as in Definition 4.23. We illustrate this algorithm in Fig. 4.4.

Algorithm 4.5: Generalized CIA decomposition algorithm with several (CIA) and (NLP_{rel}) steps for error-controlled solution of (BOCP)

```
Input : (MINLP) instance with time grid $\mathscr{G}_N$ as discretization of (BOCP), chosen binary
        control index subsets $\mathscr{S}_j$, $j \in [n_{\text{dec}}]$.
Solve (NLP($\mathscr{S}_1$)) $\rightarrow$ $\boldsymbol{x}^*, \boldsymbol{u}^*, \boldsymbol{a}^*, \mathscr{C}^*$;
for $j = 2, \ldots, n_{\text{dec}}$ do
    Solve (CIA($\mathscr{S}_{j-1}$)) $\rightarrow$ $\boldsymbol{w}^{**}$;
    Solve (NLP($\mathscr{S}_j$)) $\rightarrow$ $\boldsymbol{x}^{**}, \boldsymbol{u}^{**}, \boldsymbol{a}^{**}, \mathscr{C}^{**}$;
return: $(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}, \mathscr{C}) = (\boldsymbol{x}^{**}, \boldsymbol{u}^{**}, \boldsymbol{w}^{**}, \mathscr{C}^{**})$;
```

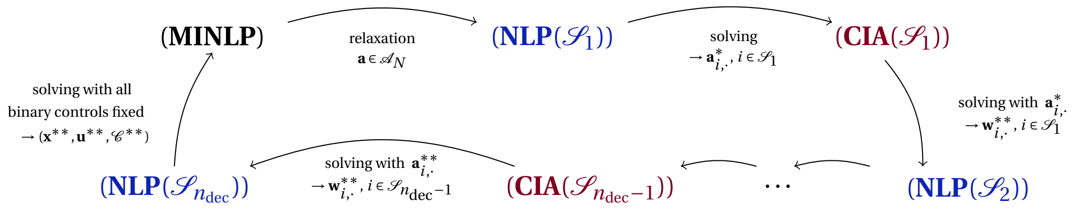

(MINLP)
relaxation
$\mathbf{a} \in \mathscr{A}_N$
(NLP($\mathscr{S}_1$))
solving
$\rightarrow \mathbf{a}^*_{i,\cdot}, i \in \mathscr{S}_1$
(CIA($\mathscr{S}_1$))
solving with all
binary controls fixed
$\rightarrow (\mathbf{x}^{**}, \mathbf{u}^{**}, \mathscr{C}^{**})$
solving with $\mathbf{a}^*_{i,\cdot}$
$\rightarrow \mathbf{w}^{**}_{i,\cdot}, i \in \mathscr{S}_1$
solving with $\mathbf{a}^{**}_{i,\cdot}$
$\rightarrow \mathbf{w}^{**}_{i,\cdot}, i \in \mathscr{S}_{n_{\text{dec}}-1}$
(NLP($\mathscr{S}_{n_{\text{dec}}}$))
(CIA($\mathscr{S}_{n_{\text{dec}}-1}$))
$\cdots$
(NLP($\mathscr{S}_2$))

Figure 4.4: Conceptual illustration of the generalized CIA decomposition in Algorithm 4.5. The original (MIOCP) instance can be transformed via outer convexification and discretization into (MINLP), as highlighted in Figure 4.1. We relax this problem by removing the integrality constraint. The algorithm involves solving an alternating sequence of (CIA) and (NLP) problems. The number of free variables in these problems is represented by $\mathscr{S}_j$ according to the *binary subset CIA-NLP sequence* in Definition 4.23 and is gradually reduced until all variables $\boldsymbol{a}$ are fixed in (NLP($\mathscr{S}_{n_{\text{dec}}}$)). The objective value of the latter problem serves as an approximation to (MINLP).

Chapter 5

Approximation properties of the CIA decomposition

This chapter is largely based on [280]. SAGER, BOCK, and DIEHL proved that the optimal solution of the modified (ROCP)[1] can be approximated with arbitrary precision by a binary control solution [225]. The proof itself is insightful since it avoids using the Krein-Milman theorem, which states the existence of a solution that may switch infinitely often. We recapitulate the proof of approximation of the differential state trajectories based on relaxed and binary controls in Section 5.1. The associated Theorem 2 from [225] has already been presented in other publications; e.g., it was transferred to the discrete setting in [135] and was improved by MANNS and KIRCHES in the sense that the new result holds under milder regularity assumptions [177]. MANNS also investigated convergence in the weak* topology of L^p spaces of the differential state trajectories for a general class of partial differential equation (PDE)-constrained problems [176]. Nevertheless, we repeat the proof of Theorem 2 from [225] as it motivates various extensions and generalizations of the CIA decomposition presented in Chapter 4. To this end, we discuss the implications of approximating the differential state trajectories for the basic combinatorial integral approximation (CIA) decomposition, different MILP variants, generalized CIA decompositions, and the inclusion of constraints in the CIA decomposition in Sections 5.2, 5.3, 5.4, and 5.5, respectively.

5.1 Approximation of differential states under integrality restrictions

The idea here is to analyze the evolution of two trajectories $\boldsymbol{x}$ and $\boldsymbol{y}$ that are based on the same ordinary differential equation (ODE) system but driven by two different controls $\boldsymbol{\alpha} \in \mathscr{A}$ and $\boldsymbol{\omega} \in \Omega$. We want to compare the distance between the two trajectories depending on the distance of the controls. The main theorem relies on a variant of Grönwall's Lemma, which we recapitulate based on [225].

Lemma 5.1 (A variant of GRÖNWALL's Lemma, see [225], Lemma 1)
Let $z_1, z_2 : \mathscr{T} \to \mathbb{R}$ be real-valued integrable functions and let z_2 also belong to $L^\infty(\mathscr{T}, \mathbb{R})$. If for a constant $L \geq 0$ the following holds:

$$z_1(t) \leq z_2(t) + L \int_{t_0}^{t} z_1(\tau)\, d\tau \qquad \text{for a.a. } t \in \mathscr{T},$$

then, we have

$$z_1(t) \leq \|z_2\|_\infty \, \mathrm{e}^{L(t-t_0)} \qquad \text{for a.a. } t \in \mathscr{T}. \tag{5.1}$$

Proof. See [225], proof of Lemma 1. □

We now state the main theorem, which is based on [280], Theorem 4.2.

[1] The exact modified setting is specified in Section 5.1.

Theorem 5.1 (Approximation of differential state trajectories)
Consider $\boldsymbol{\alpha} \in \mathscr{A}$ and $\boldsymbol{\omega} \in \Omega$. We reuse the model functions $\boldsymbol{f}_0, \boldsymbol{f}_i : \mathscr{T} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ from Definition 4.2 for $i \in [n_\omega]$. Furthermore, let $\boldsymbol{u}^* \in \mathscr{U}$ be given, where $\mathscr{U}$ is defined as in Definition 4.2. Let $\boldsymbol{x}(\cdot)$ and $\boldsymbol{y}(\cdot)$ be the unique solutions of the initial value problems (IVPs):

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_0(t, \boldsymbol{x}(t), \boldsymbol{u}^*(t)) + \sum_{i=1}^{n_\omega} \alpha_i(t) \boldsymbol{f}_i(t, \boldsymbol{x}(t), \boldsymbol{u}^*(t)), \qquad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \tag{5.2a}$$

$$\dot{\boldsymbol{y}}(t) = \boldsymbol{f}_0(t, \boldsymbol{y}(t), \boldsymbol{u}^*(t)) + \sum_{i=1}^{n_\omega} \omega_i(t) \boldsymbol{f}_i(t, \boldsymbol{y}(t), \boldsymbol{u}^*(t)), \qquad \boldsymbol{y}(t_0) = \boldsymbol{y}_0, \tag{5.2b}$$

where $\boldsymbol{x}_0, \boldsymbol{y}_0 \in \mathbb{R}^{n_x}$. Assume that there are positive constants $L, C_B \in \mathbb{R}^+$, together with a vector norm $||\cdot||$ such that for a.a. $t \in \mathscr{T}$ holds:

$$\left|\left|\boldsymbol{f}_i(t, \boldsymbol{x}(t), \boldsymbol{u}^*(t)) - \boldsymbol{f}_i(t, \boldsymbol{y}(t), \boldsymbol{u}^*(t))\right|\right| \leq L \left|\left|\boldsymbol{x}(t) - \boldsymbol{y}(t)\right|\right|, \qquad \text{for } i \in [n_\omega]_0, \tag{5.2c}$$

$$\left|\left|\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{f}_i(t, \boldsymbol{x}(t), \boldsymbol{u}^*(t))\right|\right| \leq C_B, \qquad \text{for } i \in [n_\omega]. \tag{5.2d}$$

Furthermore, let $\boldsymbol{f}_i(\cdot, \boldsymbol{x}(\cdot), \boldsymbol{u}^*(\cdot))$, $i \in [n_\omega]$ be essentially bounded by $\hat{C}_B \in \mathbb{R}^+$ on $\mathscr{T}$, and assume that for all $t \in \mathscr{T}$ and $i \in [n_\omega]$ it holds that

$$\left| \int_{t_0}^{t} \alpha_i(\tau) - \omega_i(\tau) \,\mathrm{d}\tau \right| \leq \epsilon, \tag{5.2e}$$

with the constant $\epsilon \in \mathbb{R}^+$. Then, for a.a. $t \in \mathscr{T}$ we also have

$$\left|\left|\boldsymbol{x}(t) - \boldsymbol{y}(t)\right|\right| \leq \left(\left|\left|\boldsymbol{x}_0 - \boldsymbol{y}_0\right|\right| + \epsilon n_\omega \left(\hat{C}_B + C_B(t - t_0)\right)\right) \mathrm{e}^{L(n_\omega+1)(t-t_0)}. \tag{5.2f}$$

Proof. We conclude from $\boldsymbol{\alpha} \in \mathscr{A}, \boldsymbol{\omega} \in \Omega$ that

$$|\alpha_i(t)| \leq 1, \quad |\omega_i(t)| \leq 1, \qquad \text{for } i \in [n_\omega],\ t \in \mathscr{T}. \tag{5.3}$$

For brevity, we write[2] $\boldsymbol{f}_i(\boldsymbol{x}(t))$ instead of $\boldsymbol{f}_i(t, \boldsymbol{x}(t), \boldsymbol{u}^*(t))$, and for $i \in [n_\omega]$, we introduce the abbreviation

$$\mu_i(t) := \int_{t_0}^{t} \alpha_i(\tau) - \omega_i(\tau) \,\mathrm{d}\tau.$$

Note that $|\mu_i(t)| \leq \epsilon$ holds because of (5.2e). For (5.2a, 5.2b) and $t \in \mathscr{T}$, the Lebesgue differentiation theorem yields

$$\boldsymbol{x}(t) = \boldsymbol{x}_0 + \int_{t_0}^{t} \dot{\boldsymbol{x}}(\tau) \,\mathrm{d}\tau = \boldsymbol{x}_0 + \int_{t_0}^{t} \boldsymbol{f}_0(\boldsymbol{x}(\tau)) + \sum_{i=1}^{n_\omega} \alpha_i(\tau) \boldsymbol{f}_i(\boldsymbol{x}(\tau)) \,\mathrm{d}\tau,$$

$$\boldsymbol{y}(t) = \boldsymbol{y}_0 + \int_{t_0}^{t} \dot{\boldsymbol{y}}(\tau) \,\mathrm{d}\tau = \boldsymbol{y}_0 + \int_{t_0}^{t} \boldsymbol{f}_0(\boldsymbol{y}(\tau)) + \sum_{i=1}^{n_\omega} \omega_i(\tau) \boldsymbol{f}_i(\boldsymbol{y}(\tau)) \,\mathrm{d}\tau.$$

Using this, we approximate the normed difference of $\boldsymbol{x}$ and $\boldsymbol{y}$ for a.a. $t \in \mathscr{T}$:

$$\|\boldsymbol{x}(t) - \boldsymbol{y}(t)\| = \left\| \boldsymbol{x}_0 - \boldsymbol{y}_0 + \int_{t_0}^{t} \boldsymbol{f}_0(\boldsymbol{x}(\tau)) + \sum_{i=1}^{n_\omega} \alpha_i(\tau) \boldsymbol{f}_i(\boldsymbol{x}(\tau)) - \boldsymbol{f}_0(\boldsymbol{y}(\tau)) - \sum_{i=1}^{n_\omega} \omega_i(\tau) \boldsymbol{f}_i(\boldsymbol{y}(\tau)) \,\mathrm{d}\tau \right\|$$

[2] We note that $\boldsymbol{f}_i$ stays non-autonomous due to its dependency on $\boldsymbol{u}^*(t)$

$$\begin{aligned}
&= \Bigg\| \boldsymbol{x}_0 - \boldsymbol{y}_0 + \int_{t_0}^{t} \boldsymbol{f}_0(\boldsymbol{x}(\tau)) + \sum_{i=1}^{n_\omega} \alpha_i(\tau)\boldsymbol{f}_i(\boldsymbol{x}(\tau)) - \boldsymbol{f}_0(\boldsymbol{x}(\tau)) - \sum_{i=1}^{n_\omega} \omega_i(\tau)\boldsymbol{f}_i(\boldsymbol{x}(\tau)) \\
&\quad + \boldsymbol{f}_0(\boldsymbol{x}(\tau)) + \sum_{i=1}^{n_\omega} \omega_i(\tau)\boldsymbol{f}_i(\boldsymbol{x}(\tau)) - \boldsymbol{f}_0(\boldsymbol{y}(\tau)) - \sum_{i=1}^{n_\omega} \omega_i(\tau)\boldsymbol{f}_i(\boldsymbol{y}(\tau)) \;\mathrm{d}\tau \Bigg\| \\
&\leq \|\boldsymbol{x}_0 - \boldsymbol{y}_0\| + \left\| \int_{t_0}^{t} \boldsymbol{f}_0(\boldsymbol{x}(\tau)) + \sum_{i=1}^{n_\omega} \alpha_i(\tau)\boldsymbol{f}_i(\boldsymbol{x}(\tau)) - \boldsymbol{f}_0(\boldsymbol{x}(\tau)) - \sum_{i=1}^{n_\omega} \omega_i(\tau)\boldsymbol{f}_i(\boldsymbol{x}(\tau))\,\mathrm{d}\tau \right\| \\
&\quad + \left\| \int_{t_0}^{t} \boldsymbol{f}_0(\boldsymbol{x}(\tau)) + \sum_{i=1}^{n_\omega} \omega_i(\tau)\boldsymbol{f}_i(\boldsymbol{x}(\tau)) - \boldsymbol{f}_0(\boldsymbol{y}(\tau)) - \sum_{i=1}^{n_\omega} \omega_i(\tau)\boldsymbol{f}_i(\boldsymbol{y}(\tau))\,\mathrm{d}\tau \right\| \\
&\leq \|\boldsymbol{x}_0 - \boldsymbol{y}_0\| + \left\| \int_{t_0}^{t} \sum_{i=1}^{n_\omega} (\alpha_i(\tau) - \omega_i(\tau))\boldsymbol{f}_i(\boldsymbol{x}(\tau))\,\mathrm{d}\tau \right\| \\
&\quad + \int_{t_0}^{t} \left[\left\|\boldsymbol{f}_0(\boldsymbol{x}(\tau)) - \boldsymbol{f}_0(\boldsymbol{y}(\tau))\right\| + \sum_{i=1}^{n_\omega} \left\|\boldsymbol{f}_i(\boldsymbol{x}(\tau)) - \boldsymbol{f}_i(\boldsymbol{y}(\tau))\right\| |\omega_i(\tau)| \right] \mathrm{d}\tau \\
&\leq \|\boldsymbol{x}_0 - \boldsymbol{y}_0\| + \left\| \sum_{i=1}^{n_\omega} \boldsymbol{f}_i(\boldsymbol{x}(t))\mu_i(t) - \int_{t_0}^{t} \mu_i(\tau)\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{f}_i(\boldsymbol{x}(\tau))\,\mathrm{d}\tau \right\| + \int_{t_0}^{t} \sum_{i=0}^{n_\omega} L \left\|\boldsymbol{x}(\tau) - \boldsymbol{y}(\tau)\right\| \,\mathrm{d}\tau \\
&\leq \|\boldsymbol{x}_0 - \boldsymbol{y}_0\| + \sum_{i=1}^{n_\omega} |\mu_i(t)| \left\|\boldsymbol{f}_i(\boldsymbol{x}(t))\right\| + \int_{t_0}^{t} |\mu_i(\tau)| \left\| \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{f}_i(\boldsymbol{x}(\tau)) \right\| \,\mathrm{d}\tau \\
&\quad + L(n_\omega + 1) \int_{t_0}^{t} \left\|\boldsymbol{x}(\tau) - \boldsymbol{y}(\tau)\right\| \,\mathrm{d}\tau \\
&\leq \|\boldsymbol{x}_0 - \boldsymbol{y}_0\| + \epsilon n_\omega \left(\hat{C}_B + C_B(t - t_0)\right) + L(n_\omega + 1) \int_{t_0}^{t} \left\|\boldsymbol{x}(\tau) - \boldsymbol{y}(\tau)\right\| \,\mathrm{d}\tau.
\end{aligned}$$

We added a zero in step 2, applied the norm triangle inequality in step 3, used partial integration and the Lipschitz assumption (5.2c) in step 5, and applied the other assumptions in step 7. In order to apply Lemma 5.1, we define the integrable functions z_1, z_2 for $t \in \mathcal{T}$:

$$\begin{aligned}
z_1(t) &:= \|\boldsymbol{x}(t) - \boldsymbol{y}(t)\|, \\
z_2(t) &:= \|\boldsymbol{y}_0 - \boldsymbol{x}_0\| + \epsilon n_\omega \left(\hat{C}_B + C_B(t - t_0)\right),
\end{aligned}$$

which satisfy the assumptions of the lemma. Using the result of the lemma on the last inequality yields the claim that for a.a. $t \in \mathcal{T}$

$$||\boldsymbol{x}(t) - \boldsymbol{y}(t)|| \leq \left(||\boldsymbol{x}_0 - \boldsymbol{y}_0|| + \epsilon n_\omega \left(\hat{C}_B + C_B(t - t_0)\right)\right) \mathrm{e}^{L(n_\omega + 1)(t - t_0)}. \qquad \square$$

The main consequence of Theorem 5.1 is the linear dependency of the state approximation error on the integrated difference between the two control functions $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$, which we refer to as the integral deviation gap in Definition 4.5. Hence, if we minimize the integral deviation gap, we do so for the upper bound of the differential state approximation error. This relationship is elaborated in more detail in the following sections.

Remark 5.1 (Improved regularity assumptions for Theorem 5.1)
Recently, MANNS and KIRCHES demonstrated convergence in the weak* topology of L^p spaces of the differential state trajectories for a general class of PDE-constrained problems [176], which affects Theorem 5.1. In particular, in Remark 2.4 in [176], they established that the regularity assumptions that $\boldsymbol{f}_i,\ i \in [n_\omega]_0$, is essentially bounded and bounded as in (5.2d), can be

weakened to the assumption

$$\boldsymbol{f}_i(\cdot,\boldsymbol{x}(\cdot),\boldsymbol{u}^*(\cdot)) \in L^1(\mathcal{T},\mathbb{R}^{n_x}).$$

5.2 Implications for the basic CIA decomposition

Theorem 5.1 has direct consequences for the solutions constructed by the basic CIA decomposition.

Corollary 5.1 (Approximation properties for Algorithm 4.1)
Consider (BOCP) without the vanishing constraint (4.1f) *and with dropped time-coupled combinatorial constraints, i.e., $\boldsymbol{\omega} \in \Omega$. Let $\boldsymbol{f}_i$, $i \in [n_\omega]_0$, be essentially bounded by $\hat{C}_B$ and Lipschitz continuous, and let their time derivatives be bounded by C_B, as required in Theorem 5.1. For a given discretization with grid $\mathcal{G}_N$, let $\boldsymbol{x}^*, \boldsymbol{u}^*$ denote the optimal solution of (NLP$_{\text{rel}}$) with objective value $\mathcal{C}^*$. Let $\boldsymbol{y}^*$ denote the state trajectory obtained by solving the IVP* (4.1b)–(4.1c) *with fixed $\boldsymbol{u}^*$ and $\boldsymbol{w}^*$ constructed by solving the (CIAmax) problem. Then, we have for a.a. $t \in \mathcal{T}$*

$$\|\boldsymbol{x}^*(t) - \boldsymbol{y}^*(t)\| \leq C(n_\omega)\bar{\Delta} n_\omega \left(\hat{C}_B + C_B(t-t_0)\right) \mathrm{e}^{L(n_\omega+1)(t-t_0)}, \tag{5.4}$$

where $C(n_\omega)$ is a positive constant. Assume further that the functions $\mathcal{C}$ and c_j, $j \in [n_c]$, in (4.1e) *are continuous. Then, for every $\delta > 0$, there is a grid $\mathcal{G}_N$ with grid length $\bar{\Delta}$ such that the constructed solution from the CIA decomposition (Algorithm 4.1) satisfies*

$$|\mathcal{C}(\boldsymbol{x}^*,\boldsymbol{u}^*,\boldsymbol{\omega}^*) - \mathcal{C}(\boldsymbol{x}^{**},\boldsymbol{u}^{**},\boldsymbol{\omega}^{**})| \leq \delta, \tag{5.5}$$

$$|\boldsymbol{x}^*(t_f) - \boldsymbol{x}^{**}(t_f)| \leq \delta, \tag{5.6}$$

$$|c_j(t,\boldsymbol{x}^*(t),\boldsymbol{u}^*(t)) - c_j(t,\boldsymbol{x}^{**}(t),\boldsymbol{u}^{**}(t))| \leq \delta, \quad \text{for } j \in [n_c], \text{ and a.a. } t \in \mathcal{T}. \tag{5.7}$$

Proof. We apply Theorem 5.1. The initial values are identical, i.e., $\boldsymbol{x}_0 = \boldsymbol{y}_0$, and Proposition 4.2 establishes that ϵ can be exchanged with $C(n_\omega)\bar{\Delta}$. Hence, inequality (5.4) holds true. Results (5.5)–(5.7) follow directly from (5.4), the definition of continuity, and from $\mathrm{e}^{L(n_\omega+1)(t-t_0)} \leq \mathrm{e}^{L(n_\omega+1)(t_f-t_0)}$ for all $t \leq t_f$. □

The above corollary relates the (global) optima of relaxed and binary control-related solutions to each other. In fact, Theorem 5.1 establishes an approximation bound for all feasible trajectories of (ROCP) and (BOCP) without combinatorial restrictions on $\boldsymbol{\omega}$. We explain the setting and consequences of the corollary with the following remarks.

Remark 5.2 (Convergence to the optimal solution of (BOCP))
The optimal objective value of (NLP$_{\text{rel}}$) (if solved to global optimality) represents a lower bound on the optimal objective value of (MINLP). Therefore, an arbitrarily close approximation of the optimal solution of (MINLP) can be achieved by refining the grid. This can be done by extending the basic CIA decomposition with an outer loop that checks if $\mathcal{C}^{**}$ is sufficiently close to $\mathcal{C}^*$. If not, the grid $\mathcal{G}_N$ is refined. In this sense, we establish that the constructed solution of the CIA decomposition converges to the optimal solution of (BOCP)(without combinatorial constraints) if $\bar{\Delta} \to 0$.

Remark 5.3 (Relation to the rounding gap consistency property)
The convergence result of the CIA decomposition holds true for any rounding algorithm with the *rounding gap consistency property* from Definition 4.7, and which is applied in step 2

instead of solving (CIAmax). In particular, sum-up rounding (SUR) and next-forced rounding (NFR) produce analogous approximation results.

Remark 5.4 (Convergence under combinatorial constraints)
The results in Corollary 5.1 are established for (BOCP) without combinatorial constraint restriction on $\boldsymbol{\omega}$. In [149], Theorem 3.6, KIRCHES, LENDERS, and MANNS showed that the constraint violation of vanishing constraints can also be made arbitrarily small by refining the grid. The situation is different when time-coupled combinatorial constraints, such as total variation (TV) or minimum dwell time (MDT) constraints, restrict $\boldsymbol{\omega}$, in which case the integral deviation gap does not vanish with the vanishing grid length. In Chapter 7, we investigate tight bounds on the (discretized) integral deviation gap that are independent of $\bar{\Delta}$.

5.3 Implications for different MILP variants

In this section, we investigate how the approximation results for Algorithm 4.1 behave if the rounding problems from Section 4.5.1 are applied in place of (CIAmax). We first recognize that Theorem 5.1 is applicable for any vector norm (as can be guessed from the equivalence of norms).

Corollary 5.2 (Independence from the applied (CIA) vector norm)
Consider the setting of Theorem 5.1; in particular, let the regularity assumptions on $\boldsymbol{f}_i$, $i \in [n_\omega]_0$, hold. Assume that $\boldsymbol{x}$ and $\boldsymbol{y}_{\mathrm{CIA}}$ are the solutions of the IVP (4.1b)–(4.1c)*, where $\boldsymbol{x}$ is based on a given $\boldsymbol{a} \in \mathscr{A}_N$ and $\boldsymbol{y}_{CIA}$ is based on $\boldsymbol{w}^*$, which is the optimal solution of (CIAno), no* $\in \{\max, 1\}$*, with objective value θ^*_{CIA} from Definition 4.17. Then, the state approximation error is bounded for a.a. $t \in \mathscr{T}$ by*

$$\|\boldsymbol{x}(t) - \boldsymbol{y}_{CIA}(t)\| \leq \theta^*_{CIA}(\hat{C}_B + C_B(t-t_0))\mathrm{e}^{L(t-t_0)}. \tag{5.8}$$

Proof. We recognize that θ^*_{CIA} expresses the norm of the accumulated control deviation, while ϵ from Theorem 5.1 bounds the component-wise accumulated control deviation. Thus, for applying the proof of Theorem 5.1, we rewrite the ODE with the linear mapping $\boldsymbol{F} : \mathbb{R}^{n_\omega+1} \to \mathbb{R}^{n_x}$, where $\boldsymbol{f}_i(t, \boldsymbol{x}(t), \boldsymbol{u}^*(t))$ is the ith column vector of $\boldsymbol{F}(t, \boldsymbol{x}(t), \boldsymbol{u}^*(t))$, multiplied by $(1, \boldsymbol{\alpha}(t))^\top$, respectively $(1, \boldsymbol{\omega}(t))^\top$. We eliminate with this reformulation the summation over $[n_\omega]$ in the proof and, hence, the factor n_ω in the upper bound is eliminated. In addition, μ_i is replaced by θ^*_{CIA} from the 5th step on. In this way, the proof of Theorem 5.1 can be analogously applied with the linear mapping $\boldsymbol{F}$ and the result follows. □

With the above corollary at hand, we can argue that the results from Corollary 5.1 are applicable not only for (CIAmax) but also for (CIA1).

5.3.1 Control approximation scaled with model function

The proof of Theorem 5.1 motivates the (SCIA) problems.

Corollary 5.3 (State approximation bounds via (SCIAmax),(SCIA1))
Consider the setting of Theorem 5.1, and let $\|\cdot\|_{no}$ refer to the maximum or 1-norm, i.e., no $\in \{\max, 1\}$*. Assume that $\boldsymbol{x}$ and $\boldsymbol{y}_{\mathrm{CIA}}$ are the solutions of the IVP* (4.1b)–(4.1c)*, where $\boldsymbol{x}$ is based on a*

given $\boldsymbol{a} \in \mathscr{A}_N$*, and* $\boldsymbol{y}_{CIA}$ *is driven by* $\boldsymbol{w}^*$*, which is the optimal solution of (SCIAno). Then for a.a.* $t \in \mathscr{T}$*, the state approximation error is bounded by*

$$\|\boldsymbol{x}(t) - \boldsymbol{y}_{SCIA}(t)\| \le \theta^*_{SCIA} \mathrm{e}^{L(t-t_0)} \le \theta^*_{CIA}(\hat{C}_B + C_B(t-t_0))\mathrm{e}^{L(t-t_0)}, \tag{5.9}$$

where θ^*_{SCIA} *is the optimal objective value of (SCIAno),* $no \in \{\max, 1\}$.

Proof. From the second and last inequalities in the proof of Theorem 5.1 and Corollary 5.2, it follows that for $t \in \mathscr{T}$ and any $\boldsymbol{\omega} \in \Omega$

$$\left\| \int_{t_0}^{t} \sum_{i=1}^{n_\omega} (\alpha_i(\tau) - \omega_i(\tau)) \boldsymbol{f}_i(\boldsymbol{x}(\tau)) \, \mathrm{d}\tau \right\| \le \left\| \int_{t_0}^{t} \boldsymbol{\alpha}(\tau) - \boldsymbol{\omega}(\tau) \, \mathrm{d}\tau \right\| (\hat{C}_B + C_B(t-t_0)).$$

Let $\boldsymbol{\omega}^{\text{CIA}}$ denote the control based on the optimal solution $\boldsymbol{w}^*$ of (CIAno), $no \in \{\max, 1\}$. Taking the minimum in the above inequality yields

$$\theta^*_{\text{SCIA}} \le \left\| \int_{t_0}^{t} \sum_{i=1}^{n_\omega} (\alpha_i(\tau) - \omega_i^{\text{CIA}}(\tau)) \boldsymbol{f}_i(\boldsymbol{x}(\tau)) \, \mathrm{d}\tau \right\| \le \theta^*_{\text{CIA}}(\hat{C}_B + C_B(t-t_0)). \qquad \square$$

As a consequence of Corollary 5.3, the approximation and convergence results of the CIA decomposition still hold if (SCIAno), $no \in \{\max, 1\}$, is used to construct the binary control. The approximation bound based on (SCIAno) is tighter than the existing (CIAno)-related bound. Thus, consulting these alternative binary controls for an approximation study is an obvious choice. Ideally, a binary control constructed in this way will result in an improved state approximation and objective value for (MINLP). Nevertheless, we oppose this hope next.

Remark 5.5 (Construction by (SCIA) does not guarantee superior quality)
Using (SCIAno), $no \in \{\max, 1\}$, to construct the binary control in the CIA decomposition does not necessarily result in a state approximation or objective value that is superior to that obtained using (CIAno). Using the notation from Corollaries 5.2 and 5.3, we may have that

$$\|\boldsymbol{x}(t) - \boldsymbol{y}_{\text{CIA}}(t)\| < \|\boldsymbol{x}(t) - \boldsymbol{y}_{\text{SCIA}}(t)\| < \theta^*_{\text{SCIA}}, \quad \text{for some } t \in \mathscr{T}.$$

Even if the above inequality does not hold, the computed trajectories may lead to a superior objective value for the solution based on (CIAno) compared with that based on (SCIAno) because of a non-convex objective.

5.3.2 Control approximation scaled with dual variables

We recapitulate the *cost-to-go* function and the adjoint trajectory $\boldsymbol{\lambda}$, which is its differential state derivative from Section 2.2.2. The idea of (λCIA1) does not stem from approximating the differential state values but from an approximation of the cost-to-go function values. The evaluated dual variable values of the constraints (4.1b) serve as an adjoint trajectory approximation.

Corollary 5.4 (Approximation bounds via ($\boldsymbol{\lambda}$CIA1))
Consider the setting of Theorem 5.1. In particular, let the regularity assumptions (5.2d), (5.2c), *and essential boundedness of* $\boldsymbol{f}$ *be true. Assume that* $\boldsymbol{x}$ *and* $\boldsymbol{y}_{\lambda\text{CIA}}$ *are the solutions of the IVP* (4.1b)–(4.1c)*, where* $\boldsymbol{x}$ *is based on a given* $\boldsymbol{a} \in \mathscr{A}_N$*, and* $\boldsymbol{y}_{\lambda CIA}$ *is based on* $\boldsymbol{w}^*$*, which is the optimal*

solution of (λCIA1). Let J be the cost-to-go function as defined in Definition 2.6 for (BOCP) and $\boldsymbol{\lambda}(t)$ be the adjoint vector at $t \in \mathscr{T}$. For a.a. $t \in \mathscr{T}$, it follows that

$$|J(\boldsymbol{x}(t),t) - J(\boldsymbol{y}_{\lambda CIA}(t),t)| \leq \theta^*_{\lambda CIA} \mathrm{e}^{L(n_\omega-1)(t-t_0)} + o\left(\|\boldsymbol{x}(t) - \boldsymbol{y}_{\lambda CIA}(t)\|^2\right),$$

where o refers to Landau's little-o notation.

Proof. We consider the difference of the cost-to-go functions by approximation with a partial first-order Taylor expansion around $J(\boldsymbol{x}(t),t)$. The approximation is performed with respect to the trajectories $\boldsymbol{x}$, $\boldsymbol{y}_{\lambda\text{CIA}}$. Hence, we apply Taylor's theorem for $t \in \mathscr{T}$:

$$J(\boldsymbol{x}(t),t) - J(\boldsymbol{y}_{\lambda\text{CIA}}(t),t) = \frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{x}}(\boldsymbol{x}(t),t)\left(\boldsymbol{x}(t) - \boldsymbol{y}_{\lambda\text{CIA}}(t)\right) + o\left(\left\|\boldsymbol{x}(t) - \boldsymbol{y}_{\lambda\text{CIA}}(t)\right\|^2\right). \tag{5.10}$$

As pointed out in Remark 2.1, the dual variables of the ODE constraint (4.1b) are equal to $\frac{\mathrm{d}J}{\mathrm{d}\boldsymbol{x}}(\boldsymbol{x}(t),t)$. We use the notation $\|\boldsymbol{x}(t)\|_{\lambda(t)} := \left|\sum_{k\in[n_x]} \lambda_k(t)x_k(t)\right|$ for $t \in \mathscr{T}$, which defines a semi-norm. Then for a.a. $t \in \mathscr{T}$, we transfer the proof of Theorem 5.1 to this notation and to (5.10):

$$\begin{aligned}
|J(\boldsymbol{x}(t),t) - J(\boldsymbol{y}_{\lambda\text{CIA}}(t),t)| &\leq \left\|\boldsymbol{x}(t) - \boldsymbol{y}_{\lambda\text{CIA}}(t)\right\|_{\lambda(t)} + o\left(\left\|\boldsymbol{x}(t) - \boldsymbol{y}_{\lambda\text{CIA}}(t)\right\|^2\right) \\
&\leq \ldots \text{ (as in proof of Theorem 5.1 until second inequality)} \\
&\leq \left\|\boldsymbol{x}_0 - \boldsymbol{y}_0\right\|_{\lambda(t)} + \left\|\int_{t_0}^{t} \sum_{i=1}^{n_\omega} (\alpha_i(\tau) - \omega_i^*(\tau))\boldsymbol{f}_i(\boldsymbol{x}(\tau))\,\mathrm{d}\tau\right\|_{\lambda(t)} \\
&\quad + L(n_\omega+1)\int_{t_0}^{t} \left\|\boldsymbol{x}(\tau) - \boldsymbol{y}_{\lambda\text{CIA}}(\tau)\right\|_{\lambda(t)}\,\mathrm{d}\tau + o\left(\left\|\boldsymbol{x}(t) - \boldsymbol{y}_{\lambda\text{CIA}}(t)\right\|^2\right).
\end{aligned}$$

The second summand of the last inequality represents the objective $\theta^*_{\lambda\text{CIA}}$ that is to be minimized in (λCIA1). Finally, we use $\boldsymbol{x}_0 = \boldsymbol{y}_0$ and apply the Grönwall Lemma 5.1 with the integrable functions

$$z_1(t) = \left\|\boldsymbol{x} - \boldsymbol{y}_{\lambda\text{CIA}}\right\|_{\lambda(t)}, \quad z_2(t) = \left\|\int_{t_0}^{t} \sum_{i=1}^{n_\omega} (\alpha_i(\tau) - \omega_i^*(\tau))\boldsymbol{f}_i(\boldsymbol{x}(\tau))\,\mathrm{d}\tau\right\|_{\lambda(t)},$$

proving the claim. □

We note that due to the first-order Taylor approximation, (λCIA1) requires that the relaxed trajectory $\boldsymbol{x}$ can be well approximated by a trajectory $\boldsymbol{y}$ that is based on binary controls. If there is no such trajectory in a close neighborhood of $\boldsymbol{x}$, (λCIA1) may have an unintuitive binary control as its optimal solution.

5.3.3 Backward accumulating constraints

For an (MIOCP) instance with fixed terminal state values $\boldsymbol{x}_f \in \mathbb{R}^{n_x}$ and a Lagrangian objective type, we can also apply the altered setting to Theorem 5.1. The following corollary addresses with this issue.

Corollary 5.5 (Approximation bounds via backward constraints)
Consider the setting of Theorem 5.1. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be the state trajectory solutions of the terminal value problems (4.1b) *with $\boldsymbol{x}(t_f) = \boldsymbol{x}_f$ and $\boldsymbol{y}(t_f) = \boldsymbol{y}_f$ for $\boldsymbol{x}_f, \boldsymbol{y}_f \in \mathbb{R}^{n_x}$. Assume that for all $t \in$*

$\mathcal{T}$, $\epsilon_b \in \mathbb{R}^+$, and for all $i \in [n_\omega]$ it holds that

$$\left\| \int_t^{t_f} \alpha_i(\tau) - \omega_i(\tau) \,\mathrm{d}\tau \right\| \leq \epsilon_b. \tag{5.11}$$

Then, for a.a. $t \in \mathcal{T}$ it also holds that

$$\left|\left|\boldsymbol{x}(t) - \boldsymbol{y}(t)\right|\right| \leq \left(\left|\left|\boldsymbol{x}_f - \boldsymbol{y}_f\right|\right| + \epsilon_b n_\omega \left(\hat{C}_B + C_B(t_f - t)\right)\right) \mathrm{e}^{L(n_\omega+1)(t_f-t)}. \tag{5.12}$$

Proof. The proof of Theorem 5.1 can be applied to the altered setting, where we integrate over $[t, t_\mathrm{f}]$ instead of integrating over $[t_0, t]$. □

We highlight that with the assumption that $\|\boldsymbol{x}(t_f) - \boldsymbol{y}(t_f)\|$ is small, the backward (CIA) rounding problem approach from Section 4.5.1 is not only applicable for terminal constraint problems but is also appropriate for an (MIOCP) instance with a given initial value $\boldsymbol{x}_0$ and variable final state values.

5.4 Implications for generalized CIA decompositions

As soon as (CIAmax) is included in the set $\tilde{S}^{\mathrm{CIA}}$ from Definition 4.21, the approximation results from Corollary 5.1 and Remark 5.2 for the basic CIA decomposition can be transferred to Algorithm 4.2 since the recombination algorithms are intended to improve the previous solution.

Corollary 5.6 (Approximation properties of the generalized CIA decomposition 4.2)
Let (CIAmax) $\in \tilde{S}^{\mathrm{CIA}}$ be chosen in Algorithm 4.2. If we further assume that the regularity assumptions (5.2d), (5.2c), *and essential boundedness of $\boldsymbol{f}$ hold, then the approximation quality results for differential states* (5.4), *constraint violations* (5.6)–(5.7), *and the objective value* (5.5) *from Corollary 5.1 are also true for the solution constructed by Algorithm 4.2.*

Proof. In Section 4.5.2, we assume that the recombination algorithms in $\tilde{S}^{\mathrm{REC}}$ account for the differential state approximation $\mathscr{C}^{\mathrm{track}}$ in the objective and that in this way, they solely construct solutions of (BOCP) that do not deteriorate the approximation quality with respect to the optimal relaxed trajectory $\boldsymbol{x}^*$. The claim then follows directly from Corollary 5.1. □

The example algorithms in Section 4.5.2, i.e., *GreedyTime* and *Singular arc* recombination, are designed to construct binary controls associated with (BOCP) objective values that are at least as good as those of the chosen mixed-integer linear programs (MILPs) from $\tilde{S}^{\mathrm{CIA}}$. We remark that no framework currently exists to quantify these possible improvements theoretically.

Remark 5.6 (Other bounds for the generalized CIA decomposition 4.2)
We assume (CIAmax) $\in \tilde{S}^{\mathrm{CIA}}$ in Corollary 5.6. The approximation quality established by Corollaries 5.2, 5.3, 5.4, and 5.5 for other CIA rounding problems also hold for the generalized CIA decomposition from Algorithm 4.2 if the corresponding MILPs are included in $\tilde{S}^{\mathrm{CIA}}$.

The final result in this section addresses the properties of the generalized CIA decomposition in Algorithm 4.5, where multiple steps of nonlinear program (NLP) and (CIA) problems are applied, and the number of fixed binary variables is gradually increased.

Proposition 5.1 (Approximation properties of the generalized CIA decomposition 4.5) *Consider (BOCP) without the vanishing constraint* (4.1f) *and with dropped time-coupled combinatorial constraints, i.e.,* $\boldsymbol{\omega} \in \Omega$. *Let the regularity assumptions* (5.2d), (5.2c), *and the essential boundedness of* $\boldsymbol{f}$ *be true. Furthermore, let a sequence of binary control index subsets* $\mathscr{S}_j$, $j = 1, \ldots, n_{\text{dec}}$, *be given as introduced in Definition 4.23. Assume that* $\boldsymbol{x}_{\mathscr{S}_1}$ *and* $\boldsymbol{x}_{\mathscr{S}_{n_{\text{dec}}}}$ *are the optimal differential state solutions of (NLP(*$\mathscr{S}_1$*)) and (NLP(*$\mathscr{S}_{n_{\text{dec}}}$*)), respectively, in Algorithm 4.5. Finally, assume that the normed difference of the trajectory* $\boldsymbol{y}_{\mathscr{S}_j}$, $j = 2, \ldots, n_{\text{dec}}$, *driven by the optimal solution* $\boldsymbol{w}^*$ *of (CIA(*$\mathscr{S}_j$*)), and* $\boldsymbol{x}_{\mathscr{S}_{j-1}}$ *does not increase if* $\boldsymbol{y}_{\mathscr{S}_j}$ *is replaced by* $\boldsymbol{x}_{\mathscr{S}_j}$. *Then for a.a.* $t \in \mathscr{T}$, *we have that*

$$\|\boldsymbol{x}_{\mathscr{S}_1}(t) - \boldsymbol{x}_{\mathscr{S}_{n_{\text{dec}}}}(t)\| \leq C(n_\omega)\bar{\Delta} n_\omega \left(\hat{C}_B + C_B(t - t_0)\right) \mathrm{e}^{L(n_\omega+1)(t-t_0)}, \tag{5.13}$$

where $C(n_\omega)$ *is a positive constant.*

Proof. Let θ_j^* denote the optimal objective value of (CIA($\mathscr{S}_j$)), $j = 2, \ldots, n_{\text{dec}}$. For a.a. $t \in \mathscr{T}$, we obtain

$$\begin{aligned}
\|\boldsymbol{x}_{\mathscr{S}_1}(t) - \boldsymbol{x}_{\mathscr{S}_{n_{\text{dec}}}}(t)\| &= \left\| \boldsymbol{x}_{\mathscr{S}_1}(t) - \boldsymbol{x}_{\mathscr{S}_{n_{\text{dec}}}}(t) + \sum_{j=2}^{n_{\text{dec}}-1} (\boldsymbol{x}_{\mathscr{S}_j}(t) - \boldsymbol{x}_{\mathscr{S}_j}(t)) \right\| \\
&\leq \sum_{j=2}^{n_{\text{dec}}} \left\| \boldsymbol{x}_{\mathscr{S}_{j-1}}(t) - \boldsymbol{x}_{\mathscr{S}_j}(t) \right\| \\
&\overset{(1)}{\leq} \sum_{j=2}^{n_{\text{dec}}} \theta_j^* n_\omega \left(\hat{C}_B + C_B(t - t_0)\right) \mathrm{e}^{L(n_\omega+1)(t-t_0)} \\
&\overset{(2)}{\leq} C(n_\omega)\bar{\Delta} n_\omega \left(\hat{C}_B + C_B(t - t_0)\right) \mathrm{e}^{L(n_\omega+1)(t-t_0)}.
\end{aligned}$$

In (1), we applied Corollary 5.1 and the assumption that using the NLP state solution $\boldsymbol{x}_{\mathscr{S}_j}$, instead of the (CIA($\mathscr{S}_j$))-based state solution $\boldsymbol{y}_{\mathscr{S}_j}$, does not worsen the approximation quality. In (2), we exploit the fact that every variable fixing in (CIA($\mathscr{S}_j$)) corresponds to a maximum objective value of $C(n_\omega)\bar{\Delta}$ such that the rounding error may accumulate[3] to $n_{\text{dec}} C(n_\omega)\bar{\Delta}$, which can be set as the new constant $C(n_\omega)$. □

Even though Algorithm 4.5 was not designed for grid refinement, from the dependency on $\bar{\Delta}$ we conclude that the convergence result from Remark 5.2 is still valid in this setting.

5.5 Implications for the inclusion of path constraints into the CIA decomposition

The following proposition establishes that under certain assumptions, the path constraint Taylor approximation from Definition 4.14 included in (CIA) limits the path constraint violation of the constructed differential state trajectory.

Proposition 5.2 (Approximation bound for path constraint inclusion from Definition 4.14) *Consider problem (BOCP). Let the path constraint function* $\boldsymbol{c}$ *in* (4.1e) *depend only on the differential state* $\boldsymbol{x}(t)$. *Let* $\boldsymbol{x}^*$ *be the optimal differential state solution of (NLP*$_{\text{rel}}$*). Let* $\boldsymbol{y}^*$ *denote the state trajectory obtained by solving the IVP* (4.1b)–(4.1c) *with fixed* $\boldsymbol{u}^*$ *and* $\boldsymbol{w}^*$ *constructed from*

[3] We quantify the constant $C(n_\omega)$ in Chapter 7 and find that $n_{\text{dec}} C(n_\omega)\bar{\Delta}$ is an overestimation.

the basic CIA decomposition (Algorithm 4.1) after step 2 and with the first-order approximation constraint (4.15) *from Definition 4.14 included in (CIA). We assume that for a.a. $t \in \mathscr{T}$ and for $\boldsymbol{u}^*$ obtained from solving (NLP$_{\text{rel}}$) it holds that*

$$|\boldsymbol{f}_i(t, \boldsymbol{y}^*(t), \boldsymbol{u}^*(t)) - \boldsymbol{f}_i(t, \boldsymbol{x}^*(t), \boldsymbol{u}^*(t))| \leq \delta\, \mathbf{1}_{n_x}, \quad \textit{for } i \in [n_\omega]_0, \tag{5.14}$$

where $\delta \geq 0$, and $|\cdot|$ refers here to the absolute value of each vector component. Moreover, let $\boldsymbol{R}(\boldsymbol{x}^*, \boldsymbol{y}^*)$ denote the remainder term of a first-order Taylor approximation around $\boldsymbol{c}(\boldsymbol{y}^*(t))$:

$$\boldsymbol{R}(\boldsymbol{x}^*(t), \boldsymbol{y}^*(t)) := \boldsymbol{c}(\boldsymbol{y}^*(t)) - \boldsymbol{c}(\boldsymbol{x}^*(t)) - \boldsymbol{c}_{\boldsymbol{x}}(\boldsymbol{x}^*(t))(\boldsymbol{y}^*(t) - \boldsymbol{x}^*(t)), \tag{5.15}$$

where $\boldsymbol{c}_{\boldsymbol{x}} := \frac{dc}{dx}$. Then for a.a. $t \in \mathscr{T}$, we have that

$$\boldsymbol{c}(\boldsymbol{y}^*(t)) \geq -\boldsymbol{\epsilon}(\delta, \boldsymbol{R}), \tag{5.16}$$

where $\boldsymbol{\epsilon}(\delta, \boldsymbol{R}) := |\boldsymbol{c}_{\boldsymbol{x}}(\boldsymbol{x}^*(t))(1+n_\omega)(t-t_0)\delta\, \mathbf{1}_{n_x} + \boldsymbol{R}[\boldsymbol{x}^*(t), \boldsymbol{y}^*(t)]|$ (component-wise absolute value).

Proof. We abbreviate $\boldsymbol{f}_i(\boldsymbol{y}^*(\tau)) := \boldsymbol{f}_i(\tau, \boldsymbol{y}^*(\tau), \boldsymbol{u}^*(\tau))$ and for a.a. $t \in \mathscr{T}$ calculate

$$\begin{aligned}
\boldsymbol{c}(\boldsymbol{y}^*(t)) &= \boldsymbol{c}(\boldsymbol{x}^*(t)) + \boldsymbol{c}_{\boldsymbol{x}}(\boldsymbol{x}^*(t))[\boldsymbol{y}^*(t) - \boldsymbol{x}^*(t)] + \boldsymbol{R}(\boldsymbol{x}^*(t), \boldsymbol{y}^*(t)) \\
&\geq \boldsymbol{c}_{\boldsymbol{x}}(\boldsymbol{x}^*(t)) \left[\int_{t_0}^{t} \left(\boldsymbol{f}_0(\boldsymbol{y}^*(\tau)) + \sum_{i \in [n_\omega]} \omega_i^*(\tau) \boldsymbol{f}_i(\boldsymbol{y}^*(\tau)) - \boldsymbol{f}_0(\boldsymbol{x}^*(\tau)) - \sum_{i \in [n_\omega]} \alpha_i^*(\tau) \boldsymbol{f}_i(\boldsymbol{x}^*(\tau)) \right) \mathrm{d}\tau \right] \\
&\quad + \boldsymbol{R}(\boldsymbol{x}^*(t), \boldsymbol{y}^*(t)) \\
&= \boldsymbol{c}_{\boldsymbol{x}}(\boldsymbol{x}^*(t)) \left[\int_{t_0}^{t} \left(\boldsymbol{f}_0(\boldsymbol{y}^*(\tau)) + \sum_{i \in [n_\omega]} \omega_i^*(\tau) \boldsymbol{f}_i(\boldsymbol{y}^*(\tau)) - \boldsymbol{f}_0(\boldsymbol{x}^*(\tau)) - \sum_{i \in [n_\omega]} \omega_i^*(\tau) \boldsymbol{f}_i(\boldsymbol{x}^*(\tau)) \right) \mathrm{d}\tau \right] \\
&\quad + \boldsymbol{c}_{\boldsymbol{x}}(\boldsymbol{x}^*(t)) \left[\int_{t_0}^{t} \left(\sum_{i \in [n_\omega]} (\omega_i^*(\tau) - \alpha_i^*(\tau)) \boldsymbol{f}_i(\boldsymbol{x}^*(\tau)) \right) \mathrm{d}\tau \right] + \boldsymbol{R}(\boldsymbol{x}^*(t), \boldsymbol{y}^*(t)) \\
&\geq \boldsymbol{c}_{\boldsymbol{x}}(\boldsymbol{x}^*(t)) \left[\int_{t_0}^{t} \left(\boldsymbol{f}_0(\boldsymbol{y}^*(\tau)) + \sum_{i \in [n_\omega]} \omega_i^*(\tau) \boldsymbol{f}_i(\boldsymbol{y}^*(\tau)) - \boldsymbol{f}_0(\boldsymbol{x}^*(\tau)) - \sum_{i \in [n_\omega]} \omega_i^*(\tau) \boldsymbol{f}_i(\boldsymbol{x}^*(\tau)) \right) \mathrm{d}\tau \right] \\
&\quad + \boldsymbol{R}(\boldsymbol{x}^*(t), \boldsymbol{y}^*(t)) \\
&\geq -\left| \boldsymbol{c}_{\boldsymbol{x}}(\boldsymbol{x}^*(t))(1 + n_\omega)(t - t_0)\delta\, \mathbf{1}_{n_{\mathrm{x}}} + \boldsymbol{R}[\boldsymbol{x}^*(t), \boldsymbol{y}^*(t)] \right| = -\boldsymbol{\epsilon}(\delta, \boldsymbol{R}).
\end{aligned}$$

We use $\boldsymbol{c}(\boldsymbol{x}^*(t)) \geq \mathbf{0}_{n_c}$ in the first inequality and apply (5.15). The second inequality exploits that the constraint (4.15) is included in (CIA) thus, the second summand can be dropped. Finally, we integrate and use (5.14) in the last inequality. □

We remark that $\boldsymbol{f}_i(\boldsymbol{y}^*(\tau)) \approx \boldsymbol{f}_i(\boldsymbol{x}^*(\tau))$ should hold, meaning that the constraint violation in (5.16) should be small, which motivates the idea of including (4.15) in (CIA). Moreover, solving (NLP$_{\text{bin}}$) after (CIA) is likely to be an improvement with respect to path constraint feasibility. On the other hand, the first-order Taylor approximation can be weak, and we stress that the approach is only a heuristic without guaranteed constraint feasibility.

Chapter 6

Algorithms for solving (CIA) problems

This chapter presents algorithms for solving (CIA) with and without combinatorial constraints. As an MILP, special solver programs such as `Gurobi` [109] can clearly be applied to (CIA). However, custom-made algorithms can solve (CIA) more efficiently, as shown in [224] using branch-and-bound (BnB). When time-coupled combinatorial constraints, such as TV or MDT constraints, have to be considered, tailor-made algorithms are especially useful. SUR and NFR have already been introduced in Definitions 4.8 and 4.9, respectively, as possibilities for solving (CIA) heuristically, i.e., not necessarily to optimality. This chapter discusses how these algorithms and other heuristics can be extended to combinatorial constraints.

Section 6.1 describes a heuristic for reducing the problem size based on singular arcs, which is especially useful for large problems. In Section 6.2, we establish a connection to scheduling theory and show that (CIA) is up to ϵ-optimality solvable in polynomial time if the grid is equidistant. We restate JUNG's [135] BnB algorithm, which is based on time-dependent variables and branching, in Section 6.3. In contrast, in Section 6.4, we consider the lifted problem, i.e., an extended formulation of (CIA), which works with switching-dependent variables. We formulate the associated MILP and a corresponding BnB algorithm. Different SUR variants are introduced in Section 6.5. We also introduce an MDT extension, which we also do for the NFR scheme in Section 6.6. The maximum dwell rounding (MDR) algorithm is specifically designed to (heuristically) solve the (CIA) quickly under TV constraints. It is also applicable to MDT constraints and is defined along with its properties in Section 6.7. In Section 6.8, we discuss alternative approaches before concluding the chapter with a summary in which we recommend which algorithms are most advantageous depending on the (CIA) problem.

Section 6.5 and Section 6.6 are mainly based on [282], while the adaptive maximum dwell rounding (AMDR) scheme results and the algorithm itself, presented in Section 6.7, were introduced in [222]. Moreover, Section 6.2 and Section 6.3 use ideas from [49] and [114], respectively.

6.1 A problem size reduction heuristic based on bang-bang arcs

In Section 4.5.2, we used the idea of bang-bang arcs of the relaxed solution $\boldsymbol{a}^*$, and we argued that $\boldsymbol{w}$ should attain these (almost) binary values as an optimal solution of (CIA). We recycle this idea for a heuristic that reduces the problem size. To this end, we come back to Definition 4.22, where the singular arc interval sets $\mathscr{I}_l^{\text{sing}}$ and the number of singular arcs n_{sing} are introduced for a given relaxed control $\boldsymbol{a}^*$ and a small rounding tolerance $\epsilon > 0$. Based on the identified singular arcs, our idea for problem size reduction is to fix the variables $\boldsymbol{w}_{\cdot,j}$ on complementary intervals, i.e., on the bang-bang arcs:

$$w_{i,j} = \lfloor a_{i,j}^* + \epsilon \rfloor, \qquad \text{for } i \in [n_\omega],\ j \in [N] \setminus \{\mathscr{I}_l^{\text{sing}}\}_{l \in [n_{\text{sing}}]}. \tag{6.1}$$

In the above equation, we fix the variables $w_{i,j}$ to one if the associated $a^*_{i,j}$ is in an ϵ- neighborhood of one and analogously fix the complementary case. Variable fixing in this way can greatly reduce the number of degrees of freedom in (CIA); however, it may also lead to infeasibilities with respect to combinatorial constraints, such as the TV constraint since the bang-bang arcs imply a certain minimum number of switches.

For the run time performance of algorithms such as BnB, it is beneficial to exclude the fixed variables on bang-bang arcs directly. Such a variable reduction algorithm was presented in [49], and we restate it in a similar form in Algorithm 6.1. Unlike the fixing in (6.1), we exclude only those variables on bang-bang arcs whose length exceeds a chosen MDT $C_D \geq 0$. Furthermore, we do not fix the variables at the edges of bang-bang arcs to allow more flexibility regarding the switching interval and, thus, the combinatorial constraints. For this purpose, we introduce the parameter $n_{\text{ivl}} \in \mathbb{N}$, which specifies the number of intervals at the beginning and end of a bang-bang arc, at which points the variables $w_{i,j}$ are left unfixed.

Algorithm 6.1: Problem size reduction heuristic based on bang-bang arcs

```
   Input  : Grid $\mathcal{G}_N$, relaxed control $\boldsymbol{a}^* \in \mathcal{A}_N$, singular arc interval sets $\mathcal{J}_l^{\text{sing}}$, dwell time
            parameter $C_1 \geq 0$, interval unfixing parameter $n_{\text{ivl}} \in \mathbb{N}$.
   Output: Reduced grid $\mathcal{G}_{\tilde{N}}$, reduced relaxed control $\tilde{\boldsymbol{a}}^*$.
1  Set $k_0^{\text{end}} = 0$, $k_{n_{\text{sing}}+1}^{\text{start}} = N + 1$;
2  Initialize $\mathcal{G}_{\tilde{N}} = \mathcal{G}_N$, $\tilde{\boldsymbol{a}}^* = \boldsymbol{a}^*$, $\tilde{N} = N$, $n_{\text{shift}} = 0$;
3  for $l = 1, \dots, n_{\text{sing}} + 1$ do
4      if $\sum_{j=k_{l-1}^{\text{end}}+1-n_{\text{shift}}}^{k_l^{\text{start}}-1-n_{\text{shift}}} \Delta_j \geq C_1$ then
5          if $l = 1$ then
6              Set $n_{\text{temp}} \leftarrow k_l^{\text{start}} - k_{l-1}^{\text{end}} - n_{\text{ivl}} - 1$;
7              Set $t_j \leftarrow t_{j+n_{\text{temp}}}$,   for $j = 0, \dots, N - n_{\text{temp}}$;
8              Set $\tilde{\boldsymbol{a}}^*_{\cdot,j} \leftarrow \tilde{\boldsymbol{a}}^*_{\cdot,j+n_{\text{temp}}}$,   for $j = 1, \dots, N - n_{\text{temp}}$;
9          else if  $l < n_{\text{sing}} + 1$ then
10             Set $n_{\text{temp}} \leftarrow k_l^{\text{start}} - k_{l-1}^{\text{end}} - 2n_{\text{ivl}} - 1$;
11             for $j = 1, \dots, \tilde{N} - (k_{l-1}^{\text{end}} + n_{\text{ivl}} + n_{\text{temp}})$ do
12                 Set $t_{k_{l-1}^{\text{end}}+n_{\text{ivl}}+j} \leftarrow t_{k_{l-1}^{\text{end}}+n_{\text{ivl}}+j-1} + \Delta_{k_{l-1}^{\text{end}}+n_{\text{ivl}}+j+n_{\text{temp}}}$;
13                 Set $\tilde{\boldsymbol{a}}^*_{\cdot,k^{\text{end}}+n_{\text{ivl}}+j} \leftarrow \tilde{\boldsymbol{a}}^*_{\cdot,k_{l-1}^{\text{end}}+n_{\text{ivl}}+j+n_{\text{temp}}}$;
14         else
15             $n_{\text{temp}} \leftarrow k_l^{\text{start}} - k_{l-1}^{\text{end}} - n_{\text{ivl}} - 1$;
16         Set $\tilde{N} \leftarrow \tilde{N} - n_{\text{temp}}$,   $n_{\text{shift}} \leftarrow n_{\text{shift}} + n_{\text{temp}}$;
17 return: $(\mathcal{G}_{\tilde{N}}, \tilde{\boldsymbol{a}}^*)$;
```

In Algorithm 6.1, we loop over all bang-bang arcs, i.e., the number of singular arcs plus one (and we assume that there is at least one singular arc). We check whether the chosen bang-bang arc fulfills the MDT (line 4). If so, we shift $\boldsymbol{a}^*$ and the grid points t in $\mathcal{G}_N$ by a shift parameter n_{temp}, representing the number of intervals of the bang-bang-arc minus the number of unfixed edge intervals n_{ivl} (line 5-13). Thereby, we overwrite the chosen values $\boldsymbol{a}^*$ and grid points associated with the selected bang-bang arc. Consequently, we update the number of total intervals

$\tilde{N}$ and the number of total shifted intervals n_{shift} (line 16). We recognize that this algorithm reduces the number of intervals for the (CIA) problem, the optimal solution of which will be projected back to the original grid $\mathscr{G}_N$.

6.2 On the complexity of (CIA) and the connection to scheduling theory

This section is dedicated to complexity investigations of (CIA). For this purpose, we define its decision version.

Definition 6.1 (CIA-DEC)
*Let a (CIA) problem instance be given. We denote by (**CIA-DEC**) the decision version of (CIA), which can be stated as follows: For $K \geq 0$, is there a feasible solution $\boldsymbol{w} \in \Omega_N$ of (CIA) with $\theta(\boldsymbol{w}) \leq K$?*

The similarity of the (CIA) approximation inequality (4.4) with the capacity constraint of a knapsack problem, where the grid intervals $\bar{\Delta}$ represent capacity weights, is striking. The decision version of the knapsack problem itself is NP-complete, but there exists a pseudo-polynomial time algorithm using dynamic programming [270]. We leave open the question of the complexity class of (CIA) on a general grid and remark that in the TV constrained case, i.e., (CIA-TV), a polynomial number of solutions $\mathcal{O}(N^{\sigma_{\max}})$ has been established in [137], Corollary 7, independent of the applied discretization.

In the following, we assume an equidistant grid, i.e., $\Delta_j = \bar{\Delta}$, for all $j \in [N]$, and prove that in this case (CIA) can be solved in polynomial time. First, we introduce its corresponding scheduling problem class.

On a single machine, there are $n \in \mathbb{N}$ jobs to be sequenced. We assume the *processing time* of all jobs to be equal, i.e. $p_j = p \in \mathbb{R}^+$ for $j \in [n]$. The processing of job $j \in [n]$ must begin no sooner than its *release time* $r_j \in \mathbb{N}$, shall be completed no later than its *due time* $d_j \in \mathbb{N}$, with $r_j \leq d_j$, and may not be preempted. With respect to the release times, we are interested in finding a feasible schedule that minimizes the maximum *tardiness* $T_{\max}$. The latter is defined as the maximum lateness over all jobs with respect to their due times. The described scheduling problem is formally introduced in the following definition. We refer to [105] for further details on scheduling notation and the problem class.

Definition 6.2 (CIA-Sched)
*We define the scheduling problem **(CIA-Sched)** by its scheduling notation $\left(1 \mid r_j, d_j, p_j = p \mid T_{\max}\right)$. The first field indicates that we are concerned with one machine. The second field lists the jobs characteristics. Each job $j \in [N]$ has a release time r_j, a due time d_j, and a processing time p_j which is assumed to be equal for all jobs. Finally, the third field represents the objective, which is to minimize the maximum tardiness $T_{\max}$*

$$\min\ T_{\max}, \quad \textit{with} \quad T_{\max} := \max_{j \in [n]} T_j, \ \textit{and} \ \ T_j := \max\{0, C_j - d_j\},$$

where C_j denotes the completion time of job $j \in [n]$.

It turns out that (CIA-DEC) and (CIA-Sched) are closely connected, as established by the following theorem.

Theorem 6.1 (Equivalence of (CIA-DEC) and (CIA-Sched))
(CIA-DEC) with an equidistant grid is a special case of the decision version of (CIA-Sched).

Proof. The decision version of (CIA-Sched) can be stated as follows: 'For $K_s \geq 0$, is there a feasible schedule with $T_{\max} \leq K_s$?' Such a schedule exists if and only if a feasible schedule without late jobs exists where K_s is added to the due time of each job, i.e.,

$$\tilde{d_j} := d_j + K_s.$$

The decision version thus amounts to solving the feasibility problem $\left(1 \mid r_j, \tilde{d_j}, p_j = p \mid -\right)$.

Now, we construct a specific outcome of this scheduling problem that is equivalent to (CIA-DEC) with $K \geq 0$. Assume there are n_ω given job families each with n_f, $f \in [n_\omega]$, tasks and that altogether $n = \sum_{f=1}^{n_\omega} n_f$ jobs are to be processed. Let (f,k) be the kth job of family f. We consider the problem (CIA-Sched-f):= $\left(1 \mid r_{f,k}, d_{f,k}, p_{f,k} = \bar{\Delta} \mid -\right)$ with the number of jobs per family given via

$$n_f := \max\left\{ i \in \mathbb{N} \;\middle|\; \sum_{l=1}^{N} a_{f,l} - i \geq -K/\bar{\Delta} \right\}.$$

We specify the release times and due times for each job (f,k) of the problem:

$$r_{f,k} := \min\left\{ j \geq r_{f,k-1} + 1 \;\middle|\; \sum_{l=1}^{j} a_{f,l} - k \geq -K/\bar{\Delta} \right\}, \text{ with } r_{f,0} := 0, \tag{6.2}$$

$$d_{f,k} := \begin{cases} \infty, & \text{if } \sum_{l=1}^{N} a_{f,l} - k \leq K/\bar{\Delta}, \\ \max\left\{ j \;\middle|\; \sum_{l=1}^{j} a_{f,l} - k \leq K/\bar{\Delta} \right\}, & \text{else.} \end{cases} \tag{6.3}$$

(CIA-Sched-f) is an instance of the decision version of (CIA-Sched), where we set $p = \bar{\Delta}$, $K_s = K/\bar{\Delta}$, and the number of jobs to be processed; the release times and due times are defined as above. It remains to be shown that (CIA-Sched-f) has a feasible solution if and only if the corresponding (CIA-DEC) problem has a feasible solution.

Let $\boldsymbol{w} \in \Omega_N$ be a feasible solution of (CIA-DEC). We construct a feasible schedule of (CIA-Sched-f) by providing the positions $pos_{f,k} \in [n]$ of all jobs $(f,k), f \in [n_\omega], k \in [n_f]$:

$$pos_{f,k} := \min\left\{ j \;\middle|\; \sum_{l=1}^{j} w_{f,l} = k \right\}.$$

If $\sum_{l=1}^{N} w_{f,l} < k$, we set the position $pos_{f,k} = pos$, where pos is arbitrarily chosen from among $\{N+1, \ldots, n\}$ and is not yet taken by another job. The release times and due times (6.2)–(6.3) of all jobs are satisfied since $\boldsymbol{w}$ fulfills the approximation inequality (4.4). Hence, we have created a feasible schedule. For the other direction, we assume there is a given feasible schedule of (CIA-Sched-f) with job positions $pos_{f,k}$. Let the corresponding solution of (CIA-DEC) be defined by

$$w_{i,j} := \begin{cases} 1, & \text{if } \exists (i,k) : pos_{i,k} = j \leq N, \\ 0, & \text{else.} \end{cases}$$

By the definitions of release times and deadlines (6.2)–(6.3), $\boldsymbol{w}$ is feasible for (CIA-DEC). □

Remark 6.1 ((CIA-Sched) can be efficiently solved by Horn's rule)
From the scheduling literature, the *earliest due date* heuristic, also known as Horn's rule [129],

can be stated as 'At any time, schedule an available job with the smallest due date.' The resulting schedule is known to be optimal for minimizing the maximum tardiness on single-machine problems, e.g., problems of type (CIA-Sched). Horn's rule can be executed in $\mathcal{O}(n \log n)$ time, where n is the number of jobs. Therefore, the decision version of (CIA-Sched) is in the complexity class P. Transferred to the setting of (CIA-DEC), this rule is equivalent to the NFR scheme from Definition 4.9, where $K/\bar{\Delta}$ is applied as a next-forced rounding threshold instead of $\bar{\Delta}$.

Corollary 6.1 ((CIA) solvable in polynomial time up to TOL-accuracy)
Consider (CIA) with equidistant discretization. Its optimal solution up to an objective accuracy of $TOL > 0$ can be found in polynomial time with complexity $\mathcal{O}\left(n_\omega N \log(n_\omega N) \log\left(\lceil \bar{\Delta}/TOL \rceil\right)\right)$.

Proof. Combining Theorem 6.1 with Remark 6.1 implies that we can solve (CIA-DEC) by a modified NFR scheme (i.e., Horn's rule) in the setting of (CIA-Sched-f), in $\mathcal{O}(n \log n)$, where n are the number of jobs. The optimal objective of (CIA) is bounded by $\bar{\Delta}$, see [135], Proposition 4.8. We therefore conclude that there are at most N jobs for each job family. Because n_ω job families exist, the problem involves at most $n_\omega N$ jobs. (CIA) can be solved by iteratively solving (CIA-DEC) as part of a bisection algorithm. It is sufficient to consider $K \leq \bar{\Delta}$ as a fixed objective value for (CIA-DEC) due to the boundedness of the optimal objective of (CIA) by $\bar{\Delta}$. Hence, we execute Horn's rule as part of the bisection algorithm at most $\log\left(\lceil \bar{\Delta}/TOL \rceil\right)$ times, which concludes the claimed complexity. □

6.3 Branch-and-bound with time dependent branching

Sophisticated MILP solvers such as `Gurobi` struggle to solve (CIA) efficiently, see [135]. This may be due to the fact that its canonical linear programming relaxation, i.e. (CIA) with $w_{i,j} \in [0,1]$, yields only trivial lower bounds in the case without additional combinatorial constraints. To this end, Sager, Kirches, and Jung suggested a tailored BnB scheme for solving (CIA) more efficiently, see [135, 137, 224]. Algorithm 6.2 describes the main steps. The algorithm exploits that an evaluation of the objective function up to the current grid interval yields a valid lower bound due to the maximization operator over all intermediate steps in the objective function. This lower bound is exceptionally cheap to compute, and it is tighter than canonical relaxations [137]. We select nodes from a queue Q until it is empty or until a termination criterion, such as a maximum number of iterations or a time limit (line 2), is reached. The selected node `n` is pruned if its lower bound θ is greater than the global upper bound UB (lines 4 - 5), or we update the currently best node `n*` to be `n` if its depth equals the number of intervals N (lines 6 - 7). We branch forward with respect to the interval index $j \in [N]$, whereby for each child node creation, all control entries $w_{i,j}$ become fixed with exactly one index set to be active (line 9). Nodes contain information on their depth, which is the interval number index; their so-far largest accumulated control deviation θ; and the accumulated deviation for each control realization θ_i. Depending on the imposed combinatorial constraints, we also save information about the previous $w_{i,j}$ values in the nodes and only add their child nodes if they satisfy these constraints (line 10). We note that the practical performance of a depth-first node selection strategy is usually superior to that of a θ ordered (breadth-first) node selection strategy [49]. For further details and numerical examples benchmarking Algorithm 6.2 with MILP solvers, we refer to [49, 135].

```
Algorithm 6.2: Branch-and-bound for solving (CIA)
   Input  : Relaxed control values $\boldsymbol{a}^* \in \mathscr{A}_N$, grid $\mathscr{G}_N$, termination criterion, parameters for
            combinatorial constraints.
   Output: (Optimal) solution $(\theta^*, \boldsymbol{w}^*)$ of (CIA).
1  Initialize node queue $Q$ with empty node, and set upper bound $UB$.
2  while $Q \neq \emptyset$ and termination criterion not reached do
3      Choose $\mathsf{n} \in Q$ according to node selection strategy.
4      if $n.\theta > UB$ then
5          Prune node $\mathsf{n}$.
6      else if $n.depth = N$ then
7          Set new best node $\mathsf{n}^\star \leftarrow \mathsf{n}$ and $UB = \mathsf{n}.\theta$
8      else
9          Create $n_\omega$ child nodes $\mathsf{c}_i,\ i \in [n_\omega]$ with
               $\mathsf{c}_i.depth \leftarrow d := \mathsf{n}.depth + 1,$
               $\mathsf{c}_i.w_{i,d}(k) \leftarrow \begin{cases} 1 & \text{if } k = i, \\ 0 & \text{otherwise,} \end{cases}$
               $\mathsf{c}_i.\theta_k \leftarrow \mathsf{n}.\theta_k + (a^*_{k,d} - w_{k,d}) \cdot \Delta_d$
               $\mathsf{c}_i.\theta \leftarrow \max\Big(\{\mathsf{n}.\theta\} \cup \big\{|\mathsf{c}_i.\theta_k| \mid k \in [n_\omega]\big\}\Big).$
10         Add $\mathsf{c}_i$ to $Q$ if and only if it satisfies all combinatorial constraints.
11 return: $(\theta^*, \boldsymbol{w}^*) = (\mathsf{n}^*.\theta, \mathsf{n}^*.\boldsymbol{w})$;
```

The BnB algorithm can also be adapted to solve (CIA1). In that case, one needs to apply a different objective function θ by using the modified child node objective calculation

$$\mathsf{c}_i.\theta \leftarrow \max\Big(\{\mathsf{n}.\theta\} \cup \{ \sum_{k \in [n_\omega]} |\mathsf{c}_i.\theta_k|\}\Big) \tag{6.4}$$

in line 9, the fourth update. The objective value θ of any node still serves as a lower bound to the optimal objective. However, the 1-norm structure can cause large objective function values to accumulate mostly close to the end of the time horizon, yielding a weak lower bound for early intervals compared with the lower bound of the maximum norm node. A similar modification of the objective calculation according to scaled (CIA) problem formulations such as (SCIAmax) can be analogously made.

6.4 The extended formulated (CIA) problem

The interest in extended formulations of MILPs comes from the fact that the two programs

$$\max\{g(x) \,:\, x \in \text{proj}_x(\mathscr{F})\} \quad \text{and} \quad \max\{g(x) + 0 \cdot z \,:\, (x,z)^\top \in \mathscr{F}\}$$

are equivalent, where proj_x denotes the projection of the feasibility set $\mathscr{F}$ to the domain of x [61]. However, solving the second problem is sometimes easier than solving the first problem because the set $\mathscr{F}$ may be easier to describe than its projection. Generally, a polytope $\mathscr{F}_1$ is an *extended formulation* of the polytope $\mathscr{F}_2$ if $\mathscr{F}_2$ is a projection of $\mathscr{F}_1$. We apply the concept of extended formulations to the (CIA) setting in the sense that we introduce new integer variables that indicate whether there is a switch from one mode to another on a specific interval. This reformulation can be seen as a *switching time optimization* perspective on the (CIA) problem, but without the direct opportunity to eliminate integrality constraints. We discuss two approaches to an extended formulation based on tracking the switching events by employing new variables.

We first briefly consider switching variables based on a variable switching sequence. For this, we replace $\boldsymbol{w} \in \Omega_N$ by the switching variables $z_{i_1,i_2,j}$, which indicate whether there is a switch between modes i_1 and i_2, $i_1, i_2 \in [n_\omega]$, between the $(j-1)$th and jth intervals. We clarify that switches occur solely on grid points $t_j \in \mathscr{G}_N,\ j \in [N-1]$, since the binary control $\boldsymbol{w}$ is discretized to be piecewise constant on the intervals $j \in [N]$. Because t_0 and t_f cannot be switching points, the switches can happen at the beginning of the intervals $j = 2, \dots, N$. With this explanation, we introduce the variables $z_{i_1,i_2,j}$:

$$z_{i_1,i_2,j} := \frac{1}{2}\left[(w_{i_1,j-1} - w_{i_1,j}) + (w_{i_2,j} - w_{i_2,j-1})\right], \quad \text{for } j = 2, \dots, N,\ i_1, i_2 \in [n_\omega], i_1 \neq i_2.$$

Thus, $z_{i_1,i_2,j} = 1$ if and only if there is a switch from mode i_1 to mode i_2 at the beginning of interval j, and $z_{i_1,i_2,j} = -1$ for the reverse switch from i_2 to i_1. We still need variables $w_{i,1}$, $i \in [n_\omega]$ that indicate the active mode of the first interval. With these variables, it is possible to reformulate the approximation inequality constraint (4.4) from (CIA) and hence to receive a lifted version of (CIA). However, we omit presenting it here because we postulate that an extended formulation based on a fixed switching sequence is computationally more promising. Let us assume that there is a given sequence of activated controls in the sequel and that only the switching times are sought. To do so, in Section 6.4.1 we introduce the corresponding variables that we use in Section 6.4.2 to define a (CIA)-equivalent MILP, which can be solved by a BnB algorithm from Section 6.4.3.

6.4.1 Definition of switching variables

The motivation to take the sequence of active modes as given stems from preliminary information about the solution structure. For example, bang-bang arcs in the relaxed solution $\boldsymbol{a}^*$ or a priori information about the structure of the underlying (BOCP) indicate a certain mode sequence. Nevertheless, assuming a given mode sequence is not a restriction because we can include arbitrarily many additional modes whose duration shrinks to zero if they are skipped in the optimal solution (see Remark 6.2).

Definition 6.3 (Sequence of active controls Π, active control mapping $\boldsymbol{\pi}$)
Let the given sequence of active controls be denoted by $[n_\omega]^{n_\sigma+1} \ni \Pi := (i_1, i_2, \dots)$ *with* $i_1, i_2 \in [n_\omega]$, $i_1 \neq i_2$, *where* n_σ *denotes the number of switching events, i.e.,* $|\Pi| = n_\sigma + 1$. *Furthermore, we define* $\pi : \{1, \dots, n_\omega\} \times \{1, \dots, n_\sigma + 1\} \to \{0, 1\}$ *as the mapping that indicates whether the* i*th mode is active in the* k*th position of*Π*:*

$$\pi(i,k) := \begin{cases} 1, & \textit{if } i = \Pi(k), \\ 0, & \textit{otherwise.} \end{cases} \tag{6.5}$$

We clarified above that switches may occur only on grid points $t_j \in \mathscr{G}_N,\ j \in [N-1]$. In the extended formulated (CIA) problem, however, we allow switches to occur on t_0 and $t_N = t_f$ to skip activations of the sequence Π. Tactically, we introduce switching indicator variables κ that comprise a representation of these artificial switches for this purpose.

Definition 6.4 (Switching indicator variables $\boldsymbol{\kappa}$)
For each switch $s \in [n_\sigma]$, we introduce the given set of feasible time point indices $\mathscr{J}_s^f \subseteq [N]_0$ on which s may occur. The variable $\kappa_{s,j}$, $j \in \mathscr{J}_s^f$, indicates on which time point $t_j \in \mathscr{G}_N$ the switch s takes place:

$$\kappa_{s,j} := \begin{cases} 1, & \text{if switch } s \text{ happens on } t_j, \\ 0, & \text{otherwise.} \end{cases} \tag{6.6}$$

Remark 6.2 (Artificial switches and empty activation intervals)
Rather than to stating that a switch s occurs on t_j, we could equivalently write that it occurs at the beginning of the $(j+1)$th interval. In this sense, $\kappa_{s,0} = 1$ indicates that switch s is artificial since it takes place at the beginning of the first interval (t_0). In the same way, $\kappa_{s,N} = 1$ implies that switch s appears at the beginning of interval $N+1$; hence, it is also omitted from the control problem. Furthermore, the definition of $\boldsymbol{\kappa}$ deliberately permits more than one switch on each grid point so that the duration of certain mode activations given by Π can shrink to zero.

Definition 6.5 (Activation duration $\boldsymbol{\eta}$ between two switches)
We define the auxiliary variables η_k, $k \in [n_\sigma + 1]$, which are the duration between two switches, as follows:

$$\begin{aligned} \eta_1 &:= \sum_{j \in \mathscr{J}_1^f} \kappa_{1,j} \sum_{l \in [j]} \Delta_l, \\ \eta_k &:= \sum_{j \in \mathscr{J}_k^f} \kappa_{k,j} \sum_{l \in [j]} \Delta_l - \sum_{i \in [k-1]} \eta_i, \qquad \text{for} \quad 2 \le k \le n_\sigma, \\ \eta_{n_\sigma+1} &:= t_f - t_0 - \sum_{k \in [n_\sigma]} \eta_k. \end{aligned}$$

6.4.2 MILP formulation

We derive the constraints for the extended formulated version of (CIA). First, all control activation durations need to be non-negative:

$$\eta_k \ge 0, \qquad \text{for all } k \in [n_\sigma], \tag{6.7}$$

where $\eta_{n_\sigma+1} \ge 0$ holds by Definition 6.5. Constraint (6.7) also implies that the s_1th switch does not occur before any of the previous switches $s_2 < s_1$, $s_1, s_2 \in [n_\sigma]$. Second, we require that every switch s happens on exactly one time point:

$$\sum_{j \in \mathscr{J}_s^f} \kappa_{s,j} = 1, \qquad \text{for all } s \in [n_\sigma]. \tag{6.8}$$

It remains to state the max-norm-induced approximation inequality constraint (4.4) in the lifted setting. Therefore, we make the following observation. With fixed $\boldsymbol{w}_{i,\cdot}$ the expression $\sum_{l \in [j]} (a^*_{i,l} - w_{i,l}) \Delta_l$ is monotonically increasing (if $\boldsymbol{w}_{i,\cdot} = 0$) or monotonically decreasing (if $\boldsymbol{w}_{i,\cdot} = 1$) with increasing interval j. Therefore, the absolute value of this term is maximal on an interval directly before a switch. We conclude that the approximating inequalities only need to be formulated for the respective active modes before and after the corresponding switch. We exploit this argument by formulating the approximating inequalities for all switches, expressed by the indicator variable $\kappa_{s,j}$:

$$\theta \geq \pm\kappa_{s,j}\left(\sum_{l\in[j]} a^*_{\Pi(s),l}\cdot\Delta_l - \sum_{k\in[s]}\eta_k\cdot\pi(\Pi(s),k)\right), \qquad \text{for all } s\in[n_\sigma],\ j\in\mathscr{J}^f_s, \tag{6.9}$$

$$\theta \geq \pm\kappa_{s,j}\left(\sum_{l\in[j]} a^*_{\Pi(s+1),l}\cdot\Delta_l - \sum_{k\in[s]}\eta_k\cdot\pi(\Pi(s+1),k)\right), \qquad \text{for all } s\in[n_\sigma],\ j\in\mathscr{J}^f_s. \tag{6.10}$$

We recognize that in these constraints we subtract the total activation length of mode $\Pi(s)$, respectively $\Pi(s+1)$, from the accumulated relaxed control values until interval j. Finally, one approximation inequality needs to hold at the end of the time horizon for each control – independent of its activation:

$$\theta \geq \pm\left(\sum_{l\in[N]} a^*_{i,l}\cdot\Delta_l - \sum_{k\in[n_\sigma]}\eta_k\cdot\pi(i,k)\right), \qquad \text{for all } i\in[n_\omega]. \tag{6.11}$$

With this preliminary work we are able to introduce the corresponding MILP.

Definition 6.6 (STO-CIA)
*For given $a^*\in\mathscr{A}_N$ and a sequence of modes $\Pi\in[n_\omega]^{n_\sigma+1}$, we define the problem (**STO-CIA**):*

$$\begin{aligned}\min_{\theta\geq 0,\boldsymbol{\kappa}\in\{0,1\}^{n_\sigma+1\times N}}\quad & \theta\\ \text{s.t.}\quad & \textit{Nonnegativity of activation intervals } (6.7),\\ & \textit{Switch to time point matching constraint } (6.8),\\ & \textit{Approximation inequalities } (6.9), (6.10), (6.11).\end{aligned}$$

We omit the exact proof of the equivalence of the optimal solutions of (STO-CIA) and (CIA) here but claim that $\boldsymbol{w}^*$ can be uniquely constructed from the optimal $\boldsymbol{\kappa}^*$ via

$$w^*_{i,j} := \begin{cases} 1, & \text{if there is a } k\in[n_\sigma+1] \text{ with } i=\Pi(k) \text{ and for } j \text{ holds:}\\ & \quad j\leq \sum_{l\in\mathscr{J}^f_1}\kappa^*_{1,l}l \qquad\qquad \text{for } k=1,\\ & \quad \sum_{l\in\mathscr{J}^f_{k-1}}\kappa^*_{k-1,l}l < j \leq \sum_{l\in\mathscr{J}^f_k}\kappa^*_{k,l}l \quad \text{for } 2\leq k\leq n_\sigma,\\ & \quad \sum_{l\in\mathscr{J}^f_{n_\sigma}}\kappa^*_{n_\sigma,l}l < j \qquad \text{for } k=n_\sigma+1.\\ 0, & \text{otherwise.}\end{cases}$$

The term $\sum_{l\in\mathscr{J}^f_{k-1}}\kappa^*_{k-1,l}l$ identifies the interval of the $(k-1)$th switch, which is useful for identifying the intervals j of the active mode i in the above mapping. We briefly discuss adding of the MDT and TV constraints to (STO-CIA). MDT constraints can be easily added by restricting the duration variables η_k. These variables are also useful for limiting the number of actual switches. Let $\boldsymbol{\xi}\in\{0,1\}^{n_\sigma+1}$ indicate whether the kth control activation, $k\in[n_\sigma+1]$, takes place (equivalent to $\eta_k > 0$) or is skipped (equivalent to $\eta_k = 0$). Then, the number of actual control activations, i.e. switches, can be limited by $\sigma_{\max} \in \mathbb{N}$, with $n_\sigma > \sigma_{\max}$, by means of the constraints

$$(1-\xi_k)\cdot\eta_k \leq 0, \qquad \text{for } k \in [n_\sigma+1], \tag{6.12}$$

$$\sum_{k\in[n_\sigma+1]} \xi_k \leq \sigma_{\max}+1. \tag{6.13}$$

Note that the above cardinality constraints (6.12) contain bilinear terms. Therefore, by adding these conditions to (STO-CIA), we obtain a mixed-integer quadratic program (MIQP).

We conclude this subsection with a comparison of the number of feasible solutions in (STO-CIA) and (CIA).

Remark 6.3 (Complexity reduction through (STO-CIA))
If we consider a fixed sequence Π and allow the omission of control activations (meaning $\eta_k = 0$ for some $k \in [n_\sigma+1]$), then we are left with $\binom{N-1+n_\sigma-1}{n_\sigma}$ feasible solutions $\boldsymbol{\kappa}$ for (STO-CIA). If we require the control activations to be strictly positive, this number reduces to $\binom{N-1}{n_\sigma}$ feasible solutions. For comparison, the number of feasible solutions $\boldsymbol{w}$ for (CIA) is ${n_\omega}^N$. If we restrict the number of switches by $\sigma_{\max}$, i.e., we consider (CIA-TV), the number of feasible solutions reduces to

$$|\Omega_N(\sigma_{\max})| = n_\omega \cdot \sum_{s=0}^{\sigma_{\max}} (n_\omega-1)^s \cdot \begin{pmatrix} N-1 \\ s \end{pmatrix}.$$

On the other hand, if we assume that the optimal solution takes $\sigma_{\max}$ switches, we deduce from this equation that we would have $n_\omega \cdot (n_\omega-1)^{\sigma_{\max}}\binom{N-1}{\sigma_{\max}}$ candidate sequences for (STO-CIA).

6.4.3 A switching time branch-and-bound algorithm

(STO-CIA) can be efficiently solved by a tailored BnB scheme that branches forward over all switches $s \in [n_\sigma]$. Thus, the branching rule is different from the version of BnB described in Algorithm 6.2, where the node depth indicates the associated interval index. Because the objective value θ of (STO-CIA) is extremal on an interval right before a switch, branching over switches is beneficial with respect to obtaining tight lower bounds of (STO-CIA). The BnB for (STO-CIA) is described in Algorithm 6.3. The input is similar to that for Algorithm 6.2, with the addition of the sequence of active control modes Π and of the index sets of feasible switching time points $\mathcal{J}_s^f$ for each switch $s \in [n_\sigma]$. The following information about the investigated solutions $\boldsymbol{\kappa}$ is saved in the nodes n:

- n.$depth$: the number of switches taken,
- n.t: the index j of the time point $t_j \in \mathcal{G}_N$ until $\boldsymbol{\kappa}$ is constructed,
- n.θ: the lower bound on the objective value of $\boldsymbol{\kappa}$,
- n.θ_i: mode specific accumulation values for the computation of θ,
- n.T: a vector that indicates the last activation time point index for each mode.

We select nodes from a queue Q until it is empty or a termination criterion is reached (line 2). A selected node n is pruned if its lower bound θ is greater than the global upper bound UB (lines

Algorithm 6.3: Branch-and-bound for solving (STO-CIA)

```
    Input  : Relaxed control values $\boldsymbol{a}^* \in \mathscr{A}_N$, grid $\mathscr{G}_N$, sequence of active modes $\Pi$, index
             sets of feasible switching time points $\mathscr{I}_s^f \subseteq [N]_0$ for each switch $s \in [n_\sigma]$,
             termination criterion, parameters for combinatorial constraints.
    Output: (Optimal) solution $(\theta^*, \boldsymbol{\kappa}^*)$ of (STO-CIA).
 1  Initialize node queue $Q$ with empty node and set upper bound $UB$.
 2  while $Q \neq \emptyset$ and termination criterion not reached do
 3      Choose $\mathsf{n} \in Q$ according to node selection strategy.
 4      if $n.\theta > UB$ then
 5          Prune node $\mathsf{n}$.
 6      else if $n.depth = n_\sigma$ or $n.t = N$ then
 7          Set new best node $\mathsf{n}^* \leftarrow \mathsf{n}$ and $UB = \mathsf{n}.\theta$
 8      else
 9          Create child nodes $\mathsf{c}_k,\ k \in \mathscr{I}^f_{\mathsf{n}.depth+1},\ k \geq \mathsf{n}.t$ with
              $\mathsf{c}_k.depth \leftarrow d := \mathsf{n}.depth + 1$,  and  $\mathsf{c}_k.t \leftarrow k$,
10          if $k > n.t$ then
11              $\mathsf{c}_k.T[\Pi(d)] \leftarrow k$,
12              $\mathsf{c}_k.\theta_{\Pi(d)} \leftarrow \mathsf{n}.\theta_{\Pi(d)} + \sum_{l=\mathsf{n}.t+1}^{k} (a^*_{\Pi(d),l} - 1)\Delta_l$,
13              $\mathsf{c}_k.\theta_{\Pi(d+1)} \leftarrow \mathsf{n}.\theta_{\Pi(d+1)} + \sum_{l=\mathsf{n}.T[\Pi(d+1)]+1}^{k} a^*_{\Pi(d+1),l}\Delta_l$,
14              if $k = N$ or $d = n_\sigma$ then
15                  $\mathsf{c}_k.\theta_i \leftarrow \mathsf{c}_k.\theta_i + \sum_{l=\mathsf{c}_k.T[i]+1}^{N} a^*_{i,l}\Delta_l$, for $i \in [n_\omega],\ i \neq \Pi(d+1)$
16                  $\mathsf{c}_k.\theta_{\Pi(d+1)} \leftarrow \mathsf{c}_k.\theta_{\Pi(d+1)} + \sum_{l=k+1}^{N} (a^*_{\Pi(d+1),l} - 1)\Delta_l$
17          $\mathsf{c}_k.\theta \leftarrow \max\Big(\{\mathsf{n}.\theta\} \cup \big\{|\mathsf{c}_k.\theta_i| \mid i \in [n_\omega]\big\}\Big)$.
18          Add $\mathsf{c}_k$ to $Q$ if and only if it satisfies all combinatorial constraints.
19  Reconstruct $\boldsymbol{\kappa}^*$ from switching points $\mathsf{n}^*.t$ of parent nodes of $\mathsf{n}^*$;
20  return: $(\theta^*, \boldsymbol{\kappa}^*) = (\mathsf{n}^*.\theta, \boldsymbol{\kappa}^*)$;
```

4 - 5). We update the currently best node n^* to be n if the depth of the selected node is equal to the number of switches n_σ or its time index t equals N (lines 6 - 7). Otherwise, concerning the switch index, we branch forward and create child nodes for all feasible switching points with index k associated with the dth switch (line 9 - 10). The index of the switching point is set to be $t = k$ for each child node. If the activation duration is strictly positive ($k > \mathsf{n}.t$), the time point index T of the active mode $\Pi(d)$ on which it was last activated is updated (line 11). Also, the control accumulation values θ_i are updated according to the derived approximation inequalities (6.9), (6.10), and (6.11) (line 12 - 16). Based on these values, the associated objective lower bound of the child node is computed, and the node is added to Q if and only if it satisfies the combinatorial constraints (line 17-18). We remark that the empty node initialization of Q means that all node features are set to zero.

To limit the number of switches by $\sigma_{\max}$ in Algorithm 6.3, there are two options. First, one can choose the length of the mode sequence Π to be less than or equal to $\sigma_{\max} + 1$. Second, it is possible to count the actual switches, i.e., $k > \mathsf{n}.t$ in line 10, and save the count as additional information in the nodes. MDT constraints can also accounted for by means of the activation duration $k - \mathrm{n}.t$.

Most relevant for the node selection strategy are the lower bound θ, the switch index $depth$, and the time point index t. In terms of all three parameters, it is advantageous to select nodes with maximum values, and different rankings of priorities of the three parameters make sense as a node selection strategy.

Remark 6.4 (Lower bound calculation)
In lines 12 - 16 of Algorithm 6.3, we update only the values θ_i of the active mode before and after the switch d, which are $\Pi(d)$ and $\Pi(d+1)$, respectively, except when we reach the last switch or the end of the time horizon (checked in line 14). Alternatively, we could also update θ_i for all other control modes i. This would yield a better lower bound but higher computational costs. Hence, both variants are useful.

6.5 Sum-up rounding variants

The basic SUR scheme has already been introduced in Definition 4.8. Recently, MANNS showed that it could also be successfully applied to the PDE setting, where mesh cell volumes are used as weights instead of interval lengths [176]. This section is dedicated to further modifications of SUR that are based on the different MILP formulations from Section 4.5.1 and that extend SUR to the MDT constraint context. First, we recapitulate a modification that establishes convergence properties of SUR-constructed solutions to satisfy vanishing constraints (4.1f) up to ϵ-feasibility [149].

Definition 6.7 (Sum-up rounding under vanishing constraints [149], Def. 5.4)
Let $\boldsymbol{a}^* \in \mathscr{A}_N$ be given. The SUR scheme vanishing constraint modification computes $\boldsymbol{w} \in \Omega_N$ for $j = 1, \dots, N$ and $i = 1, \dots, n_\omega$ as follows:

$$w_{i,j} := \begin{cases} 1, & \textit{if } i = \underset{k=1,\dots,n_\omega}{\operatorname{argmax}} \left\{ \sum\limits_{l=1}^{j} a^*_{k,l}\Delta_l - \sum\limits_{l=1}^{j-1} w_{k,l}\Delta_l \;\middle|\; a^*_{k,j} > 0 \right\} \quad \textit{(break ties arbitrarily)}, \\ 0, & \textit{else.} \end{cases}$$

6.5.1 Sum-up rounding based on different MILP formulations

The control approximation problem (CIA1) based on the 1-norm was introduced in Definition 4.18. Here, we suggest a SUR variant that constructs an approximate solution of the latter.

Definition 6.8 (1-norm-SUR)
Let $\boldsymbol{a}^* \in \mathscr{A}_N$ be given. We define the 1-norm SUR scheme for constructing $\boldsymbol{w} \in \Omega_N$ for $j = 1, \dots, N$ as follows (break ties arbitrarily):

$$\boldsymbol{w}_{\cdot,j} := \underset{\substack{\boldsymbol{w}_{\cdot,j} \in \{0,1\}^{n_\omega}, \\ \sum_{i \in [n_\omega]} w_{i,j} = 1}}{\operatorname{argmin}} \left\{ \sum_{i \in [n_\omega]} \left| \sum_{l=1}^{j} (a^*_{i,l} - w_{i,l})\Delta_l \right| \right\}.$$

Analogously, a modified SUR scheme can be obtained based on the evaluated model function values $\tilde{\boldsymbol{f}}$ and Definition 4.19.

Definition 6.9 (Scaled-SUR)
Let $\boldsymbol{a}^* \in \mathscr{A}_N$ and the evaluated model function values $\tilde{\boldsymbol{f}}$ from Definition 4.13 be given. We define the Scaled-SUR scheme for constructing $\boldsymbol{w} \in \Omega_N$ for $j = 1, \dots, N$ as follows:

$$w_{i^*,j} := \begin{cases} 1, & \text{if } i^* = \operatorname*{argmin}_{i=1,\dots,n_\omega} \left\{ \max_{k\in[n_x]} \left| \sum_{l=1}^{j} (a^*_{i,l} - w_{i,l})\Delta_l \tilde{f}_{i,l,k} \right| \right\} \quad \text{(break ties arbitrarily)}, \\ 0, & \text{else.} \end{cases}$$

We note that *Scaled-SUR* should be applied with caution in case $\tilde{\boldsymbol{f}}$ is zero for some controls and intervals or in case it comprises both negative and positive values. In this case, $\boldsymbol{w}$ can be discursive because the sum term in the above scheme cancels. The final definition in this subsection proposes rounding schemes that use the evaluated adjoint variables $\boldsymbol{\lambda}$ and exploit the *maximum principle* from Theorem 2.1, *(v)*. In the absence of a Lagrangian term in the objective function and assuming fixed continuous controls, the *maximum principle* yields for a.a. $t \in \mathscr{T}$

$$\boldsymbol{\omega}^*(t) = \operatorname*{argmin}_{\boldsymbol{\omega}\in\Omega} \mathscr{H}(\boldsymbol{x}^*(t), \boldsymbol{\omega}(t), \boldsymbol{\lambda}^*(t)) = \operatorname*{argmin}_{\boldsymbol{\omega}\in\Omega} \left(\boldsymbol{\lambda}^*(t)\right)^\top \left(\boldsymbol{f}_0(\boldsymbol{x}^*(t)) + \sum_{i\in[n_\omega]} \boldsymbol{\omega}_i(t) \boldsymbol{f}_i(\boldsymbol{x}^*(t)) \right).$$

Definition 6.10 ($\mathscr{H}$-Rounding, $\mathscr{H}$-SUR)
Let $\boldsymbol{a}^* \in \mathscr{A}_N$, the evaluated model function values $\tilde{\boldsymbol{f}}$ from Definition 4.13, and the evaluated dual variables $\tilde{\boldsymbol{\lambda}}$ of the ODE constraint (4.1b) *be given. The $\mathscr{H}$-Rounding constructs $\boldsymbol{w} \in \Omega_N$ for $j = 1, \dots, N$ as follows:*

$$w_{i^*,j} := \begin{cases} 1, & \text{if } i^* = \operatorname*{argmin}_{i=1,\dots,n_\omega} \left\{ \sum_{k\in[n_x]} \tilde{\lambda}_{j,k} w_{i,j} \tilde{f}_{i,j,k} \right\} \quad \text{(break ties arbitrarily)}, \\ 0, & \text{else.} \end{cases}$$

We define the $\mathscr{H}$-SUR scheme for constructing $\boldsymbol{w} \in \Omega_N$ for $j = 1, \dots, N$ as follows:

$$w_{i^*,j} := \begin{cases} 1, & \text{if } i^* = \operatorname*{argmin}_{i=1,\dots,n_\omega} \left\{ \sum_{l=1}^{j} \sum_{k\in[n_x]} \tilde{\lambda}_{l,k} (a^*_{i,l} - w_{i,l})\Delta_l \tilde{f}_{i,l,k} \right\} \quad \text{(break ties arbitrarily)}, \\ 0, & \text{else.} \end{cases}$$

While $\mathscr{H}$-Rounding aims to directly minimize the *Hamiltonian*, $\mathscr{H}$-SUR is designed to minimize the accumulated difference of the *Hamiltonian* based on the relaxed and binary control values, respectively.

6.5.2 Dwell time sum-up rounding

This subsection is based on [282], Section 6. Since the SUR scheme is very often used to find approximative solutions of (CIA), but it does not necessarily fulfill MDT constraints, in this section we discuss a canonical extension of the algorithm to this setting. To this end, we first introduce the concept of a *currently activated* control and dwell time interval blocks that depend on the initial interval and the MDT duration C_1.

Definition 6.11 (Initial interval dwell time block index sets $\mathscr{J}_k^{\mathbf{SUR}}$)
Let an MDT $C_1 \geq 0$ be given. For all intervals $k \in [N]$, we define the initial interval dependent dwell time index sets to be

$$\mathscr{J}_k^{SUR}(C_1) := \{k\} \cup \{j \mid t_{j-1} \in \mathscr{G}_N \cap [t_{k-1}, t_{k-1} + C_1)\}.$$

Definition 6.12 (Currently activated control)
We call a control index i currently activated at interval $j = 2, \dots, N$ if

$$w_{i,j-1} = 1$$

holds. Otherwise, or if $j = 1$, we say that the binary control i is currently deactivated.

A grouping of down time forbidden controls for each interval into sets $\mathscr{I}_j^{\mathrm{SUR}}$ is suggested in the following definition.

Definition 6.13 (SUR down time forbidden control set)
Let a minimum down (MD) time $C_D \geq 0$ be given. We define the set of down time forbidden controls $\mathscr{I}_j^{SUR} \subset [n_\omega]$ on interval $j \in [N]$ as follows:

$$\mathscr{I}_j^{SUR} := \{i \in [n_\omega] \mid \exists\, k < j : \ t_{j-1} \leq t_{k-1} + C_D,\ t_{k-1} \in \mathscr{G}_N \ \wedge \ w_{i,k} = 1\}.$$

We say $i \in [n_\omega]$ is MD time admissible on $j \in [N]$ if $i \notin \mathscr{I}_j^{SUR}$ holds.

Note that the above definition assumes implicitly fixed control variables for the previous intervals $[N] \ni k < j$. We have $\mathscr{I}_1^{\mathrm{SUR}} = \emptyset$ because there are no down time forbidden controls on the first interval. Moreover, the set $\mathscr{I}_j^{\mathrm{SUR}}$ may contain several controls, and it contains at most $n_\omega - 1$.

Next, we give a definition of the dwell time sum-up rounding (DSUR) scheme in Algorithm 6.4. It iterates over all intervals $j \in [N]$ and initially selects the interval representing the beginning of the time horizon, where a *currently activated* control does not yet exist. The control-dependent MDT C_i is updated in lines 3 - 4 for each iteration inside the `while` loop so that C_i equals the maximum of the minimum up (MU) time C_U and MD time C_D for a currently activated control, and otherwise it is set to the MU time C_U. The algorithm sets $C_i = C_U$ for all controls in the first `while` iteration. Next, in line 5, one searches for the *MD time admissible* control $i^\star$ with maximum forward control deviation on the upcoming intervals that cover the dwell time $C_{i^\star}$. If $i^\star$ is the *currently activated* control, we fix it to also be active on the currently selected interval j and increase the interval index (lines 7-8). Otherwise, the control is activated on the whole dwell time block represented by its interval indices $\mathscr{J}_j^{\mathrm{SUR}}(C_{i^\star})$, and the interval index is updated accordingly (lines 9-11). Finally, DSUR updates the set of *down time forbidden* controls for the next iteration in line 12. The algorithm stops as soon as the control choice has been made for the last interval N.

Clearly, $\boldsymbol{w}^{\mathrm{DSUR}}$ is a feasible solution for (CIA) because exactly one control is active per interval. It is also feasible for (CIA-U) since whenever a currently deactivated control is activated, it remains active for at least the duration C_U (lines 9-11). The solution also satisfies MD time constraints by the definition of $\mathscr{I}_j^{\mathrm{SUR}}$, making $\boldsymbol{w}^{\mathrm{DSUR}}$ an overall feasible solution for (CIA-UD).

We have transferred the main idea from the original SUR scheme to the problem setting with MDT constraints by selecting the control with the maximum forward control deviation in each

Algorithm 6.4: DSUR algorithm for approximate solution of (CIA-UD).

```
Input  : Relaxed control values $\boldsymbol{a}^* \in \mathscr{A}_N$, grid $\mathscr{G}_N$, MU time $C_U$, MD time $C_D$.
Output: Feasible solution $\boldsymbol{w}^{\mathrm{DSUR}}$ of (CIA-UD).
Initialize $\boldsymbol{w} = \boldsymbol{0}$, $j = 1$, and $\mathscr{I}_j^{\mathrm{SUR}} = \emptyset$.
while $j \leq N$ do
    Set $C_{i_a} \leftarrow \max\{C_U, C_D\}$ for the currently activated control $i_a$;
    Set $C_i \leftarrow C_U$ for all other controls $i \neq i_a$;
    Find the control with maximum deviation (break ties arbitrarily):
        $i^\star \leftarrow \arg\max \left\{ \sum_{l=1}^{j-1} (a^*_{i,l} - w_{i,l})\Delta_l + \sum_{l \in \mathscr{J}_j^{\mathrm{SUR}}(C_i)} a_{i,l}\Delta_l \mid i \in [n_\omega] \setminus \mathscr{I}_j^{\mathrm{SUR}} \right\}$;
    if $i^\star = i_a$ then
        Set $w_{i^\star,j} \leftarrow 1$ and update $j \leftarrow j+1$;
    else
        Set $w_{i^\star,l} \leftarrow 1,\ l \in \mathscr{J}_j^{\mathrm{SUR}}(C_{i^\star,1})$;
        Update $j \leftarrow \max\left\{ l \mid l \in \mathscr{J}_j^{\mathrm{SUR}}(C_{i^\star,1}) \right\} + 1$;
    Update the set of down time forbidden controls $\mathscr{I}_j^{\mathrm{SUR}}$;
return: $\boldsymbol{w}^{\mathrm{DSUR}} = \boldsymbol{w}$;
```

iteration. In the presence of MU time requirements, we need to calculate the forward accumulation for the set of next intervals with total length of at least C_U. For a given MD time larger than the MU time, Algorithm 6.4 compares the forward accumulation with length at least C_D of the *currently activated* control with that of other controls with length of at least C_U. The idea behind this approach is to prevent a situation in which a control is deactivated though it will accumulate a large control deviation during its *down time forbidden* period.

Remark 6.5 (Run time of DSUR)
Algorithm 6.4 is in $\mathscr{O}(n_\omega N^2)$ since it sums up the relaxed control values $\boldsymbol{a}^*$ on the next dwell time induced intervals on each interval and for all controls.

6.6 Dwell time next-forced rounding

This section is based on [282], Section 4. In Definition 4.9 the NFR scheme was introduced as an algorithm that can compute approximations to solutions of (CIA) in $\mathscr{O}(n_\omega N^2)$ [135] and that constructs feasible solutions of (CIA) with an objective no larger than $\bar{\Delta}$. In this section, we introduce dwell time next-forced rounding (DNFR) as a generalization, aiming for a scheme that constructs solutions that are feasible for MDT constraints and from which we derive bounds for the (CIA) objective and its MDT variants in Chapter 7. Several definitions are needed for DNFR, and we begin with a definition of sequences of intervals that are grouped into blocks in the presence of MDT constraints.

Definition 6.14 (Dwell time block interval sets)
Let an MDT $C_1 \geq 0$ be given. We iteratively define the dwell time invoked interval sets $\mathscr{J}_b$ and

their last indices l_b for $b = 1, \dots, n_b$ and with $l_0 := 0$:

$$\mathscr{J}_b := \{l_{b-1}+1\} \cup \{j \mid t_{j-1} \in \mathscr{G}_N \cap [t_{l_{b-1}}, t_{l_{b-1}} + C_1)\},$$
$$l_b := \max\{j \mid j \in \mathscr{J}_b\},$$

where $n_b := \min\{b \mid l_b = N\}$ represents the number of interval blocks.

In the following, we will sometimes loosely write *block* instead of *dwell time block* for brevity. We next establish the lengths of the dwell time blocks.

Definition 6.15 (Dwell time block length)
Let a family of dwell time block interval sets $\{\mathscr{J}_b\}_{b\in[n_b]}$ be given. We denote by $\mathscr{L}_b$ the length of dwell time block $b \in [n_b]$ and name the maximum, respectively minimum, length of all dwell time blocks $\overline{\mathscr{L}}$, respectively $\underline{\mathscr{L}}$, i.e.,

$$\mathscr{L}_b := t_{l_b} - t_{l_{b-1}}, \qquad b \in [n_b],$$
$$\overline{\mathscr{L}} := \max_{b\in[n_b]} \mathscr{L}_b, \qquad \underline{\mathscr{L}} := \min_{b\in[n_b]} \mathscr{L}_b.$$

By the definition of dwell time blocks, we see that $\mathscr{L}_b$ depends both on the time discretization $\mathscr{G}_N$ and on C_1. If $C_1 \leq \underline{\Delta}$, then the blocks are in fact the grid intervals, i.e., $\mathscr{L}_j = \Delta_j,\ j \in [N]$ and $n_b = N$. As soon as $C_1 > \underline{\Delta}$ holds, there is at least one block b with the length of two consecutive intervals $\mathscr{L}_b = \Delta_j + \Delta_{j+1},\ j \in [N-1]$. Overall, one recognizes that $\overline{\mathscr{L}}$ increases monotonically with increasing C_1, obviously stopping as soon as $C_1 > t_f - t_0$. The DNFR scheme relies crucially on the block-dependent accumulated control deviation, which is why we introduce it as an auxiliary variable in the next definition.

Definition 6.16 (Accumulated control deviation $\boldsymbol{\theta_{i,j}, \Theta_{i,b}, \gamma_{i,j}, \Gamma_{i,b}}$)
Let $\boldsymbol{a} \in \mathscr{A}_N$ and $\boldsymbol{w} \in \Omega_N$. For controls $i \in [n_\omega]$, we define the accumulated control deviation on interval $j \in [N]$ as

$$\theta_{i,j} := \sum_{l=1}^{j} (a_{i,l} - w_{i,l})\Delta_l, \qquad \gamma_{i,j} := \sum_{l=1}^{j} a_{i,l}\Delta_l - \sum_{l=1}^{j-1} w_{i,l}\Delta_l,$$

and further define $\theta_{i,0} := 0$ for all $i \in [n_\omega]$. For blocks $b \in [n_b]$, we introduce the analogous variables

$$\Theta_{i,b} := \theta_{i,l_b}, \qquad \Gamma_{i,b} := \Theta_{i,b-1} + \sum_{j\in\mathscr{J}_b} a_{i,j}\Delta_j.$$

In what follows, we sometimes write forward control deviation for control i in order to distinguish $\gamma_{i,j}, \Gamma_{i,b}$ from the (accumulated) control deviation $\theta_{i,j}, \Theta_{i,b}$.

The quantities from Definitions 6.14 – 6.16 are illustrated in Figure 6.1.

Remark 6.6 ($\boldsymbol{\Theta_{i,j}}$ as generalization of $\boldsymbol{\theta_{i,j}}$)
If a small MDT $C_1 \leq \underline{\Delta}$ is given, $\Theta_{i,j}$ trivially equals $\theta_{i,j}$, and the same holds for $\Gamma_{i,j}$ and $\gamma_{i,j}$. Nevertheless, with an MDT of $C_1 > \underline{\Delta}$, one needs interval *and* block related variables to distinguish clearly between both values.

Remark 6.7 (Link between $\theta_{i,j}$ and the (CIA) objective value)
Let $\boldsymbol{w} \in \Omega_N$ and denote by $\theta(\boldsymbol{w})$ its (CIA) objective value. By the above definition, we conclude $\theta(\boldsymbol{w}) = \max_{i\in[n_\omega], j\in[N]} |\theta_{i,j}|$. Generally, one notices that the maximum of the $|\theta_{i,j}|$ values must be assumed in an interval before a switch happens (i.e., $\boldsymbol{w}_{\cdot,j} \neq \boldsymbol{w}_{\cdot,j+1}$) or in the last interval since $|\theta_{i,j}|$ increases monotonically with constant $\boldsymbol{w}_{\cdot,j}$ and increasing j. Hence, with constant $\boldsymbol{w}_{\cdot,j}$ on the dwell time blocks, we also have that $\theta(\boldsymbol{w}) = \max_{i\in[n_\omega], b\in[n_b]} |\Theta_{i,b}|$.

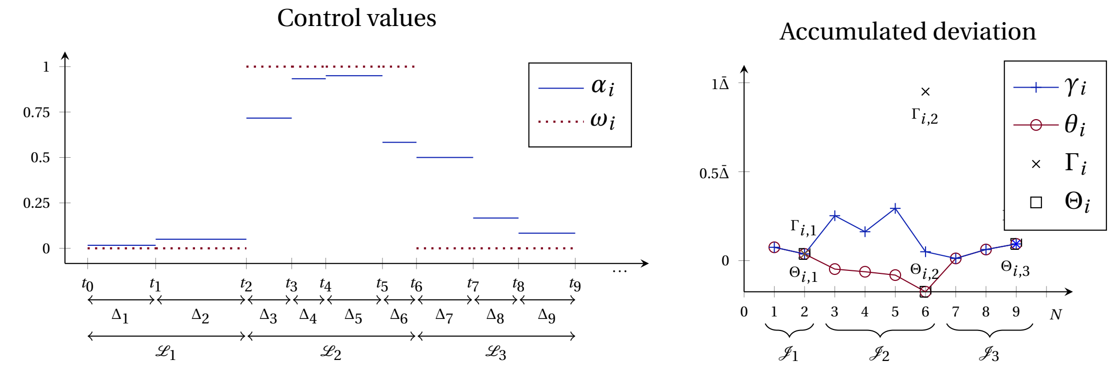


Figure 6.1: Left: Example binary and relaxed control values on nine intervals Δ_j and three blocks $\mathscr{L}_b$. Right: Corresponding accumulated control deviation. With respect to intervals, the forward control deviation γ_i is greater than or equal to θ_i with equality if the control i is inactive. Also, $\Gamma_{i,b}$ is greater than or equal to $\Theta_{i,b}$, where by definition the latter equals the last θ_i before block b begins. Note that $\Gamma_{i,2}$ sums up the weighted α values for intervals 3–6, resulting in a large value.

We introduced in Definition 4.9 the concept of a *next-forced* control that depends on the maximum grid length $\bar{\Delta}$. We generalize this concept by using blocks and a generic rounding threshold factor $C_2 > 0$, instead of always using $C_2 = 1$ as in the NFR scheme. To this end, we present a definition of different types of control variable activations that depend on the choice of prior variables and on $\boldsymbol{a}^* \in \mathscr{A}_N$.

Definition 6.17 (Admissible, forced, and future forced activation)
Let the rounding threshold factor $C_2 > 0$ and $\boldsymbol{a}^* \in \mathscr{A}_N$ be given. The choice $w_{i,j} = 1$ for $i \in [n_\omega]$, $j \in \mathscr{J}_b$, $b = 1, \dots, n_b$ is admissible if

$$\Gamma_{i,b} \geq -C_2\overline{\mathscr{L}} + \mathscr{L}_b$$

holds. Denote by Ω_a^b the set of admissible controls for block b. Similarly, the choice $w_{i,j} = 1$ for $i \in [n_\omega]$, $j \in \mathscr{J}_b$, $b = 1, \dots, n_b$ is forced if

$$\Gamma_{i,b} > C_2\overline{\mathscr{L}}$$

holds. We define a control $i \in \Omega_a^b$ on block b to be l-future forced if

$$\Theta_{i,b-1} + \sum_{k=b}^{l} \sum_{j\in\mathscr{J}_k} a^*_{i,j}\Delta_j > C_2\overline{\mathscr{L}}$$

holds, with the special case $l = b$ indicating that i is actually forced on b. If the above inequality holds for any $l \leq n_b$, we say that the control $i \in \Omega_a^b$ on block b is future forced and group these controls into the set Ω_f^b.

Definition 6.18 (Down time forbidden control)
We introduce the parameter $\chi_D \in \{0,1\}$. If the CIA problem involves an MD time constraint with $C_D > \underline{\Delta}$, we set $\chi_D = 1$ and otherwise set $\chi_D = 0$. We define i_b^D, $b = 3,\dots,n_\omega$, as the index of the control that has been activated on block $b-2$ and deactivated on block $b-1$ – if such a control exists:

$$\exists i \in [n_\omega] \;:\; w_{i,j} = 1, j \in \mathscr{J}_{b-2} \;\wedge\; w_{i,j} = 0, j \in \mathscr{J}_{b-1} \quad \Rightarrow \quad i_b^D := i.$$

Then, let $\mathscr{I}_b^D$ denote the χ_D-dependent set of the down time forbidden control:

$$\mathscr{I}_b^D := \begin{cases} \{i_b^D\}, & \text{if } \chi_D = 1, \text{ and } b \geq 3, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Note that $\mathscr{I}_b^D$ is either the empty set or contains exactly one control index. It may seem unintuitive to declare only one control per block as down time forbidden because a sufficiently large chosen MD time can comprise more than two intervals, and therefore, more than one control could be minimum down forbidden on certain blocks. However, in such situations, where several controls are minimum down forbidden, only one control may be allowed to be active, resulting in a large control deviation. Consequently, a fine granular definition is critical for deriving bounds for (CIA-UD) using the DNFR scheme. We will specify such a worst case in Example 7.1 in Section 7.3, where we argue for tolerating at most one down time forbidden control per block.

We illustrate the different control activation types of Definitions 6.17 and 6.18 in Figure 6.2.

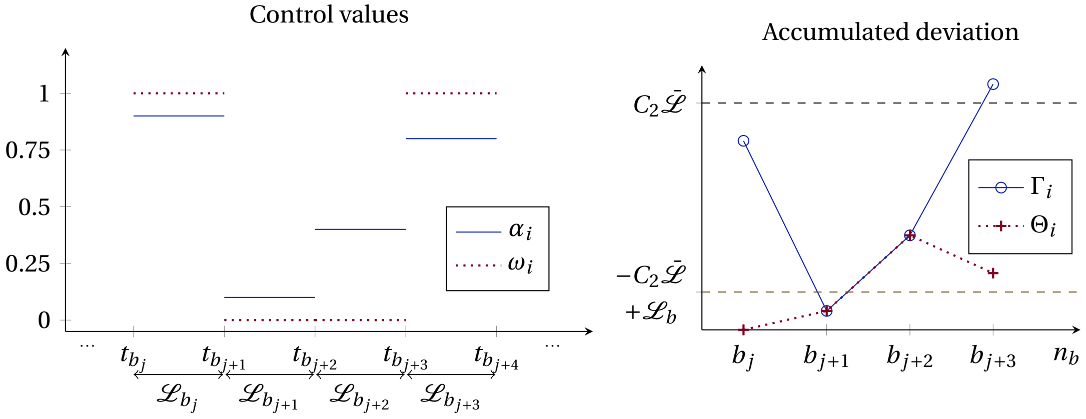


Figure 6.2: Exemplary visualization of the defined quantities. Left: Binary and relaxed control values on four blocks. Right: Corresponding block accumulated control deviation. Control i is *admissible* on block b_j, not *admissible* on block b_{j+1}, *down time forbidden* and b_{j+3}-*future forced* on block b_{j+2}, and *forced* on block b_{j+3}.

Finally, we use these control properties to declare the DNFR scheme in Algorithm 6.5. In contrast to the original NFR, we do not iterate over all intervals but rather over all dwell time

C_1 invoked blocks (line 2). We check whether there is a forced control on each block and if so, activate it (lines 3-4). Otherwise, we test if there is an earliest future forced control, and if so, we set it to be active (lines 5-8). Else, the algorithm selects the control with the maximum forward control deviation (lines 9-12), which represents a fallback to the classical SUR scheme. If the MD mode is turned on by setting $\chi_D = 1$, we exclude the set $\mathscr{I}_b^D$ from our control selection task (lines 3, 5, 11). This consideration of down time forbidden controls is a further extension of the original NFR scheme.

Algorithm 6.5: DNFR algorithm for heuristic solution of (CIA-UD).

```
   Input  : Relaxed control values $\boldsymbol{a}^* \in \mathscr{A}_N$, grid $\mathscr{G}_N$, parameters $C_1, C_2, \chi_D$.
   Output: Feasible solution $\boldsymbol{w}^{\mathrm{DNFR}}$ of (CIA-UD) with approximation quality depending on
           $C_1, C_2, \chi_D$.
 1 Initialize $\boldsymbol{w} \leftarrow \mathbf{0}$;
 2 for all dwell time blocks $b = 1, \dots n_b$ do
 3     if there is a control $i \in [n_\omega] \setminus \mathscr{I}_b^D$ with forced activation then
 4         Set $w_{i,j} \leftarrow 1,\ j \in \mathscr{J}_b$;
 5     else if it exists a future forced control, i.e., $\Omega_f^b \setminus \mathscr{I}_b^D \neq \emptyset$ then
 6         Identify control with earliest future forced activation (break ties arbitrarily):
 7         $i \leftarrow \operatorname{arg\,min} \{ b(i) \in [n_b] \mid i \in \Omega_a^b \setminus \mathscr{I}_b^D,\ i \text{ is } b(i)\text{-future forced} \}$;
 8         Set $w_{i,j} \leftarrow 1,\ j \in \mathscr{J}_b$;
 9     else
10         Find admissible control with maximum control deviation:
11         $i \leftarrow \operatorname{arg\,max} \{ \Gamma_{i,b} \mid i \in \Omega_a^b \setminus \mathscr{I}_b^D \}$ (break ties arbitrarily);
12         Set $w_{i,j} \leftarrow 1,\ j \in \mathscr{J}_b$;
13 return: $\boldsymbol{w}^{\mathrm{DNFR}} = \boldsymbol{w}$;
```

6.7 Adaptive maximum dwell rounding

This section is based on [222], Sections 4 and 5. In order to accelerate the (CIA-TV) solution process, we propose the maximum dwell rounding (MDR) scheme, which is a fast rounding heuristic. It is based on the idea of activating a chosen control mode for as long as possible without violating a desired integral deviation gap $\bar{\theta}$ and then performing this idea with the next promising mode. We apply the scheme iteratively as part of the AMDR algorithm for finding binary controls that satisfy time-coupled combinatorial constraints, such as a TV bound, and derive optimality conditions of the obtained binary control function with respect to the (CIA-TV) problem. Thus, we examine the algorithm mainly from the point of view of limited switch constraints (4.9)–(4.10) but emphasize that it can also be applied to MDT constraints. The AMDR scheme is motivated by the fact that in some instances, efficient BnB algorithms, such as the one presented in Section 6.3, struggle to find the (CIA) optimal solution quickly because the node relaxation can be relatively weak [50]. Therefore, we present a polynomial-time algorithm that constructs good initial guesses for BnB and that in some situations, even solves (CIA-TV) to optimality. We proceed by introducing MDR and AMDR in Section 6.7.1. In Section 6.7.2, we introduce auxiliary (CIA) problems that are useful for investigating the approximation quality and properties of MDR and AMDR presented in Section 6.7.3.

6.7.1 Definition of the algorithm

The concepts of *inadmissible, next forced,* and *forced* activation were introduced in the context of (dwell time) next-forced rounding in Definition 6.17. There, the rounding threshold was chosen to be the maximum dwell time block $\overline{\mathscr{L}}$ or the maximum grid length $\bar{\Delta}$. Here, in contrast, we introduce a flexible rounding threshold $\bar{\theta} > 0$ and therefore slightly modify these activation concepts.

Definition 6.19 ($\bar{\boldsymbol{\theta}}$-inadmissible, $\bar{\boldsymbol{\theta}}$-next-forced, and $\bar{\boldsymbol{\theta}}$-forced activation)
Consider a rounding threshold $\bar{\theta} > 0$ and $\boldsymbol{a}^* \in \mathscr{A}_N$. Let the values of $\boldsymbol{w} \in \Omega_N$ be given until interval $j-1$, with $j \geq 2$. The choice $w_{i,j} = 1$ for $i \in [n_\omega]$, $j \in [N]$ is $\bar{\theta}$-admissible if we have that

$$\theta_{i,j} \geq -\bar{\theta}.$$

Otherwise, we call the control i $\bar{\theta}$-inadmissible. Similarly, the choice $w_{i,j} = 1$ is $\bar{\theta}$-forced if we have that

$$\gamma_{i,j} > \bar{\theta}.$$

Further, let $\mathscr{N}_j(i) \in \{j, \ldots N\}$ denote the next interval on which control i would become $\bar{\theta}$-forced without activation after interval $j-1$:

$$\mathscr{N}_j(i) := \begin{cases} \underset{k=j,\ldots,N}{\operatorname{argmin}} \left\{\theta_{i,j-1} + \sum_{l=j}^{k} a^*_{i,l}\Delta_l > \bar{\theta}\right\}, & \textit{if } \theta_{i,j-1} + \sum_{l=j}^{N} a_{i,l}\Delta_l > \bar{\theta}, \\ \infty, & \textit{else.} \end{cases}$$

Then, we define a control $i^\star \in [n_\omega]$ on interval j to be $\bar{\theta}$-next-forced if and only if

$$\mathscr{N}_j(i^\star) = \min_{i \in [n_\omega]} \mathscr{N}_j(i) \quad \textit{and} \quad \mathscr{N}_j(i^\star) < \infty.$$

The above definition permits more than one control to be $\bar{\theta}$-*next-forced* or $\bar{\theta}$-*forced* for an arbitrary interval $j \in [N]$. This is not the standard case in our discussion, but it will be accounted for in our considerations. The guiding idea behind the above control activations is that we include more and more summands of $\boldsymbol{w}$ into the computation of $\theta(\boldsymbol{w})$ and can thereby choose the next row of $\boldsymbol{w}$ accordingly. The following definition classifies feasible control mode activations on the first interval.

Definition 6.20 (Initially admissible control)
We define a control $i \in [n_\omega]$ to be initially admissible if it is $\bar{\theta}$-admissible on the first interval and if there is no other control $i_1 \neq i$ that is $\bar{\theta}$-forced on the first interval.

We are now able to define the MDR scheme in Algorithm 6.6.
The MDR algorithm assumes a given initial control i_0 and activates it until it becomes $\bar{\theta}$-*inadmissible* or until there is another $\bar{\theta}$-*forced* control. We require the control i_0 to be *initially admissible* because otherwise, $\boldsymbol{w}^{\text{MDR}}$ would violate the control accumulation constraint (4.4). Otherwise, the control i with the maximum forward control deviation $\gamma_{i,j}$ is set to be active and remains so until it becomes $\bar{\theta}$-inadmissible or another control becomes $\bar{\theta}$-forced. This procedure is performed forward in time until the end of the time horizon N is reached. We call the

Algorithm 6.6: Maximum dwell rounding

```
Input : Relaxed control values $\boldsymbol{a}^* \in \mathscr{A}_N$, rounding threshold $\bar{\theta}$, initially admissible
        control $i_0 \in [n_\omega]$.
Initialize $\boldsymbol{w} \leftarrow \mathbf{0}$, $w_{i_0,1} \leftarrow 1$, and $i \leftarrow i_0$.
for $j = 2, \dots, N$ do
    if $i$ is $\bar{\theta}$-inadmissible or there is a $\bar{\theta}$-forced control $i_f \neq i$ then
        Set $i \leftarrow \operatorname{argmax}_{k \in [n_\omega]} \gamma_{k,j}$ (ties may be broken arbitrarily);
    Set $w_{i,j} \leftarrow 1$;
return: $\boldsymbol{w}^{\mathrm{MDR}} = \boldsymbol{w}$;
```

algorithm “maximum dwell rounding” because it tries to dwell in the current mode for as long as possible without violating the given rounding threshold.

Algorithm 6.7: Adaptive maximum dwell rounding

```
   Input : Relaxed control values $\boldsymbol{a}^* \in \mathscr{A}_N$, optimum tolerance $TOL > 0$, allowed number
           of switches $\sigma_{max}$.
1  Initialize $LB \leftarrow 0$; $UB \leftarrow \bar{\theta} = t_f - t_0$.
2  while $UB - LB > TOL$ do
3      for $i = 1, \dots, n_\omega$ do
4          if $i$ is an initially admissible control then
5              Run MDR with $i$ as initial control and set $\boldsymbol{w} \leftarrow \boldsymbol{w}^{\mathrm{MDR}(i)}$;
6              if $\boldsymbol{w}$ satisfies TV constraints (4.9)-(4.10) and $\theta(\boldsymbol{w}) < UB$ then
7                  $UB \leftarrow \theta(\boldsymbol{w})$;
8                  $\bar{\theta} \leftarrow UB - 0.5 \cdot (UB - LB)$;
9                  $\boldsymbol{w}^{\mathrm{AMDR}} \leftarrow \boldsymbol{w}$;
10                 break;
11         else if $i = n_\omega$ then
12             $LB \leftarrow \bar{\theta}$;
13             $\bar{\theta} \leftarrow LB + 0.5 \cdot (UB - LB)$;
14 return: $(\boldsymbol{w}^{\mathrm{AMDR}}, UB)$;
```

AMDR is defined in Algorithm 6.7 and can be described as a *bisection* method. It is initialized with a trivial lower bound LB and upper bound UB for (CIA-TV). The algorithm runs MDR iteratively with different thresholds $\bar{\theta}$ and initially admissible control as long as the difference between the lower and upper bounds exceeds the chosen tolerance TOL (lines 2 - 5). If the computed control function satisfies the TV constraints and exhibits a (CIA-TV) objective value that is smaller than the current UB, we update UB, reset the rounding threshold $\bar{\theta}$ via interval halving of $UB - LB$, and save the current best solution (lines 6 - 10). The evaluation $\theta(\boldsymbol{w})$ is necessary since MDR may construct a control function with an integral deviation gap larger than the desired gap $\bar{\theta}$, as discussed in the following subsections. If no computed control function $\boldsymbol{w}$ with given initial control and $\bar{\theta}$ fulfills the TV constraints, we increase the LB (lines 11 - 13).

6.7.2 (CIA−$\bar{\theta}$), (CIA−$\bar{\theta}$−init), and an associated lower bound

In this subsection, we address a problem that minimizes the switches used in $\boldsymbol{w}$ subject to a given control approximation error, i.e., the integral deviation gap $\bar{\theta}$ that is not to be exceeded by the accumulated control deviation. We then aim to construct a lower bound on its objective that will be useful in the next subsection and for which we introduce useful auxiliary variables and definitions.

Definition 6.21 ((CIA−$\bar{\boldsymbol{\theta}}$), (CIA−$\bar{\boldsymbol{\theta}}$−init))
*For given $\boldsymbol{a}^* \in \mathscr{A}_N$, $\bar{\theta} > 0$, and initial active control $i_0 \in [n_\omega]$, the problem **(CIA−$\bar{\theta}$−init)** is defined to be*

$$\sigma^\star := \min_{\boldsymbol{w}\in\Omega_N} \ \frac{1}{2}\sum_{l=1}^{N-1}\sum_{i=1}^{n_\omega} |w_{i,l+1} - w_{i,l}| \tag{6.14}$$

$$\text{s.t.} \quad \bar{\theta} \geq \pm \sum_{l=1}^{j} (a^*_{i,l} - w_{i,l})\Delta_l \qquad i \in [n_\omega],\ j \in [N], \tag{6.15}$$

$$w_{i_0,1} = 1. \tag{6.16}$$

*We define the problem **(CIA−$\bar{\theta}$)** to be **(CIA−$\bar{\theta}$−init)** without the constraint* (6.16).

Ignoring the fixed initial active control, the problems (CIA−$\bar{\theta}$) and (CIA-TV) are closely connected because the TV constraints (4.9) and (4.10) can be reinterpreted as an objective function subject to a fixed approximation error $\bar{\theta}$. This justifies the naming. In fact, (CIA−$\bar{\theta}$) is the same as problem (CIA-DEC) from Definition 6.1, but with an altered objective function and $K = \bar{\theta}$.

The MDR algorithm from the previous subsection (heuristically) solves (CIA−$\bar{\theta}$−init), and by applying this algorithm to all $i \in [n_\omega]$ as initial active controls, we exploit this relationship to solve (CIA−$\bar{\theta}$) as well, providing a link to the AMDR scheme. We stress that fixing the initial active control i_0 may seem odd. Nevertheless, fixing it reduces the problem complexity, which later yields an optimality result in Theorem 6.2, Section 6.7.3, for the solution constructed by the MDR algorithm with respect to (CIA−$\bar{\theta}$−init).

We note that (CIA−$\bar{\theta}$−init) is similar to (SCARP) from [27, 28]. (SCARP) aims to minimize the switching costs and represents a generalized objective function, whereas in (CIA−$\bar{\theta}$−init) the initial active control is fixed.

Remark 6.8 (Link to scheduling theory)
On an equidistant grid, (CIA−$\bar{\theta}$) can be reformulated into the following equivalent scheduling problem [155]: On a single machine, minimize the total setup costs (TSC) until the Nth processed job, $N \leq n$, so that n jobs (f,k) are processed within $f \in [n_\omega]$ job families subject to release times $r_{f,k}$, deadlines $d_{f,k}$, equal processing times $\bar{\Delta}$, and sequence-independent setup costs; this can be summarized in scheduling notation [105] as

$$\left(1|r_{f,k}, d_{f,k}, \mathrm{SC}_{\mathrm{si},b} = 1, p_{f,k} = \bar{\Delta}|\ \mathrm{TSC}|_1^N\right).$$

In other words, the formation of *batches* is the subject of the scheduling problem, where a batch is a set of jobs of the same family that is processed between two setups of a given schedule or between the beginning/end of the schedule and a setup. We note that the above problem is very similar to (CIA-Sched), which is not surprising given the similarity of (CIA-$\bar{\theta}$) and (CIA-DEC). In the following, we revert to scheduling-like concepts but explicitly dispense with scheduling notation to prevent confusion with the usual mixed-integer optimal control problem (MIOCP) notation.

We next need to establish several definitions to derive a lower bound for (CIA$-\bar{\theta}-$init) at the end of this subsection. We stress that we establish our results on an equidistant grid but that this assumption is dropped in some of the definitions used in later sections.

Definition 6.22 (Activations, release $r_{i,k}$, and deadline intervals $d_{i,k}$)
For each control $i \in [n_\omega]$ on an equidistant grid $\mathcal{G}_N$, we introduce the number of possible activations n_i as

$$n_i := \max\left\{ k \in \mathbb{N} \;\middle|\; \sum_{l=1}^{N} a^*_{i,l} - k \geq -\bar{\theta}/\bar{\Delta} \right\}.$$

Each activation $k \in [n_i]$ is associated with a release and a deadline interval, which are defined by

$$r_{i_0,1} := 1, \qquad d_{i_0,1} := 1, \tag{6.17}$$

$$r_{i,k} := \min\left\{ j \geq r_{i,k-1} + 1 \;\middle|\; \sum_{l=1}^{j} a^*_{i,l} - k \geq -\bar{\theta}/\bar{\Delta} \right\}, \quad \textit{with } r_{i,0} := 0, \tag{6.18}$$

$$d_{i,k} := \begin{cases} \infty, & \textit{if } \sum_{l=1}^{N} a^*_{i,l} - k \leq \bar{\theta}/\bar{\Delta}, \\ \max\left\{ j \;\middle|\; \sum\limits_{l=1}^{j} a^*_{i,l} - k \leq \bar{\theta}/\bar{\Delta} \right\}, & \textit{else.} \end{cases} \tag{6.19}$$

Finally, we call the kth activation of control i necessary if $d_{i,k} < \infty$.

Definition 6.23 (Interval of jth switch τ_j, activation block B, length of block δ)
Consider $\boldsymbol{w} \in \Omega_N$. Let the number of switches of $\boldsymbol{w}$ be defined as $n_\sigma := |TV(\boldsymbol{w})|$ with TV as introduced in (3.5). *We denote by $\tau_j \in \{2, \ldots, N\}$ the corresponding interval of the jth switch of $\boldsymbol{w}$, where we set $\tau_0 := 0$, $\tau_{n_\sigma+1} := N$. On an equidistant grid and if control $i \in [n_\omega]$ is active between two consecutive switches or between one switch and the first/last interval, we define the set of activations of i between these switches as an activation block $B \subseteq [n_i]$. On a general grid, we further define the length of the jth activation block between the $(j-1)$st switch on, i.e., τ_{j-1}, and before the jth switch, i.e., $\tau_j - 1$, via the auxiliary variable $\delta_j = \sum_{l=\tau_{j-1}}^{\tau_j - 1} \Delta_l$ for $j \in [n_\sigma + 1]$.*

We note that the switches actually occur on the grid points. However, we have indexed the variables $w_{i,j}$ according to the intervals, and therefore, for simplicity, we refer to switches on intervals in this subsection. In the following, we sometimes abbreviate activation block as block. To keep the number of used switches small and when deciding to set up a new block, it is highly relevant to know the maximum number of activations that could be included into the block, beginning with activation k. An activation $j > k$ cannot be included in the block if its release interval begins later than the deadline interval of activation k plus the number of activations between k and j. We give a definition that formalizes these deadlines for initial activation-dependent block deadlines. Based on these block deadlines, it is straightforward to introduce the notion of a *block deadline feasible* partition of activations into blocks. Due to the constraint (6.16), the first activation of control i_0 must be executed on the first interval, for which we introduce the definition of *fixed initial active control* feasibility.

Definition 6.24 ($db_{i,k}$, block deadline and fiac feasible partition)
Consider an equidistant grid. For $i \in [n_\omega]$, the deadline of a block that begins with the kth activation, $k \in [n_i]$, is defined by

$$db_{i,k} := d_{i,l}, \quad where \quad l := \max\{j \geq k \mid r_{i,j} \leq d_{i,k} + j - k\}. \tag{6.20}$$

Let P_i denote a partition of all activations $[n_i]$ for $i \in [n_\omega]$. We call P_i block deadline feasible if and only if for all subsets $B \in P_i$, i.e., all blocks, the following holds:

$$r_{i,\max\{k \in B\}} \leq d_{i,\min\{k \in B\}} + |B| - 1.$$

Furthermore, we refer to P_i as a fixed initial active control (fiac) feasible partition if and only if for all $k \in B_1$ it holds that

$$r_{i,k} = k,$$

where $B_1 \in P_i$ denotes the first activation block of P_i.

In the last definition, we provided the concept of a control specific partition of all activations. The kth activation of control $i \in [n_\omega]$ generally does not coincide with the kth interval. The following example illustrates the concepts introduced in this section and in particular, demonstrates that in total there may be *more possible* but *fewer necessary* activations than intervals N.

Example 6.1
Let the following matrices $\boldsymbol{a}^* \in \mathscr{A}_N$ and $\boldsymbol{w} \in \Omega_N$ be given for equidistant discretization:

$$\boldsymbol{a}^* := \begin{pmatrix} 1 & 1 & 0.8 & 0 & 0 & 0 & 0 & 0 & 0.5 \\ 0 & 0 & 0.2 & 0 & 0.1 & 0.8 & 1 & 1 & 0.5 \\ 0 & 0 & 0 & 1 & 0.9 & 0.2 & 0 & 0 & 0 \end{pmatrix}, \quad \boldsymbol{w} := \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

where $n_\omega = 3, N = 9$. Consider $i = 1$ to be the fixed initial active control, and set a rounding threshold of $\bar{\theta} = 1\bar{\Delta}$. Then in total, there are eleven possible activations with their release and deadline intervals:

$$\begin{aligned} i = 1: &\quad [r_{1,k}, d_{1,k}] = [1,1],\ [2,3],\ [3,9],\ [9,\infty], \\ i = 2: &\quad [r_{2,k}, d_{2,k}] = [1,6],\ [6,7],\ [7,8],\ [8,\infty], \\ i = 3: &\quad [r_{3,k}, d_{3,k}] = [1,5],\ [4,6],\ [6,\infty]. \end{aligned}$$

There are $4, 3$, and 2 activations in $\boldsymbol{w}$ for the controls $i = 1, 2$, and 3, respectively. These activations are grouped into four activation blocks so that $\boldsymbol{w}$ uses three switches. For instance, the first block of control $i = 1$ has a length of $\delta_1 = 3\bar{\Delta}$, and its deadline is $db_{1,1} = d_{1,3} = 9$. The partition $P_1 = \{\{1,2,3\},\{4\}\}$ is *fiac feasible* for $i = 1$. For control $i = 3$, the partitions $P_3 = \{\{1,2,3\}\}, \{\{1,2\},\{3\}\}$ are, amongst others, *block deadline feasible.*

As illustrated in Example 6.1, a feasible solution $\boldsymbol{w}$ of (CIA$-\bar{\theta}-$init) may not use all possible activations. To this end, we define an extension of the set of blocks of $\boldsymbol{w}$ to become a partition of $[n_i]$ for all $i \in [n_\omega]$ in the following lemma. The extension may seem arbitrary, but it is necessary to compare any $\boldsymbol{w} \in \Omega_N$ with partitions of $[n_i]$. Thereby, we establish a connection between the above feasibility concepts and a feasible solution $\boldsymbol{w}$ of (CIA$-\bar{\theta}-$init).

Lemma 6.1
For an equidistant grid, let $\boldsymbol{w} \in \Omega_N$ be feasible for (CIA–$\bar{\theta}$–init) and let P'_i denote the set of blocks of $\boldsymbol{w}$ for control $i \in [n_\omega]$. We define $\overline{P'_i} := \{k \in [n_i] \mid \nexists\, B \in P'_i : k \in B\}$ and $P_i := P'_i \bigcup_{k \in \overline{P'_i}} \{k\}$. Then, P_i is a block deadline feasible partition and if $i = i_0$, P_i is also fiac feasible.

Proof. We first argue that P_i is by definition a partition of $[n_i]$. We need to prove that these partitions are *block deadline feasible*, respectively *fiac feasible*. If for $i \in [n_\omega]$ and an activation block $B \in P_i$ it holds that

$$r_{i,\max\{k \in B\}} > d_{i,\min\{k \in B\}} + |B| - 1,$$

it implies that the $(\max\{k \in B\})$th activation of i has been processed before its release interval because B cannot be interrupted by activations from other controls. Therefore, the above inequality does not hold and *block deadline feasibility* is established. We apply the same argument to confirm *fiac feasibility*. By constraint (6.16), the first activation of i_0 is scheduled on the first interval. Hence, all activations $k \in B_1$ of the first block B_1 must be processed on the kth interval and therefore require a release interval that is no later than k. □

Remark 6.9 (Necessary condition for feasibility of (CIA–$\bar{\boldsymbol{\theta}}$–init))
The formation of activations into *block deadline* and for i_0, into *fiac feasible* partitions is a *necessary* feasibility criterion of $\boldsymbol{w} \in \Omega_N$ for (CIA–$\bar{\theta}$–init) by virtue of Lemma 6.1. Nevertheless, it is not a *sufficient* criterion since the order of the processing of blocks is not clarified. In particular, one might order the blocks such that one block contains an activation whose release interval is later than its executed interval.

Next, we formalize specific partitions of the possible activations n_i of control i, whose blocks are constructed to include as many activations as possible without violating their block deadlines. These quantities serve as tools for deriving a lower bound of necessary blocks per control independent of the blocks of other controls. This results in a lower bound for (CIA–$\bar{\theta}$–init) in Proposition 6.2. We distinguish between the case in which i is the fixed initial active control, i.e., $i = i_0$, and that in which it is not.

Definition 6.25 ($\boldsymbol{P_{i,\min}}$, $\boldsymbol{P^{\text{init}}_{i,\min}}$)
Consider an equidistant grid and the controls $i_0, i \in [n_\omega]$. Let

$$k_1 := \max\{j \le n_i \mid d_{i,j} \le db_{i,1}\}, \qquad B_{i,1} := \{1, \dots, k_1\}, \tag{6.21}$$

$$k_1^{init} := \max\{j \le n_i \mid r_{i_0,j} = j\}, \qquad B_{i_0,1}^{init} := \left\{1, \dots, k_1^{init}\right\}. \tag{6.22}$$

We write $(\cdot)_i^{(init)}$ to indicate that equations or inequalities apply to both the parameters $(\cdot)_i$ and $(\cdot)_{i_0}^{init}$. We define the blocks $B_{i,l}^{(init)}$ recursively for $l \ge 2$ and while $k_l^{(init)} < n_i$ by

$$k_l^{(init)} := \max\left\{j \le n_i \mid d_{i,j} \le db_{i,k_{l-1}^{(init)}+1}\right\}, \quad B_{i,l}^{(init)} := \left\{k_{l-1}^{(init)} + 1, \dots, k_l^{(init)}\right\}. \tag{6.23}$$

Let $nb_{i,\min}^{(init)}$ denote the number of blocks $B_{i,l}^{(init)}$ and $P_{i,\min}^{(init)}$ the partitions of $[n_i]$ constructed by the latter:

$$P_{i,\min}^{(init)} := \left\{B_{i,l}^{(init)} \mid l \in \left[nb_{i,\min}^{(init)}\right]\right\}.$$

The MDR scheme creates switches on intervals that resemble the above $k_l^{(\text{init})}$ terms. The latter, however, only expresses the grouping of activations, whereas the switches also explicitly specify the corresponding intervals. It turns out that the partitions $P_{i,\min}$ and $P^{\text{init}}_{i_0,\min}$ are minimal in the number of blocks, as indicated in the following proposition.

Proposition 6.1 ($P_{i,\min}$ and $P^{\text{init}}_{i_0,\min}$ are minimal in the number of blocks)
For $i_0, i \in [n_\omega]$, let the partitions $P_{i,\min}$, $P^{init}_{i_0,\min}$ be given as in Definition 6.25. For any partition P_i of $[n_i]$, with $i = i_0$ included, we define its restriction to the first $\tilde{n}_i \leq n_i$ activations as

$$P_i|_{\tilde{n}_i} := \{B \cap [\tilde{n}_i] \mid B \in P_i\}.$$

Then for any $\tilde{n}_i \leq n_i$, the partition $P_{i,\min}$, respectively $P^{init}_{i_0,\min}$, consists of a minimal number of blocks on the first $\tilde{n}_i$ activations compared with all other block deadline feasible, respectively both block deadline and fiac feasible, partitions P_i:

$$\left| P^{(init)}_{i,\min}\Big|_{\tilde{n}_i} \right| \leq \left| P_i|_{\tilde{n}_i} \right|. \tag{6.24}$$

Proof. We consider first $P_{i,\min}\big|_{\tilde{n}_i}$. It is *block deadline feasible* because the deadline of the last activation for each block is defined in (6.21) and (6.23) to be less than or equal to the corresponding block deadline. Assume there is a *block deadline feasible* partition P_i for the control $i \in [n_\omega]$ with $\left|P_i|_{\tilde{n}_i}\right| < \left|P_{i,\min}\big|_{\tilde{n}_i}\right|$. In other words, there exists a subset of the first j blocks of $P_i|_{\tilde{n}_i}$ that includes more activations than the ones included into the first j blocks of $P_{i,\min}\big|_{\tilde{n}_i}$. We consider the minimal number of blocks j with this property:

$$j := \min\left\{ l \in [nb_{i,\min}] \mid B_{i,l} \in P_{i,\min}\big|_{\tilde{n}_i}, B'_{i,l} \in P_i|_{\tilde{n}_i} : \max\{k \in B_{i,l}\} < \max\{k \in B'_{i,l}\} \right\}. \tag{6.25}$$

The block index j is unique since the association of activations to blocks is monotonically increasing, meaning that there are no k_1th, k_2th activations, $k_1 < k_2$, with $k_1 \in B_{i,l_1}, k_2 \in B_{i,l_2}$, and $l_1 > l_2$. We conclude

$$\min\{k \in B'_{i,j}\} \leq \min\{k \in B_{i,j}\}, \quad B'_{i,j} \in P_i|_{\tilde{n}_i}, B_{i,j} \in P_{i,\min}\big|_{\tilde{n}_i}, \tag{6.26}$$

such that the first activation k' of block $B'_{i,j}$ is smaller than or equal to k, which marks the earliest activation of $B_{i,j}$. The definition of release intervals (6.18) implies $r_{i,k'} \leq r_{i,k}$ for $k' \leq k$. Similarly, the definition of block deadlines (6.20) implies $db_{i,k'} \leq db_{i,k}$ for $r_{i,k'} \leq r_{i,k}$, and in particular, with (6.26) we find

$$db_{i,\min\{k \in B'_{i,j}\}} \leq db_{i,\min\{k \in B_{i,j}\}}. \tag{6.27}$$

On the other hand, the definition of $P_{i,\min}$ in (6.23) implies

$$db_{i,k_{j-1}+1} = \max\{k \in B_{i,j}\}. \tag{6.28}$$

Then, the definition of j yields

$$db_{i,\min\{k \in B_{i,j}\}} \overset{(6.23)}{=} db_{i,k_{j-1}+1} \overset{(6.28)}{=} \max\{k \in B_{i,j}\} < \max\{k \in B'_{i,j}\} \leq db_{i,\min\{k \in B'_{i,j}\}}, \tag{6.29}$$

where the last inequality must hold due to the assumption that P_i is *block deadline feasible*. Inequality (6.27) contradicts inequality (6.29); equivalently, there are no such partitions P_i and $P_{i,\min}$ that indeed use a minimal number of blocks on any $[\tilde{n}_i] \subseteq [n_i]$.

For $j \geq 2$, the same argumentation in equation (6.25) can be applied in order to prove the result for $P^{\text{init}}_{i_0,\min}$ since P_{i_0} is also assumed to be *block deadline feasible* in this case; the same holds for $P^{\text{init}}_{i_0,\min}$ from the second block on. We still need to deal with the case when $j = 1$, i.e., if $B_{i_0,j}$, respectively $B'_{i_0,j}$, is the first block of control i_0. Here, $\max\{k \in B_{i_0,1}\} < \max\{k \in B'_{i_0,1}\}$ cannot occur since P_{i_0} is assumed to be *fiac feasible* and the construction of the first block of $P^{\text{init}}_{i_0,\min}$ implies that no further activation can be added to $B_{i_0,1}$ without violating *fiac feasibility*. Thus, $j = 1$ is impossible in (6.25), and $P^{\text{init}}_{i_0,\min}$ is also minimal in the number of blocks. □

Corollary 6.2 ($P_{i,\min}$ and $P^{\text{init}}_{i_0,\min}$ involve the minimal number of blocks until N)
Consider the setting of Proposition 6.1 and the controls $i_0, i \in [n_\omega]$. We define

$$\tilde{n}_{i,N} := \max\{k \mid d_{i,k} \leq N\}, \quad nb^N_{i,\min} := \left| P_{i,\min}\big|_{\tilde{n}_{i,N}} \right|, \quad nb^{N,init}_{i_0,\min} := \left| P^{init}_{i_0,\min}\Big|_{\tilde{n}_{i_0,N}} \right|.$$

There is no block deadline feasible partition, respectively block deadline and fiac feasible partition, that uses less than $nb^N_{i,\min}$ blocks on $[\tilde{n}_{i,N}]$, respectively $nb^{N,init}_{i_0,\min}$ blocks on $\left[\tilde{n}_{i_0,N}\right]$.

Proof. The result follows directly from Proposition 6.1 with $\tilde{n}_i = \tilde{n}_{i,N}$ and $\tilde{n}_{i_0} = \tilde{n}_{i_0,N}$. □

As final result for this section, we establish a lower bound for (CIA−$\bar{\theta}$−init) that will be useful in Theorem 6.2.

Proposition 6.2 (Lower bound for (CIA−$\bar{\theta}$−init))
Let $\sigma^\star$ be the objective of (CIA−$\bar{\theta}$−init) with equidistant discretization and i_0 be the fixed initial active control, as defined in Definition 6.21. Let $nb^N_{i,\min}$ for all $i \neq i_0$ and $nb^{N,init}_{i_0,\min}$ be given as in Corollary 6.2. This gives the result:

$$\sum_{i \in [n_\omega], i \neq i_0} nb^N_{i,\min} + nb^{N,init}_{i_0,\min} - 1 \leq \sigma^\star. \tag{6.30}$$

Proof. By virtue of Lemma 6.1, a feasible solution of (CIA−$\bar{\theta}$−init) satisfies the *necessary* condition of generating only *block deadline feasible* partitions P_i, and if $i = i_0$, the activation partition P_i is also *fiac feasible*. Moreover, all activations are executed no later than their deadline interval. In particular, this holds for those activations that are due no later than N. Hence, we can apply Corollary 6.2 and thus conclude that the minimum number of blocks of a feasible solution until the Nth activation is $nb^N_{i,\min}$, respectively $nb^{N,\text{init}}_{i_0,\min}$. Finally, we obtain the claim (6.30) by summing over all controls and using the fact that the setup of the first block does not count as a switch. □

6.7.3 Solution quality and properties of MDR and AMDR

Although the MDR algorithm may seem simple, it generates optimal solutions $\boldsymbol{w}^{\text{AMDR}}$ for (CIA−$\bar{\theta}$−init) under certain conditions, for which we need the following definition.

Definition 6.26 (Canonical switch)
Consider a given $\bar{\theta} > 0$. We define a switch j to be canonical if on interval τ_j the following holds: exactly one control i_1 is $\bar{\theta}$-inadmissible, and exactly one control $i_2 \neq i_1$ is $\bar{\theta}$-forced.

We build the theoretical results in this section primarily on the following assumption.

Assumption 6.1 (MDR uses only canonical switches)
Suppose $\boldsymbol{w}^{\text{MDR}} \in \Omega_N$ has been generated by MDR. We assume that all switches of $\boldsymbol{w}^{\text{MDR}}$ are *canonical.*

Properties of the MDR algorithm

Assumption 6.1 may seem restrictive, although it is satisfied under certain conditions. The following lemma is not only useful for Proposition 6.3 but is also required for Lemma 7.9 on page 127.

Lemma 6.2 (Properties of Control Accumulation $\boldsymbol{\gamma}$ and $\boldsymbol{\theta}$)
Consider $\boldsymbol{a}^* \in \mathscr{A}_N$ and $\boldsymbol{w} \in \Omega_N$. For each $j \in [N]$ it holds that

$$\sum_{i\in[n_\omega]} \theta_{i,j} = 0, \qquad \sum_{i\in[n_\omega]} \gamma_{i,j} = \Delta_j.$$

Proof. These equations follow directly from Definition 6.16 of $\boldsymbol{\theta}$ and $\boldsymbol{\gamma}$ as well as from the convexity property of $\boldsymbol{a}^*$ and $\boldsymbol{w}$. □

Proposition 6.3 (MDR with $n_\omega = 2$ and $\bar{\theta} \geq \frac{1}{2}\bar{\Delta}$ uses canonical switches)
Consider $n_\omega = 2$, $\boldsymbol{a}^* \in \mathscr{A}_N$, and any grid $\mathscr{G}_N$. If we choose $\bar{\theta} \geq \frac{1}{2}\bar{\Delta}$, then the control function $\boldsymbol{w}^{MDR}$ constructed by the MDR scheme uses only canonical switches.

Proof. We have to prove

1. If control i_1 is $\bar{\theta}$-*forced* on interval $j \geq 2$, then it is $\bar{\theta}$-admissible.
2. For all intervals $j \geq 2$, control i_1 is $\bar{\theta}$-*inadmissible* if and only if $i_2 \neq i_1$ is $\bar{\theta}$-*forced.*

The first statement follows from the definition of $\bar{\theta}$-*forced* activation and from $\bar{\theta} \geq \frac{1}{2}\bar{\Delta}$:

$$\theta_{i_1,j} = \theta_{i_1,j-1} + a^*_{i_1,j}\Delta_j - \Delta_j > \bar{\theta} - \Delta_j \geq -\frac{1}{2}\bar{\Delta} \geq -\bar{\theta}.$$

To prove the second statement, assume i_1 is $\bar{\theta}$-forced on $j \in [N]$, i.e., $\gamma_{i_1,j} > \bar{\theta}$. By virtue of Lemma 6.2 for $\gamma_{i_2,j}$, we derive

$$\theta_{i_2,j-1} + a^*_{i_2,j}\Delta_j = \gamma_{i_2,j} < -\bar{\theta} + \Delta_j,$$

which means that i_2 is $\bar{\theta}$-*inadmissible* on j. Conversely, if i_1 is $\bar{\theta}$-*inadmissible* on j, we conclude from $\theta_{i_1,j-1} + (a^*_{i_1,j} - 1)\Delta_j < -\bar{\theta}$ and from the equation for $\boldsymbol{\theta}$ in Lemma 6.2 that $\gamma_{i_2,j} = \theta_{i_2,j-1} + a^*_{i_2,j}\Delta_j > \bar{\theta}$. Therefore, i_2 is $\bar{\theta}$-*forced.* □

Remark 6.10
Assumption 6.1 is not necessarily true for a control problem that involves more than two binary controls. It may, however, hold for special cases of such problems. For instance, if the relaxed values are of the bang-bang type and $\bar{\theta}$ is chosen to be smaller than the smallest activation block, then the situation resembles *the case n_ω* = 2, and Assumption 6.1 may hold (without proof). On the other hand, Example 6.3 demonstrates that this assumption can indeed be quite restrictive.

Assumption 6.1 allows us to favorably prove properties of control functions obtained by MDR and AMDR. The first result demonstrates that the MDR scheme indeed produces control functions that exhibit an integral deviation gap smaller than or equal to $\bar{\theta}$.

Lemma 6.3 (MDR solution satisfies $\bar{\boldsymbol{\theta}}$ bound)
Let Assumption 6.1 hold, and let $\boldsymbol{w}^{MDR} \in \Omega_N$ be constructed by MDR with given threshold $\bar{\theta}$. Then, we obtain $\theta(\boldsymbol{w}^{MDR}) \leq \bar{\theta}$.

Proof. As soon as the activated control becomes $\bar{\theta}$-inadmissible or there is a $\bar{\theta}$-forced control on interval $j \geq 2$, by the definition of MDR, $\boldsymbol{w}^{\mathrm{MDR}}$ has a switch. By Assumption 6.1, the newly activated control is both $\bar{\theta}$-forced and $\bar{\theta}$-admissible. Hence, $\theta_{i,j} \geq -\bar{\theta}$, and there is no other $\bar{\theta}$-forced control on j; thus, $\theta_{i,j} \leq \bar{\theta}$. □

The following example demonstrates that $\theta(\boldsymbol{w}^{\mathrm{MDR}}) > \bar{\theta}$ may generally appear without Assumption 6.1.

Example 6.2 (Counterexample for Assumption 6.1)
Consider an equidistant discretization and $\boldsymbol{a}^* \in \mathscr{A}_N$ with the first values given as $a^*_{1,1} = 1$, $a^*_{2,1} = 0$, $a^*_{1,2} = 0.5$, $a^*_{2,2} = 0.5$. For this relaxed value, let $\boldsymbol{w}^{\mathrm{MDR}} \in \Omega_N$ be the corresponding binary control function computed by MDR with given threshold $\bar{\theta} = 0.4\bar{\Delta}$ and initial control $i = 1$. Then, $w^{\mathrm{MDR}}_{1,2} = 0$, $w^{\mathrm{MDR}}_{2,2} = 1$ holds since the second control becomes $\bar{\theta}$-forced on the second interval. At the same time, control $i = 2$ is $\bar{\theta}$-inadmissible on the second interval; hence Assumption 6.1 is violated, and it results $\theta_{2,2} = -0.6\bar{\Delta} < -\bar{\theta}$.

The following theorem states that under certain assumptions MDR constructs a binary control that uses a minimum number of switches. We reuse concepts from the previous subsection, especially *activations* and their grouping into *blocks*, and build the proof upon Proposition 6.2.

Theorem 6.2 (Least switches property of MDR)
Let Assumption 6.1 hold. For given $\boldsymbol{a}^* \in \mathscr{A}_N$ and an equidistant grid, let $\boldsymbol{w}^{MDR}$ be constructed by MDR with i as initial control and any $\bar{\theta} > 0$, where we assume that i is initially admissible. Let $\sigma(\boldsymbol{w}^{MDR})$ denote the number of switches used by $\boldsymbol{w}^{MDR}$. Then, for the optimal objective value σ^* of (CIA$-\bar{\theta}-$init) with $i_0 = i$ as initial control, it holds that

$$\sigma^* = \sigma\left(\boldsymbol{w}^{MDR}\right). \tag{6.31}$$

Proof. We can conclude $\boldsymbol{w}^{\mathrm{MDR}}$ is a feasible solution of (CIA$-\bar{\theta}-$init) by Lemma 6.3. Combining this with Proposition 6.2 yields

$$\sum_{i\in[n_\omega], i\neq i_0} nb^N_{i,\min} + nb^{N,\mathrm{init}}_{i_0,\min} - 1 \leq \sigma^* \leq \sigma\left(\boldsymbol{w}^{\mathrm{MDR}}\right). \tag{6.32}$$

To prove optimality, we note that $\boldsymbol{w}^{\mathrm{MDR}}$ constructs partitions of the activations $[n_i], i \in [n_\omega]$, that are due no later than N and denote these partitions by P^{MDR}_i. Using the notation from Corollary 6.2, we want to show that these partitions coincide with the partitions constructed in Definition 6.25:

$$P^{\mathrm{MDR}}_{i_0} = P^{\mathrm{init}}_{i_0,\min}\Big|_{\tilde{n}_{i_0,N}}, \quad P^{\mathrm{MDR}}_i = P_{i,\min}\big|_{\tilde{n}_{i,N}}, \text{ for } i \neq i_0 \tag{6.33}$$

because Corollary 6.2 would then imply that $\boldsymbol{w}^{\mathrm{MDR}}$ has $\sum_{i\in[n_\omega],i\neq i_0} nb^N_{i,\min} + nb^{N,\mathrm{init}}_{i_0,\min}$ activation blocks or equivalently that

$$\sigma\left(\boldsymbol{w}^{\mathrm{MDR}}\right) = \sum_{i\in[n_\omega],i\neq i_0} nb^N_{i,\min} + nb^{N,\mathrm{init}}_{i_0,\min} - 1.$$

The claim follows from inequality (6.32). Consider the first blocks $B_1 \in P^{\mathrm{MDR}}_{i_0}$ and $B^{\mathrm{init}}_1 \in P^{\mathrm{init}}_{i_0,\min}$. By Assumption 6.1, the MDR algorithm activates i_0 until it becomes $\bar{\theta}$-*inadmissible* on interval τ_1 of the first switch:

$$\theta_{i_0,\tau_1} = \sum_{l=1}^{\tau_1} a_{i_0,l} - w_{i_0,l} < -\bar{\theta}/\bar{\Delta}.$$

We compare this inequality with Definition 6.22 of release intervals and note that either the next activation τ_1 of control i_0 has a release interval that is later than τ_1 or there is no further possible activation. Thus, if the τ_1th activation exists, then its release interval has not yet been reached:

$$r_{i_0,\tau_1} = r_{i_0,\max\{k\in B_1\}+1} > \max\{k \in B_1\} + 1.$$

By the definition of $P^{\mathrm{init}}_{i_0,\min}$, we conclude that $B_1 = B^{\mathrm{init}}_1$. We now consider the jth blocks $B_j \in P^{\mathrm{MDR}}_{i_0}$ and $B^{\mathrm{init}}_j \in P^{\mathrm{init}}_{i_0,\min}\Big|_{\tilde{n}_{i_0,N}}$, where $j \geq 2$. Again by the definition of MDR and Assumption 6.1, i_0 is $\bar{\theta}$-*forced* on interval τ_j, which is equivalent to $d_{i_0,\min\{k\in B_j\}} = \tau_j$. The MDR scheme activates i_0 either until N (at which point $B_j = B^{\mathrm{init}}_j$ trivially) or until it becomes $\bar{\theta}$-*inadmissible* on interval τ_{j+1} (by Assumption 6.1). With the argument for $j = 1$, $\bar{\theta}$-inadmissible means hereby the $(\max\{k \in B_j\} + 1)$th activation has a release interval greater than τ_{j+1}. Because the first activation τ_j of the block is processed on its deadline interval $d_{i_0,\min\{k\in B_j\}}$, we get

$$r_{i_0,\max\{k\in B_j\}+1} > d_{i_0,\min\{k\in B_j\}} + |B_j| - 1.$$

The above inequality demonstrates that B_j contains as many activations as possible without violating its block deadline $db_{i_0,\min\{k\in B_j\}}$ and by construction of $P^{\mathrm{init}}_{i_0,\min}$, this is equivalent to $B_j = B^{\mathrm{init}}_j$. This settles the case $i = i_0$ in (6.33). We can reuse the above arguments about $\bar{\theta}$-*forced* and $\bar{\theta}$-*inadmissible* activation for $j \geq 2$ in order to analogously prove the case $i \neq i_0$ in (6.33). □

Remark 6.11 (Equidistant discretization is critical)
Theorem 6.2 is predicated on the assumption of an equidistant grid. We stress that after grid refinement of the MIOCP, i.e., after several rounds of applying the CIA decomposition, this might be a restriction.

The following corollary establishes a way to find the optimum of (CIA$-\bar{\theta}-$init) in the setting of Theorem 6.2.

Corollary 6.3 (Using MDR to find a control function with the minimum number of switches)
Consider the setting of Theorem 6.2. A control function $\boldsymbol{w}^*$ that uses a minimum number of switches, i.e., $\sigma(\boldsymbol{w}^*) = \sigma^*$, can be found by running MDR.

Proof. Let i be the initial control of $\boldsymbol{w}^*$. Execute MDR with i as the initial control so that the result follows directly from Theorem 6.2. □

It is not clear which initial active control is optimal for minimizing the number of switches. In practice, MDR must be executed for all controls $i \in [n_\omega]$, one after the other, as the initial active control. This is expressed by the following corollary.

Corollary 6.4 (Link between MDR and (CIA–$\bar{\theta}$))
Consider the setting of Theorem 6.2. We assume that for all $i \in [n_\omega]$ as initial active controls the MDR algorithm constructs the control functions $\boldsymbol{w}^{MDR}$ that use only canonical switches. Then, there is a minimizing control $\boldsymbol{w}^* \in \Omega_N$ for (CIA–$\bar{\theta}$) that only uses canonical switches. Moreover, there exists $i_0 \in [n_\omega]$ such that running MDR with i_0 as the initial control produces $\boldsymbol{w}^{MDR} \in \Omega_N$ that minimizes (CIA–$\bar{\theta}$).

Proof. If the MDR algorithm produces $\boldsymbol{w}^{\text{MDR}}$, which only uses canonical switches, $\boldsymbol{w}^{\text{MDR}}$ is optimal by Theorem 6.2 for (CIA–$\bar{\theta}$–init) with the corresponding initial control fixed. Then, the result follows from the fact that the optimal solution of (CIA–$\bar{\theta}$) is contained in the set of optimal solutions for the set of problems (CIA–$\bar{\theta}$–init) with each control $i \in [n_\omega]$ initially fixed. □

Lemma 6.4 (Implication of $\theta(\boldsymbol{w}^{\text{MDR}}) > \bar{\theta}$ for $n_\omega = 2$)
Consider $\boldsymbol{a}^* \in \mathscr{A}_N$ on an equidistant grid, and assume $n_\omega = 2$. Let $\boldsymbol{w}^{MDR}$ denote the control function constructed by MDR with $\bar{\theta} > 0$ and given initial control. If $\theta(\boldsymbol{w}^{MDR}) > \bar{\theta}$, then there is no control function $\boldsymbol{w} \in \Omega_N$ with the same initial active control and $\theta(\boldsymbol{w}) \leq \bar{\theta}$.

Proof. We consider the first interval j on which the accumulated control deviation of $\boldsymbol{w}^{\text{MDR}}$ is greater than $\bar{\theta}$. Let control i_1 be active on j. By definition of the MDR scheme, $|\theta_{i_1,j}| > \bar{\theta}$ or $|\theta_{i_2,j}| > \bar{\theta}$ can only appear if there is a switch on interval j and

1. i_1 is both $\bar{\theta}$-*forced* and $\bar{\theta}$-*inadmissible* on interval j, or
2. both i_1 and i_2 are $\bar{\theta}$-*inadmissible* on interval j.

Proposition 6.3 establishes that $\boldsymbol{w}^{\text{MDR}}$ uses only canonical switches for $\bar{\theta} \geq \frac{1}{2}\bar{\Delta}$, and thus, the above cases *cannot* appear for $\bar{\theta} \geq \frac{1}{2}\bar{\Delta}$. Let us focus on $\bar{\theta} < \frac{1}{2}\bar{\Delta}$. In order to create a control function $\boldsymbol{w}$ that fulfills $\theta(\boldsymbol{w}) \leq \bar{\theta}$, we need to change at least one activation of $\boldsymbol{w}^{\text{MDR}}$ on an earlier interval $l < j$. However, we recognize that any earlier change of activation is not possible:

- We cannot extend an activation block at its end since the active control is $\bar{\theta}$-*inadmissible*.
- If the active control i_1 is $\bar{\theta}$-*admissible* on l, then the other control i_2 is not $\bar{\theta}$-*forced* on l – otherwise it would be active in the MDR scheme. This means that $\theta_{i_2,l-1} + a_{i_2,l}\bar{\Delta} \leq \bar{\theta}$. Activating i_2 on l results in

  $$\theta_{i_2,l} = \theta_{i_2,l-1} + (a_{i_2,l} - 1)\bar{\Delta} \leq \bar{\theta} - \bar{\Delta} < -\frac{1}{2}\bar{\Delta} < -\bar{\theta},$$

  where we applied $\bar{\theta} < \frac{1}{2}\bar{\Delta}$. This indicates that the integral deviation gap of $\boldsymbol{w}$ is again greater than $\bar{\theta}$.

Hence, no previous activation $\boldsymbol{w}^{\text{MDR}}$ can be changed such that there is no $\boldsymbol{w}$ with $\theta(\boldsymbol{w}) \leq \bar{\theta}$. □

Properties of the AMDR algorithm

Theorem 6.3 states that the AMDR algorithm can find the optimal solution of (CIA-TV) for $n_\omega = 2$ and equidistant discretization. Otherwise, strict assumptions are required for optimality, and in general, the feasible solution that is found represents only a promising upper bound.

Theorem 6.3 (Properties of AMDR)
AMDR terminates for given $\boldsymbol{a}^* \in \mathscr{A}_N$, $TOL > 0$, and $\sigma_{\max} \in \mathbb{N}$ after a finite number of iterations. Furthermore, consider an equidistant grid $\mathscr{G}_N$. Let $\boldsymbol{w}^{AMDR}$ denote the solution constructed by AMDR. It follows that

1. *$\boldsymbol{w}^{AMDR}$ is a feasible solution of (CIA-TV).*
2. *a) If $n_\omega = 2$, for the optimum θ^* of (CIA-TV), we have $\theta(\boldsymbol{w}^{AMDR}) \leq \theta^* + TOL$.*

   b) Let $n_\omega > 2$. We assume that in every run the AMDR scheme uses only canonical switches. Furthermore, suppose we have the following: If the MDR scheme constructs a solution with $\theta(\boldsymbol{w}^{MDR}) > \bar{\theta}$, then there is no control function $\boldsymbol{w} \in \Omega_N$ with the same initial active control and $\theta(\boldsymbol{w}) \leq \bar{\theta}$. With these assumptions, for the optimum θ^* of (CIA-TV), we obtain $\theta(\boldsymbol{w}^{AMDR}) \leq \theta^* + TOL$.
3. *AMDR has time complexity $\mathscr{O}(n_\omega \cdot C_{MDR} \cdot \log_2(\lceil (t_f - t_0)/TOL \rceil))$, where $C_{MDR} \in \mathscr{O}(N)$ denotes the time complexity of the MDR scheme.*

Proof. AMDR is a bisection algorithm that either decreases UB (lines 7 - 8) or increases LB (lines 12 - 13) by at least one half of $(UB - LB)$ in every `while` loop iteration (line 2). Because of this and $TOL > 0$, we conclude that the `while` loop and AMDR as a whole terminate after finitely many iterations.

1. The objective of (CIA-TV) cannot be greater than $t_f - t_0$, even when no switches are allowed, i.e., $\sigma_{\max} = 0$. Since we initialize the AMDR algorithm with $UB = t_f - t_0$, it finds a feasible solution in any case.
2. Every $\boldsymbol{w}^{\mathrm{MDR}}$ generated by MDR in line 5 that satisfies the TV constraints together with $\theta(\boldsymbol{w}^{\mathrm{MDR}}) < UB$ represents an upper bound on θ^*, i.e., $UB = \theta(\boldsymbol{w}^{\mathrm{MDR}}) \geq \theta^*$. To prove that AMDR constructs valid lower bounds LB on θ^*, we exploit that $\boldsymbol{w}^{\mathrm{MDR}}$ only uses canonical switches for $n_\omega = 2$ and by assumption for $n_\omega > 2$, making Corollary 6.4 applicable. For given $\bar{\theta}$, if there is no initial control $i \in [n_\omega]$ for which MDR produces $\boldsymbol{w}^{\mathrm{MDR}}$ that uses fewer or an equal number of switches as required by the TV constraints, we conclude that there exists no such $\boldsymbol{w} \in \Omega_N$ for this specific threshold $\bar{\theta}$ by Corollary 6.4. Therefore, $LB = \bar{\theta} \leq \theta^\star$ is a true lower bound on the optimal (CIA-TV) objective value. Moreover, if for a given $\bar{\theta}$ and all initial controls $i \in [n_\omega]$, MDR constructs control functions $\boldsymbol{w}^{\mathrm{MDR}}$ with $\theta(\boldsymbol{w}^{\mathrm{MDR}}) > \bar{\theta}$, Lemma 6.4 and the assumption in (b) guarantee that this $\bar{\theta}$ is also a true lower bound on the optimal (CIA-TV) objective value. Altogether, AMDR iteratively generates valid lower bounds LB and upper bounds UB for θ^* and produces a feasible solution that is optimal up to the chosen tolerance TOL.
3. MDR runs forward in time and computes solely the accumulated control deviation γ and θ for all intervals $j \in [N]$; therefore, $C_{\mathrm{MDR}} \in \mathscr{O}(N)$. The interval halving in AMDR ensures that we execute the `while` loop a maximum of $\log_2((t_f - t_0)/TOL)$ times. Inside this loop, in the worst case, we need to run the MDR scheme with all n_ω controls as initial controls. Combining these findings yields the asserted complexity. □

Remark 6.12 (AMDR modifications)
Several meaningful modifications are available for the AMDR algorithm. We may also use it to find control functions that fulfill other combinatorial constraints, such as MDT constraints, by checking them together with the TV constraint in line 5. As part of the MDR scheme, the control with maximum forward control deviation γ is activated if the previously active control is $\bar{\theta}$-*inadmissible*. One might choose a less greedy variant instead. For instance, we could activate the $\bar{\theta}$-*next-forced* and $\bar{\theta}$-*admissible* control. Lastly, the initial upper bound UB can be reduced, as we point out in Section 7.5.

Remark 6.13 (AMDR viewed as a generalization of SUR)
If we drop the TV constraint on $\boldsymbol{w}$, the AMDR scheme finds a control function with the same objective value as that obtained by the control function of SUR (without proof).

Most of the results in this section are based on the assumption of an equidistant discretization, which is common in practice. The assumption of dealing only with *canonical* switches in the produced control function is critical. The following example demonstrates that a control function generated by MDR with non-canonical switches may use more switches than needed or may not satisfy the rounding bound $\bar{\theta}$.

Example 6.3 (Possible MDR outcome without Assumption 6.1)
Consider an equidistant grid. Let the two relaxed values $\boldsymbol{a}^1, \boldsymbol{a}^2 \in \mathscr{A}_N$ be defined as

$$\left(a^1_{i,j}\right)_{i\in[3],j\in[3]} := \begin{pmatrix} 1 & 0.25 & 0 \\ 0 & 0.375 & 0.5 \\ 0 & 0.375 & 0.5 \end{pmatrix}, \qquad \left(a^2_{i,j}\right)_{i\in[3],j\in[3]} := \begin{pmatrix} 1 & 0.2 & 0 \\ 0 & 0.4+\epsilon & 0 \\ 0 & 0.4-\epsilon & 1 \end{pmatrix}, \qquad 0<\epsilon<0.4.$$

Then, MDR with $i = 1$ as the initial control and $\bar{\theta}^1 = 0.75\bar{\Delta}$, respectively $\bar{\theta}^2 = (0.6+\epsilon)\bar{\Delta}$, constructs the following control functions:

$$\left(w^{\text{MDR},1}_{i,j}\right)_{i\in[3],j\in[3]} := \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \left(w^{\text{MDR},2}_{i,j}\right)_{i\in[3],j\in[3]} := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

In the first example, two controls are simultaneously $\bar{\theta}$-*forced* on the third interval; the (CIA-TV) objective value would be smaller if $w_{3,2} = 1$ was chosen. In the second example, the MDR constructs a control function that uses two switches, although activating the third control on the second interval would result in only one switch with almost the same (CIA-TV) objective value. The examples have the use of *non-canonical* switches in common. Hence, the improved control functions would be

$$\left(w^{\text{OPT},1}_{i,j}\right)_{i\in[3],j\in[3]} := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \qquad \left(w^{\text{OPT},2}_{i,j}\right)_{i\in[3],j\in[3]} := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}.$$

6.8 Other approaches

A common approach to solving MILPs that has not been mentioned in this chapter is Branch-and-Cut. This method investigates the polyhedral structure of the constraints of the MILP.

JUNG used the software Porta [58] to find the describing facets of the polytope that corresponds to the feasible set of (CIA) [135]. However, it turns out that the facets to the approximation inequality constraints (4.4) contain many non-zero coefficients and are abundant. Therefore, JUNG concluded “we assume that there is no good cutting plane approach for this problem class” (page 122 in [135]).

Recently, BESTEHORN et al. [28] proposed a shortest-path approach for solving (CIA−$\bar{\theta}$), using a generalized objective function that aims to minimize a switching costs function F_C over $\boldsymbol{w} \in \Omega_N$:

$$F_C(\boldsymbol{w}) := F_c^s(\boldsymbol{w}_{\cdot,1}) + \sum_{j=2}^{N} F_c^j(\boldsymbol{w}_{\cdot,j-1}, \boldsymbol{w}_{\cdot,j}) + F_c^e(\boldsymbol{w}_{\cdot,N}),$$

where F_c^s, F_c^e, and F_c^j denote the initial, terminal, and switching costs, respectively. This objective can also be used to minimize the total number of switches by setting $F_c^j = 1$ and $F_c^s = F_c^e = 0$. The authors of [28] derive a parameterized directed acyclic graph that represents the feasible solutions of the altered (CIA−$\bar{\theta}$) problem. The graph is constructed with the following properties:

- The vertices represent different activations of control modes in $\boldsymbol{w}$.
- Each vertex of the jth layer of the graph, with $j \in [N]$, indicates a feasible combination of fixed activations in $\boldsymbol{w}$ up to and including the jth interval.
- There are only directed arcs from nodes of the jth layer to nodes of the $(j+1)$st layer.
- An arc exists between two nodes $\boldsymbol{w}_1, \boldsymbol{w}_2$ if and only if $\boldsymbol{w}_2$ equals $\boldsymbol{w}_1$ up to and including the jth interval and the realization of $\boldsymbol{w}_2$ on the $(j+1)$st interval is feasible with respect to the approximation inequality constraint (6.15), i.e., $\theta_{i,j+1} \leq \bar{\theta}$ for all $i \in [n_\omega]$.
- Apart from the initial and terminal costs, the costs of a path in the graph are associated with arcs that connect vertices with different control mode activations on intervals j and $(j+1)$.

Assuming an equidistant grid, one can exploit that for any control mode $i \in [n_\omega]$, and for any layer, respectively interval, the order of the activations on previous intervals is irrelevant for checking the feasibility of outgoing arcs to the next layer because only the number of intervals on which i is active is necessary for evaluating the approximation inequality constraint (6.15). By this assumption and observation, the size of the graph can be reduced considerably since nodes with the same number of activations per control mode can be aggregated to an equivalent node in the layer. Only the following information the jth layer needs to be stored in the nodes themselves:

- its predecessor node,
- a label vector that includes the number of activations $\sum_{l\in[j]} w_{i,l}$ per mode i up to and including the jth interval, and
- the associated costs.

With these ideas at hand, a shortest path algorithm similar to Dijkstra’s algorithm can be applied to on the graph. The numerical results reported in [28] are promising with respect to the run time. Moreover, it is mentioned that the graph can model MDTs.

6.9 Summary

This chapter has introduced a variety of algorithms that solve (CIA) under different restrictions and situations. The complexity reduction heuristic 6.1 is distinct in that it does not solve (CIA), but reduces the problem size. We use Table 6.1 to classify the algorithms presented in this chapter in terms of their applicability. The algorithms are listed in ascending order with respect to their anticipated run time, e.g., SUR should (heuristically) solve a (CIA) problem faster than NFR does.

For the sake of completeness, as an algorithmic option, we also list standard MILP solvers, which are usually based on BnB methods but are not tailor-made for (CIA) problems like the BnB methods from Sections 6.3 and 6.4. An MILP model can generally include a large class of constraints, making MILP solvers a powerful tool for testing new constraint models. However, because of the shorter computation times of custom-made algorithms [135, 50], they are recommended over MILP solvers when possible.

SUR is widely applied to heuristically solve the unconstrained (CIA). Its success is due to its speed (run time complexity $\mathcal{O}(N)$) and robust approximation properties. For the case $n_\omega = 2$ and equidistant discretization, it even computes optimal solutions. If an optimal solution on an equidistant grid is desired, a bisection using NFR with an adaptive rounding threshold should be applied; it performs in polynomial time, see Corollary 6.1. On a general grid, the BnB algorithms, which are often still very fast (with an order of magnitude of 10^{-2} seconds for medium-size problems [50]), compute optimal solutions. We group the algorithms BnB and STO-BnB together since their run times have not yet been intensively compared. STO-BnB is advantageous for problems with few switches, whereas BnB is preferable for a large number of switches. The shortest path approach involves a lower anticipated run time than the BnB algorithms because it was designed specifically for equidistant problems.

For (CIA) problems with MDT constraints, DNFR and DSUR are the methods of choice if computational speed is the top priority, e.g., in the context of nonlinear model predictive control (MPC). AMDR is also a heuristic algorithm but constructs a binary control solution in polynomial time, too. For the general (CIA-UD) problem, BnB and STO-BnB compute optimal solutions in reasonable times and are therefore recommended.

The situation with TV constraints is similar to that with MDT constraints, though the AMDR algorithm is custom-made for the (CIA-TV) problem class and constructs optimal solutions for $n_\omega = 2$ and equidistant discretization. Thus, AMDR represents a valid alternative to BnB and STO-BnB for large problem sizes.

We list the shortest path approach with a question mark for (CIA) problems with general combinatorial constraints since the constraints that it can handle are not discussed in detail in [28]. We assume that it is a beneficial algorithm for the equidistant case. The BnB algorithm implementation of `pycombina` (see Chapter 8) accommodates a range of combinatorial constraints beyond MDT and TV constraints and is therefore recommended for the general (multiple) constrained case.

To construct binary controls that incorporate further data from (NLP$_{\text{rel}}$), such as the evaluated model function $\tilde{\boldsymbol{f}}$ or the cost-to-go $\tilde{\boldsymbol{\lambda}}$, as suggested in Section 4.5.1, so far only MILP models are available, apart from the heuristic rounding schemes $\mathscr{H}$-SUR, $\mathscr{H}$-Rounding, and Scaled-SUR. We also use the MILP model to test the first-order Taylor path constraint approximation approach from Definition 4.14 since it provides high flexibility for further constraints.

Problem type	Algorithms
(CIA)	SUR‡, NFR*, Bisection-NFR†, Shortest path†, BnB/ STO-BnB, MILP solver
(CIA-U), (CIA-D), (CIA-UD)	DNFR*, DSUR*, AMDR*, Shortest path†, BnB/ STO-BnB, MILP solver
(CIA-TV)	AMDR‡, Shortest path†, BnB/ STO-BnB, MILP solver
(CIA) with multiple combinatorial constraints	Shortest path†?, BnB/ STO-BnB, MILP solver
Including further data from (NLP$_{\text{rel}}$) into (CIA)	$\mathscr{H}$-SUR*, Scaled-SUR*, MILP solver

Table 6.1: List of (CIA) rounding problems and the corresponding applicable algorithms. The problems and their features are marked in dark-blue, and algorithms are highlighted in burgundy red. We distinguish between the unconstrained problem (CIA); the minimum dwell time-constrained problems (CIA-U), (CIA-D), (CIA-UD); and the total variation constrained problem (CIA-TV) in the first, second, and third rows, respectively. The fourth row includes algorithms that can deal with multiple combinatorial constraints, including minimum dwell time and total variation constraints. In the last row, we list algorithms designed to incorporate further information from (NLP$_{\text{rel}}$), such as data with respect to the path constraints or the evaluated model function $\tilde{\boldsymbol{f}}$, into the rounding problem. Bisection-NFR refers to Remark 6.1 and Corollary 6.1, where we pointed out that (CIA) can be solved via bisection and NFR. The shortest path approach refers to the one proposed by BESTEHORN et al., discussed in Section 6.8. By MILP solver, we refer to a black-box solver program such as `Gurobi`. The algorithms are listed in ascending order with respect to their anticipated run time. Heuristic algorithms are marked with *, optimal algorithms that work only for equidistant grids are marked with †, and algorithms that construct optimal solutions for $n_{\omega} = 2$ and equidistant grids are marked with ‡; all other algorithms construct optimal solutions.

Chapter 7

Approximation results for the integral deviation gap

This chapter investigates approximation bounds of the binary controls constructed by the new algorithms introduced in Chapter 6. For this, we return to the *integral deviation gap* $\theta(\boldsymbol{w})$ for $\boldsymbol{w} \in \Omega_N$ as defined in Definition 4.5. The aim of (CIA) is to minimize the integral deviation gap, i.e., $\min_{\boldsymbol{w}\in\Omega_N} \theta(\boldsymbol{w})$. Since the lower bound on the integral deviation gap is trivially zero (and can be reached), when we refer to bounds in this chapter, we always mean upper bounds. Thus, we investigate how large the objective function value θ^* of (CIA) and its combinatorial constraint variants can become. In other words, we examine

$$\theta^{\max} := \max_{\boldsymbol{a}\in\mathscr{A}_N} \min_{\boldsymbol{w}\in\Omega_N} \max_{i\in[n_\omega], j\in[N]} |\,\theta_{i,j}\,| \quad \text{s.t.} \quad \text{combinatorial constraints.} \tag{7.1}$$

We observe the trivial upper bound

$$\theta^{\max} \leq \sum_{j\in[N]} \Delta_j = t_f - t_0$$

and another weak result in the following remark.

Remark 7.1 (Upper bound for $\boldsymbol{\theta^{\max}}$ from Minimax theory)
Neglecting combinatorial constraints for a moment, it is known from Minimax theory [268] that

$$\max_{\boldsymbol{a}\in\mathscr{A}_N} \min_{\boldsymbol{w}\in\Omega_N} \max_{i\in[n_\omega], j\in[N]} |\theta_{i,j}| \;\leq\; \min_{\boldsymbol{w}\in\Omega_N} \max_{\boldsymbol{a}\in\mathscr{A}_N} \max_{i\in[n_\omega], j\in[N]} |\theta_{i,j}|$$

holds. On the right side of the inequality, we maximize over $\boldsymbol{a}$ for given $\boldsymbol{w}$ and check which value of $\boldsymbol{w}$ leads to an overall minimum objective. In this way, $\boldsymbol{a}$ manipulates the control deviation to be as large as possible. That is, for a given $\boldsymbol{w}$, it is possible to set $a_{i^{\min},j} = 1, j \in [N]$, where $i^{\min}$ is the control with the smallest total accumulation $\sum_{j\in[N]} \boldsymbol{w}_{i,j}\Delta_j$ such that we obtain the (CIA) objective value $\theta^* = \sum_{j\in[N]}(1 - \boldsymbol{w}_{i^{\min},j})\Delta_j$. With these arguments, one can derive

$$\theta^{\max} \leq \left(N - \left\lfloor \frac{N}{n_\omega} \right\rfloor\right) \bar{\Delta}.$$

We omit the exact proof since this bound is generally weak, as we will see later in this chapter.

In this chapter, we derive bounds for $\theta^{\max}$ for (CIA) under the predominantly investigated combinatorial constraints: the MDT constraints (4.11) and (4.12) as well as the TV constraints (4.9) and (4.10). Section 7.1 establishes auxiliary lemmata for our results. In Sections 7.2 and 7.3, we investigate the integral deviation gaps of DSUR and DNFR each for MDT constraints. It emerges that the DNFR scheme is particularly suitable for deriving integral deviation gap bounds on (CIA) and its MDT constraint variants. We state approximation results for (CIA), (CIA-U), (CIA-D), and (CIA-UD) by means of DNFR-constructed solutions in Section 7.4. For the AMDR algorithm, we have already proven approximation properties related to the optimal solution in Section 6.7. We use these properties and the MDR scheme to derive bounds on the integral deviation gap of (CIA-TV) in Section 7.5. Finally, we summarize the obtained bounds and previously established approximation results for the rounding algorithms in Section 7.6.

7.1 Auxiliary lemmata

We present a series of lemmata that are necessary for the proofs of Theorem 7.3 and Theorem 7.4, which establish DNFR-related approximation results. These lemmata are based on [282], Section 5, as well as Appendix A.1 and build on Definitions 6.14- 6.17.

Lemma 7.1 (Family of dwell time block sets)
The family of dwell time block interval sets $\{\mathscr{I}_b\}_{b\in[n_b]}$, as introduced in Definition 6.14 is a partition of the set of all interval indices $[N]$.

Proof. This follows directly from the definition of dwell time block interval sets. □

Lemma 7.2 (Block-accumulated control deviation properties of $\Gamma_{i,b}$ and $\Theta_{i,b}$)
For all $b \in [n_b]$ we have that

$$\sum_{i\in[n_\omega]} \Gamma_{i,b} = \mathscr{L}_b, \qquad \sum_{i\in[n_\omega]} \Theta_{i,b} = 0.$$

Proof. From Lemma 6.2 we know that

$$\sum_{i\in[n_\omega]} \theta_{i,j} = 0 \qquad \text{for all } j \in [N] \tag{7.2}$$

holds. We use this and rearrange the sums in order to prove the first assertion:

$$\sum_{i\in[n_\omega]} \Gamma_{i,b} = \sum_{i\in[n_\omega]} \left(\theta_{i,l_{b-1}} + \sum_{j\in\mathscr{I}_b} a^*_{i,j}\Delta_j\right) = 0 + \sum_{i\in[n_\omega]}\sum_{j\in\mathscr{I}_b} a^*_{i,j}\Delta_j = \sum_{j\in\mathscr{I}_b}\sum_{i\in[n_\omega]} a^*_{i,j}\Delta_j \overset{\text{(Conv)}}{=} \sum_{j\in\mathscr{I}_b} \Delta_j$$
$$= \mathscr{L}_b.$$

The auxiliary result is also useful for the second statement:

$$\sum_{i\in[n_\omega]} \Theta_{i,b} = \sum_{i\in[n_\omega]} \theta_{i,l_{b-1}} \overset{(7.2)}{=} 0.$$ □

Lemma 7.3 (Accumulated difference of Γ and Θ over active controls)
Let $b_1, b_2 \in [n_b]$, and define S_{b_1,b_2} as the set of active controls between b_1 and b_2:

$$S_{b_1,b_2} := \{i \in [n_\omega] \mid \exists b\colon\ b_1 < b < b_2 \text{ with } w_{i,j} = 1,\ \forall j \in \mathscr{I}_b\}.$$

Then, we have

$$\sum_{i\in S_{b_1,b_2}} \left(\Gamma_{i,b_2} - \Theta_{i,b_1}\right) \leq \overline{\mathscr{L}}.$$

Proof. Using Definition 6.16 of Γ, Θ and rearranging the sums yields

$$\begin{aligned}
\sum_{i \in S_{b_1,b_2}} \left(\Gamma_{i,b_2} - \Theta_{i,b_1}\right) &= \sum_{i \in S_{b_1,b_2}} \left(\sum_{b=b_1+1}^{b_2} \sum_{j \in \mathscr{J}_b} a^*_{i,j}\Delta_j - \sum_{b=b_1+1}^{b_2-1} \sum_{j \in \mathscr{J}_b} w_{i,j}\Delta_j \right) \\
&= \sum_{b=b_1+1}^{b_2} \sum_{j \in \mathscr{J}_b} \Delta_j \underbrace{\sum_{i \in S_{b_1,b_2}} a^*_{i,j}}_{\leq 1} - \sum_{b=b_1+1}^{b_2-1} \sum_{j \in \mathscr{J}_b} \Delta_j \underbrace{\sum_{i \in S_{b_1,b_2}} w_{i,j}}_{=1} \\
&\leq \sum_{b=b_1+1}^{b_2} \sum_{j \in \mathscr{J}_b} \Delta_j - \sum_{b=b_1+1}^{b_2-1} \sum_{j \in \mathscr{J}_b} \Delta_j \\
&= \mathscr{L}_{b_2} \leq \overline{\mathscr{L}}.
\end{aligned}$$

□

Note that S_{b_1,b_2} is trivially the empty set if $b_2 \leq b_1 + 1$, but the result remains true in this case. We employ the concept of S_{b_1,b_2} for a contradiction in the proofs of Theorem 7.3 and Theorem 7.4.

Lemma 7.4 (Control with negative Γ value has not been future forced)
Let (C_1, C_2, χ_D) be given, and assume that after executing DNFR (Algorithm 6.5), the forward control deviation of a control $i \in [n_\omega]$ and a block $b_2 \geq 2$ satisfies:

$$\Gamma_{i,b_2} \leq C_2\overline{\mathscr{L}} - \overline{\mathscr{L}}, \qquad \text{and} \qquad \Gamma_{i,b_2} < 0.$$

Then, there is an earlier activation of i on some block $b_1 < b_2$, and this activation has not been b_2-future forced on b_1.

Proof. Note that $\Gamma_{i,b}$ is monotonically increasing in b for deactivated controls i. We conclude from this and $\Gamma_{i,b_2} < 0$ that there is an earlier activation of i on some block $b_1 < b_2$. We take a closer look at the forward control deviation on block b_2:

$$C_2\overline{\mathscr{L}} - \overline{\mathscr{L}} \geq \Gamma_{i,b_2} = \sum_{k=1}^{b_2} \sum_{j \in \mathscr{J}_k} a^*_{i,j}\Delta_j - \sum_{k=1}^{b_1} \sum_{j \in \mathscr{J}_k} w_{i,j}\Delta_j = \sum_{k=1}^{b_2} \sum_{j \in \mathscr{J}_k} a^*_{i,j}\Delta_j - \sum_{k=1}^{b_1-1} \sum_{j \in \mathscr{J}_k} w_{i,j}\Delta_j - \mathscr{L}_{b_1}.$$

Rearranging the terms implies

$$\sum_{k=1}^{b_2} \sum_{j \in \mathscr{J}_k} a^*_{i,j}\Delta_j - \sum_{k=1}^{b_1-1} \sum_{j \in \mathscr{J}_k} w_{i,j}\Delta_j \leq C_2\overline{\mathscr{L}} - \overline{\mathscr{L}} + \mathscr{L}_{b_1} \leq C_2\overline{\mathscr{L}}.$$

The last inequality shows that i has been not b_2-future forced on b_1. □

We introduce a grid on which an MDT $C_1 > 0$ overlaps the grid points by a small $\epsilon > 0$. This grid represents a worst case for the accumulated control difference and is therefore useful for evaluating the tightness of bounds.

Definition 7.1 (Minimal C_1-overlapping grid)
Consider a non-degenerate MDT length, i.e., $C_1 > 0$, and let for ϵ hold $C_1 \gg \epsilon > 0$. Further, let a time horizon $[t_0, t_f]$ be given with length at least $4C_1$, i.e.,

$$t_f \geq t_0 + 4C_1.$$

We define a minimal C_1-overlapping grid $\mathscr{G}_N$ recursively as follows:

$$\begin{aligned} t_1 &:= t_0 + C_1 - \epsilon, \\ t_2 &:= t_1 + C_1, \\ t_j &:= \begin{cases} t_{j-1} + C_1 - \epsilon, & \text{if } j \text{ odd}, \\ t_{j-1} + C_1, & \text{if } j \text{ even}, \end{cases} \qquad \text{for } j = 3, \dots, N-1, \end{aligned}$$

where we set $N-1 := \max\{\, j \mid t_j < t_f \}$, so that $\mathscr{G}_N$ consists of N intervals.

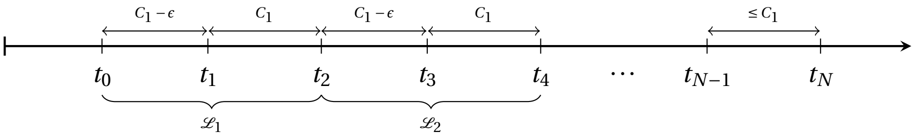


Figure 7.1: Visualization of a *minimal C_1-overlapping grid.*

We determine the length of the resulting dwell time blocks in the following lemma.

Lemma 7.5 (Block length in a minimal C_1-overlapping grid)
The dwell time-invoked blocks $\mathscr{J}_b$, $b \in [n_b]$, of a minimal C_1-overlapping grid, as introduced in Definition 7.1, have the length

$$\begin{aligned} \mathscr{L}_b &= 2C_1 - \epsilon, \qquad\qquad b \in [n_b - 1], \\ \mathscr{L}_{n_b} &= t_f - (t_0 + (n_b - 1)(2C_1 - \epsilon)). \end{aligned}$$

Moreover, we have

$$\bar{\Delta} = C_1, \qquad \overline{\mathscr{L}} = 2C_1 - \epsilon.$$

Proof. We refer to Definition 6.14 from which we deduce $\mathscr{J}_1 = \{1, 2\}$ because $t_0 + C_1$ (minimally) overlaps t_1. The next dwell time block begins at $t_2 = t_0 + 2C_1 - \epsilon$ and again consists of two intervals. This argumentation can be extended to the first $n_b - 1$ blocks, and by the definition of block lengths, we conclude $\mathscr{L}_b = 2C_1 - \epsilon$. The length of the last block $\mathscr{L}_{n_b}$ is directly computed from the definition of $N-1$ to be the last index of the grid point recursion before t_f. Finally, the definition of a *minimal C_1-overlapping grid* and the obtained block lengths imply

$$\bar{\Delta} = C_1, \qquad \overline{\mathscr{L}} = 2C_1 - \epsilon. \qquad \square$$

The following technical lemma is taken from [222], Lemma 8, and is used in the proof of Theorem 7.6.

Lemma 7.6
For $N, \sigma_{\max} \in \mathbb{N}$, where $1 \leq \sigma_{\max} \leq N - 2$, let $R \in \mathbb{Q}$ be defined by

$$R := \frac{N}{3 + 2\sigma_{\max}}.$$

We have

$$2\lceil R \rfloor_{0.5} - 1 \leq \left\lfloor \frac{N - \lceil R \rceil}{\sigma_{\max} + 1} \right\rfloor, \tag{7.3}$$

where we indicate by $\lceil x \rceil_{0.5}$ *the rounding up of* $x \in \mathbb{R}$ *to the next multiple of* 0.5 *as defined in Appendix A.*

Proof. Since R is a rational number with $3+2\sigma_{\max}$ in the denominator, we have

$$\lceil R \rceil_{0.5} \le \frac{N}{3+2\sigma_{\max}} + 0.5\left(1 - \frac{1}{3+2\sigma_{\max}}\right). \tag{7.4}$$

Moreover, using basic properties of the floor and ceiling functions yields

$$\lceil R \rceil \le \lceil R \rceil_{0.5} + 0.5, \tag{7.5}$$

$$\frac{N-\lceil R \rceil}{\sigma_{\max}+1} - \frac{\sigma_{\max}}{\sigma_{\max}+1} \le \left\lfloor \frac{N-\lceil R \rceil}{\sigma_{\max}+1} \right\rfloor. \tag{7.6}$$

Next, we calculate

$$\begin{aligned}
2\left(\lceil R \rceil_{0.5} - 1\right) &= \frac{(2\lceil R \rceil_{0.5} - 1)(\sigma_{\max}+1) - 1}{(\sigma_{\max}+1)} - \frac{\sigma_{\max}}{\sigma_{\max}+1} \\
&= \frac{(2\lceil R \rceil_{0.5} - 1)(\sigma_{\max}+1) + \lceil R \rceil_{0.5} - 1}{(\sigma_{\max}+1)} - \frac{\lceil R \rceil_{0.5}}{(\sigma_{\max}+1)} - \frac{\sigma_{\max}}{\sigma_{\max}+1} \\
&= \frac{\lceil R \rceil_{0.5}(3+2\sigma_{\max}) - (\sigma_{\max}+1) - 1}{(\sigma_{\max}+1)} - \frac{\lceil R \rceil_{0.5}}{(\sigma_{\max}+1)} - \frac{\sigma_{\max}}{\sigma_{\max}+1} \\
&\overset{(7.4)}{\le} \frac{\left(\frac{N}{3+2\sigma_{\max}} + 0.5 - \frac{1}{2(3+2\sigma_{\max})}\right)(3+2\sigma_{\max}) - (\sigma_{\max}+2)}{(\sigma_{\max}+1)} - \frac{\lceil R \rceil_{0.5}}{(\sigma_{\max}+1)} - \frac{\sigma_{\max}}{\sigma_{\max}+1} \\
&= \frac{N-0.5}{(\sigma_{\max}+1)} - \frac{\lceil R \rceil_{0.5}}{(\sigma_{\max}+1)} - \frac{\sigma_{\max}}{\sigma_{\max}+1} - \frac{1}{2(\sigma_{\max}+1)} \\
&\overset{(7.5)}{\le} \frac{N}{(\sigma_{\max}+1)} - \frac{\lceil R \rceil}{(\sigma_{\max}+1)} - \frac{\sigma_{\max}}{\sigma_{\max}+1} - \frac{1}{2(\sigma_{\max}+1)} \\
&= \frac{N-\lceil R \rceil}{(\sigma_{\max}+1)} - \frac{\sigma_{\max}}{\sigma_{\max}+1} - \frac{1}{2(\sigma_{\max}+1)} \\
&\overset{(7.6)}{\le} \left\lfloor \frac{N-\lceil R \rceil}{\sigma_{\max}+1} \right\rfloor - \frac{1}{2(\sigma_{\max}+1)} \\
&< \left\lfloor \frac{N-\lceil R \rceil}{\sigma_{\max}+1} \right\rfloor.
\end{aligned}$$

Both $2\left(\lceil R \rceil_{0.5} - 1\right) \in \mathbb{Z}$ and $\left\lfloor \frac{N-\lceil R \rceil}{\sigma_{\max}+1} \right\rfloor \in \mathbb{Z}$ are valid, so from the above inequality we can deduce that

$$2\left(\lceil R \rceil_{0.5} - 1\right) \le \left\lfloor \frac{N-\lceil R \rceil}{\sigma_{\max}+1} \right\rfloor - 1.$$

□

7.2 Bounds for dwell time sum-up rounding

This section is based on Chapter 6 and Appendix C in [282]. We investigate bounds of DSUR under minimum up (MU) and minimum down (MD) time constraints. Kirches et al. [149] proved the tightest possible bound on the integral deviation gap for SUR. From this, we can derive implications for DSUR in the absence of MD conditions.

Theorem 7.1 (Tight bound for SUR integral deviation gap, Theorem 6.1 in [149])
Let $\boldsymbol{w}^{SUR}$ be constructed from $\boldsymbol{a}^* \in \mathscr{A}_N$ by means of SUR for an equidistant discretization of $[t_0, t_f]$, and denote by $\theta(\boldsymbol{w}^{SUR})$ its (CIA) objective value. Then, the rounding gap is bounded by

$$\theta(\boldsymbol{w}^{SUR}) \leq \bar{\Delta} \sum_{i=2}^{n_\omega} \frac{1}{i},$$

which is the tightest possible upper bound.

Corollary 7.1 (Tight bound for DSUR integral deviation gap without MD times)
For the MD time C_D, let $C_D < \underline{\Delta}$ hold, and let an MU time $C_U > 0$ be given. Let the time horizon $[t_0, t_f]$ be discretized with a minimal C_U-overlapping grid, and let $\boldsymbol{w}^{DSUR}$ be constructed from $\boldsymbol{a}^* \in \mathscr{A}_N$ by means of DSUR. Then, the rounding gap $\theta(\boldsymbol{w}^{DSUR})$ of the (CIA-UD) objective value is bounded by

$$\theta(\boldsymbol{w}^{DSUR}) \leq (C_U + \bar{\Delta}) \sum_{i=2}^{n_\omega} \frac{1}{i},$$

which is the tightest possible upper bound.

Proof. As in the proof of Theorem 7.1 in [149], a dynamic programming argument can be applied, here with an equidistant dwell time block length of $(C_U + \bar{\Delta} - \epsilon)$, as derived in Lemma 7.5. With a time horizon length of $n_\omega(C_U + \bar{\Delta} - \epsilon)$, analogous to the proof of Theorem 7.1, we may construct an example that indicates the tightness of the bound as follows:

$$a_{i,j} := \begin{cases} 0, & \text{if } 2i+1 \leq j \leq N, \\ 1/(n_\omega + 1 - j/2), & \text{if } j \text{ is even}, \\ 1/(n_\omega + 1 - (j+1)/2), & \text{if } j \text{ is odd}, \end{cases} \qquad 1 \leq j \leq N = 2n_\omega.$$

For this example, the DSUR scheme constructs a binary control solution that switches directly after each dwell time block with length $(C_U + \bar{\Delta} - \epsilon)$. Moreover, the controls $i \in [n_\omega - 1]$ are each active on dwell time block i so that the last control n_ω accumulates the asserted rounding gap until the end of dwell time block $n_\omega - 1$. □

Remark 7.2 (DSUR as a generalization of SUR)
The last corollary implicitly states that DSUR can be seen as a generalization of the original SUR algorithm since it reduces to the latter for a negligible MDT $C_U, C_D \leq \underline{\Delta}$.

Theorem 7.1 does not allow a direct conclusion for the case with absent MU times and an active MD time $C_D > \underline{\Delta}$. It is at least possible to provide worst-case examples for $\boldsymbol{a} \in \mathscr{A}_N$ to get a rough idea of how large the upper bound can be for the DSUR integral deviation gap without MU times. This is expressed in the following theorem.

Theorem 7.2 (Integral deviation gap for DSUR without MU times)
Consider an inactive MU time constraint, i.e., $C_U \leq \underline{\Delta}$ and an equidistant grid $\mathscr{G}_N$. We assume for the MD time

$$C_D > (2(n_\omega - 1) - 1)\bar{\Delta}. \tag{7.7}$$

For the grid, let the following hold:

$$N \geq (n_\omega - 1)(1 + M_D) + \lceil M_D/2 \rceil - 1, \tag{7.8}$$

where M_D denotes the number of MD time intervals constructed by C_D, i.e., $M_D := \lceil C_D/\bar{\Delta}\rceil$. Then, there is an $\boldsymbol{a} \in \mathscr{A}_N$ that yields a (CIA-D) objective value $\theta(\boldsymbol{w}^{DSUR})$ of $\boldsymbol{w}^{DSUR}$ constructed by DSUR with

$$\theta(\boldsymbol{w}^{DSUR}) \geq \left(\frac{M_D}{2} + (n_\omega - 2)\right)\bar{\Delta}. \tag{7.9}$$

Proof. Using (7.7) and the definition of M_D, we obtain $M_D \geq 2(n_\omega - 1)$. Notice that even if $C_D/\bar{\Delta} \notin \mathbb{N}$, for the cardinality of the dwell time index sets, we still find $|\mathscr{J}_k^{\mathrm{SUR}}(C_D)| = M_D \in \mathbb{N}$ for $k \leq N - M_D$ because we are using an equidistant grid. Hence, we calculate the forward control deviation of the currently activated control in the DSUR algorithm (line 3) on the subsequent M_D intervals.

We prove the claim by proving the following claims: for any $n_\omega \geq 2$, C_D, and N fulfilling (7.7) and (7.8), there is an $\boldsymbol{a} \in \mathscr{A}_N$ with

$$a_{i,j} = 0, \qquad \text{for} \quad i = 2,\dots,n_\omega,\ j = 1,\dots,i-1, \tag{7.10}$$

resulting in a constructed $\boldsymbol{w}^{\mathrm{DSUR}}$ with

$$w_{i,j}^{\mathrm{DSUR}} = 0, \qquad \text{for} \quad i = 2,\dots,n_\omega,\ j = 1,\dots,i-1, \tag{7.11}$$

and

$$\theta_{2,j} = \left(\frac{M_D}{2} + (n_\omega - 2)\right)\bar{\Delta}, \quad j = (n_\omega - 1)(1 + M_D) + \lceil M_D/2\rceil - M_D. \tag{7.12}$$

This implies Claim (7.9) by the definition of the objective value of (CIA-D). We proceed via induction.

Base case:

$n_\omega = 2$: By assumption we have $M_D \geq 2\bar{\Delta}$ and thus, a nontrivial MD time. We construct $\boldsymbol{a} \in \mathscr{A}_N$ on $N = (1 + M_D) + \lceil M_D/2\rceil - 1$ intervals. If Claim (7.12) is true for this $\boldsymbol{a}$, it also holds for $N \geq (1 + M_D) + \lceil M_D/2\rceil - 1$ because we can extend $\boldsymbol{a}$ by inserting arbitrary unit vector columns after the last column without affecting Claim (7.12). We consider

$$\boldsymbol{a} := \begin{cases} \begin{pmatrix} 1 & 0 \cdots 0 & 1 \cdots 1 \\ 0 & \underbrace{1 \cdots 1}_{j \in \tilde{\mathscr{J}}_1} & \underbrace{0 \cdots 0}_{j \in \tilde{\mathscr{J}}_2} \end{pmatrix}, & \begin{aligned} \tilde{\mathscr{J}}_1 &= \{2,\dots,M_D/2+1\}, \\ \tilde{\mathscr{J}}_2 &= \{M_D/2+2,\dots,N\}, \end{aligned} \quad \text{if } M_D \text{ even}, \\ \begin{pmatrix} 1 & 0 \cdots 0 & 0.5 & 1 \cdots 1 \\ 0 & \underbrace{1 \cdots 1}_{j \in \tilde{\mathscr{J}}_1} & 0.5 & \underbrace{0 \cdots 0}_{j \in \tilde{\mathscr{J}}_2} \end{pmatrix}, & \begin{aligned} \tilde{\mathscr{J}}_1 &= \{2,\dots,\lceil M_D/2\rceil\}, \\ \tilde{\mathscr{J}}_2 &= \{\lceil M_D/2\rceil+2,\dots,N\}, \end{aligned} \quad \text{if } M_D \text{ odd}. \end{cases}$$

Because $a_{2,1} = 0$, (7.10) is true. The DSUR algorithm activates the first control on interval $j = 1$. Then, $i = 1$ is the *currently activated* control. Assuming $i = 1$ is active until $2 \leq k - 1 \leq M_D/2$ when M_D is even, respectively $2 \leq k - 1 \leq \lceil M_D/2\rceil$ when M_D is odd, its dwell time block index set is $\mathscr{J}_k^{\mathrm{SUR}}(C_D) = \{k,\dots,k + M_D - 1\}$ and its forward control deviation on interval k, as given in line 5 of DSUR amounts to

$$\theta_{1,k-1} + \sum_{l \in \mathscr{J}_k^{\mathrm{SUR}}(C_D)} a_{1,l}\bar{\Delta} = -(k-2)\bar{\Delta} + (M_D/2 + (k-2))\,\bar{\Delta} = \frac{M_D}{2}\bar{\Delta}.$$

On the other hand, the forward control deviation for $i = 2$ on interval k amounts to

$$\gamma_{2,k} = \theta_{2,k-1} + a_{2,k}\bar{\Delta} = \begin{cases} (k-2)\bar{\Delta} + 0.5\bar{\Delta} = M_D/2, & \text{if } M_D \text{ odd and } k-1 = \lceil M_D/2 \rceil, \\ (k-2)\bar{\Delta} + 1\bar{\Delta} = (k-1)\bar{\Delta} \leq M_D/2, & \text{else.} \end{cases}$$

We observe that for all intervals k, the forward control deviation for control $i = 1$ is greater than or equal to that of $i = 2$, and we let DSUR deliberately choose $i = 1$ to be active in case of equality. Hence, $w_{1,j}^{\text{DSUR}} = 1$, for $j \in [N]$. This implies that control $i = 2$ stays inactive and in particular, that (7.11) is true. Combining this with the above forward control deviation for $i = 2$ yields

$$\theta_{2,1+\lceil M_D/2 \rceil} = \frac{M_D}{2}\bar{\Delta},$$

which settles Claim (7.12) for $n_\omega = 2$.

Inductive step: We show that if the claim holds for $n_\omega - 1$, then it is also true for n_ω. Let $\boldsymbol{a}^{(n_\omega-1)} \in [0,1]^{(n_\omega-1)\times((n_\omega-2)(1+M_D)+\lceil M_D/2 \rceil-1)}$ denote a matrix for which DSUR constructs a $\boldsymbol{w}^{\text{DSUR}}$ that satisfies Claims (7.10)-(7.12) for $n_\omega - 1$ controls. We construct $\boldsymbol{a} \in \mathscr{A}_N$ on $N = (n_\omega - 1)(1 + M_D) + \lceil M_D/2 \rceil - 1$ intervals and with n_ω controls. As for the base case, we can argue for neglecting the case $N > (n_\omega - 1)(1 + M_D) + \lceil M_D/2 \rceil - 1$. Let $\boldsymbol{I}_k$ denote the identity matrix of dimension $k \times k$, and let $\boldsymbol{0}_k$ denote the zero matrix of dimension $k \times n$, where n is specified by the dimension of the block matrix below the zero matrix. We consider the following matrix

$$(a_{i,j})_{i\in[n_\omega],j\in[N]} := \left(\begin{array}{c|c|c|c} \boldsymbol{I}_{n_\omega} & \boldsymbol{I}_{n_\omega-1} & \boldsymbol{0}_{n_\omega-1} & \boldsymbol{a}^{(n_\omega-1)} \\ \cline{2-4} & 0 \;\cdots\; 0 & \underbrace{1 \;\cdots\; 1}_{j \in \tilde{\mathscr{J}}} & 0 \;\cdots\; 0 \end{array} \right), \qquad \tilde{\mathscr{J}} = \{2n_\omega, \ldots, M_D + 1\},$$

where the third block of columns may be nonexistent if $2n_\omega > M_D + 1$. The first two blocks of columns, however, are well-defined since $M_D \geq 2(n_\omega - 1)$ by (7.7) and thus, $2n_\omega - 1 \leq M_D + 1$. The above matrix is defined on N intervals, with N chosen as above, since we add $M_D + 1$ intervals to the existing $(n_\omega - 2)(1 + M_D) + \lceil M_D/2 \rceil - 1$ intervals from $\boldsymbol{a}^{(n_\omega-1)}$. We first note that (7.10) is satisfied by $\boldsymbol{a}$. Second, we claim that DSUR constructs the following $\boldsymbol{w}^{\text{DSUR}} \in W$:

$$(w_{i,j}^{\text{DSUR}})_{i\in[n_\omega],j\in[N]} := \left(\begin{array}{c|c|c} \boldsymbol{I}_{n_\omega} & \boldsymbol{0}_{n_\omega-1} & \boldsymbol{w}^{\text{DSUR},(n_\omega-1)} \\ \cline{2-3} & \underbrace{1 \;\cdots\; 1}_{j \in \tilde{\mathscr{J}}} & 0 \;\cdots\; 0 \end{array} \right), \qquad \tilde{\mathscr{J}} = \{n_\omega + 1, \ldots, M_D + 1\},$$

where $\boldsymbol{w}^{\text{DSUR},(n_\omega-1)}$ denotes the solution obtained by DSUR for $\boldsymbol{a}^{(n_\omega-1)}$. We first justify this value for the intervals $k = 1, \ldots, n_\omega$:

- $k = 1$: DSUR selects control $i = 1$ because $a_{1,1} = 1$.
- $k = 2$: Control $i = 1$ is currently activated with a forward control deviation of $\bar{\Delta}$, calculated on the subsequent M_D intervals. The forward control deviation for control $i = 2$ amounts to $\gamma_{2,2} = \theta_{2,1} + a_{2,2}\bar{\Delta} = 0 + \bar{\Delta}$. Therefore, DSUR may set the control $i = 2$ to be active.
- $k = 3$: We use the inductive hypothesis for $\boldsymbol{a}^{(n_\omega-1)}$ and Claim (7.10), which yields $a_{2,M_D+2}^{(n_\omega-1)} = 0$. Thus, the forward control deviation of control $i = 2$ is $\bar{\Delta}$, which is the same as for $i = 3$. We let DSUR deliberately set the control $i = 3$ to be active.

- $k = 4, \dots, n_\omega$: We argue analogously to the case $k = 3$.

Hence, (7.11) is established. After control $i = n_\omega$ has been activated on interval $k = n_\omega$, all other controls are *down time forbidden* until interval $M_D + 1$. Thus, control $i = n_\omega$ stays active up to and including interval $M_D + 1$. Because the controls $i = 1, \dots, n_\omega - 1$ are only active once before interval $M_D + 1$, but $\sum_{k \in [M_D+1]} a_{i,k}\bar{\Delta} = 2\bar{\Delta}$, we conclude that $\theta_{i,M_D+1} = \bar{\Delta}$. This justifies why DSUR constructs $\boldsymbol{w}^{\mathrm{DSUR},(n_\omega-1)}$ after interval $M_D + 1$:

- The controls $i = 2, \dots, n_\omega - 1$ are *down time forbidden* on the intervals $k = (M_D + 1) + 1, \dots, (M_D + 1) + i - 1$, but they are not activated these intervals in $\boldsymbol{w}^{\mathrm{DSUR},(n_\omega-1)}$ anyway, according to the inductive hypothesis (7.11).
- The control deviation for control n_ω is negative, i.e., $\theta_{n_\omega,k} = -(M_D + 1 - n_\omega)\bar{\Delta}$ for $k \geq M_D + 1$, so control n_ω is not activated after interval $M_D + 1$.
- All other controls $1, \dots, n_\omega - 1$ start with the same control deviation $\theta_{i,M_D+1} = \bar{\Delta}$ when DSUR iterates on interval $M_D + 2$. Thus, DSUR constructs the same $\boldsymbol{w}$ from $\boldsymbol{a}^{(n_\omega-1)}$ as it would construct from $\boldsymbol{a}^{(n_\omega-1)}$ starting with the first interval and $\theta_{i,0} = 0$. By the inductive hypothesis, this implies that DSUR generates $\boldsymbol{w}^{\mathrm{DSUR},(n_\omega-1)}$.

The inductive hypothesis regarding (7.12) implies for $\boldsymbol{w}^{\mathrm{DSUR},(n_\omega-1)}$

$$\theta_{2,j} = \left(\frac{M_D}{2} + ((n_\omega - 1) - 2)\right)\bar{\Delta}, \quad j = ((n_\omega - 1) - 1)(1 + M_D) + \lceil M_D/2 \rceil - M_D.$$

We argued that this control deviation value is increased by $\bar{\Delta}$ in $\boldsymbol{w}^{\mathrm{DSUR}}$ and that before the choice of $\boldsymbol{w}^{\mathrm{DSUR},(n_\omega-1)}$ there exist $M_D + 1$ columns in $\boldsymbol{w}^{\mathrm{DSUR}}$. So, (7.12) is also true for n_ω controls. □

Remark 7.3 (Rounding gap for DSUR with MU and MD constraints)
Generally, when the problem setting involves both MU and MD time constraints, i.e., $C_D, C_U > \underline{\Delta}$, the worst-case integral deviation gap constructed by the DSUR scheme is at least the maximum of the bounds obtained in Corollary 7.1 and Theorem 7.2.

7.3 Bounds for dwell time next-forced rounding

We investigate the integral deviation gap for binary controls constructed by DNFR with specified parameter choices for C_2 and χ_D. These investigations are presented as two theorems; the proofs of which follow a similar approach as that for Proposition 4.8 in [135]. In Theorem 7.3, we examine how large the control deviation can become as part of the DNFR algorithm during an MD time phase. Based on this result, in Theorem 7.4, we derive that DNFR constructs (CIA) feasible solutions with objective bounds that depends on the rounding threshold C_2 and on whether down time forbidden controls are allowed, i.e., $\chi_D = 1$. This section reproduces results from Section 5.1 in [282].

Theorem 7.3 (Γ of a *down time forbidden* control as part of DNFR)
Let $\boldsymbol{a}^* \in \mathscr{A}_N$, $(C_2, \chi_D) = \left(\frac{3}{2}, 1\right)$, and $C_1 \geq 0$ be given and assume there is a down time forbidden control $i_D \in \mathscr{I}_b^D$ on dwell time block $b \geq 3$ after DNFR has been executed. Then, the forward control deviation satisfies

$$\Gamma_{i_D,b} \leq \tfrac{3}{2}\overline{\mathscr{L}}.$$

Proof. We proceed via induction.
Base case: We consider the first block b on which a down time forbidden control $i_D \in [n_\omega]$ appears and assume that

$$\Gamma_{i_D,b} > \tfrac{3}{2}\overline{\mathscr{L}} \tag{7.13}$$

holds; we prove that this results in a contradiction. It follows from Lemma 7.2 that

$$\tfrac{3}{2}\overline{\mathscr{L}} \geq \mathscr{L}_b = \sum_{i \neq i_D} \Gamma_{i,b} + \Gamma_{i_D,b},$$

so there must be a control $i_1 \neq i_D$ with negative forward control deviation on b:

$$\exists\, i_1 \neq i_D : \, \Gamma_{i_1,b} < 0.$$

We apply Lemma 7.4 to the last inequality: i_1 has not been b-future forced on its last activation, and we denote the dwell time block of this activation by b_1. In other words, we know that there is at least one dwell time block b_1 and one control i_1 that was not b-future forced on b_1 and was still activated on b_1. We denote by i_1 the control of this property with the last activation before b. By this definition, we observe that all controls that are activated after b_1 would become forced until b. We notate

$$F_{b_1,b} := \{i \in [n_\omega] \mid \exists k(i) : \, b_1 < k(i) \leq b \text{ on which } i \text{ is forced or } b\text{-future forced}\}.$$

In particular, we have $i_D \in F_{b_1,b}$. For $i \in F_{b_1,b} \backslash \{i_D\}$ we conclude

$$\Gamma_{i,b} = \Theta_{i,b-1} + \sum_{j \in \mathscr{J}_b} a_{i,j}\Delta_j = \sum_{k=1}^{b} \sum_{j \in \mathscr{J}_k} a_{i,j}\Delta_j - \sum_{k=1}^{k(i)-1} \sum_{j \in \mathscr{J}_k} w_{i,j}\Delta_j > \tfrac{3}{2}\overline{\mathscr{L}},$$

and therefore,

$$\Gamma_{i,b} > \tfrac{3}{2}\overline{\mathscr{L}} - \mathscr{L}_{k(i)}, \qquad i \in F_{b_1,b} \backslash \{i_D\}. \tag{7.14}$$

The last inequality holds, since control i was last activated at dwell time block $k(i)$. For dwell time block b_1, we know that i_1 was chosen, despite not being b-future forced. We use this observation and our assumption that $b > b_1$ is the first dwell time block with a down time forbidden control to conclude that all controls from $F_{b_1,b}$ were *inadmissible* on b_1. Hence, for $i \in F_{b_1,b}$, it results that

$$\Gamma_{i,b_1} < -\tfrac{3}{2}\overline{\mathscr{L}} + \mathscr{L}_{b_1}, \qquad \Rightarrow \qquad \Theta_{i,b_1} < -\tfrac{3}{2}\overline{\mathscr{L}} + \mathscr{L}_{b_1}. \tag{7.15}$$

We sum up the inequalities (7.13) and (7.14) over $F_{b_1,b}$ and similarly for (7.15), which yields

$$\sum_{i \in F_{b_1,b}} \Gamma_{i,b} \; > \; \tfrac{3}{2}\overline{\mathscr{L}} + (|F_{b_1,b}| - 1)\left(\tfrac{3}{2}\overline{\mathscr{L}} - \mathscr{L}_{b_2}\right), \tag{7.16}$$

$$\sum_{i \in F_{b_1,b}} \Theta_{i,b_1} \; < \; |F_{b_1,b}|\left(-\tfrac{3}{2}\overline{\mathscr{L}} + \mathscr{L}_{b_1}\right), \tag{7.17}$$

where we set $b_2 := \arg\min\{\mathscr{L}_k \mid b_1 < k \leq b\}$ and denote by $|F_{b_1,b}|$ the cardinality of $F_{b_1,b}$. Subtracting (7.17) from (7.16) results in

$$\begin{aligned}\sum_{i\in F_{b_1,b}} \left(\Gamma_{i,b}-\Theta_{i,b_1}\right) &> \tfrac{3}{2}\overline{\mathscr{L}}+(|F_{b_1,b}|-1)\left(\tfrac{3}{2}\overline{\mathscr{L}}-\mathscr{L}_{b_2}\right)-|F_{b_1,b}|(-\tfrac{3}{2}\overline{\mathscr{L}}+\mathscr{L}_{b_1})\\ &= \tfrac{3}{2}\overline{\mathscr{L}}+(2|F_{b_1,b}|-1)\tfrac{3}{2}\overline{\mathscr{L}}-(|F_{b_1,b}|-1)\mathscr{L}_{b_2}-|F_{b_1,b}|\mathscr{L}_{b_1}\\ &\geq \tfrac{3}{2}\overline{\mathscr{L}}+(2|F_{b_1,b}|-1)\tfrac{1}{2}\overline{\mathscr{L}}\\ &> \overline{\mathscr{L}}. \end{aligned} \tag{7.18}$$

We used $\mathscr{L}_{b_1},\mathscr{L}_{b_2} \leq \overline{\mathscr{L}}$ in the second inequality. We finish our calculations by considering the property of $F_{b_1,b}$ comprising all control activations between dwell time block b_1+1 and $b-1$. Therefore, we can apply Lemma 7.3 with $F_{b_1,b} = S_{b_1,b}$ and obtain

$$\sum_{i\in F_{b_1,b}} \left(\Gamma_{i,b}-\Theta_{i,b_1}\right) \leq \overline{\mathscr{L}}, \qquad\qquad \lightning \tag{7.19}$$

which contradicts inequality (7.18).
Inductive step: Let the assertion hold until any dwell time block $b-1\in[n_b]$; we prove that the statement holds for b. Again, we consider $i_D\in[n_\omega]$ and assume that

$$\Gamma_{i_D,b} > \tfrac{3}{2}\overline{\mathscr{L}} \tag{7.20}$$

holds; we prove that this results in a contradiction. With similar argumentation as in the base case we deduce that there is a control i_1 that has not been b-future forced on dwell time block $b_1<b$ and reuse the definition of $F_{b_1,b}$. Thus, inequality (7.14) still holds. Now, we distinguish between two cases in which the controls from $F_{b_1,b}$ were not activated on b_1. If all controls $i\in F_{b_1,b}$ were *inadmissible* on b_1, we can argue as in the base case. Hence, we focus on the other case: there is an $i_2\in F_{b_1,b}$ that was *down time forbidden* on b_1, while all other controls $i\in F_{b_1,b}\backslash\{i_2\}$ were inadmissible. By the inductive hypothesis and the previously derived inequality (7.15) we have

$$\Theta_{i_2,b_1}\leq \tfrac{3}{2}\overline{\mathscr{L}}, \qquad\qquad \Theta_{i,b_1} < -\tfrac{3}{2}\overline{\mathscr{L}}+\mathscr{L}_{b_1}, \quad i\in F_{b_1,b}\backslash\{i_2\}.$$

Summing up these inequalities over $F_{b_1,b}$ therefore results in

$$\sum_{i\in F_{b_1,b}} \Theta_{i,b_1} \;<\; \tfrac{3}{2}\overline{\mathscr{L}}+|F_{b_1,b}-1|\left(-\tfrac{3}{2}\overline{\mathscr{L}}+\mathscr{L}_{b_1}\right). \tag{7.21}$$

Next, we argue that $F_{b_1,b}$ contains at least two controls: the case $b_1=b-1$ is not possible since i_D is forced and down time forbidden on b by assumption and hence admissible on $b-1$. Therefore, $b_1\leq b-2$, and there is a control $i\neq i_D$, $i\in F_{b_1,b}$, that is activated on $b-1$. Altogether, we have $|F_{b_1,b}|\geq 2$. With this observation we subtract inequality (7.21) from (7.16):

$$\begin{aligned}\sum_{i\in F_{b_1,b}} \left(\Gamma_{i,b}-\Theta_{i,b_1}\right) &> \tfrac{3}{2}\overline{\mathscr{L}}+(|F_{b_1,b}|-1)\left(\tfrac{3}{2}\overline{\mathscr{L}}-\mathscr{L}_{b_2}\right)-\left(\tfrac{3}{2}\overline{\mathscr{L}}+(|F_{b_1,b}|-1)(-\tfrac{3}{2}\overline{\mathscr{L}}+\mathscr{L}_{b_1})\right)\\ &= 3(|F_{b_1,b}|-1)\overline{\mathscr{L}}-(|F_{b_1,b}|-1)\mathscr{L}_{b_2}-(|F_{b_1,b}|-1)\mathscr{L}_{b_1}\\ &\geq (|F_{b_1,b}|-1)\overline{\mathscr{L}}\\ &\geq \overline{\mathscr{L}}. \end{aligned}$$

Note that $|F_{b_1,b}| \geq 2$ is used in the last inequality. Finally, we again build on Lemma 7.3, where it is justified to set $F_{b_1,b} = S_{b_1,b}$. The above inequality thus contradicts the inequality from the lemma, and we have shown that the assertion holds for all $b \in [n_b]$ on which a down time forbidden control exists. □

With the last theorem, we already have a statement about the control deviation for down time forbidden controls. The next result goes further and establishes a connection between DNFR and (CIA).

Theorem 7.4 (Rounding gap of solution constructed by DNFR)
Let $\boldsymbol{a}^* \in \mathscr{A}_N$ and the following parameter settings be given:

I. $(C_2, \chi_D) = \left(\frac{2n_\omega - 3}{2n_\omega - 2}, 0\right)$,

II. $(C_2, \chi_D) = \left(\frac{3}{2}, 1\right)$,

and $C_1 \geq 0$. Then, $\boldsymbol{w}^{DNFR}$ obtained by DNFR is a feasible solution of (CIA) for both cases with approximation quality

$$\theta(\boldsymbol{w}^{DNFR}) \leq C_2 \overline{\mathscr{L}}.$$

Proof. The assertion can be shown in a very similar way for the parameter choices I and II, and we therefore prove both cases in parallel. Since for each dwell time block $b \in [n_\omega]$, the algorithm activates either a *forced*, *future forced*, or *admissible* control and the family of dwell time blocks is a partition of $[N]$ by Lemma 7.1, exactly one control is activated on each interval $j \in [N]$. Therefore, the (Conv) constraint is satisfied. Hence, DNFR guarantees the feasibility of $\boldsymbol{w}^{\text{DNFR}}$. If down time forbidden controls are neglected, i.e., $\chi_D = 0$, $\boldsymbol{w}^{\text{DNFR}}$ yields an objective value with at most the claimed upper bound by the definitions of admissible and forced activation. The same holds for the choice $\chi_D = 1$ since by Theorem 7.3, the control deviation does not exceed the claimed upper bound during an MD time phase. Therefore, we only need to prove that DNFR always provides a solution. To this end, we show that for each interval there is 1.) at least one admissible control and 2.) at most one forced control.

1.) We prove by contradiction that there exists at least one admissible control. Assume that there is no admissible activation for dwell time block $b \in [n_b]$ and distinguish between the following cases:

I. With $C_2 = \frac{2n_\omega - 3}{2n_\omega - 2}$, we assume

$$\Gamma_{i,b} < -\frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} + \mathscr{L}_b, \qquad i \in [n_\omega],$$

and we prove that this results in a contradiction. It follows from summing up all controls and from Lemma 7.2 that

$$\mathscr{L}_b = \sum_{i \in [n_\omega]} \Gamma_{i,b} < n_\omega \left(-\frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} + \mathscr{L}_b\right) = -n_\omega \frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} + n_\omega \mathscr{L}_b.$$

Subtracting $n_\omega \mathscr{L}_b$ from the right-hand side yields

$$(1-n_\omega)\overline{\mathscr{L}} \leq (1-n_\omega)\mathscr{L}_b < -n_\omega \frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} = -n_\omega \overline{\mathscr{L}} + \frac{n_\omega}{2(n_\omega - 1)}\overline{\mathscr{L}} \overset{n_\omega \geq 2}{\leq} (1-n_\omega)\overline{\mathscr{L}}. \quad ↯$$

II. If there is no down time forbidden control on b, we can proceed as in I. Otherwise, there may be one control i_D that is down time forbidden. We assume all other controls are *inadmissible*, i.e.,

$$\Gamma_{i,b} < -\tfrac{3}{2}\overline{\mathscr{L}} + \mathscr{L}_b, \qquad i \in [n_\omega],\ i \neq i_D,$$

and we prove that this results in a contradiction. By Lemma 7.2

$$\mathscr{L}_b = \sum_{i\in[n_\omega]} \Gamma_{i,b} = \sum_{i\neq i_D} \Gamma_{i,b} + \Gamma_{i_D,b} < (n_\omega - 1)(-\tfrac{3}{2}\overline{\mathscr{L}} + \mathscr{L}_b) + \Gamma_{i_D,b},$$

and therefore,

$$\tfrac{3}{2}(n_\omega - 1)\overline{\mathscr{L}} - (n_\omega - 2)\mathscr{L}_b \leq \tfrac{3}{2}\overline{\mathscr{L}} < \Gamma_{i_D,b}. \qquad ↯$$

The last inequality is a contradiction of Theorem 7.3.

We conclude that there must be an admissible activation for all dwell time blocks and thereby for all intervals.

2.) If there were more than one forced controls at a time step, the algorithm would be ambiguous in lines 3-4. Moreover, in this case, DNFR would provide a solution that does not satisfy the upper bound on the objective. Therefore, we prove that this case is impossible and again do so by contradiction. Assume that $b \in [n_b]$ is the dwell time block with the smallest index on which at least two controls i_1, i_2 are forced, i.e.,

$$I.\quad \Gamma_{i_1,b}, \Gamma_{i_2,b} > \tfrac{2n_\omega-3}{2n_\omega-2}\overline{\mathscr{L}}, \qquad II.\quad \Gamma_{i_1,b}, \Gamma_{i_2,b} > \tfrac{3}{2}\overline{\mathscr{L}}. \tag{7.22}$$

In the proof of Theorem 7.3, we showed how to obtain a contradiction with only one forward control deviation $\Gamma_{i,b}$ greater than the rounding threshold, which settles case *II.* We thus focus on case *I.* for which we proceed very similarly as in the proof of Theorem 7.3. We first apply Lemma 7.2:

$$\overline{\mathscr{L}} \geq \mathscr{L}_b = \sum_{i\in[n_\omega]} \Gamma_{i,b} = \sum_{\substack{i\in[n_\omega],\\ i\neq i_1,i_2}} \Gamma_{i,b} + \sum_{i=i_1,i_2} \Gamma_{i,b} > \sum_{\substack{i\in[n_\omega],\\ i\neq i_1,i_2}} \Gamma_{i,b} + 2\frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}}.$$

Hence, we have

$$\sum_{i\in[n_\omega], i\neq i_1,i_2} \Gamma_{i,b} < \overline{\mathscr{L}} - 2\,\frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} = -\frac{2n_\omega - 4}{2n_\omega - 2}\overline{\mathscr{L}},$$

which implies that there is at least one control i_3 such that

$$\Gamma_{i_3,b} < -\frac{1}{n_\omega - 2}\frac{2n_\omega - 4}{2n_\omega - 2}\overline{\mathscr{L}} = -\frac{2}{2n_\omega - 2}\overline{\mathscr{L}}.$$

Then by Lemma 7.4, there is an earlier activation of i_3 on some dwell time block $b_3 < b$, and this activation has not been b-future forced on b_3. Let i_3 denote the control of this property with the last activation before b. This definition implies that all controls that are active between b_3 and b become forced until b. We reuse the notation

$$F_{b_3,b} := \{i \in [n_\omega] \mid \exists k(i):\ b_3 < k(i) \leq b \text{ on which } i \text{ is forced or } b\text{-future forced.}\}.$$

In particular, we find $i_1, i_2 \in F_{b_3,b}$. For $i \in F_{b_3,b} \backslash \{i_1, i_2\}$, we apply the definition of $F_{b_3,b}$ and Γ:

$$\Gamma_{i,b} = \Theta_{i,b-1} + \sum_{j\in\mathscr{I}_b} a_{i,j}\Delta_j = \sum_{k=1}^{b}\sum_{j\in\mathscr{I}_k} a_{i,j}\Delta_j - \sum_{k=1}^{k(i)}\sum_{j\in\mathscr{I}_k} w_{i,j}\Delta_j.$$

Since control i was last activated on dwell time block $k(i)$ and b- future forced on $k(i)$, we have

$$\Gamma_{i,b} > \frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} - \mathscr{L}_{k(i)}, \qquad i \in F_{b_3,b}\backslash\{i_1, i_2\}. \tag{7.23}$$

For dwell time block b_3, we know that i_3 has been chosen even though it is not b-future forced. This implies that i_3 was selected on b_3 because none of the controls from $F_{b_3,b}$ were admissible at this dwell time block. Hence, for $i \in F_{b_3,b}$, it results that

$$\Gamma_{i,b_3} < -\frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} + \mathscr{L}_{b_3}, \qquad \Rightarrow \qquad \Theta_{i,b_3} < -\frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} + \mathscr{L}_{b_3}. \tag{7.24}$$

Now, we consider the sum of inequalities (7.23) and (7.22) and sum up (7.24) over $F_{b_3,b}$, yielding

$$\begin{aligned}
\sum_{i\in F_{b_3,b}} \Gamma_{i,b} &> 2\frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} + (|F_{b_3,b}| - 2)\left(\frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} - \mathscr{L}_{b_2}\right), && (7.25)\\
\sum_{i\in F_{b_3,b_3}} \Theta_{i,b_3} &< |F_{b_3,b}|\left(-\frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} + \mathscr{L}_{b_3}\right), && (7.26)
\end{aligned}$$

where $b_2 := \arg\min\{\mathscr{L}_k \mid b_3 < k \leq b\}$. Subtracting (7.26) from (7.25), we obtain

$$\begin{aligned}
\sum_{i\in F_{b_3,b}} \left(\Gamma_{i,b} - \Theta_{i,b_3}\right) &> 2\frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} + (|F_{b_3,b}| - 2)\left(\frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} - \mathscr{L}_{b_2}\right)\\
&\quad - |F_{b_3,b}|\left(-\frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} + \mathscr{L}_{b_3}\right)\\
&= 2|F_{b_3,b}|\frac{2n_\omega - 3}{2n_\omega - 2}\overline{\mathscr{L}} - (|F_{b_3,b}| - 2)\mathscr{L}_{b_2} - |F_{b_3,b}|\ \mathscr{L}_{b_3}\\
&\geq \overline{\mathscr{L}}\left(2|F_{b_3,b}|\frac{2n_\omega - 3}{2n_\omega - 2} - 2|F_{b_3,b}| + 2\right) && (7.27)\\
&= \overline{\mathscr{L}}\left(2 - \frac{|F_{b_3,b}|}{n_\omega - 1}\right)\\
&\geq \overline{\mathscr{L}}. && (7.28)
\end{aligned}$$

In (7.27) we used $\mathscr{L}_{b_2}, \mathscr{L}_{b_3} \leq \overline{\mathscr{L}}$, while the last inequality holds since $|F_{b_3,b}| \leq n_\omega - 1$. As in the proof of Theorem 7.3, we invoke Lemma 7.3 with $F_{b_3,b} = S_{b_1,b}$ to raise a contradiction with inequality (7.28). Overall, we have shown that there is at most one forced activation per dwell time block and thereby per interval. This completes the proof. □

Remark 7.4 (Known integral deviation gap result for NFR as a special case of Theorem 7.4) On closer inspection, the proof of Theorem 7.4 shows us that DNFR provides a solution with control deviation bounded by $C_2\overline{\mathscr{L}}$ in the absence of MD time constraints, i.e., $\chi_D = 0$, and for any chosen rounding threshold $C_2 \geq \frac{2n_\omega - 3}{2n_\omega - 2}$ and any dwell time block length parameter $C_1 \geq 0$. This implies that the previously known result for NFR in Proposition 4.8, [135], $\theta(\boldsymbol{w}^{\text{NFR}}) \leq \bar{\Delta}$, is a special case of DNFR with $C_1 = 0$, and $C_2 = 1$.

7.4 (CIA) with and without minimum dwell time constraints

We deduce specific bounds on the integral deviation gap for (CIA) and (CIA-U) as well as for (CIA-D) and (CIA-UD) in Sections 7.4.1 and 7.4.2, respectively. The DNFR scheme and the associated results from Section 7.3 are crucial for this section, the content of which comes from Sections 5.2 and 5.3 in [282].

7.4.1 Implications for (CIA) and (CIA-U)

Theorem 7.4 states only generic approximation results for (CIA) with an MDT parameter C_1. We assess the consequences for (CIA-U) by specifying C_1 and proving the tightness of the resulting upper bound. Clearly, (CIA) is a special case of (CIA-U), where $C_U = 0$, so results for (CIA-U) are inherited by (CIA).

Proposition 7.1 (Upper bound for (CIA-U))
Let any MU time $C_U \geq 0$, grid $\mathscr{G}_N$ and $\boldsymbol{a}^* \in \mathscr{A}_N$ be given. Then, for (CIA-U) the following holds:

$$\theta^* \leq \frac{2n_\omega - 3}{2n_\omega - 2}\left(C_U + \bar{\Delta}\right).$$

Proof. We consider the DNFR scheme with $(C_1, C_2, \chi_D) = \left(C_U, \frac{2n_\omega - 3}{2n_\omega - 2}, 0\right)$. Then, $\boldsymbol{w}^{\text{DNFR}}$ is a feasible solution by Theorem 7.4 and according to the property that DNFR activates dwell time blocks of intervals with length at least $C_1 = C_U$. From the definition of block length, we conclude $\overline{\mathscr{L}} < C_U + \bar{\Delta}$, and the assertion follows directly from Theorem 7.4. □

We show that the deduced MU time bound is tight.

Proposition 7.2 (Tightness of the bound for (CIA-U))
Let an MU time $C_U \geq 0$ and a grid $\mathscr{G}_N$ be given with

$$t_f - t_0 \geq 2\,C_U(n_\omega - 1).$$

Then, the objective bound for (CIA-U) mentioned in Proposition 7.1 is the tightest possible bound.

Proof. We first consider $C_U > 0$ and construct an example with the desired objective value by means of a *minimal C_1-overlapping grid*, where we set $C_1 = C_U$. The proposition assumes a time horizon length of at least $2C_1(n_\omega - 1)$, so the grid induced by Lemma 7.5 consists of at least $n_b \geq n_\omega - 1$ dwell time blocks. Let $\boldsymbol{a}^* \in \mathscr{A}_N$ be given as

$$(a_{i,j})^*_{i\in[n_\omega],j\in[N]} := \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & \cdots & 0 \\ \frac{1}{2n_\omega-2} & \frac{1}{2n_\omega-2} & \frac{1}{n_\omega-1} & \cdots & \frac{1}{n_\omega-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \underbrace{\frac{1}{2n_\omega-2} \quad \frac{1}{2n_\omega-2}}_{j\,\in\,\mathscr{J}_1} & & \frac{1}{n_\omega-1} & \cdots & \frac{1}{n_\omega-1} \end{pmatrix}.$$

Consequently, in $\boldsymbol{a}^*$, all controls $i \in [n_\omega]$, $i \neq 1$, assume the same values on each interval. After the first dwell time block, we set control $i = 1$ to zero, while all other variables are set to $\frac{1}{n_\omega-1}$ for the remaining intervals, i.e., dwell time blocks. Next, we discuss how the optimal solution of (CIA-U) on the first $n_\omega - 1$ dwell time blocks might be chosen. We calculate the control deviation if we were to activate a control $i = 2 \ldots n_\omega$ on the first dwell time block:

$$\Theta_{i,1} = \left| \sum_{j\in\mathscr{J}_1} \frac{1}{2n_\omega - 2}\Delta_j - \mathscr{L}_1 \right| = \frac{2n_\omega - 3}{2n_\omega - 2}\mathscr{L}_1 = \frac{2n_\omega - 3}{2n_\omega - 2}(2C_U - \epsilon) = \frac{2n_\omega - 3}{2n_\omega - 2}(C_U + \bar{\Delta} - \epsilon).$$

In the second and third equalities, we used Lemma 7.5. For $i = 2\ldots n_\omega$ and dwell time blocks $1, \ldots, n_\omega - 1$, the values of the relaxed controls $\boldsymbol{a}^*$ sum up to

$$\begin{aligned} \sum_{b\in[n_\omega-1]} \sum_{j\in\mathscr{J}_b} a^*_{i,j}\Delta_j &= \frac{1}{2n_\omega - 2}\mathscr{L}_1 + \sum_{b=2,\ldots,n_\omega-1} \frac{1}{n_\omega - 1}\mathscr{L}_b \\ &= \frac{1}{2n_\omega - 2}(C_U + \bar{\Delta} - \epsilon) + \sum_{b=2,\ldots,n_\omega-1} \frac{1}{n_\omega - 1}(C_U + \bar{\Delta} - \epsilon) \\ &= \frac{2n_\omega - 3}{2n_\omega - 2}(C_U + \bar{\Delta} - \epsilon). \end{aligned}$$

Thus, there are $n_\omega - 1$ controls with this control accumulation on $n_\omega - 1$ dwell time blocks; however, activating any of these controls on the first dwell time block yields the same control deviation. Hence, the objective value of (CIA-U) with this $\boldsymbol{a}^*$ is at least $\frac{2n_\omega-3}{2n_\omega-2}(C_U + \bar{\Delta} - \epsilon)$, where ϵ is arbitrarily small. If we combine this result with Proposition 7.1, we find that $\frac{2n_\omega-3}{2n_\omega-2}(C_U + \bar{\Delta})$ is the tightest possible bound. We argue for the degenerate case, $C_U = 0$, that we can create an example with the length of all dwell time blocks set to $\bar{\Delta}$ and obtain the same tight bound. □

Corollary 7.2 (Tight bound on the integral deviation gap for (CIA))
Consider $\mathscr{G}_N$ and $\boldsymbol{a}^* \in \mathscr{A}_N$. The optimal objective value of (CIA) is bounded by

$$\theta^* \leq \frac{2n_\omega - 3}{2n_\omega - 2}\bar{\Delta}.$$

If $N \geq n_\omega - 1$ holds, then this bound is tight.

Proof. The bound follows from Proposition 7.1 with $C_U = 0$ and if $N \geq n_\omega - 1$, we are able to construct the same worst-case example as in the proof of Proposition 7.2, with intervals applied instead of dwell time blocks. □

7.4.2 Implications for the objectives of (CIA-D) and (CIA-UD)

The bound obtained for (CIA-U) can be transferred in a straightforward manner to (CIA-D) by using $C_1 = C_D$ as the MDT in the DNFR scheme. However, we note the increased number of degrees of freedom when dealing with MD times rather than MU times: only the down time forbidden control is fixed for a specific time duration in contrast to the MU time constraint situation in which all controls are fixed due to the fixed active control. With this observation, we introduced in the DNFR scheme the min down mode $\chi_D = 1$ and subsequently deduce an alternative upper bound to that obtained for DNFR with $\chi_D = 0$. As will be shown, this alternative bound is independent of n_ω but is not always an improvement. We therefore declare the minimum of both bounds as the upper bound in the following proposition.

Proposition 7.3 (Bounds on the objectives of (CIA-D) and (CIA-UD))
Consider any grid $\mathcal{G}_N$ and $\boldsymbol{a}^* \in \mathcal{A}_N$. Let the MU and MD times $C_U, C_D \geq 0$ be given. Then

1. *(CIA-D) is bounded by*

$$\theta^* \leq \min\left\{\frac{3}{4}C_D + \frac{3}{2}\bar{\Delta}, \frac{2n_\omega - 3}{2n_\omega - 2}\left(C_D + \bar{\Delta}\right)\right\}.$$

2. *(CIA-UD) is bounded by*

$$\theta^* \leq \begin{cases} \frac{2n_\omega-3}{2n_\omega-2}\left(C_U + \bar{\Delta}\right), & \text{if } C_U \geq C_D, \\ \min\left\{\frac{3}{2}C_U + \frac{3}{2}\bar{\Delta}, \frac{2n_\omega-3}{2n_\omega-2}\left(C_D + \bar{\Delta}\right)\right\}, & \text{if } C_D > C_U > C_D/2, \\ \min\left\{\frac{3}{4}C_D + \frac{3}{2}\bar{\Delta}, \frac{2n_\omega-3}{2n_\omega-2}\left(C_D + \bar{\Delta}\right)\right\}, & \text{if } C_D/2 \geq C_U. \end{cases}$$

Proof. Generally, if $C_D > C_U$ or if MU constraints are absent, we may apply the DNFR scheme with $(C_1, C_2, \chi_D) = \left(C_D, \frac{2n_\omega-3}{2n_\omega-2}, 0\right)$, this constructs feasible solutions for (CIA-D), respectively (CIA-UD), with the objective bound $\frac{2n_\omega-3}{2n_\omega-2}(C_D + \bar{\Delta})$. We are left with the case $\chi_D = 1$:

1. If we set $C_1 = \frac{1}{2}C_D$, we have $\overline{\mathcal{L}} < \frac{1}{2}C_D + \bar{\Delta}$. With this MDT and the choice $\chi_D = 1$, the DNFR scheme constructs a feasible solution for (CIA-D). Then, by virtue of Theorem 7.4, case *II.*, with $C_2 = \frac{3}{2}$ we deduce the bound $\frac{3}{4}C_D + \frac{3}{2}\bar{\Delta}$.

2. a) If $C_U \geq C_D$ is given, we can set $C_1 = C_U$, and all block lengths are at least as large as those of the MD time C_D. Therefore, the binary control solution constructed by DNFR with $\chi_D = 0$ and $C_2 = \frac{2n_\omega-3}{2n_\omega-2}$ fulfills both the MU and MD time constraints.

   b) We set $\chi_D = 1$, $C_1 = C_U$, and $C_2 = \frac{3}{2}$, when $C_D > C_U > C_D/2$ is given. By this choice, the solution of DNFR fulfills an MD time of $2C_U$ because

$$2\underline{\mathcal{L}} > 2C_U > 2C_D/2 = C_D.$$

   Furthermore, by setting $C_1 = C_U$, it is clear that $\boldsymbol{w}^{\text{DNFR}}$ does not violate the MU time.

   (c) $C_D/2 \geq C_U$: With down time configuration $\chi_D = 1$ and $C_1 = C_D/2 \geq C_U$, $C_2 = \frac{3}{2}$, DNFR can be executed without violating the MU time constraint. □

For the problems (CIA-D) and (CIA-UD), it is not as straightforward to obtain tightness results as for (CIA-U). Nevertheless, we discuss the quality of the bounds obtained in Proposition 7.3 in terms of the DNFR scheme.

Proposition 7.4 (Tightness of the bound for (CIA-D))
Assume the MD time constraint is active, i.e., $C_D > \underline{\Delta}$ is given. Then the following is true:

1. *The bound for (CIA-D) stated in Proposition 7.3 cannot be improved by the DNFR scheme with $\chi_D = 1$ for $n_\omega \geq 3$.*

2. *The bound for (CIA-D) is tight up to at most the constant $\frac{1}{4}C_D + \bar{\Delta}$.*

Proof. The assumption of an active MD time constraint ensures that the bound cannot be improved by the bound for MU times from Proposition 7.2. We again use the concept of a *minimal C_1-overlapping grid*, here with $C_1 = C_D/2$.

1. We want to prove that the DNFR scheme with $\chi_D = 1$ and $C_2 < \frac{3}{2}$ may provide solutions with a (CIA-D) objective value greater than $C_2\overline{\mathscr{L}}$. First consider $C_2 \leq \frac{3}{2} - \epsilon_1$, with $0 < \epsilon_1 \leq 0.5$. We present example values for $\boldsymbol{a}^* \in \mathscr{A}_N$ with a time horizon length of at least $12C_1$, so that by Lemma 7.5 at least six blocks with length $\overline{\mathscr{L}}$ exist. Let $0 < \epsilon_2 < \epsilon_1$ be small, and let the relaxed control values $\boldsymbol{a}^*$ be given as

$$(a^*_{i,b})_{i\in[n_\omega],b\in[n_b]} := \begin{pmatrix} 1 & 0.5-\epsilon_1+\epsilon_2 & 1-\epsilon_2 & 2\epsilon_1-2\epsilon_2 & 0.5 & 0.5 & \cdots & 0.5 \\ 0 & 0.5+\epsilon_1-\epsilon_2 & 0 & 1-2\epsilon_1+2\epsilon_2 & 0.5 & 0.5 & \cdots & 0.5 \\ 0 & 0 & \epsilon_2 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}.$$

   With these example values, we discuss the thereby constructed DNFR solution as well its objective quality.

   - First dwell time block: i_1 is 2-future forced and activated.
   - Second dwell time block: Both i_1 and i_2 are 4-future forced. The DNFR algorithm breaks ties arbitrarily, so activating i_2 is legitimate.
   - Third dwell time block: i_1 is *down time forbidden*, while i_2 is not *admissible*. DNFR therefore activates i_3.
   - Fourth dwell time block: i_1 is activated since it is *forced*.
   - Fifth dwell time block: In the meantime we have $\Theta_{i_1,4} = (0.5+\epsilon_1-2\epsilon_2)\overline{\mathscr{L}}$ and $\Theta_{i_2,4} = (0.5-\epsilon_1+\epsilon_2)\overline{\mathscr{L}}$. Since ϵ_2 satisfies $\epsilon_2 < \epsilon_1$, both controls are 6-future forced on the fifth block. Let DNFR activate i_2.
   - Sixth dwell time block: i_1 is still *down time forbidden* and cannot be active, which implies
   $$\Theta_{i_1,6} = (0.5+\epsilon_1-2\epsilon_2+1)\overline{\mathscr{L}} > (\tfrac{3}{2}-\epsilon_1)\overline{\mathscr{L}} = C_2\overline{\mathscr{L}},$$
   so the proposed control deviation bound is not fulfilled.

Finally, if $\epsilon_1 > 0.5$ and thus, $C_2 < 1$, we can construct a similar example for which the control i_1 is already forced on the first dwell time block and the control deviation again exceeds $C_2\overline{\mathscr{L}}$.

2. The MD time constraints are equivalent to MU time constraints with $C_U = C_D$ for a problem with only two controls $n_\omega = 2$. Proposition 7.2 provides an example for this case, where $\theta^* \geq \frac{1}{2}(C_U + \bar{\Delta})$ holds. This example can also be applied for more than two controls by setting the relaxed control values $a^*_{i,b}$ to zero, for $i > 2$. Then, the difference from the upper bound $\frac{3}{4}C_D + \frac{3}{2}\bar{\Delta}$ in Proposition 7.3 is as stated in the assertion. □

Proposition 7.4 tells us that the DNFR scheme with $C_2 = \frac{3}{2}$ and $\chi_D = 1$ cannot be improved. The following example motivates why we have chosen the set of the *down time forbidden control* in Definition 6.18 such that the active control can only be changed after a duration of $C_D/2$ at the earliest. If it is already possible to switch after one interval Δ_j, DNFR may construct greedy solutions with a large control deviation at long MD times C_D. The following example illustrates why this occurs.

Example 7.1 (Modified DNFR scheme iterating over intervals yields no improved bounds) Let $n_\omega = 3$ and a rounding threshold $C_2 \geq \frac{2n_\omega - 3}{2n_\omega - 2}$ be given. We alter DNFR in the following way: instead of iterating over dwell time blocks, we iterate forward over all intervals. For a given MD time $C_1 = C_D$, we keep the threshold $C_2\overline{\mathscr{L}}$ for forced, future forced, and admissible activation. To construct feasible solutions for (CIA-D), we extend Definition 6.18 of $\mathscr{I}_b^D$ by letting all controls that are inactive and were active in the previous period of length C_D be *down time forbidden.* Next, we construct exemplary relaxed values for this modified DNFR scheme with a large control deviation. We first recursively introduce the indices

$$j_i := \min\left\{ j \in [N] \mid \textstyle\sum_{l=j_{i-1}}^{j} \Delta_l > C_2\overline{\mathscr{L}} \right\}, \quad i = 1, 2, 3,$$

where $j_0 := 1$. Let the relaxed values be given as follows:

$$(a_{i,j})^*_{i\in[n_\omega], j\in[N]} := \begin{pmatrix} 1 & \cdots & \overbrace{1}^{j_1} & 0 & \cdots & \overbrace{0}^{j_2} & 0 & \cdots & \overbrace{0}^{j_3} & \cdots \\ 0 & \cdots & 0 & 1 & \cdots & 1 & 0 & \cdots & 0 & \cdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & 1 & \cdots & 1 & \cdots \end{pmatrix}.$$

Then, the modified DNFR can construct the following binary control:

$$(w_{i,j})_{i\in[n_\omega], j\in[N]} = \begin{pmatrix} 1 & 0 & 0 & \cdots & \overbrace{0}^{j_4} & \cdots \\ 0 & 1 & 0 & \cdots & 0 & \cdots \\ 0 & 0 & 1 & \cdots & 1 & \cdots \end{pmatrix},$$

where $j_4 := \max\{j \in [N] \mid \sum_{l\in[j]} \Delta_l < C_D + \Delta_1\}$ is the last index before the MD phase of i_1 ends. At first, i_1 is the earliest future forced control, but it is not after being active on $j = 1$. Then i_2 is activated on the second interval before i_3 becomes the earliest future forced control and needs to stay active until j_4 since the other controls are down time forbidden. We notice that with small Δ_1, Δ_2, it can result that $j_4 \leq j_2$, and therefore

$$|\theta_{i_3,j_4}| = \left| \sum_{l=3}^{j_4} (0-1)\Delta_l \right| \leq |C_D + \bar{\Delta} - \Delta_2| \leq C_D + \bar{\Delta} - \underline{\Delta}.$$

If we compare the term on the right side of the inequality with the bound from Proposition 7.3, i.e., $\frac{2n_\omega - 3}{2n_\omega - 2}(C_D + \bar{\Delta}) = \frac{3}{4}(C_D + \bar{\Delta})$, we conclude that the former is smaller only if $C_D < 4\underline{\Delta} - \bar{\Delta}$. Since $\underline{\Delta}$ can be arbitrarily small and we assumed C_D to be big compared with the grid length, the modified DNFR scheme will not construct improved bounds. Similar “greedy” examples can be constructed for $n_\omega > 3$ and dwell time block lengths greater than $\underline{\Delta}$.

Comments on these tightness properties are in order.

Remark 7.5 (Quality of the bound for (CIA-D))
The MD time configuration of DNFR, i.e., $\chi_D = 1$, only yields smaller upper bounds than the DNFR algorithm with MU time configuration, i.e. $\chi_D = 0$ and $C_1 = C_D$, for instances with more than three controls and a large MD time C_D compared with the grid length $\bar{\Delta}$. In fact, for any n_ω, we conjecture that the upper bound on (CIA-D) is $\theta^{\max} = \frac{1}{2}C_D + \bar{\Delta}$, that is, only slightly greater than that for $n_\omega = 2$. By taking $\frac{1}{2}C_D + \bar{\Delta}$ as an activation threshold as part of DNFR, there would be no forced control until the first down time forbidden control appears. We postulate that active controls that become forced without activation during the next C_D time duration may stay active without other controls becoming forced. Of course, this argumentation does not constitute a proof – together with Example 7.1, Proposition 7.4 states that a generic solution that fulfills this bound cannot be found by means of the DNFR scheme, and it is presumably hard, if not impossible, to construct it by another polynomial-time algorithm.

Remark 7.6 (Quality of the bound for (CIA-UD))
As stated in Proposition 7.3, the integral deviation gap bound for (CIA-UD) is tight for $C_U \geq C_D$ by the result of Proposition 7.2. For $C_U < C_D$, the bound is not necessarily tight, but it is again difficult to prove tight bounds due to the combinatorial structure of the problem.

Remark 7.7 (Implications of MDT as a multiple of the grid intervals)
If we deal with an MDT C_1 that begins and ends exactly on the grid points, the upper bounds become $\frac{2n_\omega - 3}{2n_\omega - 2}C_U$ for (CIA-U), $\frac{3}{4}C_D$ for (CIA-D), and are accordingly reduced for (CIA-UD).

7.5 (CIA) with bounded discrete total variation

This section establishes bounds on the integral deviation gap for (CIA-TV). For this, we use the AMDR algorithm and the associated concepts introduced in Section 6.7. In particular, we argue with *activation blocks*, which are not to be confused with dwell time blocks in the context of MDT, the corresponding lengths δ, and the switching interval variables τ, all of which were introduced in Definition 6.23. We prove the tightest possible upper bound on the integral deviation gap for equidistant discretization and the case of two binary controls in Corollary 7.3. We distinguish between the cases $n_\omega = 2$ and $n_\omega > 2$ in Subsections 7.5.1 and 7.5.2, respectively. This section is based on Sections 6 and 7 from [222].

7.5.1 Upper bounds on (CIA-TV) with $n_\omega = 2$

Here, we use the MDR algorithm and previous results from Section 6.7.3 in order to deduce bounds on (CIA-TV) for two binary control modes, i.e., $n_\omega = 2$. We consider a given (CIA-TV) problem with grid $\mathscr{G}_N$, relaxed value $\boldsymbol{a}^* \in \mathscr{A}_N$, and maximum number of switches $\sigma_{\max} > 0$. The idea in the following is to construct a generic control function $\boldsymbol{w}^{\text{MDR}}$ that bounds the objective of (CIA-TV). For finding an appropriate initial active control for the MDR scheme, we introduce an auxiliary grid $\widetilde{\mathscr{G}}_N$ that ends at $\tilde{t}_f$ and has $\tilde{N}$ intervals:

$$\widetilde{\mathscr{G}}_N := \mathscr{G}_N \cap \left[t_0, t_0 + \frac{5(t_f - t_0)}{3 + 2\sigma_{\max}} \right], \quad \tilde{N} := |\widetilde{\mathscr{G}}_N| - 1, \quad \tilde{t}_f := \max\{t_j \mid t_j \in \widetilde{\mathscr{G}}_N\}.$$

In the definition of $\widetilde{\mathscr{G}}_N$, we intersect two sets because we consider the given $\mathscr{G}_N$ and $\sigma_{\max}$. To specify the rounding down of a value $t_0 \leq t$ to the next grid point, we utilize the following brackets notation

$$\lfloor t \rfloor_{\mathscr{G}_N} := \max\{t_j \in \mathscr{G}_N \mid t_j \leq t\}.$$

If we are dealing with an equidistant grid, we can prove a sharp bound for (CIA-TV). We distinguish between the cases with and without an equidistant grid in the upcoming results and introduce the following constant:

$$\tilde{C}_1 := \begin{cases} \frac{1}{3+2\sigma_{\max}}, & \text{if } \mathscr{G}_N \text{ equidistant,} \\ 0, & \text{else.} \end{cases} \tag{7.29}$$

We propose applying the rounding threshold

$$\bar{\theta} := \frac{t_f - t_0}{3 + 2\sigma_{\max}} + \frac{1}{2}\bar{\Delta} - \frac{\tilde{C}_1}{2}\bar{\Delta} \tag{7.30}$$

in the MDR scheme, and claim that this choice is beneficial for proving upper bounds on (CIA-TV). Next, we establish useful properties of rounding to the next grid point $\lfloor \cdot \rfloor_{\mathscr{G}_N}$.

Lemma 7.7 (Distance to the next grid points for $\lfloor \cdot \rfloor_{\mathscr{G}_N}$)
Consider $\sigma_{\max} > 0$ and the rounding threshold $\bar{\theta}$ defined as in (7.30). *The following holds:*

1. $\lfloor t_0 + j\bar{\theta} \rfloor_{\mathscr{G}_N} \geq t_0 + j\bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}$, *for* $j \in [2]$,
2. $\left\lfloor t_0 + \frac{5(t_f - t_0)}{3+2\sigma_{\max}} \right\rfloor_{\mathscr{G}_N} \geq t_0 + \frac{5(t_f - t_0)}{3+2\sigma_{\max}} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}$.

Proof. 1. We first consider the non-equidistant case. If $t_0 + j\bar{\theta} \leq t_f$, we deduce that the maximum distance from $t_0 + j\bar{\theta}$ to the next smaller or equal grid point is $\bar{\Delta}$. If $t_0 + j\bar{\theta} > t_f$, we have $\lfloor t_0 + j\bar{\theta} \rfloor_{\mathscr{G}_N} = t_f$, and obtain

$$t_0 + j\bar{\theta} \leq t_0 + 2\bar{\theta} \leq t_0 + 2\frac{t_f - t_0}{3 + 2 \cdot 1} + \bar{\Delta} = \frac{3}{5}t_0 + \frac{2}{5}t_f + \bar{\Delta} < t_f + \bar{\Delta}.$$

This settles the non-equidistant case: $\lfloor t_0 + j\bar{\theta} \rfloor_{\mathscr{G}_N} \geq t_0 + j\bar{\theta} - \bar{\Delta}$. For the equidistant case, we observe

$$\bar{\theta} = \frac{t_f - t_0}{3 + 2\sigma_{\max}} + \frac{1}{2}\bar{\Delta} - \frac{1}{2(3 + 2\sigma_{\max})}\bar{\Delta} = \frac{N\bar{\Delta} + (\sigma_{\max} + 1)\bar{\Delta}}{3 + 2\sigma_{\max}} = \frac{(N + \sigma_{\max} + 1)\bar{\Delta}}{3 + 2\sigma_{\max}}.$$

We note that the numerator of the right fraction consists of a product of an integer and $\bar{\Delta}$, whereas the denominator is the integer $3+2\sigma_{\max}$. Thus, the maximum cut-off by rounding down to the closest grid point is $\frac{3+2\sigma_{\max}-1}{3+2\sigma_{\max}}\bar{\Delta}$, which is equal to $\bar{\Delta}-\tilde{C}_1\bar{\Delta}$, proving the claim.

2. This follows from a similar argument as that for claim *1*. For the non-equidistant case, we only need to consider $t_0+\frac{5(t_f-t_0)}{3+2\sigma_{\max}}\leq t_f$; for the equidistant case, we again take advantage of $t_f-t_0=N\bar{\Delta}$. □

We continue with a lemma that quantifies the length of the activation blocks in $\boldsymbol{w}^{\mathrm{MDR}}$.

Lemma 7.8 (Length of activation blocks $\boldsymbol{\delta_l}$)
Consider a feasible control solution for (CIA–$\bar{\theta}$) that only uses canonical switches. Then, for the length of its activation block δ_l, $2\leq l\leq\sigma_{\max}$, it follows that

$$\delta_l \geq 2\bar{\theta}-\bar{\Delta}+C_1\bar{\Delta}.$$

Proof. Let i be the active control on activation block l. We use the assumption regarding canonical switches twice. First, control mode i is $\bar{\theta}$-forced for the earlier switch $l-1$:

$$\theta_{i,\tau_{l-1}-1}+a^*_{i,\tau_{l-1}}\Delta_{\tau_{l-1}}>\bar{\theta}, \tag{7.31}$$

and, second, it is $\bar{\theta}$-inadmissible on interval τ_l:

$$\theta_{i,\tau_{l-1}-1}+\sum_{j=\tau_{l-1}}^{\tau_l-1}(a^*_{i,j}-1)\Delta_j+(a^*_{i,\tau_l}-1)\Delta_{\tau_l}=\theta_{i,\tau_l-1}+(a^*_{i,\tau_l}-1)\Delta_{\tau_l}<-\bar{\theta}. \tag{7.32}$$

By Definition 6.23 of activation blocks, we have $\delta_l=\sum_{j=\tau_{l-1}}^{\tau_l-1}\Delta_j$, so by rearranging (7.32), we obtain

$$\delta_l>\theta_{i,\tau_{l-1}-1}+\bar{\theta}+\sum_{j=\tau_{l-1}}^{\tau_l-1}a^*_{i,j}\Delta_j+(a^*_{i,\tau_l}-1)\Delta_{\tau_l}.$$

Plugging (7.31) into the above inequality yields

$$\delta_l>2\bar{\theta}-a^*_{i,\tau_{l-1}}\Delta_{\tau_{l-1}}+\sum_{j=\tau_{l-1}}^{\tau_l-1}a^*_{i,j}\Delta_j+(a^*_{i,\tau_l}-1)\Delta_{\tau_l}=2\bar{\theta}+\sum_{j=\tau_{l-1}+1}^{\tau_l}a^*_{i,j}\Delta_j-\Delta_{\tau_l}\geq 2\bar{\theta}-\bar{\Delta},$$

which settles the non-equidistant case. For an equidistant grid, we compute

$$2\bar{\theta}-\bar{\Delta}=\frac{2N\bar{\Delta}}{3+2\sigma_{\max}}+\bar{\Delta}-\tilde{C}_1\bar{\Delta}-\bar{\Delta}=\frac{2N-1}{3+2\sigma_{\max}}\bar{\Delta},$$

and because δ_l is a multiple of $\bar{\Delta}$, it follows from $\delta_l>2\bar{\theta}-\bar{\Delta}$ that

$$\delta_l\geq\frac{2N-1}{3+2\sigma_{\max}}\bar{\Delta}+\frac{1}{3+2\sigma_{\max}}\bar{\Delta}=2\bar{\theta}-\bar{\Delta}+\tilde{C}_1\bar{\Delta}.$$

□

Next, in Algorithm 7.1, we propose a specification of the initial active control i_0 for the MDR scheme.
We observe that at most one switch occurs on $\widetilde{\mathscr{G}}_N$, as quantified in the following lemma.

Algorithm 7.1: Detecting the initial active control for MDR that results in at most one switch on $\widetilde{\mathscr{G}}_N$.

```
Input : Relaxed control values $\boldsymbol{a}^* \in \mathscr{A}_N$, where $n_\omega = 2$, rounding threshold $\bar{\theta}$ from (7.30).
if there is a control $i_1$ with $\sum_{j=1}^{\tilde{N}} a^*_{i_1,j}\Delta_j \le \bar{\theta}$ then
    Set $i_0 = i_2 \neq i_1$;
else if there is a control $i_1$ with $\sum_{j=1}^{\tilde{N}} a^*_{i_1,j}\Delta_j \le 2\bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}$ then
    Set $i_0 = i_1$;
else
    Set $i_0 = i_1$, where $i_1$ is a $\bar{\theta}$-next forced control on interval $j = 1$;
return: $i_0$ as initial control;
```

Lemma 7.9 ($\boldsymbol{w}^{\mathrm{MDR}}$ has at most one switch on $\widetilde{\mathscr{G}}_N$)
The MDR algorithm applied to the auxiliary grid $\widetilde{\mathscr{G}}_N$ with rounding threshold $\bar{\theta}$ from (7.30), *$n_\omega = 2$, and i_0 from Algorithm 7.1 as the initial control, constructs a control function $\boldsymbol{w}^{MDR}$ that uses at most one switch on $\widetilde{\mathscr{G}}_N$.*

Proof. We distinguish between the three possibilities for the initial control in Algorithm 7.1.

1. If MDR is initialized with i_2, and for i_1, $\sum_{j=1}^{\tilde{N}} a^*_{i_1,j}\Delta_j \le \bar{\theta}$ holds, i_1 does not become $\bar{\theta}$-*forced* on $\widetilde{\mathscr{G}}_N$. For this reason there is no switch.

2. If i_1 with $\sum_{j=1}^{\tilde{N}} a^*_{i_1,j}\Delta_j \le 2\bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}$ is the initial active control, a switch must occur when i_2 is $\bar{\theta}$-*forced* on some interval $\tau_1 \in [\tilde{N}]$. We need to prove that i_1 does not become $\bar{\theta}$-*forced* after the first switch. This is equivalent to i_2 not becoming $\bar{\theta}$-inadmissible due to $n_\omega = 2$ and Lemma 6.2 because in that case, there is no other switch. For this, we derive a lower bound on the length of the first activation block δ_1, where i_1 is active. At the earliest, the control i_1 becomes $\bar{\theta}$-*inadmissible* when it has been active on intervals j with $a^*_{i_1,j} = 0$ whose lengths sum up to be more than $\bar{\theta}$, i.e., a length of $\lfloor t_0 + \bar{\theta} \rfloor_{\mathscr{G}_N} - t_0$. With this observation and Lemma 7.7.1, we derive

$$\delta_1 = \sum_{j=1}^{\tau_1 - 1} \Delta_j \ge \lfloor t_0 + \bar{\theta} \rfloor_{\mathscr{G}_N} - t_0 \overset{\text{Lemma 7.7.1}}{\ge} \bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}.$$

Note that $\gamma_{i_1,j}$ is monotonically increasing with the increasing interval $j > \tau_1$ as long as i_1 is inactive, i.e., $w_{i_1,j-1} = 0$. Hence, if we are able to prove $\gamma_{i_1,\tilde{N}} \le \bar{\theta}$ when $w_{i_1,j} = 0$ for $j > \tau_1$, we also have that $\gamma_{i_1,j} \le \bar{\theta}$ for any $j > \tau_1$, meaning there is no second switch. Altogether, with the above inequality, we obtain

$$\gamma_{i_1,\tilde{N}} = \sum_{j=1}^{\tilde{N}} a^*_{i_1,j}\Delta_j - \delta_1 \le 2\bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta} - (\bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}) \le \bar{\theta},$$

so $\boldsymbol{w}^{\mathrm{MDR}}$ switches no more than once on $\widetilde{\mathscr{G}}_N$.

3. Otherwise, we have

$$\sum_{j=1}^{\tilde{N}} a^*_{i,j}\Delta_j > 2\bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}, \qquad \text{for} \quad i = i_1, i_2, \tag{7.33}$$

in the `else` case. We can argue similarly to the previous case, which is why we only have to prove $\gamma_{i_1,\tilde{N}} \leq \bar{\theta}$. Since control mode i_1 is a $\bar{\theta}$-*next forced control* on the first interval, there is an interval $l \leq \tau_1$ with $\sum_{j=1}^{l} a^*_{i_1,j}\Delta_j > \bar{\theta}$. This implies that the interval τ_1 of the earliest possible switch is given by

$$\tau_1 = \underset{l\in[\tilde{N}]}{\operatorname{argmin}} \left\{ \sum_{j=1}^{l} (a^*_{i_1,j} - 1)\Delta_j < -\bar{\theta} \;\middle|\; \sum_{j=1}^{l} a^*_{i_1,j}\Delta_j > \bar{\theta} \right\}$$

from which we find $\sum_{j=1}^{\tau_1}\Delta_j > 2\bar{\theta}$. We conclude for the grid point $t_{\tau_1} = \lfloor t_0 + \sum_{j=1}^{\tau_1}\Delta_j \rfloor_{\mathscr{G}_N} > \lfloor t_0 + 2\bar{\theta} \rfloor_{\mathscr{G}_N}$, which implies $t_{\tau_1 - 1} = \lfloor t_0 + \sum_{j=1}^{\tau_1 - 1}\Delta_j \rfloor_{\mathscr{G}_N} \geq \lfloor t_0 + 2\bar{\theta} \rfloor_{\mathscr{G}_N}$. This is equivalent to $\sum_{j=1}^{\tau_1-1}\Delta_j \geq \lfloor t_0 + 2\bar{\theta} \rfloor_{\mathscr{G}_N} - t_0$ and thus

$$\delta_1 = \sum_{j=1}^{\tau_1 - 1}\Delta_j \geq \lfloor t_0 + 2\bar{\theta} \rfloor_{\mathscr{G}_N} - t_0 \overset{\text{Lemma 7.7.1}}{\geq} 2\bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}. \tag{7.34}$$

Using the (Conv) property yields $\sum_{j=1}^{\tilde{N}} a^*_{i_1,j}\Delta_j = \sum_{j=1}^{\tilde{N}}\Delta_j - \sum_{j=1}^{\tilde{N}} a^*_{i_2,j}\Delta_j$, and therefore,

$$\begin{aligned}
\gamma_{i_1,\tilde{N}} &\leq \sum_{j=1}^{\tilde{N}} a^*_{i_1,j}\Delta_j - \delta_1 \overset{(7.34)}{\leq} \sum_{j=1}^{\tilde{N}}\Delta_j - \sum_{j=1}^{\tilde{N}} a^*_{i_2,j}\Delta_j - (2\bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}) \\
&\overset{(7.33)}{<} \tilde{t}_f - t_0 - (2\bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}) - (2\bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}) \\
&= \left\lfloor t_0 + \frac{5(t_f - t_0)}{3 + 2\sigma_{\max}} \right\rfloor_{\mathscr{G}_N} - t_0 - \frac{4(t_f - t_0)}{3 + 2\sigma_{\max}} \\
&\leq t_0 + \frac{5(t_f - t_0)}{3 + 2\sigma_{\max}} - t_0 - \frac{4(t_f - t_0)}{3 + 2\sigma_{\max}} \\
&= \frac{(t_f - t_0)}{3 + 2\sigma_{\max}} \\
&< \bar{\theta},
\end{aligned}$$

where we used $\tilde{t}_f = \left\lfloor t_0 + \frac{5(t_f - t_0)}{3 + 2\sigma_{\max}} \right\rfloor_{\mathscr{G}_N}$ in the third equation. To conclude, there is again at most one switch. □

The above three lemmata are crucial for the following theorem, which provides an upper bound on the integral deviation gap for (CIA-TV).

Theorem 7.5 (Bound on the integral deviation gap for (CIA-TV) and $n_\omega = 2$)
Consider any grid $\mathscr{G}_N$, relaxed values $\boldsymbol{a}^* \in \mathscr{A}_N$, and maximum number of switches $\sigma_{\max} > 0$. The optimal objective value of (CIA-TV) is bounded by

$$\theta^* \leq \frac{N}{3 + 2\sigma_{\max}}\bar{\Delta} + \frac{1}{2}\bar{\Delta} - \frac{\tilde{C}_1}{2}\bar{\Delta}.$$

Proof. We want to prove that the control function $\boldsymbol{w}^{\text{MDR}}$ constructed by MDR with rounding threshold $\bar{\theta}$ from (7.30) and initial control from Algorithm 7.1 is feasible and that it satisfies the claimed bound. We observe $\bar{\theta} \geq \frac{1}{2}\bar{\Delta}$ from its definition in (7.30) and the definition of $\tilde{C}_1$ in (7.29). Thus, we can apply Proposition 6.3 in connection with Lemma 6.3 so that $\boldsymbol{w}^{\text{MDR}}$ indeed fulfills the claimed bound:

$$\theta(\boldsymbol{w}^{\text{MDR}}) \leq \bar{\theta} = \frac{t_f - t_0}{3 + 2\sigma_{\max}} + \frac{1}{2}\bar{\Delta} - \frac{\tilde{C}_1}{2}\bar{\Delta}.$$

What remains to be shown is that $\boldsymbol{w}^{\text{MDR}}$ is a feasible solution for (CIA-TV), i.e., that it does not use more than $\sigma_{\max}$ switches. In the sequel, we write $n = \sigma_{\max}$ in the variable indices to improve the readability of the latter. We assume that $\sigma_{\max}$ switches have already been taken in $\boldsymbol{w}^{\text{MDR}}$ and calculate the maximum length of a final possible activation block, i.e., $\delta_{n+1} = t_f - t_{\tau_n - 1}$. In Lemma 7.9, we derived that at most one switch is used until $\tilde{t}_f$ on the reduced grid $\widetilde{\mathscr{G}}_N$, but another switch may follow shortly afterward, i.e., $\tau_2 \geq \tilde{N} + 1$. For the remaining $\sigma_{\max} - 2$ activation blocks until $t_{\tau_n - 1}$, we can apply Lemma 7.8 since Proposition 6.3 states that MDR uses *canonical* switches for $n_\omega = 2$. Lemma 7.8 establishes that

$$\delta_l \geq 2\theta - \bar{\Delta} + C_1\bar{\Delta}, \qquad \text{for } 3 \leq l \leq \sigma_{\max}.$$

Combining these findings and using Lemma 7.7.2 results in

$$\begin{aligned}
t_f - t_{\tau_n - 1} &= t_f - \sum_{j=1}^{\sigma_{\max}} \delta_j \leq t_f - \left[\tilde{t}_f + (\sigma_{\max} - 2)(2\bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta})\right] \\
&\leq t_f - \left\lfloor t_0 + \frac{5(t_f - t_0)}{3 + 2\sigma_{\max}} \right\rfloor_{\mathscr{G}_N} - (\sigma_{\max} - 2)(2\bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}) \\
&\leq t_f - (t_0 + \frac{5(t_f - t_0)}{3 + 2\sigma_{\max}} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}) - (\sigma_{\max} - 2)(2\bar{\theta} - \bar{\Delta} + \tilde{C}_1\bar{\Delta}) \\
&= \frac{(3 + 2\sigma_{\max})(t_f - t_0)}{3 + 2\sigma_{\max}} - \frac{5(t_f - t_0)}{3 + 2\sigma_{\max}} + \bar{\Delta} - \tilde{C}_1\bar{\Delta} - (\sigma_{\max} - 2)\left(\frac{2(t_f - t_0)}{3 + 2\sigma_{\max}}\right) \\
&= \frac{2(t_f - t_0)}{3 + 2\sigma_{\max}} + \bar{\Delta} - \tilde{C}_1\bar{\Delta}.
\end{aligned} \tag{7.35}$$

Let i denote the control mode that is active after the $\sigma_{\max}$th switch of $\boldsymbol{w}^{\text{MDR}}$. Note that $\theta_{i,j}$ is monotonically decreasing with the increasing interval $j \geq \tau_n$ since i is chosen to be active on j. Hence, if we are able to show that control i is $\bar{\theta}$*-admissible* on interval N, then it is also $\bar{\theta}$*-admissible* on earlier intervals, and there will be no further switch until N. For this, let us assume control i is $\bar{\theta}$*-inadmissible* on interval N. We obtain

$$\begin{aligned}
-\bar{\theta} &> \theta_{i,N-1} + (a^*_{i,N} - 1)\Delta_N \\
&= \underbrace{\theta_{i,\tau_n - 1} + a^*_{i,\tau_n}\Delta_{\tau_n}}_{>\bar{\theta}} - \Delta_{\tau_n} + \sum_{j=\tau_n+1}^{N-1} (a^*_{i,j} - 1)\Delta_j + (a^*_{i,N} - 1)\Delta_N \\
&> \bar{\theta} + \sum_{j=\tau_n+1}^{N} a^*_{i,j}\Delta_j - \sum_{j=\tau_n}^{N} \Delta_j
\end{aligned}$$

(Continuation of the estimate of $-\bar{\theta} > \ldots$):

$$\begin{aligned}
&\geq \bar{\theta} + 0 - (t_f - t_{\tau_n - 1}) \\
&\overset{(7.35)}{\geq} \bar{\theta} - \left(\frac{2(t_f - t_0)}{3 + 2\sigma_{\max}} + \bar{\Delta} - \tilde{C}_1 \bar{\Delta} \right) \\
&= -\left(\frac{t_f - t_0}{3 + 2\sigma_{\max}} \right) - \frac{1}{2}\bar{\Delta} + \frac{\tilde{C}_1}{2}\bar{\Delta} \\
&= -\bar{\theta}. \qquad ↯
\end{aligned}$$

In the second inequality, we used that control i is $\bar{\theta}$*-forced* on interval τ_n of the nth, respectively $\sigma_{\max}$th, switch. With this contradiction, there cannot be a further switch after τ_n. In other words, $\boldsymbol{w}^{\text{MDR}}$ uses at most $\sigma_{\max}$ switches and is a feasible solution of (CIA-TV). This completes the proof. □

The obvious question arises of whether the upper bound from Theorem 7.5 is sharp. The following theorem gives a positive answer. We exclude the case $\sigma_{\max} \geq N - 1$ because otherwise, the TV constraints (4.9)-(4.10) would be no longer restrictive.

Theorem 7.6 (Lower bound on $\theta^{\max}$ for (CIA-TV) and $n_\omega = 2$)
For $N, \sigma_{\max} \in \mathbb{N}$, where $1 \leq \sigma_{\max} \leq N - 2$, there is an equidistant grid $\mathcal{G}_N$ and an $\boldsymbol{a}^* \in \mathcal{A}_N$ such that (CIA-TV) has an optimal objective value of

$$\theta^* \geq \left\lceil \frac{N}{3 + 2\sigma_{\max}} \right\rceil_{0.5} \bar{\Delta}. \tag{7.36}$$

Proof. If $\sigma_{\max} + 2 \leq N < 3 + 2\sigma_{\max}$, we can define $\boldsymbol{a}^*$ by specifying the values of control mode i_1 for the intervals $j \in [N]$ as

$$a^*_{i_1,j} := \begin{cases} 1, & \text{if } j \text{ odd}, \\ 0, & \text{if } j \text{ even}. \end{cases}$$

Since there are more intervals N than the maximum number of switches $\sigma_{\max}$ plus one, there is an interval j on which the optimal solution $\boldsymbol{w}$ of (CIA-TV) has the value $w_{i_1,j} = 1$, while $a^*_{i_1,j} = 0$ holds. This results in $\theta^* \geq 1\bar{\Delta} = \lceil \frac{N}{3+2\sigma_{\max}} \rceil_{0.5} \bar{\Delta}$.
Otherwise, if $N \geq 3 + 2\sigma_{\max}$, we proceed as follows:

1. We construct a specific matrix $\boldsymbol{a}^*$ that depends on the choice of $\sigma_{\max}$ and N.
2. We prove that for both initial active controls, for this $\boldsymbol{a}^*$ matrix and with a rounding threshold of

$$\bar{\theta} := \left\lceil \frac{N}{3 + 2\sigma_{\max}} \right\rceil_{0.5} \bar{\Delta} - \epsilon, \qquad \text{for any } 0 < \epsilon < \left\lceil \frac{N}{3 + 2\sigma_{\max}} \right\rceil_{0.5} \bar{\Delta}, \tag{7.37}$$

the MDR scheme constructs control functions $\boldsymbol{w}^{\text{MDR}}$ that use more than $\sigma_{\max}$ switches. Then, we return to the idea of the AMDR scheme and Theorem 6.3.1., which states that $\boldsymbol{w}^{\text{AMDR}}$ is feasible for (CIA-TV), i.e., that it uses at most $\sigma_{\max}$ switches, and conclude

$$\left\lceil \frac{N}{3 + 2\sigma_{\max}} \right\rceil_{0.5} \bar{\Delta} \leq \theta\left(\boldsymbol{w}^{\text{AMDR}}\right).$$

Theorem 6.3, 2. (a), also provides a statement about the relation of AMDR to the optimal solution of (CIA-TV):

$$\theta\left(\boldsymbol{w}^{\text{AMDR}}\right) \le \theta^* + TOL.$$

Because the tolerance TOL can be arbitrarily small, we conclude that the optimal solution of (CIA-TV) involves an objective value of at least $\left\lceil \frac{N}{3+2\sigma_{\max}} \right\rceil_{0.5} \bar{\Delta}$.

1. We reuse the notation of R from Lemma 7.6 and introduce the auxiliary constant $n_I \in \mathbb{N}$:

$$R := \frac{N}{3+2\sigma_{\max}}, \qquad n_I := \left\lfloor \frac{N - \lceil R \rceil}{\sigma_{\max}+1} \right\rfloor.$$

Next, we are interested in designing a specific $\boldsymbol{a}^* \in \mathscr{A}_N$ that has the property of enforcing an improper covering by any $\boldsymbol{w} \in \Omega_N$ that involves a (CIA-TV) objective value of at most $\bar{\theta}$. By improper covering, we mean that $\boldsymbol{w} \in \Omega_N$ has to use more than $\sigma_{\max}$ switches in order to yield the desired (CIA-TV) objective value of at most $\bar{\theta}$. We create sets of consecutive intervals for $\boldsymbol{a}^*$ on which either $\boldsymbol{a}^*_{i_1,\cdot}$ or $\boldsymbol{a}^*_{i_2,\cdot}$ is set to one (and the other control is thereby set to zero). Here, we call these sets of consecutive intervals with the same value *index sections*. We generate $\sigma_{\max}+2$ index sections, where the two control modes are alternately set to one in $\boldsymbol{a}^*$, implementing the idea that a feasible solution $\boldsymbol{w}$ of (CIA-TV) with at most $\sigma_{\max}$ switches shall contain at most $\sigma_{\max}+1$ activation blocks. The first index section includes $\lfloor R \rfloor$ intervals, followed by index sections with n_I intervals. The last index section arises from the remaining intervals until N is reached. After conveying some intuition of the specific $\boldsymbol{a}^* \in \mathscr{A}_N$, we continue with a technical definition of the index set $\mathscr{J}^{i_1}$ that specifies the index sections on which $\boldsymbol{a}^*_{i_1,\cdot}$ is set to one:

$$\mathscr{J}^{i_1}_{\text{even}} := [\lfloor R \rfloor] \cup \left\{ j \mid \lceil R \rceil + (2k-1)n_I + 1 \le j \le \lceil R \rceil + 2kn_I,\ k \in [\lfloor \sigma_{\max}/2 \rfloor] \right\},$$

$$\mathscr{J}^{i_1} := \begin{cases} \mathscr{J}^{i_1}_{\text{even}}, & \text{if } \sigma_{\max} \text{ is even}, \\ \mathscr{J}^{i_1}_{\text{even}} \cup \{ j \mid \lceil R \rceil + (2\lfloor \sigma_{\max}/2 \rfloor + 1)n_I + 1 \le j \le N \}, & \text{if } \sigma_{\max} \text{ is odd}. \end{cases}$$

With these definitions, we introduce $\boldsymbol{a}^*$ by fixing the values of control i_1.

$$a^*_{i_1,j} = \begin{cases} 1, & \text{if } j \in \mathscr{J}^{i_1}, \\ 0.5, & \text{if } j = \lceil R \rceil, \text{ and } \lfloor R \rfloor < R \le \lfloor R \rfloor + 0.5, \\ 1, & \text{if } j = \lceil R \rceil, \text{ and } R > \lfloor R \rfloor + 0.5, \\ 0, & \text{else.} \end{cases} \tag{7.38}$$

The value of $\boldsymbol{a}^*_{i_1,\cdot}$ on the $(\lceil R \rceil)$th interval may seem unintuitive in the second and third case. The idea of this construction is that it results in $\theta_{i_1,\lceil R \rceil} = \lceil R \rceil_{0.5} \bar{\Delta}$ if control i_1 is neither active on the first index section nor on the $(\lceil R \rceil)$th interval. In this way, control i_1 needs to already be active on the first index section to maintain an (CIA-TV) objective value of at most $\bar{\theta}$.

2. We want to prove that the MDR scheme with the rounding threshold from (7.37) and with $\boldsymbol{a}^*$ defined in (7.38) constructs a control function that uses more than $\sigma_{\max}$ switches, independent of the initial active control. For this, we are going to establish the following claim:

a) If i_1 is the initial active control, the kth switch of $\boldsymbol{w}^{\text{MDR}}$ happens before the $(\lceil R \rceil + kn_I)$th interval, where $k \in [\sigma_{\max}+1]$.

b) If i_2 is the initial active control, the kth switch of $\boldsymbol{w}^{\mathrm{MDR}}$ happens before the $(\lceil R\rceil + (k-1)n_I)$th interval, where $k \in [\sigma_{\max}+1]$.

Assuming that the claim is true, $\boldsymbol{w}^{\mathrm{MDR}}$ indeed uses more than $\sigma_{\max}$ switches because the $(\lceil R\rceil + (\sigma_{\max}+1)n_I)$th interval exists; i.e., it is smaller than or equal to N:

$$\lceil R\rceil + (\sigma_{\max}+1)n_I = \lceil R\rceil + (\sigma_{\max}+1)\left\lfloor \frac{N-\lceil R\rceil}{\sigma_{\max}+1}\right\rfloor \le \lceil R\rceil + (\sigma_{\max}+1)\frac{N-\lceil R\rceil}{\sigma_{\max}+1} = N.$$

The above inequality shows that there are indeed $\sigma_{\max}+2$ index sections for $\boldsymbol{a}^*$ as described above. With this information, we deduce that $\bar{\theta} < \frac{1}{2}\bar{\Delta}$ directly results in more than $\sigma_{\max}$ switches or in control solutions that do not satisfy the claimed optimal (CIA-TV) objective value from (7.36):

- If $\boldsymbol{a}^*$ consists only of zeros and ones and $\bar{\theta} < \frac{1}{2}\bar{\Delta}$, the MDR algorithm creates switches on all intervals j for which $\boldsymbol{a}^*_{\cdot,j} \ne \boldsymbol{a}^*_{\cdot,j-1}$ holds true. In this way, the activation blocks of $\boldsymbol{w}^{\mathrm{MDR}}$ match the index sections of $\boldsymbol{a}$, i.e. $\boldsymbol{w}^{\mathrm{MDR}} = \boldsymbol{a}^*$. Because we derived $\sigma_{\max}+2$ index sections for $\boldsymbol{a}$, there are $\sigma_{\max}+2$ blocks for $\boldsymbol{w}^{\mathrm{MDR}}$ and therefore, $\sigma_{\max}+1$ switches.

- If $a^*_{i_1,\lceil R\rceil} = 0.5$, then there is no $\boldsymbol{w}$ with $\theta(\boldsymbol{w}) < \frac{1}{2}\bar{\Delta}$ regardless of which control is active on interval $\lceil R\rceil$ since $\boldsymbol{a}^*$ is either zero or one on all other intervals. Hence, we can exclude the case $\bar{\theta} < \frac{1}{2}\bar{\Delta}$ from further consideration.

Thus, we are left with the case $\bar{\theta} \ge \frac{1}{2}\bar{\Delta}$, for which we can apply Proposition 6.3 and conclude that we deal only with canonical switches. We now return to proving the claim and proceed via induction.
Base case:
a) We consider $k=1$ and conclude from $N \ge 3+2\sigma_{\max}$ that $\lceil R\rceil_{0.5} \ge 1$ holds. Plugging this into inequality (7.3) from Lemma 7.6 results in $\lceil R\rceil_{0.5} < n_I$, and thus

$$\bar{\theta} < n_I\bar{\Delta}. \tag{7.39}$$

By construction of $\boldsymbol{a}^*$, the values $a^*_{i_1,j}$ are equal to one for $1 \le j \le \lfloor R\rfloor$. The value $a^*_{i_1,\lceil R\rceil}$ is either 0.5 or 1. Therefore, $-0.5\bar{\Delta} \le \theta_{i_1,\lceil R\rceil} \le 0$ holds for the accumulated control deviation of $\boldsymbol{w}^{\mathrm{MDR}}$ with i_1 as the initial active control. After the $(\lceil R\rceil)$th interval, n_I intervals follow on which $a^*_{i_1,j}$ is zero. We conclude that i_1 becomes $\bar{\theta}$-inadmissible before interval $\lceil R\rceil + n_I$ by (7.39) and that the first switch thus appears before this interval.
b) We demonstrate the claim for the first two switches because we are interested in a switch that occurs after interval $\lceil R\rceil$ in the inductive step. Let $k=1$. Since

$$\sum_{j=1}^{\lceil R\rceil}(a^*_{i_2,j}-1)\bar{\Delta} = (\lceil R\rceil - \lceil R\rceil_{0.5} - \lceil R\rceil)\,\bar{\Delta} < -\lceil R\rceil_{0.5}\,\bar{\Delta} + \epsilon = -\bar{\theta},$$

we conclude that control i_2 becomes $\bar{\theta}$-inadmissible at the latest on interval $\lceil R\rceil$ when it is the initial active control and equivalently, that $\boldsymbol{w}^{\mathrm{MDR}}$ has a switch on interval $\lceil R\rceil$ at the latest. This is equivalent to at least one activation of control i_1 up to and including interval $\lceil R\rceil$, which we use for proving the assertion for $k=2$. Assume that the second switch happens on or after interval $\lceil R \rceil + n_I$. This implies that i_1 is $\bar{\theta}$-admissible on that interval, and we derive

$$\begin{aligned}
-\bar{\theta} \le \theta_{i_1, \lceil R \rceil + n_I} \quad &= \quad \theta_{i_1, \lceil R \rceil} + \sum_{l = \lceil R \rceil + 1}^{\lceil R \rceil + n_I} (0-1)\bar{\Delta} \\
&\overset{\text{Case } k=1}{\le} \quad \lceil R \rceil_{0.5}\, \bar{\Delta} - \bar{\Delta} - n_I \bar{\Delta} \\
&\overset{\text{Lemma 7.6}}{\le} \quad \lceil R \rceil_{0.5}\, \bar{\Delta} - \bar{\Delta} - (2\lceil R \rceil_{0.5} - 1)\bar{\Delta} \\
&= \quad -\lceil R \rceil_{0.5}\, \bar{\Delta} < -\bar{\theta}. \qquad \lightning
\end{aligned}$$

Consequently, the second switch happens before the $(\lceil R \rceil + n_I)$th interval.

Inductive step:
Assume that the assertion holds for $k-1 \le \sigma_{\max}$; we show that it is also true for k. We first prove an auxiliary result. For $i \in [2]$ and $j \ge \lceil R \rceil$, we have that

$$\theta_{i,j} = \lceil R \rceil_{0.5}\, \bar{\Delta} + z\bar{\Delta}, \qquad \text{for some } \; z \in \mathbb{Z}. \tag{7.40}$$

We prove Equation (7.40) by computing the accumulated control deviation:

$$\theta_{i_1,j} = \bar{\Delta}\left(\sum_{l=1}^{\lceil R \rceil} a^*_{i_1,l} + \sum_{l=1+\lceil R \rceil}^{j} a^*_{i_1,l} - \sum_{l=1}^{j} w_{i_1,l}\right) = \lceil R \rceil_{0.5}\, \bar{\Delta} + \left(\sum_{l=1+\lceil R \rceil}^{j} a^*_{i_1,l} - \sum_{l=1}^{j} w_{i_1,l}\right)\bar{\Delta}.$$

For $j > \lceil R \rceil$ we defined $a^*_{i_1,j} \in \{0,1\}$, so (7.40) holds with $z = \left(\sum_{l=1+\lceil R \rceil}^{j} a^*_{i_1,l} - \sum_{l=1}^{j} w_{i_1,l}\right)$. On the other hand, for the other control i_2, it holds that

$$\theta_{i_2,j} = \bar{\Delta}\left(\sum_{l=1}^{\lceil R \rceil} a^*_{i_2,l} + \sum_{l=1+\lceil R \rceil}^{j} a^*_{i_2,l} - \sum_{l=1}^{j} w_{i_2,l}\right) = (\lceil R \rceil - \lceil R \rceil_{0.5})\, \bar{\Delta} + \left(\sum_{l=1+\lceil R \rceil}^{j} a^*_{i_2,l} - \sum_{l=1}^{j} w_{i_2,l}\right)\bar{\Delta},$$

and therefore, (7.40) is satisfied with $z = \left(\lceil R \rceil - 2\lceil R \rceil^{0.5} + \sum_{l=1+\lceil R \rceil}^{j} a^*_{i_2,l} - \sum_{l=1}^{j} w_{i_2,l}\right)$.

To make use of the established auxiliary result for the induction step, we need to argue that the $(k-1)$st switch happens after the interval $\lceil R \rceil$. In case a), the MDR algorithm will not deactivate i_1 since $a^*_{i_1,j} = 1$ before the $\lceil R \rceil$th interval. So it does on the $\lceil R \rceil$th interval if $a^*_{i_1,\lceil R \rceil} = 0.5$ because we have established $\bar{\theta} \ge \frac{1}{2}\bar{\Delta}$. In case b), we use the base case for the second switch. We consider the interval τ_1 of the first switch in case a) and compare the two accumulated control deviations for cases a) and b) on τ_1: we obtain $\theta_{i_1,\tau_1}(b) \ge \theta_{i_1,\tau_1}(a)$ because i_2 has already been activated in case b) unlike in case a). Since $\tau_1 > \lceil R \rceil$, we are done.

Now, without loss of generality, let i_1 be the active control after the switch on interval τ_{k-1}. We know that i_2 is active, and thus admissible, on interval $\tau_{k-1} - 1$:

$$-\bar{\theta} \le \theta_{i_2, \tau_{k-1}-1},$$

which for the control mode i_1 implies the following by Lemma 6.2:

$$\theta_{i_1,\tau_{k-1}-1} \le \bar{\theta} = \lceil R \rceil_{0.5}\,\bar{\Delta} - \epsilon.$$

We exploit the above inequality in the sense that by equation (7.40), for some $z_{i_1} \ge 1$, we have

$$\theta_{i_1,\tau_{k-1}-1} = \lceil R \rceil_{0.5}\,\bar{\Delta} - z_{i_1}\bar{\Delta} \le (\lceil R \rceil_{0.5} - 1)\bar{\Delta}. \tag{7.41}$$

The control i_2 is $\bar{\theta}$-inadmissible on interval τ_{k-1} as there are only canonical switches. If $a^*_{i_2,\tau_{k-1}} = 1$ were true, then i_2 would have already been $\bar{\theta}$-inadmissible on interval $\tau_{k-1} - 1$. Furthermore, $a^*_{i_2,\tau_{k-1}} = 0.5$ is not possible because we derived $\tau_{k-1} > \lceil R \rceil$. We conclude that $a^*_{i_2,\tau_{k-1}} = 0$. From this and the inductive hypothesis, which states that the $(k-1)$st switch appears before the $(\lceil R \rceil + (k-1)n_I)$th interval, it follows that $a^*_{i_1,j} = 1$ for the intervals j between τ_{k-1} and $(\lceil R \rceil + (k-1)n_I)$. Hence, $\theta_{i_1,\lceil R \rceil+(k-1)n_I} \le (\lceil R \rceil_{0.5} - 1)\bar{\Delta}$ holds due to (7.41). Finally, we assume that i_1 can stay active up to and including interval $\lceil R \rceil + kn_I$ without becoming *$\bar{\theta}$-inadmissible*. This and $a^*_{i_1,j} = 0$ for $\lceil R \rceil + (k-1)n_I + 1 \le j \le \lceil R \rceil + kn_I$ imply that

$$\begin{aligned}
-\bar{\theta} &\le \theta_{i_1,\lceil R \rceil+kn_I} \\
&= \theta_{i_1,\lceil R \rceil+(k-1)n_I} + \sum_{l=\lceil R \rceil+(k-1)n_I+1}^{\lceil R \rceil+kn_I} (a^*_{i_1,l} - 1)\bar{\Delta} \\
&\le (\lceil R \rceil_{0.5} - 1)\bar{\Delta} + 0\bar{\Delta} - n_I\bar{\Delta} \\
&\overset{\text{Lemma 7.6}}{\le} (\lceil R \rceil_{0.5} - 1)\bar{\Delta} + 0 - (2\lceil R \rceil_{0.5} - 1)\bar{\Delta} \\
&< -\bar{\theta} \qquad ↯.
\end{aligned}$$

Thus, control i_1 is *not* active until the $(\lceil R \rceil + kn_I)$th interval, and, with an analogous computation for case b), control i_1 is *not* active until the $(\lceil R \rceil + (k-1)n_I)$th interval. We have thereby shown that the assertion holds for k. Altogether, the constructed control function $\boldsymbol{w}^{\mathrm{MDR}}$ uses more than $\sigma_{\max}$ switches for the chosen rounding threshold $\bar{\theta}$ so that the optimal (CIA-TV) objective value is at least $\bar{\theta}$, and we conclude that the claimed theorem is true. □

We complete this subsection by concluding Theorems 7.5 and 7.6.

Corollary 7.3 (Tightest possible bound on the integral deviation gap for (CIA-TV) and $n_\omega = 2$) *Consider an equidistant grid $\mathcal{G}_N$, $\boldsymbol{a}^* \in \mathcal{A}_N$, and $1 \le \sigma_{\max} \le N-2$. The optimal objective of (CIA-TV) is bounded by*

$$\theta^* \le \frac{N + \sigma_{\max} + 1}{3 + 2\sigma_{\max}}\bar{\Delta}, \tag{7.42}$$

which is the tightest possible bound.

Proof. The inequality (7.42) is achieved by applying Theorem 7.5 to the equidistant case and rearranging the terms:

$$\theta^* \le \left(\frac{N}{3+2\sigma_{\max}} + \frac{1}{2} - \frac{1}{2(3+2\sigma_{\max})}\right)\bar{\Delta} = \frac{N + \sigma_{\max} + 1}{3 + 2\sigma_{\max}}\bar{\Delta}.$$

It is the tightest possible bound by Theorem 7.6 and the case $N = k(3+2\sigma_{\max})+2+\sigma_{\max}$, $k \in \mathbb{N}_0$:

$$\theta^* \geq \left\lceil \frac{k(3+2\sigma_{\max})+2+\sigma_{\max}}{3+2\sigma_{\max}} \right\rceil_{0.5} \bar{\Delta} = (k+1)\bar{\Delta} = \frac{N+\sigma_{\max}+1}{3+2\sigma_{\max}} \bar{\Delta}. \qquad \square$$

7.5.2 Upper bounds on (CIA-TV) with $n_\omega > 2$

The number of feasible solutions (CIA-TV) increases significantly for $n_\omega > 2$ compared with the case $n_\omega = 2$, making it challenging to derive bounds for this setting. Recall the definition of $\theta^{\max}$ in (7.1), and here let it denote the maximum optimal objective value of any (CIA-TV) problem instance. First, we use known results to derive lower and upper bounds on $\theta^{\max}$. Then, we dedicate ourselves to the continuous relaxation of (CIA-TV), allowing us to prove a sharper lower bound. Based on this, we state a conjecture about the actual value of $\theta^{\max}$.

Corollary 7.4 (Lower bound on $\theta^{\max}$ for $n_\omega > 2$)
Let $1 \leq \sigma_{\max} \leq N-2$ and $n_\omega > 2$. For (CIA-TV) it holds that $\theta^{\max} \geq \frac{N+\sigma_{\max}+1}{3+2\sigma_{\max}}\bar{\Delta}$.

Proof. This bound was established in Theorem 7.6 and Corollary 7.3 for the case $n_\omega = 2$. The example provided in the proof of Theorem 7.6 can also be applied to the case $n_\omega > 2$ by setting the values of the relaxed controls $\boldsymbol{a}^*_{i,\cdot}$ to zero for all $i \in [n_\omega]$ with $i > 2$. □

Corollary 7.5 (Upper bound on $\theta^{\max}$ for $n_\omega > 2$)
Let $1 \leq \sigma_{\max} \leq N-2$ and $n_\omega > 2$. We have that $\theta^{\max} \leq \frac{2n_\omega-3}{2n_\omega-2}\left(\frac{t_f-t_0}{\sigma_{\max}+1} + \bar{\Delta}\right)$ holds for (CIA-TV).

Proof. In Proposition 7.1, we provided a sharp bound for (CIA-U):

$$\theta^* \leq \frac{2n_\omega-3}{2n_\omega-2}\left(C_U + \bar{\Delta}\right).$$

If, for a binary control solution $\boldsymbol{w}$, we require that an activated control remains active for a time period of at least $\frac{t_f-t_0}{\sigma_{\max}+1}$, at most $\sigma_{\max}$ switches take place. Thus, the TV constraints (4.9)–(4.10) serve as a relaxation of the MU time constraint. This implies that $\theta^{\max}$ for (CIA-TV) is smaller than or equal to $\theta^{\max}$ for (CIA-U). □

We tighten the above results by investigating the continuous version of the (CIA-TV) problem.

Definition 7.2 (CCIA-TV)
*Let $\boldsymbol{\alpha} \in \mathscr{A}$, a time horizon $\mathscr{T}$, and $\sigma_{\max} \in \mathbb{N}$ be given. Then, we define the continuous combinatorial integral approximation problem (**CCIA-TV**) subject to total variation (TV) constraints to be*

$$\min_{\theta \geq 0,\, \boldsymbol{\omega} \in \Omega} \quad \theta \tag{7.43}$$

$$\text{s.t.} \quad \theta \geq \pm \int_{t_0}^{t} [\alpha_i(s) - \omega_i(s)]\, ds, \quad \textit{for all } i \in [n_\omega],\ t \in \mathscr{T}, \tag{7.44}$$

$$\sigma_{\max} \geq TV(\boldsymbol{\omega}), \tag{7.45}$$

where TV is defined as in (3.5).

Obviously, the problem (CCIA-TV) is a reformulation of

$$\min_{\boldsymbol{\omega}\in\Omega}\max_{t\in\mathcal{T}}\left\|\int_{t_0}^{t}(\boldsymbol{\alpha}(s)-\boldsymbol{\omega}(s))\,\mathrm{d}s\right\|,\quad \text{s.t. TV constraint (7.45).}$$

We stress that for (CCIA-TV) the given data $\boldsymbol{\alpha}$ does not live in $\mathscr{A}_N$ but in $\mathscr{A}$, and analogously, we try to find a binary control function $\boldsymbol{\omega}\in\Omega$. We obtain a lower bound for the maximum optimal objective of (CCIA-TV) over all $\boldsymbol{\alpha}\in\mathscr{A}$ by constructing a specific instance, as indicated in the following proposition.

Proposition 7.5 (Lower bound on the maximum optimal objective of (CCIA-TV))
There is a relaxed control function $\boldsymbol{\alpha}\in\mathscr{A}$ and a time horizon $\mathcal{T}$ such that for the optimal objective value θ^* of (CCIA-TV) the following holds:

$$\theta^* \geq \begin{cases} \frac{t_f-t_0}{\sigma_{\max}+2}, & \textit{if}\, \sigma_{\max}\leq n_\omega-2,\\ \frac{t_f-t_0}{2\sigma_{\max}+4-n_\omega}, & \textit{else.}\end{cases} \tag{7.46}$$

Proof. We first prove the inequality for $\sigma_{\max}\leq n_\omega-2$. We abbreviate $\tilde{t}:=\frac{t_f-t_0}{\sigma_{\max}+2}$ and construct the following instance:

$$\alpha_i(t):=\begin{cases}1, & \text{for } t\in[t_0+(i-1)\cdot\tilde{t},\ t_0+i\cdot\tilde{t}),\\ 0, & \text{else,}\end{cases}\qquad \text{for all } i\in[\sigma_{\max}+2],$$
$$\alpha_i(t):=0 \qquad \text{for all } i=\sigma_{\max}+3,\dots,n_\omega, \text{ and } t\in\mathcal{T}.$$

We conclude that $\boldsymbol{\alpha}\in\mathscr{A}$ since the convex combination constraint is satisfied on the entire time horizon $\mathcal{T}$. It results that $\int_{\mathcal{T}}\alpha_i(s)\,\mathrm{d}s=\tilde{t}$ for all $i\in[\sigma_{\max}+2]$. Thus, we would need to activate each control ω_i for some time to achieve a smaller objective value than $\tilde{t}$. However, because the number of switches is restricted to be at most $\sigma_{\max}$, this is not possible. Hence, $\theta^*\geq\tilde{t}$.

Next, we consider $\sigma_{\max}>n_\omega-2$ and again construct a specific instance with the claimed objective value of $\theta^*\geq\frac{t_f-t_0}{2\sigma_{\max}+4-n_\omega}$. We abbreviate $\bar{t}:=\frac{t_f-t_0}{2\sigma_{\max}+4-n_\omega}$. Similar to the above example, we first let the relaxed controls α_i be active sequentially for a period of length $\bar{t}$. After all relaxed controls have been active once and while the end of the time horizon has not yet been reached, we activate each control for a period of length $2\bar{t}$ according to the ascending index $i\in[n_\omega]$ until we reach the end of the time horizon. We express this idea by introducing the domains of activation for all $i\in[n_\omega]$:

$$\begin{aligned}\mathcal{T}_i^A:=&[t_0+(i-1)\cdot\bar{t},\ t_0+i\cdot\bar{t})\\ &\cup\left\{\left[\,t_0+[n_\omega+2(j-1)]\bar{t},\ t_0+(n_\omega+2j)\bar{t}\right)\ \middle|\ j\in[\sigma_{\max}+2-n_\omega],\ j\equiv i \mod n_\omega\right\}.\end{aligned}$$

Based on these domains we define the functions $\alpha_i(t)$ via

$$\alpha_i(t):=\begin{cases}1, & \text{for } t\in\mathcal{T}_i^A,\\ 0, & \text{else,}\end{cases}\qquad \text{for all } i\in[n_\omega].$$

Fig. 7.2 provides a visualization of the specifically defined control function $\boldsymbol{\alpha}$, which depends

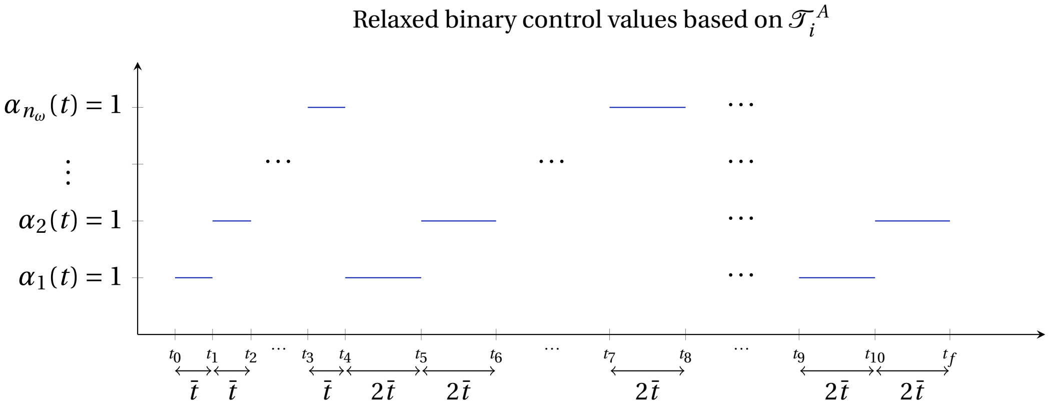


Figure 7.2: Exemplary visualization of the relaxed control function $\boldsymbol{\alpha}$ from the proof to Proposition 7.5; it results in an optimal (CCIA-TV) objective value of $\bar{t}$. Since $i = 2$ is the last activated control mode in this example, the maximum number of allowed switches is $\sigma_{\max} = kn_\omega + 2 - 2 = kn_\omega$, for $k \geq 2$.

on $\sigma_{\max}$. We have

$$n_\omega\, \bar{t} + (\sigma_{\max} + 2 - n_\omega)2\bar{t} = t_f - t_0,$$

such that $\cup_{i\in[n_\omega]}\mathscr{T}_i^A = \mathscr{T}$ follows, and because the intervals $\mathscr{T}_i^A$ are all disjoint, we obtain $\alpha_i(t) = 1$ for exactly one control mode i and for all $t \in \mathscr{T}$. Hence, $\boldsymbol{\alpha} \in \mathscr{A}$. The next observation about $\boldsymbol{\alpha}$ is that it consists of $n_\omega + (\sigma_{\max} + 2 - n_\omega) = \sigma_{\max} + 2$ activation blocks (interpreted in this continuous setting), meaning that there are $\sigma_{\max} + 1$ changes of the active control. Now, assume we can approximate $\boldsymbol{\alpha}$ with a binary control function $\boldsymbol{\omega} \in \Omega$ that yields a (CCIA-TV) objective value that is less than $\bar{t}$. We have

$$\int\limits_{t_0}^{t_0+i\bar{t}} \alpha_i(t)\, \mathrm{d}t = \bar{t}, \qquad \text{for all } i \in [n_\omega].$$

Each control ω_i, $i \in [n_\omega]$ thus needs to be active for some time up to and including $t_0 + i\,\bar{t}$, resulting in at least $n_\omega - 1$ switches up to and including $t_0 + n_\omega\,\bar{t}$. Then, we have that

$$\int\limits_{t_0}^{t_0+n_\omega\bar{t}} \alpha_i(t) - \omega_i(t)\, \mathrm{d}t < \bar{t}, \qquad \text{for all } i \in [n_\omega].$$

This, and using that the next activation blocks of $\boldsymbol{\alpha}$ last for a period of $2\,\bar{t}$, implies that each control ω_i needs to be activated again up to and including $t_0 + (n_\omega + 2i)\bar{t}$. If it were possible for some control $i \in [n_\omega]$ to skip the activation of ω_i without violating the control deviation bound $\bar{t}$, it would result in

$$\left| \int\limits_{t_0}^{t_0+(n_\omega+2i)\bar{t}} \alpha_i(t) - \omega_i(t)\, \mathrm{d}t \right| < \bar{t},$$

while it would simultaneously hold that

$$\int_{t_0+(n_\omega+2(i-1))\bar{t}}^{t_0+(n_\omega+2i)\bar{t}} \omega_i(t)\ \mathrm{d}t = 0,$$

which implies

$$\left| \int_{t_0}^{t_0+(n_\omega+2(i-1))\bar{t}} \alpha_i(t) - \omega_i(t)\ \mathrm{d}t \right| > \bar{t}$$

because of $\int_{t_0+(n_\omega+2(i-1))\bar{t}}^{t_0+(n_\omega+2i)\bar{t}} \alpha_i(t)\ \mathrm{d}t = 2\,\bar{t}$. We apply this argument for all activation blocks of $\boldsymbol{\alpha}$ until t_f and conclude that $\boldsymbol{\omega}$ must use at least one switch for each activation block of $\boldsymbol{\alpha}$ after $t_0+n_\omega\,\bar{t}$, i.e., it must use at least $(\sigma_{\max}+2-n_\omega)$ switches. Overall, there are at least $n_\omega-1+(\sigma_{\max}+2-n_\omega)=\sigma_{\max}+1$ switches. Therefore, any $\boldsymbol{\omega}\in\Omega$ that uses at most $\sigma_{\max}$ switches involves an (CCIA-TV) objective value of at least $\bar{t}$, which settles the claim for the case $\sigma_{\max}>n_\omega-2$. □

(CIA-TV) can be interpreted as a discretized version of (CCIA-TV). Thereby, we deduce the following corollary.

Corollary 7.6 (Lower bound on $\theta^{\max}$ for $n_\omega > 2$ deduced from (CCIA-TV))
Let $1\le\sigma_{\max}\le N-2$ and $n_\omega>2$. For the maximum optimal objective value $\theta^{\max}$ of (CIA-TV), we obtain

$$\theta^{\max} \geq \begin{cases} \frac{t_f-t_0}{\sigma_{\max}+2}, & \textit{if}\, \sigma_{\max}\le n_\omega-2,\\ \frac{t_f-t_0}{2\sigma_{\max}+4-n_\omega}, & \textit{else.}\end{cases} \tag{7.47}$$

Proof. (CCIA-TV) is a relaxation of (CIA-TV) since every feasible solution of (CIA-TV) corresponds to a feasible solution of (CCIA-TV). Thus, the claim follows from Proposition 7.5. □

The lower bound in Corollary 7.6 is sharp in the sense that there are combinations of n_ω, $\sigma_{\max}$, and $\mathscr{G}_N$ such that $\theta^{\max}$ equals the claimed lower bound. The following example illustrates this relationship.

Example 7.2 (Tightness of the bound in Corollary 7.6)
Let the grid be equidistant with $N=3$, and assume $n_\omega=3$. Consider the following two instances:

$$(a^1_{i,j})_{i\in[3],j\in[3]} := \begin{pmatrix} 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}, \qquad (a^2_{i,j})_{i\in[3],j\in[3]} := \begin{pmatrix} 1 & 0.5 & 0\\ 0 & 0.25 & 0.5\\ 0 & 0.25 & 0.5\end{pmatrix}.$$

Consider $\sigma_{\max}=1$ in the first example. Then, $\theta^*=\bar{\Delta}$, and $\theta^{\max}\ge\bar{\Delta}$ follows from the above corollary. Any asymmetric modification of $(a^1_{i,j})$ with unequal control accumulation $\sum_{j=1}^3 a^1_{i_1,j} \neq \sum_{j=1}^3 a^1_{i_2,j}$ would result in a binary control function $\boldsymbol{w}^*$ that activates the control mode with the largest control accumulation and hence $\theta^*<\bar{\Delta}$. We conclude that the claimed bound is sharp, i.e., $\theta^{\max}=\bar{\Delta}$.

Assume $\sigma_{\max}=2$ for the second instance. Then, $w^*_{i,j}=1$ for $(i,j)=(1,1),(1,2),(2,3)$ and thus $\theta^*=0.75\bar{\Delta}$. Therefore, the bound in Corollary 7.6, which amounts to $\theta^{\max}\ge\frac{3}{5}\bar{\Delta}$, is not tight for this instance.

Finding the exact value of $\theta^{\max}$ is difficult due to the nonconvex objective and the tremendously increased number of different $\boldsymbol{\omega} \in \Omega$ when $n_\omega > 2$. Nevertheless, we conjecture that the lower bound in Proposition 7.5 cannot be improved. We recognize the symmetry of the constructed $\boldsymbol{\alpha}$ in the proof: any modification of $\boldsymbol{\alpha}$ that alters the length of its activation blocks would result either in fewer than $\sigma_{\max}+2$ activation blocks or in at least one block with a length smaller than its previous length. The length of the latter activation block would be smaller than $\frac{t_f-t_0}{2\sigma_{\max}+4-n_\omega}$ if the block were the activation of the first control, respectively smaller than $2\cdot\frac{t_f-t_0}{2\sigma_{\max}+4-n_\omega}$ else. Using the argumentation from the proof of Proposition 7.5, this would allow us to choose a control function $\boldsymbol{\omega} \in \Omega$ with a (CCIA-TV) objective value that is smaller than $\frac{t_f-t_0}{2\sigma_{\max}+4-n_\omega}$. Furthermore, we argue that the optimal objective value of (CCIA-TV) is smaller than that of (CIA-TV) by at most $\frac{1}{2}\bar{\Delta}$ because the switching times of the optimal $\boldsymbol{\omega} \in \Omega$ differ by at most one half of the maximum grid length from the optimal $\boldsymbol{w} \in \Omega_N$. We close this section by summarizing these thoughts in the following conjecture.

Conjecture 7.1 (True value of $\boldsymbol{\theta}^{\max}$ for (CIA-TV) with $\boldsymbol{n_\omega} > 2$)
Let $1 \leq \sigma_{\max} \leq N-2$. We conjecture that for the maximum optimal objective value $\theta^{\max}$ of all (CIA-TV) instances the following holds:

$$\theta^{\max} = \begin{cases} \frac{t_f-t_0}{\sigma_{\max}+2} + \frac{1}{2}\bar{\Delta}, & \text{if } \sigma_{\max} \leq n_\omega - 2, \\ \frac{t_f-t_0}{2\sigma_{\max}+4-n_\omega} + \frac{1}{2}\bar{\Delta}, & \text{else.} \end{cases} \tag{7.48}$$

7.6 Summary

This chapter established upper bounds on the integral deviation gap for the (CIA) problem itself and under MDT and TV constraints. We used the rounding algorithms DNFR and AMDR as tools for constructing generic binary control solutions that comprise a bounded integral deviation gap, thereby implying bounds for the integral deviation gaps of (CIA-UD) and (CIA-TV). We then investigated the tightness of the bounds by providing examples $\boldsymbol{a}^* \in \mathscr{A}_N$ that result in the proven integral deviation gap. We similarly examined the behavior of DSUR in terms of the integral deviation gap.

We summarize the established integral deviation gap bounds for the NFR and SUR schemes from the literature in the form of $\theta(\boldsymbol{w}) \leq C(n_\omega)\bar{\Delta}$ in Table 7.1.

	NFR	SUR	SUR with vanishing constraints
$C(n_\omega) =$	1, see [135],	$\sum_{i=2}^{n_\omega} \frac{1}{i}$, see [149],	$\lfloor n_\omega/2 \rfloor$, see [179].

Table 7.1: Integral deviation gap bounds from the literature for binary control approximation algorithms.

Table 7.2 lists the results from this chapter for the various (CIA) problem settings. We use the notation from Equation (7.1) for $\theta^{\max}$, indicating the maximum integral deviation gap over all instances $\boldsymbol{a} \in \mathscr{A}_N$.

We remark that an algorithm that solves (CIA) to optimality is a rounding gap consistent algorithm as introduced in Definition 4.7, while the problems with MDT or TV constraints do not satisfy this property. Consequently, the convergence property for the CIA decomposition

Problem type	Established result	Statement
(CIA)	$\theta^{\max} = \frac{2n_\omega-3}{2n_\omega-2}\bar{\Delta}$	Corollary 7.2
(CIA-U)	$\theta^{\max} = \frac{2n_\omega-3}{2n_\omega-2}(C_U+\bar{\Delta})$	Proposition 7.1-7.2
(CIA-D)	$\theta^{\max} \leq \min\left\{\frac{3}{4}C_D + \frac{3}{2}\bar{\Delta}, \frac{2n_\omega-3}{2n_\omega-2}\left(C_D+\bar{\Delta}\right)\right\}$	Proposition 7.3
(CIA-UD)		Proposition 7.3
if $C_U \geq C_D$:	$\theta^{\max} \leq \frac{2n_\omega-3}{2n_\omega-2}\left(C_U+\bar{\Delta}\right)$	
if $C_D > C_U > C_D/2$:	$\theta^{\max} \leq \min\left\{\frac{3}{2}C_U + \frac{3}{2}\bar{\Delta}, \frac{2n_\omega-3}{2n_\omega-2}\left(C_D+\bar{\Delta}\right)\right\}$	
if $C_D/2 \geq C_U$:	$\theta^{\max} \leq \min\left\{\frac{3}{4}C_D + \frac{3}{2}\bar{\Delta}, \frac{2n_\omega-3}{2n_\omega-2}\left(C_D+\bar{\Delta}\right)\right\}$	
(CIA-TV)		
$n_\omega = 2$	$\theta^{\max} \leq \left(\frac{N}{3+2\sigma_{\max}} + \frac{1}{2}\right)\bar{\Delta}$	Theorem 7.5
$n_\omega = 2, \Delta_j = \bar{\Delta}$	$\theta^{\max} = \frac{N+\sigma_{\max}+1}{3+2\sigma_{\max}}\bar{\Delta}$	Corollary 7.3
$n_\omega > 2$	$\frac{N+\sigma_{\max}+1}{3+2\sigma_{\max}}\bar{\Delta} \leq \theta^{\max} \leq \frac{2n_\omega-3}{2n_\omega-2}\left(\frac{t_f-t_0}{\sigma_{\max}+1}+\bar{\Delta}\right)$	Corollary 7.4-7.5
$n_\omega > 2, \sigma_{\max} \leq n_\omega - 2$	$\theta^{\max} \geq \frac{t_f-t_0}{\sigma_{\max}+2}$	Corollary 7.6
$n_\omega > 2, \sigma_{\max} > n_\omega - 2$	$\theta^{\max} \geq \frac{t_f-t_0}{2\sigma_{\max}+4-n_\omega}$	Corollary 7.6
$n_\omega > 2, \sigma_{\max} \leq n_\omega - 2$	$\theta^{\max} \overset{?}{=} \frac{t_f-t_0}{\sigma_{\max}+2} + \frac{1}{2}\bar{\Delta}$	Conjecture 7.1
$n_\omega > 2, \sigma_{\max} > n_\omega - 2$	$\theta^{\max} \overset{?}{=} \frac{t_f-t_0}{2\sigma_{\max}+4-n_\omega} + \frac{1}{2}\bar{\Delta}$	Conjecture 7.1

Table 7.2: Established results for the integral deviation gap of the problems (CIA), (CIA-U), (CIA-D), (CIA-UD), and (CIA-TV) with references to the corresponding statement. If $\theta^{\max}$ equals the upper bound, we proved a tight upper bound on the integral deviation gap. We indicate that the bound is conjectured with a question mark.

achieved in Chapter 5 does not hold if (BOCP) is restricted by these time-coupled combinatorial constraints. In particular, in this situation, the optimal solution of (BOCP) cannot be approximated arbitrarily close by the solution constructed by the (CIA) decomposition and by refining the discretization grid. Nevertheless, the proven integral deviation gaps are useful for quantifying the approximation errors.

Table 7.3 presents the integral deviation gap results for the algorithms DSUR, DNFR, and AMDR. Our analysis did not include other possible combinatorial constraints, such as the prefixing (3.15) or the mode transition (3.14) constraint. Without providing rigorous proof for these constraints, we note that the integral deviation gap can theoretically become very large (i.e., $t_f - t_0$) when there are indices $i \in [n_\omega]$, $j \in [N]$ with $a_{i,j} = 1$ and $w_{i,j} = 0$ is fixed.

Algorithm	Established result	Statement
DSUR		
$C_D < \underline{\Delta}$	$\theta(\boldsymbol{w}) \leq (C_U + \bar{\Delta}) \sum_{i=2}^{n_\omega} \frac{1}{i}$	Corollary 7.1
$\Delta_j = \bar{\Delta}$, $C_U < \bar{\Delta}$	$\exists\, \boldsymbol{a}^* : \ \theta(\boldsymbol{w}) \geq \left(\frac{\lceil C_D / \bar{\Delta} \rceil}{2} + (n_\omega - 2) \right) \bar{\Delta}$	Theorem 7.2
DNFR	$\theta(\boldsymbol{w}) \leq C_2 \overline{\mathscr{L}}$, $(C_2, \chi_D) = \left(\frac{2n_\omega - 3}{2n_\omega - 2}, 0 \right), \left(\frac{3}{2}, 1 \right)$	Theorem 7.4
AMDR, $\Delta_j = \bar{\Delta}$, if $n_\omega > 2$, assume only canonical switches	$\theta(\boldsymbol{w}) \leq \theta^* + TOL$	Theorem 6.3

Table 7.3: Established results for the integral deviation gap of the constructed binary control $\boldsymbol{w}$ of the rounding algorithms DSUR, DNFR, and AMDR. Further details on the assumptions and parameters are given in the referenced result statement.

Part III

Implementation, numerical results, and applications

Chapter 8

Implemented software: pycombina

To solve mixed-integer optimal control problems (MIOCPs) using the combinatorial integral approximation (CIA) decomposition, the solutions of nonlinear programs (NLPs) and a mixed-integer linear program (MILP) are necessary. While advanced programs for modeling and solving NLPs like `Casadi` [9] and `Ipopt` [263] already exist, the (CIA) problem, representing an MILP, can be highly complex to solve. Therefore, tailor-made algorithms are beneficial for this decomposition step. pycombina is a software package designed for the formulation and solving of (CIA) rounding problems. The tool is mainly developed in Python, facilitating integration into existing projects. The performance-critical parts of the BnB solver are written in C++, and we use `pybind11` [133] to interface Python with the solver.

pycombina emerged as part of the publications [48, 49, 50]. This chapter is inspired by [50], where the tool has already been described. The author of this thesis contributed to the conception and development of the software package and implemented solver classes related to this thesis. ADRIAN BÜRGER implemented most parts of the Python interface and the BnB solver, while MIRKO HAHN contributed implementation-related improvements of the BnB solver.

Section 8.1 introduces the underlying code design and the usage of the pycombina tool. The different available combinatorial constraint options and solver classes are discussed in Section 8.2 and 8.3, respectively, before we exemplify applications of the tool in Section 8.4.

8.1 Basic code design and usage structure

Figure 8.1 illustrates the usage of pycombina in a flowchart pattern. We assume that (NLP_{rel}) has been solved and that the optimal relaxed controls $\boldsymbol{a}^*$ are thus available. This matrix of relaxed control values and a vector of the time grid points are necessary for setting up a (CIA) problem. We implemented the Python class `BinApprox` to store and handle (CIA) problem data. Initialization of a `BinApprox` object is performed by providing the relaxed control value and grid data. pycombina computes internally auxiliary variables, such as the number of control modes and the grid lengths. If additional combinatorial constraints restrict the (CIA) problem, the corresponding constraint parameters need to be provided. The `BinApprox` class facilitates methods to set up these constraints. We give more details on available combinatorial constraints in Section 8.2.

The next task is to choose a solver. Most of the algorithms mentioned in Chapter 6 are available as part of pycombina, for instance, `CombinaSUR` and `CombinaBnB`, which respectively represent SUR and BnB. The solvers are accessible via dedicated solver classes, all of which initialize a solver object with the `BinApprox` object data. Afterward, the (CIA) problem data is internally preprocessed; e.g., inactive controls are removed. These tasks are performed by the class `BinApproxPreprocessed`, which is a child class of `BinApproxBase`, like `BinApprox`.

After a solver object has been initialized, the final step is to solve the problem (depending on the chosen solver, it may only be heuristically), using the `solve()`-method of the solver class.

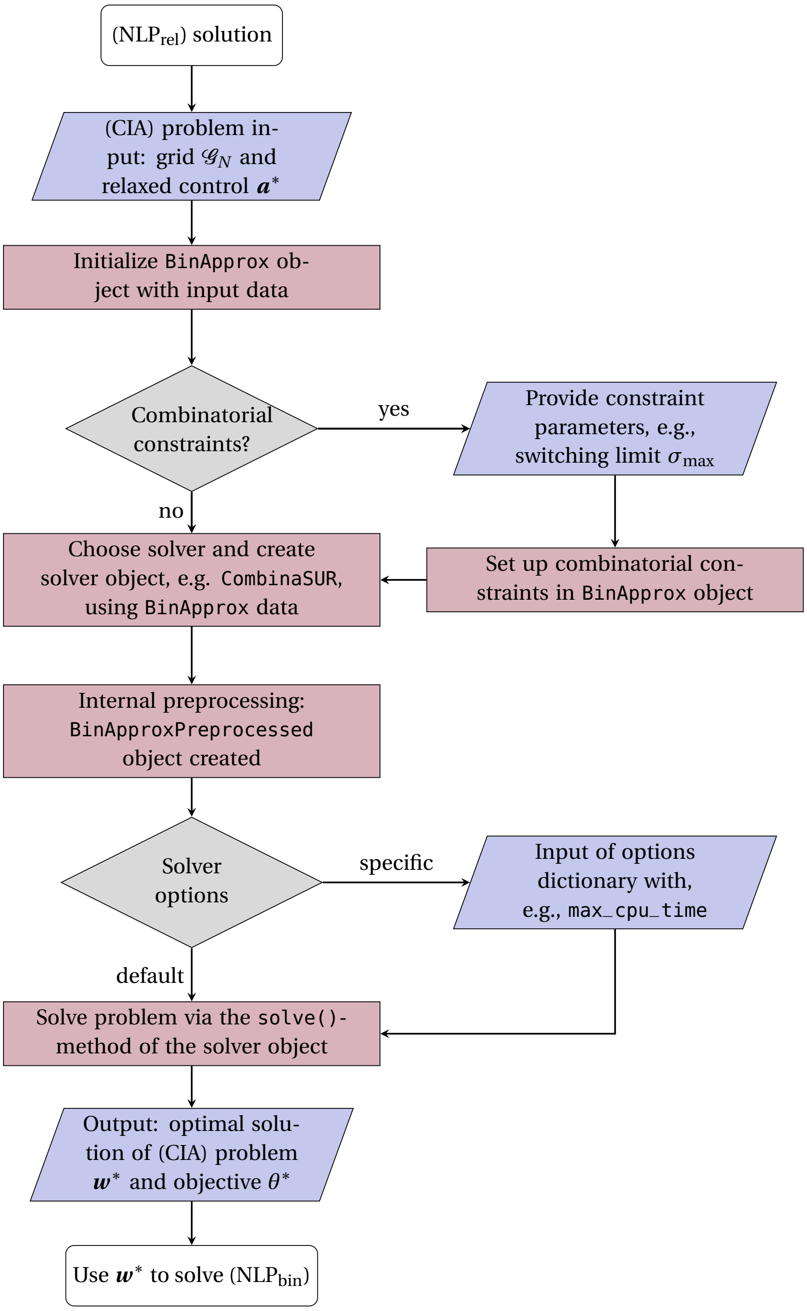

($\mathrm{NLP}_{\mathrm{rel}}$) solution
(CIA) problem input: grid $\mathcal{G}_N$ and relaxed control $\boldsymbol{a}^*$
Initialize BinApprox object with input data
Combinatorial constraints?
yes
no
Provide constraint parameters, e.g., switching limit $\sigma_{\max}$
Set up combinatorial constraints in BinApprox object
Choose solver and create solver object, e.g. CombinaSUR, using BinApprox data
Internal preprocessing: BinApproxPreprocessed object created
Solver options
specific
default
Input of options dictionary with, e.g., max_cpu_time
Solve problem via the solve()-method of the solver object
Output: optimal solution of (CIA) problem $\boldsymbol{w}^*$ and objective θ^*
Use $\boldsymbol{w}^*$ to solve ($\mathrm{NLP}_{\mathrm{bin}}$)

Figure 8.1: Flowchart overview of the application of pycombina. Stadium (white), rectangle (red), rhombus (gray), and rhomboid (blue) respectively indicate the beginning/ending of the process, a set of operations, a process of inputting/outputting data, and a conditional operation.

Specific solver options can be chosen for some of the solver classes, e.g., a maximum run time can be set up via `max_cpu_time` and passed to the solve function within an options dictionary. The optimization output can be accessed via the solution class attributes `eta` for the objective value and `b_bin` for the resulting $\boldsymbol{w}$ value. The latter is used to solve ($\mathrm{NLP}_{\mathrm{bin}}$).

8.2 Available combinatorial constraints

We list the different pycombina functions for setting up combinatorial constraints, and creating a connection to Section 3.2:

- `set_n_max_switches`: This function allows us to set up mode-specific limiting switching constraints as defined in Constraint (3.7). It requires switching limits, i.e., positive integers, as the input vector for every control mode.

- `set_total_n_max_switches` is equivalent to Constraint (3.8). Thus, in the discretized setting, it represents the total variation (TV) constraint. We need to provide $\sigma_{\max} \in \mathbb{N}$ as input.

- `set_min_up_times` enforces mode-specific minimum up (MU) time spans as defined in Constraint (3.9). The MU time span for each mode needs to be provided.

- `set_min_down_times` is analogous to `set_min_up_times` but with minimum down (MD) time spans. It is the implementation of Constraint (3.10), where a vector of MD times is needed for all control modes.

- `set_max_up_times`: This sets the maximum up time per control, i.e., the maximum time that a control mode can stay active once it has been activated, as defined in Constraint (3.12). The user needs to provide as input a vector of time spans for the corresponding modes.

- `set_total_max_up_times` is the pycombina function for enforcing Constraint (3.13). A vector of mode-specific `float` numbers is taken as input, limiting the mode-specific activation time over the entire time horizon.

- `set_valid_controls_for_interval`: As introduced in Constraint (3.15), certain control modes can be excluded on specific time intervals. The function takes a time interval and a binary vector indicating the allowed modes on this interval as input. A “zero” in the nth position of the vector indicates that the nth control mode is forbidden on the chosen interval.

- `set_valid_control_transitions` corresponds to Constraint (3.14) in pycombina. This function requires the control mode index i for which valid mode transitions are defined together with a binary vector that indicates the allowed transitions. A “zero” in the nth position of this vector indicates that the nth control mode cannot be activated directly after control mode i has been active.

- `set_b_bin_pre` defines an active control mode on the hypothetical interval before t_0. With this pre-activated control mode, a change of activation on t_0 can be counted as a switch. This feature appears to be relevant in model predictive control (MPC) applications [49]. As input, it requires a binary vector, indicating which mode was pre-activated.

8.3 Available solver classes

We differentiate between rounding, MILP, and branch-and-bound (BnB) solver classes. In the following, we discuss available solver algorithms according to these groups and in connection with Chapter 6.

1. *Rounding algorithms:* We list the various rounding algorithms with their solver options and functionality to construct binary controls fulfilling combinatorial constraints.
   - `CombinaSUR` and `CombinaNFR` correspond to the classical SUR and NFR schemes, respectively. They are not designed to construct combinatorial constraint feasible solutions.
   - `CombinaDSUR` and `CombinaDNFR` are the implementations of DSUR and DNFR, respectively. These solvers can handle MU and MD time constraints. Since these algorithms are designed to include mode-independent MU and MD times, the current implementation uses the maximum MU and MD times of all modes as the dwell time parameter.
   - `CombinaAMDR` is the solver implementation of AMDR. The constructed solution satisfies TV constraints and MU times. It is possible to specify the TOL parameter via `eta_tol`. The rule for choosing the next active control as part of the MDR algorithm can be set to either "`argmax`" or "`adm_next_forced`" via the parameter `mdr_strategy` . While the first rule selects the control with the maximum γ value (as defined in MDR), the admissible next-forced control is chosen as part of the second rule. If the admissible next-forced control is chosen as activation strategy, the solver is equivalent to Bisection-NFR from Chapter 6. Thus, the (CIA) problem can be solved to optimality up to a tolerance TOL by using `CombinaAMDR` with this activation strategy.
2. *Interface to MILP solver programs:* `CombinaMILP` automatically sets up an MILP that can be solved by `Gurobi` [109]. The implementation relies on `Gurobi`'s Python interface `gurobipy`. All the combinatorial constraints mentioned in this thesis are available for this solver option. This solver class is, in particular, suitable for testing new constraint types due to the straightforward implementation of constraints in `gurobipy`. The constraint method `set_state_cnstr_apprx` provides the possibility of setting up the first-order Taylor path constraint approximation from Definition 4.14. For this purpose, the corresponding constraint derivative and model function weights are provided as input. `Gurobi`'s regular solver options, such as a time limit, can be passed to `Gurobi` via `CombinaMILP`. The extended formulated (CIA) problem from Section 6.4.2 can be addressed with the class `CombinaSTOMILP`, though only TV constraints are currently available. Before applying this solver, a predefined active control sequence needs to be provided through the function `set_predef_control_seq`.
3. *BnB algorithms:* `CombinaBnB` is the solver corresponding to the BnB algorithm from Section 6.3; it has proven beneficial for real-world applications [49]. All the mentioned combinatorial constraints are available for this solver. The algorithm can take considerable time for big problems, so the solver options `max_iter` and `max_cpu_time` are available to abort it prematurely. The choice of the node selection strategy can have a significant

influence on the solution time. Example implementations are depth-first and best-first strategy accessible via the solver option `bnb_search_strategy`. A draft of the extended formulated BnB algorithm, i.e., STO-BnB from Section 6.4.3, is implemented as the solver class `CombinaSTOBnB`.

8.4 Exemplary application of the tool

Figure 8.2 presents a code snippet that illustrates the usage of the Python interface. We use the relaxed control data from Problem (P2), Section 9.3. We follow the main problem set up and solution steps described in Section 8.1. To demonstrate the usage of combinatorial constraints, we add mode-specific switching limits for the three control modes. The example problem instance is solved with the NFR scheme and with the BnB algorithm to include TV constraints. The constructed control solutions, together with the relaxed control solution, are depicted in Figure 8.3. While the control solution constructed by NFR switches frequently, the number of switches in the BnB solution has been reduced to the induced switching limits.

```
1    import pycombina
2
3    # (CIA) problem data:
4    G = [0.0, 0.075, 0.15, 0.225, 0.3, ..., 11.925, 12]
5    a = [[0.0, 0.0, 0.0, 0.0, 0.0, ..., 0.2773, 0.3874],
6         [1.0, 1.0, 1.0, 1.0, 1.0, ..., 0.7227, 0.6126],
7         [0.0, 0.0, 0.0, 0.0, 0.0, ..., 0.0, 0.0]]
8
9    cia_problem = pycombina.BinApprox(G,a)
10
11   # First, solve with NFR:
12   nfr_solver = pycombina.CombinaNFR(cia_problem)
13   nfr_solver.solve()
14
15   # Second, limit the number of switches and solve with BnB:
16   cia_problem.set_n_max_switches([4,6,4])
17   bnb_solver = pycombina.CombinaBnB(cia_problem)
18   bnb_opts = {"max_iter": 1e8}
19   bnb_solver.solve(**bnb_opts)
```

Figure 8.2: Example Python code to illustrate the usage of pycombina. By providing the time grid `G` and the relaxed control value matrix `a`, a `BinApprox` object can be initialized. This (CIA) problem instance, referred to as `cia_problem`, is used to create an NFR solver object in line 12. The problem can be (heuristically) solved by applying the `solve()` method of the initiated solver object. The `cia_problem` can be modified with optional combinatorial constraints, as performed in line 16 with limiting switching constraints. We use the BnB solver `CombinaBnB` to solve the modified (CIA) problem. Here, specific solver options, such as an iteration limit, are available.

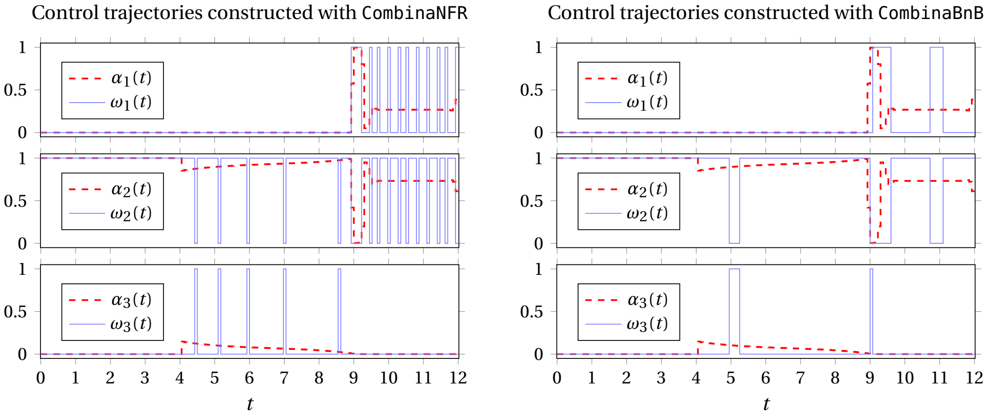


Figure 8.3: Control trajectory results for the example code from Figure 8.2 and data from the test problem (P2). Left: Constructed binary control solutions based on NFR. Right: Constructed binary control solutions based on BnB with TV constraints.

Chapter 9

Numerical results

This chapter presents the computational results of solving MIOCPs with the CIA decomposition based on the methods that have been described in the previous chapters. In particular, we address

1. The generalized CIA decomposition based on different MILP formulations and recombination heuristics in Section 9.1,
2. The incorporation of path constraints into the (CIA) problem in Section 9.2,
3. MIOCPs under minimum dwell time constraints in Section 9.3, and
4. MIOCPs under bounded discrete total variation constraints in Section 9.4.

All computational experiments were executed on a workstation with 4 Intel i5-4210U CPUs (1.7 GHz) and 7.7 GB RAM. We tested the proposed algorithms with benchmark examples from the `https://mintOC.de` library [221]. We used the `AMPL` [79] code `ampl_mintoc`, which is a modeling framework for handling optimal control problems, to produce the results in Section 9.1. Sections 9.2–9.4 involve implementations in `Python` v3.7.1, where we applied `CasADi` v3.4.5 [9] to parse the NLP models with an efficient derivative calculation of the Jacobians and Hessians for the solver `IPOPT` v3.12.3 [264]. Details on the applied rounding problem-solving procedures are given in the individual sections. Sections 9.1, 9.3, and 9.4 are based on the numerical results chapters of the publications [280], [282], and [222], respectively.

9.1 Benchmark computations of different MILPs and recombination heuristics

We test the computational performance of the generalized CIA decomposition from Section 4.5 here; in particular, we test the different MILP formulations (Section 4.5.1) and recombination as postprocessing (Section 4.5.2). In accordance with Section 4.5, we emphasize that we use the term (CIAmax) to highlight the application of the maximum norm, in contrast to other applicable norms, in the (CIA) problem.

The outline of this section is as follows. We describe the practical implementation of the tests in Section 9.1.1, present results for the different MILP formulations in Sections 9.1.2–9.1.4, and present results for the postprocessing heuristics in Section 9.1.5 before performing an analysis of the run times in Section 9.1.6. Details on the discretization of the instances and individual problem results are given in Section 9.1.7 and 9.1.8.

9.1.1 Practical implementation and instances

The `ampl_mintoc` code used here is advantageous for our purpose since it includes automatic differentiation, it interfaces to MILP and NLP solvers, and its problem implementation stays close to the mathematical formulation. Furthermore, AMPL is beneficial since it provides the evaluated dual variables $\tilde{\boldsymbol{\lambda}}$. We used the Radau collocation from [31] for discretizing the (BOCP) problems. Throughout the numerical study in this section, we applied Gurobi v8.1 [109] as an MILP solver and IPOPT v3.12.4 as an NLP solver. We assumed that the choice of the MILP solver has little influence on the solution quality and verified this by comparison with CPLEX v12.9. We discretized the test problems with a control grid with N intervals and a finer grid for the differential states with M intervals. We chose M such that the objective value changes only in the 5th decimal place with a finer discretization and constant N. Afterward, we varied N with fixed M to create several instances. For further details, we refer to Section 9.1.7. The problems involving path or terminal constraints can result in infeasible solutions after solving the binary approximation problem and solving (NLP$_{\text{bin}}$) with fixed binary controls. To this end, we relaxed these constraints and applied a merit function that penalizes constraint violations as part of the objective with a sufficiently high penalty factor.

9.1.2 Control approximation scaled with the model function

We hypothesized that the MILPs based on the model function-scaled combinatorial integral approximation, i.e., (SCIAmax) and (SCIA1), perform the best on instances where the binary control enters the control-dependent right-hand side terms $\boldsymbol{f}_i$ of the ordinary differential equation (ODE) in an affine way:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}_0(t, \boldsymbol{x}(t)) + \sum_{i=1}^{n_\omega} \omega_i(t) c_i, \qquad \text{for a.e. } t \in \mathscr{T}, \tag{9.1}$$

where $c_i \in \mathbb{R}$. If $\boldsymbol{f}_i$ depends on $\boldsymbol{x}(t)$, it may change rapidly over time resulting in possibly inaccurate $\boldsymbol{\omega}$ solutions because we only have the discretized state trajectory $\boldsymbol{x}(t)$ value at hand. We identified the MIOCPs "Double tank (Multimode)" and "Lotka Volterra (absolute fishing variant)" as candidate problems from https://mintOC.de with the above right-hand side structure and calculated their solutions for different discretizations, both with and without the TV constraints (4.9)–(4.10). In Section 9.1.7, we list the detailed discretization and maximum switches parameters. The results are presented in Fig. 9.1.

We chose to evaluate the (BOCP) solutions based on both (CIA) and (SCIA) binary controls according to the distance in the $\|\cdot\|_\infty$-norm of the differential state trajectories corresponding to binary and relaxed controls. The theoretical justification of the CIA decomposition is built on this distance, as pointed out in Chapter 5, and particularly, the proximity of objective values and constraint satisfaction follows. The performance plot shows that the differential state trajectories based on the (SCIAmax) and (SCIA1) binary control solutions are significantly closer to the relaxed solution than are their (CIA) counterparts. There are hardly any differences between the $\|\cdot\|_1$- and $\|\cdot\|_\infty$-norm results, although the $\|\cdot\|_\infty$-variants tend to perform better.

We examined whether the solutions constructed by (CIAmax) or (CIA1) outperform those computed by (SCIAmax) and (SCIA1) if the state trajectory distance is measured by the $\|\cdot\|_1$-norm or if the objective value deviation from the relaxed solution is considered: the (SCIA) variants are the clear winners. The result is similar when comparing the algorithms solely for instances with and without TV constraints.

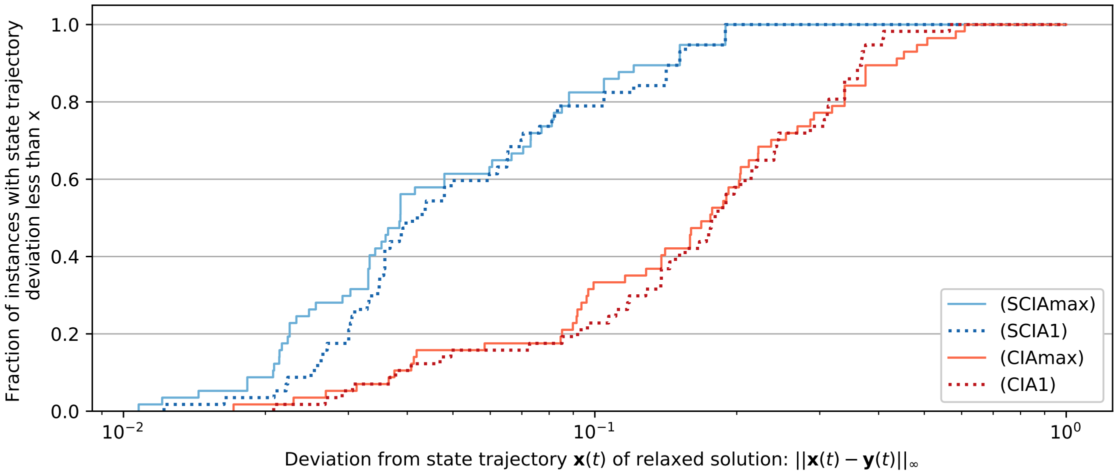


Figure 9.1: Performance profile comparing the deviation of the differential states based on the (SCIA) and (CIA) solutions from the differential states based on relaxed control values: The difference is evaluated in terms of the maximum norm in log-scale. The results are based on the instances "Double tank (Multimode)" and "Lotka Volterra (absolute fishing variant)" from the mintoc.de benchmark library. Using (SCIAmax) or (SCIA1) can significantly improve the performance of the CIA decomposition.

9.1.3 Control approximation scaled with dual variables

We derived (λCIA1) as an approximation approach of the cost-to-go function difference based on the binary and relaxed control solutions. Since this approximation is linear, we assume that the (CIAmax) approach is more suitable for most (nonlinear) MIOCPs. We postulated that the situation differs when a regularization term enters the objective function, accounting for the cost of activating binary controls in the form of, e.g.,

$$\mathscr{C}(\boldsymbol{x}, \boldsymbol{\omega}) = \Phi(\boldsymbol{x}(t_\mathrm{f})) + \int\limits_{t\in\mathscr{T}} \sum_{i=1}^{n_\omega} \omega_i(\tau) c_i \; \mathrm{d}\tau,$$

with constants $c_i \in \mathbb{R}$. The problem "Quadrotor (binary variant)" from `https://mintOC.de` includes a cost function in the above form, so we used it with different discretizations and both with and without the TV constraints (4.9) and (4.10) to compare the (λCIA1)-constructed (BOCP) solutions with those obtained from (CIAmax) and (CIA1). In Fig. 9.2, we present the computational results. In contrast to scaling with the model function, here we compared the objective deviations of the (BOCP)-feasible solutions to the relaxed solution in terms of percent since (λCIA1) aims to improving the objective values directly. We remark that (λCIA1) performs worse than (CIAmax) and (CIA1) when the distance to the relaxed solution is measured in differential state space.

The performance plot shows that in some instances (λCIA1) provides solutions with an improved objective value when compared with those of (CIAmax) and (CIA1) but in many others, it does not. Since (λCIA1) provided even weaker approximations of the relaxed solutions for other MIOCPs, as shown in Section 9.1.5, we do not, in general, recommend using it as a single rounding problem approximation step. It does serve, however, as a beneficial candidate

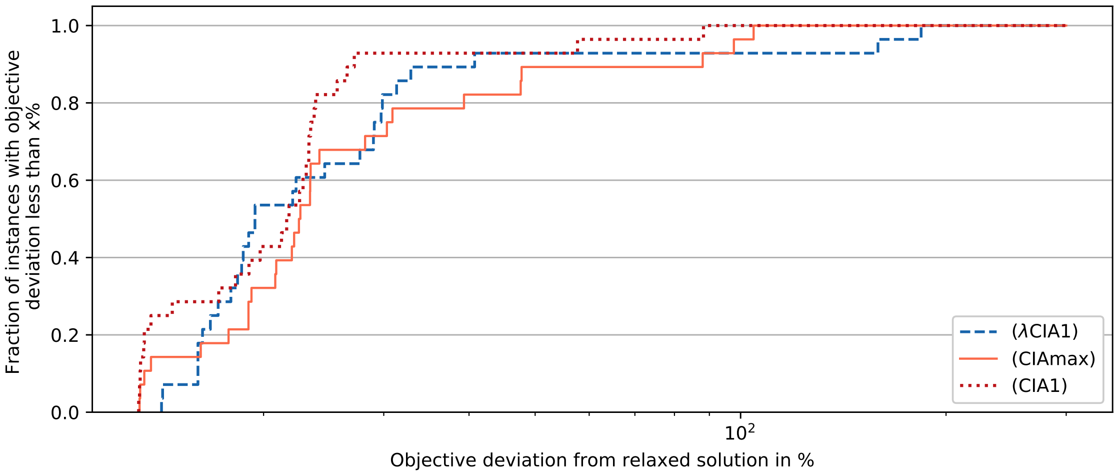


Figure 9.2: Performance profile comparing the objective value deviation of the (λCIA1)-, (CIAmax)-, and (CIA1)-based solutions to the relaxed control-based solution in terms of percent in log-scale. The results are based on different discretization instances of "Quadrotor (binary variant)" from the `https://mintOC.de` benchmark library. (λCIA1) appears to provide no clear improvement over the (CIA) solutions.

solution for recombination and might be useful for as of yet unexplored problem classes.

9.1.4 (CIA) with backward accumulating constraints

We hypothesize that the MILP variants based on backward accumulated constraints from Equation (4.32) are beneficial for MIOCP involving terminal equality constraints on the differential states. The standard (CIAmax) approach may construct a binary control that results in an infeasible terminal constraint because the deviation from the relaxed solution is too large. However, the direct incorporation of terminal constraints into (NLP_{rel}) may already lead to numerical difficulties. We decided to deal with soft constraints, meaning that we introduce auxiliary variables to penalize deviation of the differential states from a desired terminal value. We identified the candidate problem "Lotka Volterra (terminal constraint violation)" from `https://mintOC.de` and calculated its solutions for different discretizations, both with and without the TV constraints (4.9) and (4.10). Fig. 9.3 illustrates the objective deviation from the relaxed solution in percent of (CIA) and its backward variant solutions.

We chose the objective deviation as the performance metric for our comparison study since the objective accounts for violations of the terminal constraints via an auxiliary variable penalty term. The graphs of (CIAmaxB) and (CIA1B) indicate that their corresponding (BOCP) solutions yield lower objective values than their forward accumulation (CIA) counterparts. We observe that this result seems to be independent of the chosen norm since the performance differences between (CIAmaxB) to (CIA1B) are negligible.

9.1.5 Recombination heuristics

We used the MILP solutions constructed by (CIAmax), (CIA1), (SCIAmax), (λCIA1), and (CIAmaxB) as a base to run the recombination heuristics from Section 4.5.2 on a set of thirteen MIOCPs from the benchmark collection site `https://mintOC.de` with different discretizations

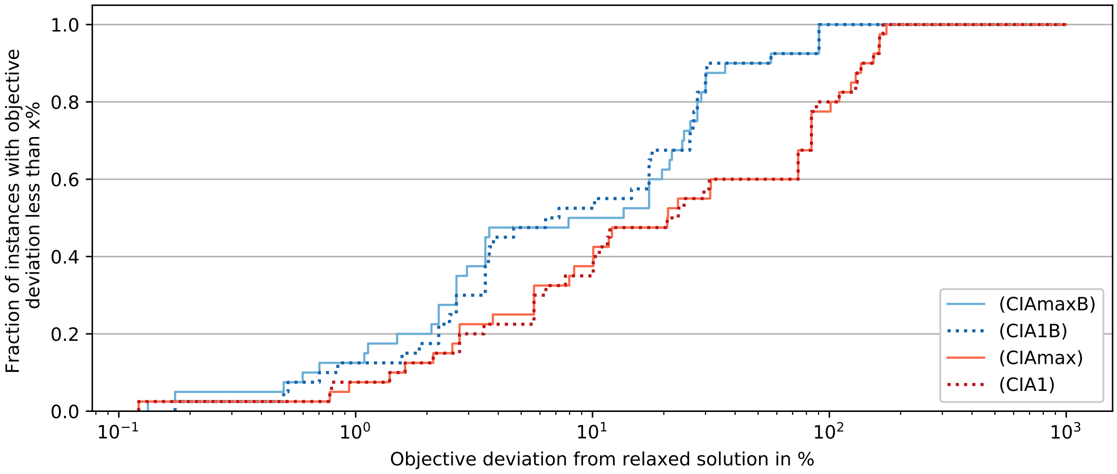


Figure 9.3: Performance profile comparing the objective deviation of (CIA) and its backward variant solutions from the relaxed solution in percent and log-scale: The results are based on the instance "Lotka Volterra (terminal constraint violation)" from the mintoc.de benchmark library. Using (CIA1B) or (CIAmaxB) can substantially improve the performance of the CIA decomposition.

(see Section 9.1.7 for details). The box plot in Figure 9.4 depicts the numerical results with respect to the percent deviation of the (BOCP) objective value of each algorithm from the objective value of the relaxed solution.

The boxes for the (SCIAmax), (SCIA1), and backward approaches appear to have slightly higher median values than those of the (CIA) MILPs, while their mean values and outliers appear to be slightly lower. The numerical study revealed several instances in which (SCIAmax) and (SCIA1) encounter a binary control solution with active controls on some intervals that have relaxed values close to zero. Under the assumption that the combinatorial approximation is mainly made on singular arcs, these cases might be called *degenerate*. We found underperforming (BOCP) objective values for (SCIAmax)- and (SCIA1)-based solutions in the case of degenerate control values, explaining some of the instances of low performance. We conclude that (SCIAmax), (SCIA1), and (CIAmaxB) should be used with caution. For specific problem classes, as shown in the previous subsections, they may suitable. We have not specifically selected such problems, and for the general problem class used here, there is no guarantee that these algorithms provide any real improvement. In the future, more sophisticated quadrature rules than the rectangle rule should be tested for the approximation of $\boldsymbol{f}$ in (SCIA).

The solutions of (λCIA1) clearly underperform, but we stress their importance for recombination, as mentioned in Section 9.1.3. For comparison, we have also computed the solutions based on SUR and see that it provides good solutions that are similar, albeit somewhat worse, than those of (CIAmax) and (CIA1). Note that depending on the selected algorithm, some instances resulted in an objective deviation of more than 100%, which can be attributed to highly penalized infeasible solutions of path- or terminal-constrained problems.

The depicted recombination heuristics provide significantly better binary control solutions in terms of the objective value than the MILP solutions. The median values are reduced by a factor of about two (Singular Arc Recombination) to a factor of three (GreedyTime) in comparison with (CIAmax) and (CIA1). The other characteristics, such as the mean values, quartiles,

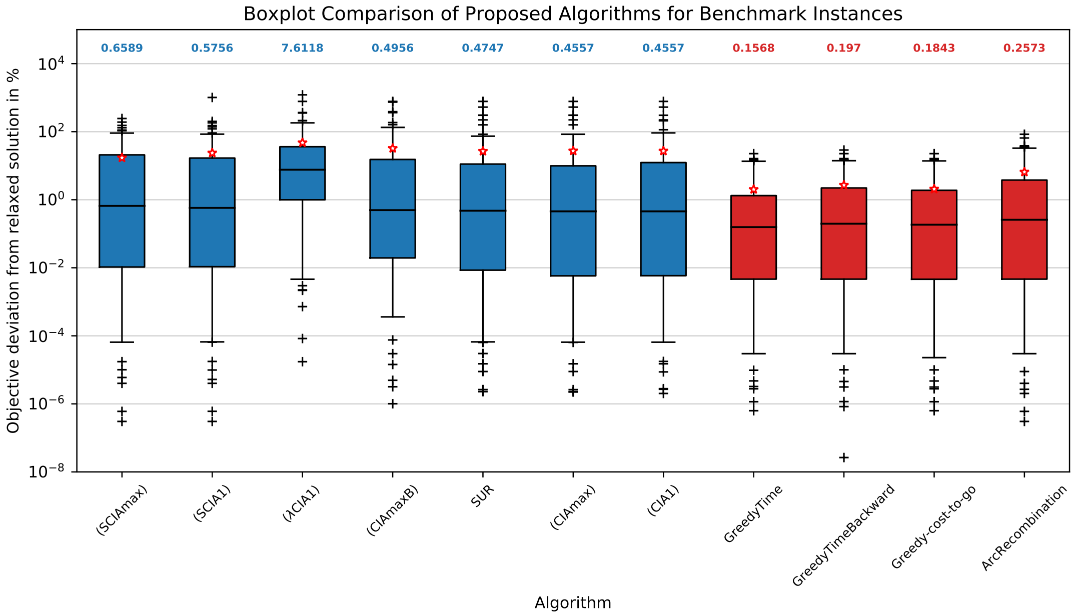


Figure 9.4: Box plot comparing the (BOCP) objective value deviation of several MILP- (blue) and recombination heuristic- (red) based solutions from the relaxed solution in percent and log-scale. The results are based on the instances listed in Subsection 9.1.7. The box borders are the first and third quartiles, and the whiskers represent the 1/20 and 19/20-quantiles. We visualize the median values as black lines in the boxes and also display them numerically above the boxes. We represent the average values for the algorithms with red asterisks and outliers with black crosses. ArcRecombination refers to the *Singular Arc Recombination* heuristic from Algorithm 4.4, while *GreedyTime* represents Algorithm 4.3 with *GreedyTimeBackward* and *Greedy-cost-to-go* being its modifications from Remark 4.7. The boxes for the recombination strategies indicate lower objective values compared those for the (CIA) algorithms: recombination can thus substantially improve the CIA decomposition performance.

and outliers, also reflect improvements. Particularly noteworthy is the GreedyTime heuristic, which is robust against outliers and on average constructs solutions with small objective values. The *Singular Arc Recombination* heuristic selects the solution of the MILP algorithms with the smallest objective value when there is only one singular arc. Since many of the selected problems involve only one singular arc, the box plot indicates that over all MILP-constructed solutions, this minimum can provide a significant improvement.

9.1.6 Run time evaluation

Figure 9.5 exemplifies the relationship between the run time and objective function values for the Lotka Volterra multimode problem with $M = 12000$ and varying N. We compare the relative (BOCP) objective value of the binary control solution constructed from (CIAmax) with those constructed from the minimum over all MILP solutions (denoted by $\text{Opt}(S^{\text{CIA}})$), the *GreedyTime* solution, and the solution obtained by the MINLP solver `Bonmin` v1.8.6. For a fair comparison, we ran Bonmin with its four main algorithms `B-BB`, `B-OA`, `B-QG`, and `B-Hyb` and depicted the shortest run time of these algorithms. The elapsed real time from AMPL represents the run time in our computations since the CPU time appeared to be very similar for our Bonmin calculations, while Gurobi is known to be a multi-threaded solver. First, the illustration shows that the spread from the objective value to the relaxed solution vanishes with increasing N regardless of the selected approach. Second, for some instances (CIAmax) was already quite close to Bonmin ($N = 25, 50$) in terms of objective quality, so GreedyTime could not provide much improvement. For other discretizations with a considerable gap between the (CIAmax) and Bonmin objective value solutions, GreedyTime was able to close most of the gap while being two orders of magnitude faster than Bonmin.

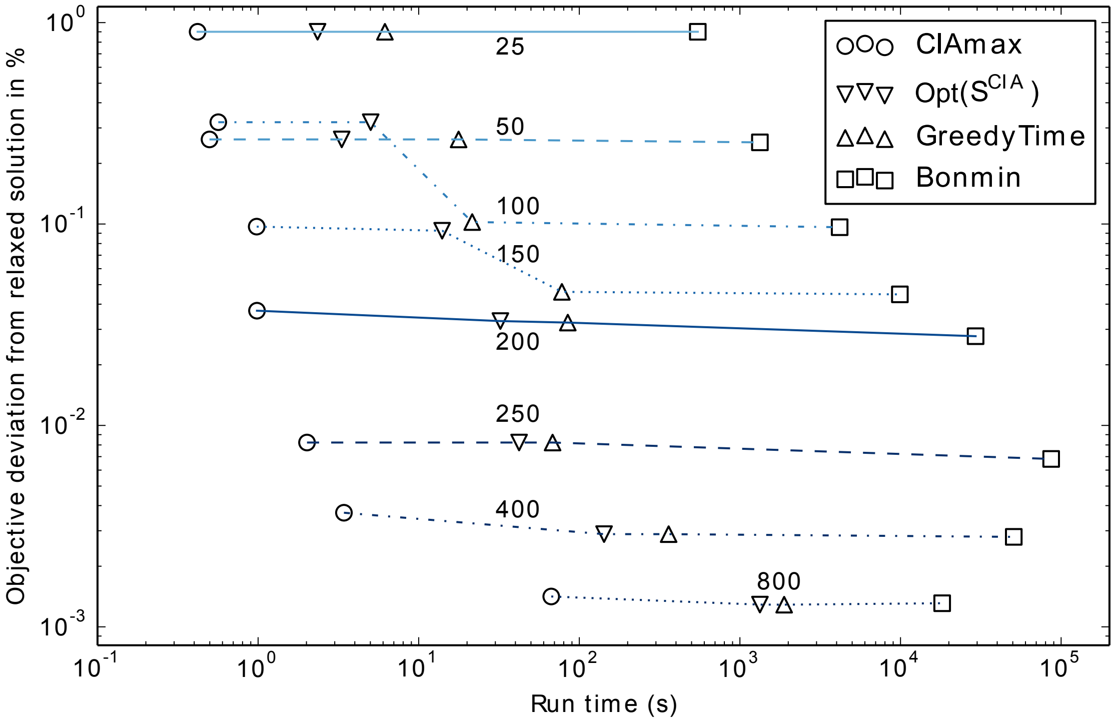


Figure 9.5: Log-plot of the run time and (BOCP) objective value deviation from the relaxed solution for the Lotka-Volterra multimode problem, with state discretization of $M = 12000$ intervals. The numbers indicate the corresponding number of control grid intervals N, and by Opt(S^{CIA}), we denote the minimal objective deviation over all MILP solutions. Note the convergence of all approaches towards the lower bound provided by the relaxed solution and closure of the gap between the (CIAmax) and Bonmin solutions for a fixed discretization. GreedyTime is roughly two orders of magnitude slower than (CIAmax) but faster than Bonmin.

The average run time over all instances for (CIAmax) was a few seconds and increased slightly for (SCIAmax), see Section 9.1.8. On average, Gurobi needed more than one minute for the MILPs with the 1-norm and thus took considerably more time. For instances involving a fine discretization, the run time increased enormously, and we thus set a time limit of 30 minutes.

The greedy heuristics and *Singular Arc Recombination* are to be used with caution since inputting numerous MILPs solutions leads to a high number of recombinations that must be evaluated. Singular Arc Recombination is relatively inexpensive and computes a solution that is at least as good as the best one from the MILPs. The algorithm is most beneficial in the case of several singular arcs in contrast to most applied problem instances where there is only one arc. The greedy algorithm variants are quite expensive (with run times of up to 15 minutes) yet provide solutions with objective function values very close to those of the relaxed problem. When it comes to MINLP solvers, run times of days or even weeks cast a positive light on the proposed generalized CIA decomposition Algorithm 4.2, which includes recombination.

9.1.7 Problem discretization details

To generate the performance plots and box plots in Section 9.1, we applied Algorithm 4.2, which is the generalized CIA decomposition, on the following discretized problems from `https://mintOC.de`:

"Lotka Volterra (absolute fishing variant)":
$M = 12000$, $N \in \{25, 50, 75, 80, 100, 120, 150, 160, 200\}$, $\sigma_{\max} \in \{10, 20, \infty\}$

"Quadrotor (binary variant)":
$M = 12000$, $N \in \{25, 50, 60, 80, 100, 150, 200, 300\}$, $\sigma_{\max} \in \{4, 10, 20, \infty\}$

"Lotka Volterra (terminal constraint violation)":
$M = 12000$, $N \in \{20, 30, 40, 50, 60, 100, 120, 200, 240, 300, 400, 600\}$,
$\sigma_{\max} \in \{4, 10, 20, \infty\}$

"F-8 aircraft (AMPL variant)":
$M = 6000$, $N \in \{30, 40, 50, 60, 100, 120, 150, 200, 240, 300, 400, 500\}$

"Egerstedt standard problem":
$M = 6000$, $N \in \{20, 30, 40, 60, 100, 120, 150, 200, 240, 300\}$

"Double Tank":
$M = 18000$, $N \in \{25, 50, 100, 180, 250, 300, 360, 720\}$

"Double Tank multimode":
$M = 12000$, $N \in \{20, 25, 50, 100, 200, 250, 300, 400, 600\}$, $\sigma_{\max} \in \{10, 20, \infty\}$

"Lotka Volterra fishing problem":
$M = 12000$, $N \in \{20, 30, 40, 60, 100, 120, 200, 300, 400, 600\}$

"Lotka Volterra multi-arcs problem":
$M = 18000$, $N \in \{25, 50, 100, 150, 200, 250, 300, 400, 600\}$

"Lotka Volterra multimode problem":
$M = 12000$, $N \in \{25, 50, 100, 150, 200, 250, 300, 400, 800\}$

"Van der Pol Oscillator (binary variant)":
$M = 6000$, $N \in \{20, 30, 40, 50, 60, 100, 120, 150, 200, 300\}$

"D'Onofrio chemotherapy model":
Scenarios 1,2, and 3 with $M = 6000$, $N \in \{20, 30, 40, 50, 60, 100, 120, 150, 200, 300\}$,
Only $N \in \{20, 30, 60, 120\}$ for Scenario 1, $N = 100$ for Scenario 2, and $N \in \{40, 100\}$ for Scenario 3 resulted in feasible relaxed solutions and were included.

"Catalyst Mixing problem":
$M = 3000$, $N \in \{10, 15, 20, 30, 50, 60, 75, 100, 120, 150\}$

9.1.8 Average performance indicators and individual problem results

Table 9.1: Comparison of the mean and standard deviation (σ) of the objective deviation, switch values, and run time for different approaches. The objective deviation is the percent deviation from the relaxed objective, and run time refers to the elapsed real time.

Approach	**obj. dev [%]**	**switches [#]**	**run time [s]**	**σ(obj. dev)**	**σ(switches)**	**σ(run time)**
CIAmax	27.32	40.08	8.84	95.91	40.16	29.60
CIA1	27.08	39.54	106.11	93.60	38.99	385.77
SCIAmax	17.13	31.12	12.38	38.06	31.54	42.35
SCIA1	23.71	30.94	78.17	96.18	29.91	317.55
λCIA1	47.51	28.08	54.91	139.97	45.10	290.47
CIAmaxB	32.37	40.41	19.15	110.28	40.10	166.22
GreedyTime	2.06	33.36	106.26	4.27	34.47	133.61
GreedyTimeB	2.68	33.61	103.05	5.11	33.64	131.84
Greedy-Cost-to-go	2.01	34.05	117.24	4.24	34.09	172.88
ArcRecombination	6.53	35.34	11.26	13.55	37.01	37.12

Table 9.2: Results for the Lotka Volterra multimode problem with differential states discretization $M = 12000$ and varying N. The tables list the objective values, difference from the relaxed objective, number of switches (S), and run time (R) in seconds required for the different approaches to construct the binary controls.

	(CIAmax)				**(CIA1)**			
N	**Obj.**	**Diff. to rel.**	**S [#]**	**R [s]**	**Obj.**	**Diff. to rel.**	**S [#]**	**R [s]**
25	1.84519	0.00920032	6	0.419997	1.84519	0.00920032	6	0.323492
50	1.83353	0.00189968	9	0.498163	1.83353	0.00189968	9	0.526022
100	1.83458	0.00470921	15	0.564993	1.83458	0.00470921	15	0.849123
150	1.83049	0.00129738	20	0.979946	1.83058	0.00138375	20	3.37327
200	1.8294	0.000412465	23	0.983907	1.8294	0.000412465	23	9.61383
250	1.82887	8.52473e-05	30	2.01582	1.82887	8.52473e-05	30	6.84566
300	1.82884	2.1597e-05	33	1.87382	1.82884	2.1597e-05	33	27.4496
400	1.82879	3.40892e-05	47	3.42292	1.82879	3.40892e-05	47	45.0224
800	1.82875	2.58672e-05	87	66.8739	1.82875	2.58672e-05	87	484.285
	(SCIAmax)				**(SCIA1)**			
25	1.84519	0.00920032	6	0.313487	1.84519	0.00920032	6	0.925208
50	1.83399	0.00235793	8	0.493533	1.83399	0.00235793	8	0.723298
100	1.91199	0.0821278	16	1.05474	1.91199	0.0821278	16	3.62938
150	1.8834	0.0542079	20	2.84568	1.8834	0.0542079	20	9.7413
200	1.86972	0.0407389	25	7.90383	1.86972	0.0407389	25	36.7948
250	1.82887	8.52473e-05	30	11.6632	1.82887	8.5189e-05	30	65.8286
300	1.82887	4.80446e-05	32	8.75161	1.82887	4.80449e-05	32	55.7173
400	1.82877	1.94567e-05	47	30.4913	1.82877	1.94577e-05	47	188.408
800	1.83859	0.00987316	88	233.701	1.8381	0.00937638	89	1479.19
	(λCIA1)				**(CIAmaxB)**			
25	1.84543	0.0094458	5	0.443169	1.87559	0.0395975	7	0.511785
50	1.84372	0.0120927	5	0.581385	1.84076	0.00912746	9	0.574335
100	1.8533	0.0234329	16	0.839147	1.8347	0.00483784	15	0.735999
150	1.85038	0.0211798	23	2.50497	1.83041	0.00121583	19	0.840867
200	1.83509	0.00610253	30	2.52277	1.82932	0.000336555	25	1.53587
250	1.8289	0.000119317	25	13.3181	1.82894	0.000159443	31	2.02886
300	1.8553	0.0264818	29	6.28425	1.82887	5.56473e-05	35	2.94341
400	2.07161	0.242853	128	12.5695	1.82878	2.53493e-05	47	5.77022
800	3.44174	1.61302	420	18.8157	1.82875	3.20174e-05	89	34.2567

Table 9.3: Second part of Table 9.3: Results for the recombination heuristics solving the Lotka Volterra multimode problem.

	GreedyTime				**GreedyTimeBackward**			
N	**Obj.**	**Diff. to rel.**	**S [#]**	**R [s]**	**Obj.**	**Diff. to rel.**	**S [#]**	**R [s]**
25	1.84519	0.00920032	6	3.81442	1.84519	0.00920032	6	4.55248
50	1.83353	0.00189968	9	14.398	1.83353	0.00189968	9	14.5728
100	1.83059	0.000723242	15	16.5069	1.83117	0.00130419	13	16.7239
150	1.82956	0.000364781	19	63.9035	1.83	0.000802598	20	60.1375
200	1.82931	0.000326273	24	52.5325	1.82932	0.000336555	25	53.6014
250	1.82887	8.52473e-05	30	25.9954	1.82887	8.52473e-05	30	25.6038
300	1.82884	2.1597e-05	33	81.1383	1.82884	2.1597e-05	33	82.31
400	1.82877	1.94567e-05	47	217.31	1.82877	1.94567e-05	47	179.64
800	1.82874	2.35655e-05	87	553.42	1.82874	2.3582e-05	87	605.012
	ArcRecombination				**Greedy-cost-to-go**			
25	1.84519	0.00920032	6	0.6978	1.84519	0.00920032	6	3.89079
50	1.83353	0.00189968	9	0.5322	1.83353	0.00189968	9	14.4531
100	1.83458	0.00470907	15	0.3819	1.83318	0.00331505	17	27.2192
150	1.83041	0.00121583	19	0.8785	1.82965	0.00045483	17	67.8054
200	1.82932	0.000336555	25	0.6278	1.82931	0.000326273	24	60.569
250	1.82887	8.52473e-05	30	0.6946	1.82887	8.52473e-05	30	25.6708
300	1.82884	2.1597e-05	33	0.9826	1.82884	2.1597e-05	33	103.187
400	1.82877	1.94567e-05	47	0.5933	1.82877	1.94567e-05	47	302.851
800	1.82874	2.3582e-05	87	0.7660	1.82874	2.35652e-05	87	1166.96

9.2 Incorporation of the path constraint data into (CIA)

In this section, we return to MIOCPs with path constraint restrictions of type (4.1e). In Definition 4.14, we proposed constraints to be included in (CIA) based on a first-order TAYLOR approximation of path constraints and justified this approach in Section 5.5 with an error bound result. Here, we present a case study of benchmarking the proposed basic CIA decomposition, i.e., Algorithm 4.1, where Constraint (4.15) is included in (CIA). The MIOCP we are dealing with is the *Lotka Volterra fishing problem*[1], which has been used in other benchmark studies [280, 28]. The problem formulation reads

[1] See `https://mintoc.de/index.php/Lotka_Volterra_fishing_problem` for further details on e.g., the problem interpretation.

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{\omega}\in\Omega} \int_{t_0}^{t_f} & (x_1(t)-k_1)^2 + (x_2(t)-k_2)^2 \,\mathrm{d}t \\
\text{s.t. } \dot{x}_1(t) &= x_1(t) - x_1(t)x_2(t) - k_3\omega_1(t)x_1(t) - k_4\omega_2(t)x_1(t), && \text{for a.a. } t\in[t_0,t_f], \\
\dot{x}_2(t) &= -x_2(t) + x_1(t)x_2(t) - k_5\omega_1(t)x_2(t) - k_6\omega_2(t)x_2(t), && \text{for a.a. } t\in[t_0,t_f], \\
\boldsymbol{x}(0) &= \boldsymbol{x_0}, \\
0 &\le (C_{\mathrm{ub}} - x_1(t)), && \text{for a.a. } t\in[t_0,t_f].
\end{aligned} \tag{P1}
$$

For our computations, the parameters are set to

$$(k_1,k_2,k_3,k_4,k_5,k_6) := (1,1,0.4,0,0.2,0), \qquad \boldsymbol{x_0} := (0.5,0.7)^\top, \qquad [t_0,t_f] := [0,12].$$

Furthermore, we add a path constraint in which x_1 is bounded from above by C_{ub}. The parameter C_{ub} is varied in our computations and specified in the sequel. In order to apply the first-order Taylor approximation constraint, we use the notation from Definition 4.14 and for $t\in\mathcal{T}$ obtain

$$c(x_1(t)) := C_{\mathrm{ub}} - x_1(t), \qquad c_{x_1}(x_1(t)) = -1, \qquad \boldsymbol{f}_1(\boldsymbol{x}(t)) := (0.4x_1(t), 0.2x_2(t)^\top,$$

where we neglect the zero terms $\boldsymbol{f}_2$ and c_{x_2}. The discretized and evaluated $c_{\boldsymbol{x}}$ and $\boldsymbol{f}_1$ are needed in Constraint (4.15). We solved the resulting (CIA) problem with `Gurobi` v9.0, which can be addressed via `pycombina` with an add-on implementation for these constraints.

Figure 9.6 shows example numerical results for a discretization with multiple shooting and $N = 200$ intervals. The parameter C_{ub} is set to 1.5. We compare the control and state trajectories based on relaxed controls with those based on (CIA) and (CIA) with Constraint (4.15).

We omit the presentation of ω_2 since it is complementary to ω_1. The relaxed state trajectory x_1 satisfies the path constraint by decreasing the fishing rate ω_1 between $t = 2$ and $t = 4$. Rounding the relaxed control values via (CIA) results in a state trajectory x_1 that does not fulfill the path constraint. The maximum violation of the upper bound C_{ub} amounts to 0.0197. This violation can be decreased to 0.0049 by using the path constraint approximation approach. This improvement in feasibility comes with an increase in the (BOCP) objective value from 1.3576 for (CIA) to 1.3609 when including Constraint (4.15). We recognize an unexpectedly late activation for the latter approach with ω_1 around $t = 10$, which may unnecessarily worsen the objective function value.

To investigate the constraint satisfaction of our approach quantitatively, we solved (P1) with different discretizations N and C_{ub} parameter values. The range for C_{ub} was chosen to be $[1.42, 1.63]$ because (NLP$_{\mathrm{rel}}$) becomes infeasible for smaller values, and the path constraint is not restricting for larger values. With an increment of 0.01, all values in this range (22 values) were tested for C_{ub}. As an infeasibility criterion, we chose the weighted accumulated infeasibility over all intervals, i.e.,

$$\nu(\boldsymbol{x}_1, N) := \frac{t_f - t_0}{N} \sum_{j\in[N]} \max\left\{0, x_{1,j} - C_{\mathrm{ub}}\right\}.$$

Figure 9.7 illustrates the numerical results of the constraint violation ν for (CIA) with and with-

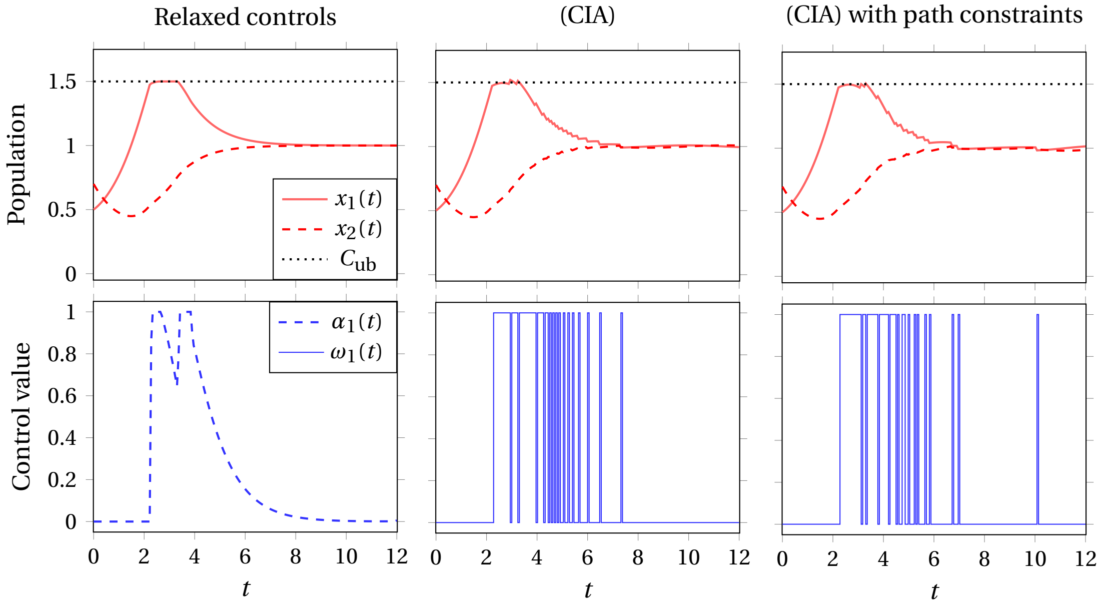


Figure 9.6: Differential state and control trajectories for the test problem (P1) solved with the CIA decomposition. The two plots on the left show the results after solving (NLP$_{\text{rel}}$). The control and state trajectories obtained after solving (CIA) are depicted in the middle two plots. The plots on the right illustrate the control and trajectories constructed by (CIA), including with the path constraint approach from Definition 4.14. The problem was discretized with Multiple Shooting and $N = 200$ intervals. We observe a slight path constraint violation for the x_1 trajectory constructed by (CIA); it is significantly reduced by including the constraints proposed in Definition 4.14.

out Constraint (4.15).

We observe that the constraint violation decreases with refinement of the grid intervals N – independent of the chosen approach. Fulfilling the path constraint is already challenging for small upper bound values in the relaxed solution, leading to high constraint violation values for the binary control solutions. The plot demonstrates that the infeasibility v can be significantly reduced by including Constraint (4.15). This is particularly true for the finer discretizations, i.e., $N = 200, 400$, and large upper bound values. We remark, however, that the solution constructed by (CIA) under Constraint (4.15) does not guarantee path constraint feasibility since in most instances, a small violation remains. In some instances, the violation is even worse than that of (CIA). Nevertheless, we conclude that the proposed approach can be useful for obtaining an (almost) (BOCP)-feasible solution in terms of path constraints. This could prove especially beneficial when grid refinement is not possible or desirable. Otherwise, path constraint satisfaction can be achieved by applying finer discretizations, as pointed out earlier. Future work in this area should treat more problem instances, including those with combinatorial constraints and nonlinear and more general path constraint types.

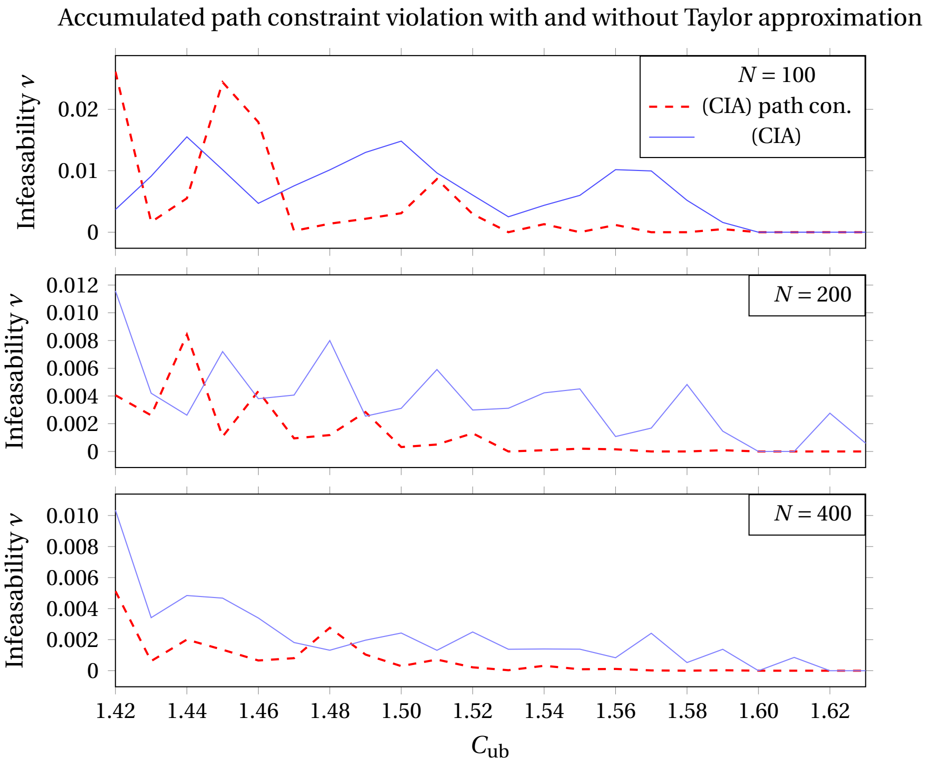


Figure 9.7: Evaluation of the path constraint violation metric $v(\boldsymbol{x}_1, N)$ for (P1), different (CIA) approaches, varying upper bound C_{ub}, and varying discretizations N. The definition of $v(\boldsymbol{x}_1, N)$ is given in the text. We compare the state trajectory feasibility constructed by (CIA) and that constructed by (CIA) including the path constraint approach from Definition 4.14. The latter approach reduces the infeasibility of the obtained solution, although it does not guarantee feasibility.

9.3 Mixed-integer optimal control under minimum dwell time constraints

This section consists of a case study for solving an MIOCP under minimum dwell time (MDT) constraints with the basic CIA decomposition, Algorithm 4.1; it is based on [282], Section 7. We consider a three-tank flow system problem with three controlling modes to evaluate the integral deviation gap in practice and to test the proposed rounding methods DSUR and DNFR. The problem models the dynamics of upper-, middle-, and lower-level tanks, connected to each other with pipes. The goal is to minimize the deviation of certain fluid levels k_2, k_4, in the middle, respectively lower, level tank. This problem type has been discussed in a variety of publications on the optimal control of constrained switched systems [60] and is taken from the benchmark `https://mintOC.de` library [221]. The problem reads

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{\omega}\in\Omega} \int_{t_0}^{t_f} & k_1(x_2(t)-k_2)^2 + k_3(x_3(t)-k_4)^2 \,\mathrm{d}t \\
\text{s.t. } \dot{x}_1(t) &= -\sqrt{x_1(t)} + c_1\omega_1(t) + c_2\omega_2(t) - \omega_3(t)\sqrt{c_3x_1(t)}, && \text{for a.a. } t\in[t_0,t_f], \\
\dot{x}_2(t) &= \sqrt{x_1(t)} - \sqrt{x_2(t)}, && \text{for a.a. } t\in[t_0,t_f], \\
\dot{x}_3(t) &= \sqrt{x_2(t)} - \sqrt{x_3(t)} + \omega_3(t)\sqrt{c_3x_1(t)}, && \text{for a.a. } t\in[t_0,t_f], \\
\boldsymbol{x}(0) &= \boldsymbol{x_0}.
\end{aligned}
\tag{P2}
$$

The additional parameters are

$$
\boldsymbol{k} := (2,3,1,3)^T, \qquad \boldsymbol{c} := (1,2,0.8)^T, \qquad \boldsymbol{x_0} := (2,2,2)^T, \qquad [t_0,t_f] := [0,12].
$$

Furthermore, we add MU and MD time constraints (4.11) and (4.12) to the three-tank problem with varying C_U and C_D parameters. We applied Direct Multiple Shooting for temporal discretization with a varying number of control grid intervals N together with a fourth-order Runge-Kutta scheme to obtain the evolution of the differential state and thus the objective value.[2] To find the optimal solution of the resulting (CIA) problem and its MDT variants we used the BnB solver of *pycombina* [49]. We published the `Python` source code for solving (P2) via the basic CIA decomposition online.[3]

We stress that the obtained feasible solutions for (P2) via the CIA decomposition are, in general, not globally optimal solutions. In fact, Problem (P2) appears to be nonconvex such that `IPOPT` may construct a local solution, like rounding via (CIA) may do. Nevertheless, finding a globally optimal solution is computationally expensive, as argued before.

Figure 9.8 depicts the state and control trajectories constructed by the CIA decomposition with relaxed (ROCP) and binary (BOCP) control values and a required MU time of $C_U = 0.3$. We remark that the objective value of the binary solution under the MU time constraint is about 1.3% larger than that of the relaxed solution and is about 1.2% larger than the objective value of the binary solution without the MU time constraint.

In Figure 9.9, we illustrate the effect of an increasing MU time on the optimal objective values of (CIA-U) and (BOCP). As expected, the finer the discretization grid and the shorter the required MDT time, the better the objective values of both problems become. A short MU time results in a weak restriction for (BOCP), making its objective value close to that of (ROCP), which is $\mathscr{C} = 8.776$. However, for $C_U \geq 0.7$, refinement of the grid cannot compensate for the MU time restriction, and the (BOCP) objective values are about 25% larger than those of (ROCP). Interestingly, the objective value hardly increases for $C_U > 0.7$; it even decreases slightly after $C_U = 0.7$ before increasing again and then remaining constant from $C_U \approx 2.0$ on. We observe a few outlier instances for which the objective value appears to be unexpectedly large: e.g.,

[2] When applying the fourth-order Runge-Kutta scheme, for the differential states, we need $\boldsymbol{x} \in C^5(\mathscr{T}, R^{n_x})$ to generate a fourth-order error term. It is common to only require $\boldsymbol{x} \in W^{1,\infty}(\mathscr{T}, R^{n_x})$; however, we can assume stronger regularity thanks to the piecewise continuously differentiable control functions from Definition 4.3. Nevertheless, the algorithms presented here are independent of the chosen numerical integration scheme, and one may choose a more accurate scheme according to the dynamical system at hand.

[3] See `https://mintoc.de/index.php/Three_Tank_multimode_problem_(python/casadi)`

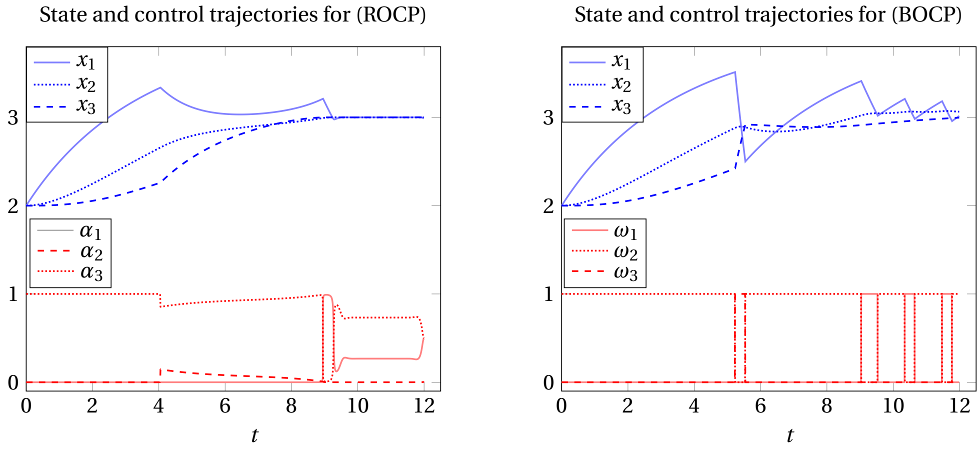


Figure 9.8: Differential state and control trajectories for the test problem (P2): on the left, with relaxed binary controls, i.e., problem (ROCP), and on the right, with approximated binary controls, i.e., problem (BOCP), with MU time $C_U = 0.3$ and a temporal discretization with $N = 1280$ intervals. The optimal objective value for (ROCP) is $\mathscr{C} = 8.776$, while that for (BOCP) is $\mathscr{C} = 8.888$.

$N = 20$ with $C_U = 1.2$ or $N = 40$ with $C_U \in \{0.4, 0.5\}$. This can be explained by the coarse grid choices, thereby highlighting the importance of a fine time discretization for the stability of the obtained solution for (BOCP).

Conversely, the objective value of (CIA-U) increases roughly linearly in C_U on fine grids until reaching $\theta^* \approx 0.87$, which seems to be the maximum value for (P2) in this setting. Thus, while small values of the objective of (CIA-U) correspond to promising objective values of (BOCP), (CIA-U) and (BOCP) appear to be uncorrelated for $C_U \geq 0.7$. We computed similar results for (P2) with MD time constraints (not shown). We also tested whether including the relaxed MDT constraints into the (NLP$_{\text{rel}}$) has a significant impact on the solution and found that this was not the case.

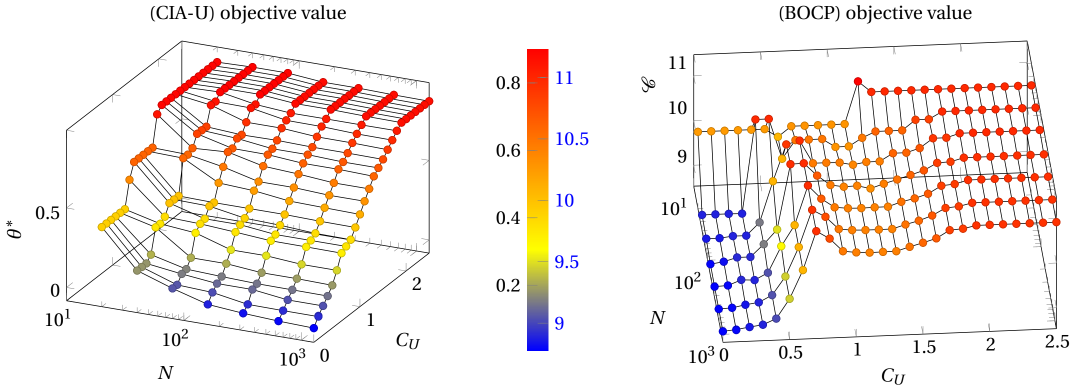


Figure 9.9: Objective values of (CIA-U) and (BOCP) based on the test problem (P2) and on different control discretizations N and MU time durations C_U.

We analyze the performance of DNFR and DSUR for both MU and MD time constraints and with respect to θ^* in Figure 9.10. The obtained solutions are compared with the global minima for (CIA-UD) from the BnB of `pycombina`, Algorithm 6.2. We observe that DNFR seems to perform better for MU time constraints, while DSUR performs better for the instances with MD time requirements. We also plotted the theoretical upper bounds (UB) from Propositions 7.1 and 7.3: $\frac{3}{4}(C_1 + \bar{\Delta})$, $C_1 = C_U, C_D$ here. In agreement with Figure 9.9, the minima of (CIA-U) and (CIA-D) hardly increase for large MDTs and therefore diverge from their theoretical upper bounds. We explain this behavior by the problem-specific given relaxed values, which induce an objective value of $\theta^* \approx 0.87$ for (CIA-U) and (CIA-D) even if no switches are used in the binary solution.

We also show the upper bound derived for DSUR with MU time constraints from Corollary 7.1, i.e. $\frac{5}{6}(C_U + \bar{\Delta})$, and the lower bound for the upper bound for DSUR with MD time constraints by Theorem 7.2, i.e. $\frac{1}{2}(C_D + \bar{\Delta})$. While the solutions constructed by DSUR may violate the upper bounds for (CIA-U) and (CIA-D), as happening for the MU time case, the bounds for DSUR are not violated. We observe that the (CIA-D) objective values are not only by far smaller than their upper bound, but even smaller or equal to the DSUR bound by Theorem 7.2.

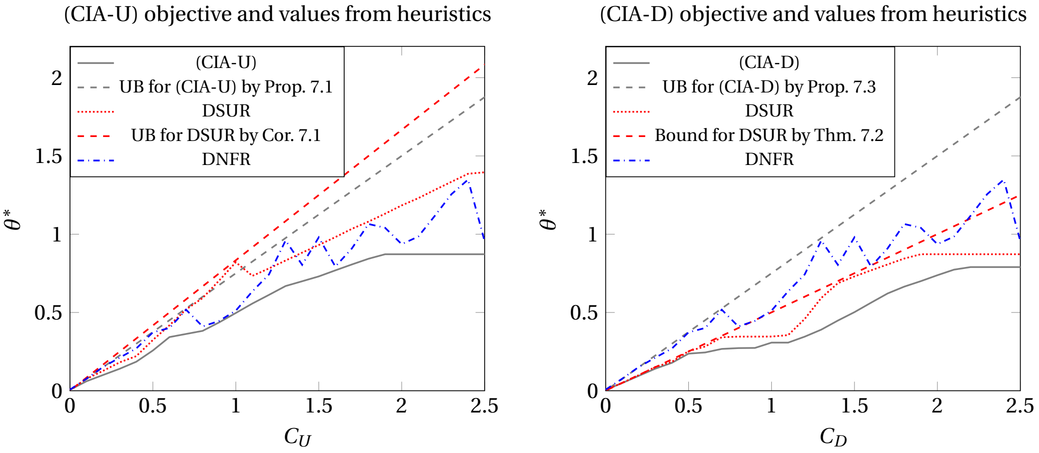


Figure 9.10: (CIA) objective function values θ^* for test problem (P2) with time discretization $N = 1280$ and varying MU time C_U (left), respectively varying MD time C_D (right). The optimal objective values for (CIA-U), respectively (CIA-D), are obtained via the BnB algorithm of `pycombina` (Algorithm 6.2) and are compared with the objective value solutions constructed by DNFR and DSUR. We also show the upper bound (UB) for (CIA-U) respectively (CIA-D) from Propositions 7.1–7.3 and the bounds derived for DSUR from Corollary 7.1 and Theorem 7.2. We note that although Theorem 7.2 derives only a lower bound for the upper bound of DSUR with MD time constraints, this bound is not violated by the computational results.

Since in all instances the execution of the heuristics took no more than 0.02 seconds, we conclude that the heuristics can be used to quickly generate robust solutions with competitive objective values. They might also be useful for initializing BnB algorithms of `pycombina` with a good upper bound. However, a numerical study is needed to verify the added benefit, which could be elaborated in future work.

9.4 Mixed-integer optimal control with bounded discrete total variation

This section investigates the numerical behavior of the basic CIA decomposition, Algorithm 4.1, when solving MIOCPs under TV constraints. We focus on a performance evaluation of the AMDR algorithm compared to the BnB Algorithm 6.2 when solving (CIA-TV). Furthermore, we review the proven upper bounds for (CIA-TV) from Section 7.5 to examine their behavior in practice. The implemented AMDR scheme in `pycombina` was applied with the tolerance parameter set to $TOL = 0.0001$. We test the basic CIA decomposition with a benchmark example from the `https://mintOC.de` library, a real-world adsorption cooling machine problem [49], and generic data in Sections 9.4.1, 9.4.2, and 9.4.3, respectively. Finally, we discuss the results in Section 9.4.4. The content of this section is based on [222], Section 8.

9.4.1 Multimode mixed-integer optimal control problem

We consider the following MIOCP, which is a modified version of the *Egerstedt standard problem* from `https://mintOC.de`:

$$
\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{\omega}\in\Omega}\ & x_1(t_f)^2 + x_2(t_f)^2 \\
\text{s.t.}\ \ & \text{for a.a. } t\in[0,1]: \\
& \dot{x}_1(t) = -x_1(t)\omega_1(t) + (x_1(t)+x_2(t))\omega_2(t) + (x_1(t)-x_2(t))\omega_3(t), \\
& \dot{x}_2(t) = (x_1(t)+2x_2(t))\omega_1(t) + (x_1(t)-2x_2(t))\omega_2(t) + (x_1(t)+x_2(t))\omega_3(t), \\
& \boldsymbol{x}(0) = \boldsymbol{x_0}.
\end{aligned}
\tag{P3}
$$

Obviously, the problem includes 3 different modes, i.e., $n_\omega = 3$. As initial values, we use $\boldsymbol{x_0} := (0.5, 0.5)^T$. Furthermore, we add the TV constraint (3.8) to (P3) with a varying maximum number of switches $\sigma_{\max}$. Fig. 9.11 illustrates the differential state and control trajectories for $\sigma_{\max} = 20$, with relaxed binary controls as well as binary controls based on SUR, BnB, and AMDR. We remark that the control function constructed by SUR uses 70 switches and is therefore infeasible with respect to $\sigma_{\max} = 20$. The relaxed control values are greater than zero and less than one around $t \approx 0.45$ and for $t \geq 0.8$ such that the corresponding approximated state trajectories of BnB and AMDR are slightly different from the relaxed one for $t \approx 0.45$ on. We set the BnB iteration limit to $5 \cdot 10^6$; it stopped after 15.3 s with a (CIA-TV) objective value of $\theta = 9.1 \cdot 10^{-3}$ and $\mathscr{C} = 0.991855$ as the (P3) objective value. The execution of AMDR took 0.2 s and resulted in improved objective values of $\theta = 4.6 \cdot 10^{-3}$, respectively $\mathscr{C} = 0.991509$, which can be explained by the more uniform distribution of switches compared with the BnB solution.

Table 9.4 shows that for small instances, e.g., $N = 200$, the BnB algorithm constructs better (CIA-TV) objective values than AMDR if enough time is available. If the BnB scheme finds a good solution, it usually does so after a few million iterations. While the θ values of AMDR are close to those of BnB for $N = 200$, AMDR clearly outperforms BnB for larger instances. Its run time only increases slightly with the refinement of the grid, from about 0.1 s to at most 0.6 s. A `C++` implementation could further improve the run time since we used a prototype implementation in `Python`. It appears that selecting the next-forced control rather than the control with the maximum γ value is beneficial as part of the AMDR algorithm and tends to yield the solution with the smallest (CIA-TV) objective value.

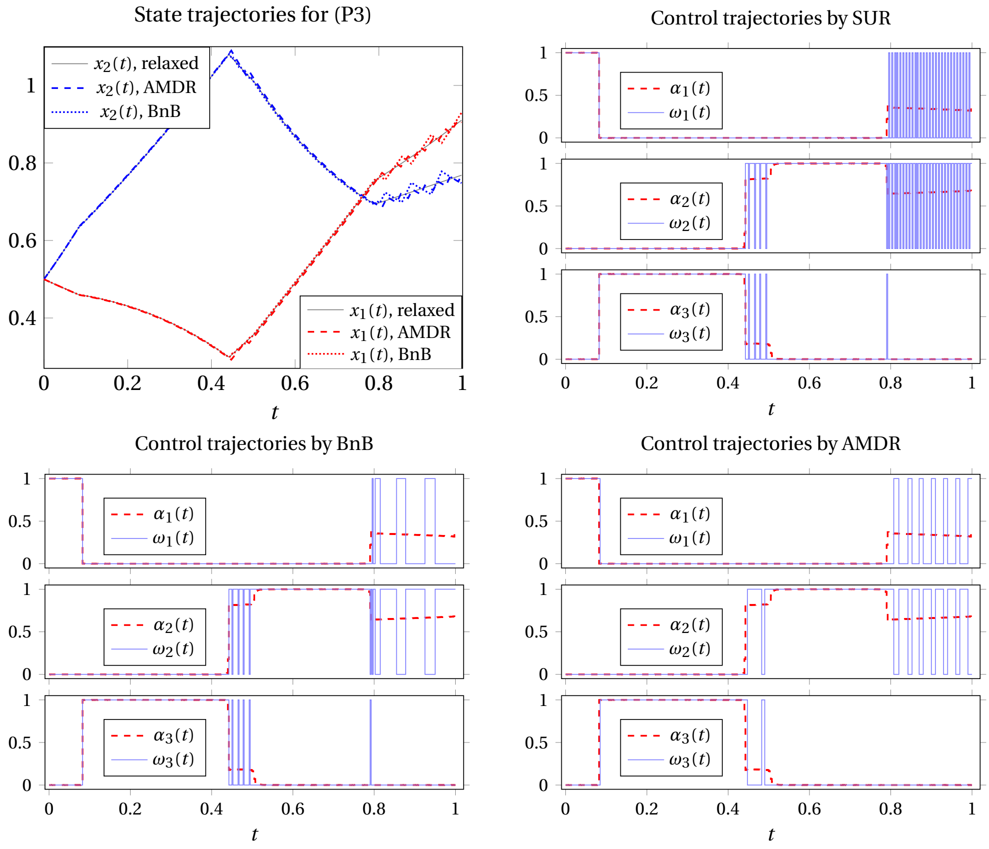


Figure 9.11: Differential state and control trajectories for test problem (P3): The problem was discretized with Direct Multiple Shooting and $N = 400$ intervals. The state trajectories based on SUR are very similar to the relaxed ones (i.e., those based on $\boldsymbol{\alpha}$), and we therefore omit their presentation.

9.4.2 Adsorption cooling machine problem

In [48, 49], a complex renewable energy system in the form of a solar thermal climate system with nonlinear system behavior is introduced as an MIOCP. The core of the system is an adsorption cooling machine, which can be switched on to intensify cooling of the ambient temperature. The goal is to control the room temperature within a comfort zone while minimizing the energy costs. We restrict the problem to two modes of the adsorption cooling machine, i.e., $n_\omega = 2$, and assume an entire day time horizon with control adjustment every four minutes, i.e., $N = 360$. We omit a detailed description of the system but refer to [49] and consider the relaxed binary control values as given therein and illustrated on the left of Fig. 9.12.

We use the AMDR scheme to calculate a candidate solution of the (CIA-TV) problem depending on $\sigma_{\max}$, which is optimal by Theorem 6.3.2(a). The right plot in Fig. 9.12 compares the optimal solutions with the (CIA-TV) objective values of BnB solutions with an increasing limit on the number of iterations. For small and large numbers of allowed switches, the deviation of the BnB solutions is small. One explanation for this is the limited degrees of freedom for small $\sigma_{\max}$, greatly limiting the width of the BnB tree. On the other hand, for large $\sigma_{\max}$, solutions can be found quickly with small θ values, with which many nodes can be pruned.

	AMDR		AMDR-NF		BnB, 5 million iter.		BnB, 50 million iter.		Bounds	
$\sigma_{\max}$	θ	CPU [s]	θ	CPU [s]	θ	CPU [s]	θ	CPU [s]	θ^*	$\theta^{\max}$
N = 200										
5	0.017899	0.11	0.017899	0.26	0.015212	14.35	0.015212	145.22	0.015212	0.09341
10	0.007622	0.12	0.007622	0.23	0.010297	14.29	0.007254	144.99	0.007254	0.05012
20	0.004368	0.16	0.004368	0.22	0.004368	14.73	0.004368	15.03	0.004368	0.02689
30	0.003828	0.1	0.003828	0.15	0.003828	0.58	0.003828	0.56	0.003828	0.01889
40	0.002991	0.13	0.002991	0.17	0.002991	0.0	0.002991	0.0	0.002991	0.01485
N = 400										
5	0.016816	0.21	0.016816	0.27	0.015164	14.67	0.014283	148.12	0.014283	0.09216
10	0.007562	0.28	0.007562	0.35	0.011264	14.34	0.008186	144.9	0.007562	0.04887
20	0.004607	0.18	0.004226	0.26	0.006600	14.28	0.005554	143.56	0.004202	0.02564
30	0.004607	0.18	0.003248	0.29	0.005212	14.36	0.004039	145.64	0.003075	0.01764
40	0.002604	0.31	0.002578	0.24	0.003764	14.56	0.003208	152.65	0.002351	0.01360
N = 800										
5	0.015952	0.58	0.015952	0.58	0.020405	15.21	0.013830	147.94	0.013830	0.09153
10	0.007535	0.47	0.007535	0.56	0.018007	14.87	0.010561	149.56	0.006933	0.04824
20	0.005405	0.47	0.005025	0.54	0.018007	14.96	0.010561	147.74	0.004116	0.02502
30	0.003315	0.4	0.002620	0.47	0.008500	14.69	0.006292	145.04	0.002620	0.01702
40	0.002018	0.57	0.002018	0.59	0.008390	14.39	0.005850	146.8	0.001914	0.01297

Table 9.4: Comparison of the (CIA-TV) objective values and run times of different solving methods for (P3) with varying $\sigma_{\max}$. AMDR refers to Algorithm 6.7, while AMDR-NF is a modification in which the *admissible* and *next-forced* control is selected to be active in Line 4 of Algorithm 6.6. By BnB, we refer to Algorithm 6.2 with the depth-first node selection strategy, where we set an iteration limit of 5 and 50 million nodes. We highlight the best obtained objective values in red. The last two columns show the optimal objective values and the upper bounds from Conjecture 7.1.

The deviation from the optimal solution is particularly striking for medium-sized $\sigma_{\max}$. For some instances, especially $10 \le \sigma_{\max} \le 20$, an increase in the iteration limit leads to negligible improvement because the BnB algorithm seems to remain in a suboptimal branch. We also compare the optimal solution of (CIA-TV) with the upper bound from Corollary 7.3 and find that the latter appears to be between 200 and 600 percent larger.

9.4.3 Upper bound evaluation for (CIA-TV) based on generic data

The two investigated MIOCPs exhibited a relatively large deviation of the optimal (CIA-TV) objective values from the derived upper bounds. We therefore generated uniformly distributed random values $\boldsymbol{a} \in \mathscr{A}_N$ for $N = 40$ equidistant intervals and $n_\omega = 2,3$ controls and examined the resulting ratio of the objective values to the upper bounds. We illustrate this comparison in Fig. 9.13, where we use the upper bound from Corollary 7.3 for $n_\omega = 2$ and that from Conjecture 7.1 for $n_\omega = 3$. The objective values θ, θ^* and bounds $\theta^{\max}$ decrease logarithmically with the increase in $\sigma_{\max}$, as expected. In contrast to the above MIOCPs, the (CIA-TV) objective values come close to the upper bounds, in particular for small $\sigma_{\max}$, but a relevant gap remains for larger $\sigma_{\max}$. The gap may be further reduced by using a larger sample size: here we considered only 1000 (CIA-TV) instances per $\sigma_{\max}$ value. We also note that the values generated by the

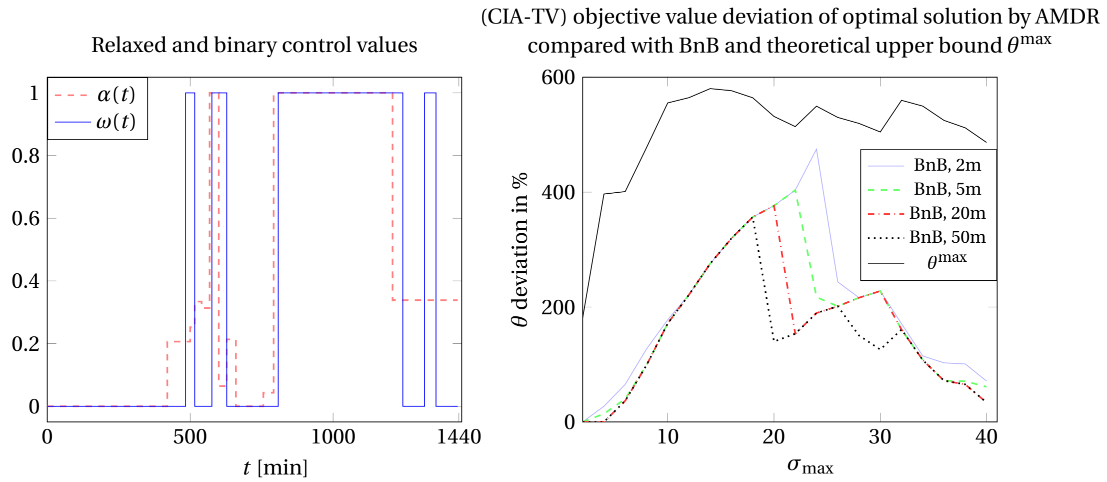


Figure 9.12: Left: Relaxed binary control values $\boldsymbol{\alpha}$ for the adsorption cooling machine problem for the time horizon of an entire day and exemplary approximated binary values $\boldsymbol{\omega}$ obtained by AMDR with $\sigma_{\max} = 8$. Right: Comparison of the (CIA-TV) objective values based on the upper bound $\theta^{\max}$ from Corollary 7.3 and BnB solutions with the optimal solutions constructed by AMDR and varying $\sigma_{\max}$ values. We report the deviation in percent. The number next to BnB in the legend indicates the maximum number of iterations, e.g., 2 million.

AMDR algorithm are very close to the optimal ones.

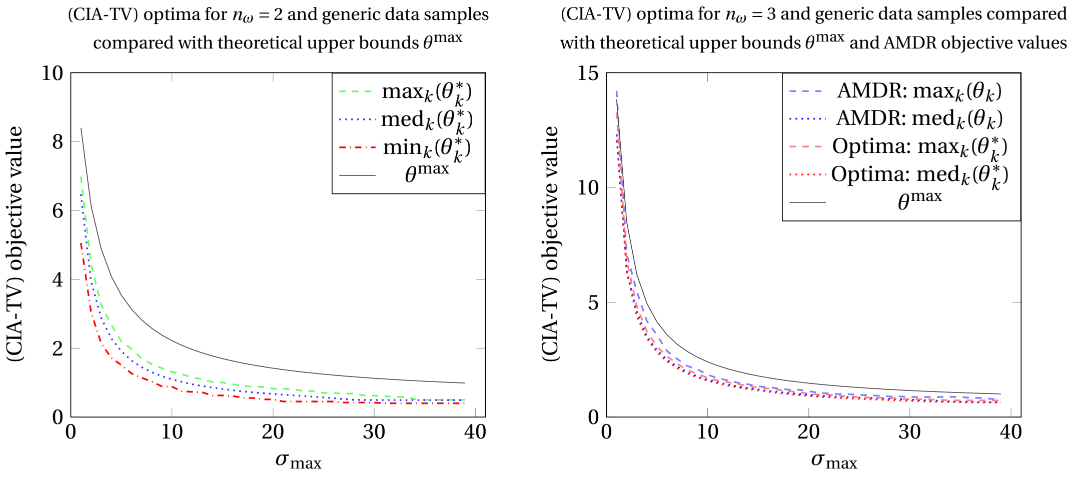


Figure 9.13: Optimal objective values θ_k^* of (CIA-TV) for randomly generated values $\boldsymbol{a} \in \mathscr{A}_N$ with $k \in [1000]$ samples, $N = 40$, and varying $\sigma_{\max}$ values compared with the derived upper bounds $\theta^{\max}$. We display the maximal, median, and minimal objective values of the samples for each $\sigma_{\max}$. Left: The optima are computed with the AMDR algorithm for the case $n_\omega = 2$. Right: Comparison of the values constructed by the AMDR algorithm with the optimal values obtained by the BnB algorithm.

9.4.4 Discussion

As expected based on the polynomial run time complexity, our prototype implementation of AMDR constructs (CIA-TV) feasible solutions quickly. The θ values mostly outperform those obtained by the BnB algorithm or are at least close to those of the BnB algorithm for problems with more than two binary controls. Consequently, the AMDR solution is itself a promising (CIA-TV) feasible solution. Alternatively, it can be a fast way to initialize the BnB algorithm with a competitive upper bound. As stated in Remark 6.12, the AMDR algorithm may also include combinatorial constraints other than the TV constraints.

For comparison with the BnB method, one restriction was that we only used the depth-first node selection strategy and could have tuned it more to achieve more competitive feasible solutions of (CIA-TV). Moreover, the BnB algorithm can accommodate a variety of combinatorial conditions of the (CIA-TV) problem, so it is generally advantageous.

We also note that our analysis mainly examines the (CIA-TV) objective value because it correlates with the (BOCP) objective value. With similar (CIA-TV) objective values, however, the smaller value may lead to a worse (BOCP) objective value – and vice versa. In particular, there may be several binary control functions with the same (CIA-TV) objective value but different (BOCP) objective values. In some instances, we observed that the AMDR algorithm generates a control function with suboptimal (BOCP) objective value since its switches are structurally delayed compared to the switches on bang-bang-arcs of the relaxed binary values. In this case, we tested, as a heuristic, shifting the AMDR binary values backward in time by $\lfloor \theta / \bar{\Delta} \rfloor$ intervals so that the control function is more similar to the relaxed binary values, which worked well.

Chapter 10

Applications

We present two application-driven case studies to illustrate the real-world modeling capacity of mixed-integer optimal control problems (MIOCPs). Section 10.1 deals with finding a minimum-fuel energy management strategy (EMS) for a hybrid electric vehicle (HEV) and is based on [212]. To the best of the author's knowledge, there are very only a few studies to date that investigate the application of mixed-integer optimal control (MIOC) in the field of cardiology. Section 10.2 provides a contribution to fill this gap and is based on [281]. We will deviate from the previous variable notation in some places to accommodate engineering conventions, especially concerning the integer control $\boldsymbol{v}$ that will indicate the velocity in this chapter.

10.1 Multiphase mixed-integer optimal control of hybrid electric vehicles

Automotive manufacturers and research centers have been significantly investing resources and efforts into the development of alternative propulsive technologies to lower fuel consumption and pollutant emissions in passenger and commercial vehicles. HEVs represent a concrete answer to address these problems. HEVs can reduce greenhouse gas emissions and fuel consumption while guaranteeing driver pleasure. Notwithstanding this, the growing complexity and degrees of freedom of current hybrid powertrain architectures impose a tailored supervisory EMS to unleash the full potential of the HEV in terms of fuel economy and driveability. Different gear choices and engine on/off modes give rise to a problem with discrete variables. To this end, we introduce a mode control function $m_{\mathrm{c}}(t)$ that represents the current driving mode at time t. Therefore, this study addresses the problem of finding the off-line EMS for the following MIOCP:

Problem 10.1
Find the continuous torque split control factor u over a compact set and the integer mode choice control m_{c} that minimizes the fuel consumption $\dot{m}_{\mathrm{f}}$

$$\mathscr{C} := \int_{t_0}^{t_f} \dot{m}_{\mathrm{f}}(t)\,\mathrm{d}t,$$

over the given time horizon $t \in [t_0, t_f] \subset \mathbb{R}$ and subject to:

$$\begin{aligned}
\text{Multiphase ODE:} \quad & \dot{\boldsymbol{x}}(t) && = \boldsymbol{f}(s(t), \boldsymbol{x}(t), u(t), m_{\mathrm{c}}(t)),\\
\text{Boundary Conditions:} \quad & \boldsymbol{x}(0) && = \boldsymbol{x}_0,\ \ \boldsymbol{x}(t_f) = \boldsymbol{x}_{\mathrm{f}},\\
\text{Path Constraints:} \quad & \boldsymbol{0}_{n_c} && \le \boldsymbol{c}(s(t), \boldsymbol{x}(t)),\\
\text{Vanishing Constraints:} \quad & \boldsymbol{0}_{n_d} && \le \boldsymbol{d}(\boldsymbol{x}(t), u(t), m_{\mathrm{c}}(t)),\\
\text{Combinatorial Constraints:} \quad & m_{\mathrm{c}} && \in \mathscr{V}.
\end{aligned}$$

where the differential states $\boldsymbol{x}$ include besides $\dot{m}_{\mathrm{f}}$ the battery state-of-charge and the internal combustion engine (ICE) cooling water temperature. $\boldsymbol{x}_0, \boldsymbol{x}_\mathrm{f}$ denote the initial and final state conditions. We use the smooth function $\boldsymbol{f}$ to describe the right-hand side of the powertrain model's ordinary differential equation (ODE). The system dynamics change according to the given phases $s(t) \in [n_p]$, with $n_p \in \mathbb{N}$ phases; thus, we write *multiphase*. The functions $\boldsymbol{c}$ and $\boldsymbol{d}$ represent here path and vanishing constraint mappings, respectively. The combinatorial restrictions on m_c are implied by the corresponding feasible control space $\mathcal{V}$.

The above problem will be specified with all of its variables and constraints in more detail in the Subsections 10.1.1 and 10.1.2.

Related work. Similar problems to Problem 10.1 were investigated with a fixed time horizon and an on-line EMS setting based on past research ([209, 111]) that serves as a comprehensive overview. Discrete dynamic programming has been extensively used for solving the nonlinear EMS problem due to its straightforward implementation and global optimality guarantees, with prominent examples in [76, 265]. However, the curse of dimensionality restricts this method to solving problems with a small number of states. Examples based on indirect optimal control methods can be found in [144, 234, 230].

Convex optimization methods have been proven to be beneficial to solve the off-line EMS problem ([211, 73]). However, their main drawback is the simplification of nonlinearities when it is applied as a linear or quadratic model. To cope with the nonlinear effects, in [167, 267] the authors propose the direct transcription of the optimal control problem (OCP) into an nonlinear program (NLP) that can be solved by using standard NLP solvers. Both approaches deal with mixed-integer problems, which arise when both continuous and discrete variables are embedded into the OCP. This leads to NP-hard problems that are computationally intractable for standard solvers when considering long time horizons.

There are several studies about control theory in the automotive field that builds upon the combinatorial integral approximation (CIA) decomposition ([37, 148, 184]); however, most neglect the combinatorial constraints by using rounding schemes such as sum-up rounding (SUR). A rare example application of multiphase MIOCP can be found in [38].

Contributions. This study investigates Problem 10.1 under real-world requirements, specifically:

- The powertrain operates in different modes depending on a given speed profile, which imposes the multiphase setting of the ODE.
- The dual clutch gearbox allows only a specific switching structure that this study proposes for *mode transition constraints*.
- Switching between the electric and hybrid driving mode during arbitrarily short periods of time is impossible, which translates into *minimum dwell time (MDT) constraints*.

We apply and test the generalized CIA decomposition that uses several NLP and mixed-integer linear program (MILP) steps from Section 4.5.3 with the idea to construct a feasible solution with a promising objective value for the complex MIOCP that entail multiphase, vanishing, state, and combinatorial constraints. Furthermore, we come back to the approach proposed in Section 4.4.3 for the inclusion of multiphase dynamics into (CIA).

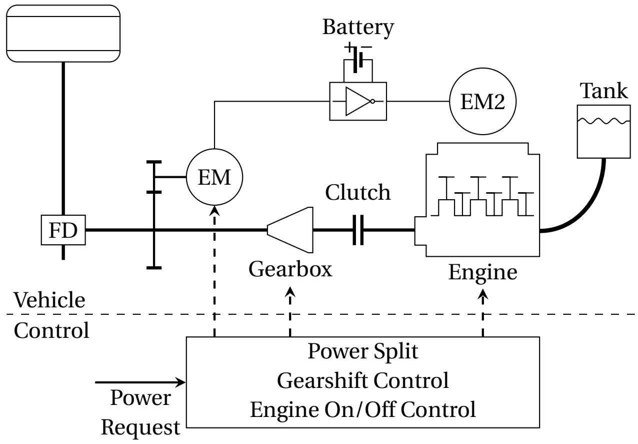


Figure 10.1: Schematic representation of the hybrid electric vehicle (HEV) with energy management strategy (EMS). Abbreviations indicate internal combustion engine (ICE), electric motor (EM), second electric motor (EM2) and final drive (FD).

Outline. Subsection 10.1.1 describes the powertrain model with its variables and constraints. Subsection 10.1.2 introduces the combinatorial constraints. Finally, the numerical case studies are discussed followed by the conclusions in Subsection 10.1.3 and 10.1.4, respectively. We provide details on the ODE model in the Supplemental material 10.1.5.

10.1.1 Model description

This section presents the powertrain shown in Figure 10.1. It consists of an ICE, an EM that provides boosting and regenerative braking, and an EM2 connected to the ICE through a belt. This can be used to recharge the battery when the vehicle stands still. The engine is connected to a 7-speed dual clutch gearbox, while the EM is coupled to the output shaft of the gearbox with an additional gear set. The FD and the differential transmit the propulsive power to the wheels. The fuel tank and the battery are used for on-board energy storage.

In order to correctly evaluate the fuel consumption and the battery's state-of-charge while retaining a simple and fast estimation, we use a backward quasi-static modeling approach ([111, 85]) to describe the non-causal relationships between the powertrain subsystems. By making this choice, the number of states needed to describe the powertrain were reduced. We consider the speed profile $v(t),\ t \in [t_0, t_f]$, of a given driving cycle as an exogenous variable and drop the driver model that would have otherwise been necessary to follow a reference speed profile; thus, reducing the complexity of the HEV model. The efficiencies and parameters of the main subsystems were introduced by means of look-up tables; hence, making it possible to implement a model with nonlinear data. Furthermore, we cast the model into a *multiphase* problem, in which a different set of model functions applies for each phase.

Dividing the time horizon into phases

Given a speed profile $v(t) \in \mathbb{R}_{\geq 0}$ for $t \in [t_0, t_f]$, we assume a sampling time of one second and discretize the driving cycle accordingly with N intervals and the grid set $\mathcal{G}_N = \{t_0 < t_1 \ldots < t_N = t_f\}$, where the generic interval length is $\Delta_j = 1\text{s},\ \ j \in [N]$. In each period Δ_j, we consider the

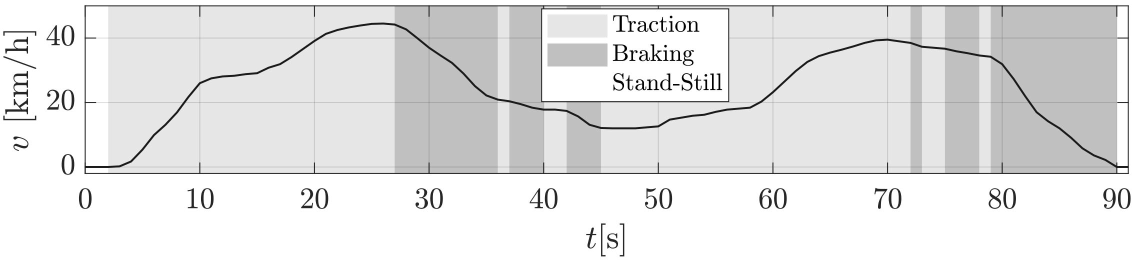


Figure 10.2: Velocity profile and phases in an exemplary driving cycle.

first-order differential equation describing the vehicle's longitudinal dynamics ([111]):

$$m_{\text{eq}} \cdot \frac{\mathrm{d}v}{\mathrm{d}t}(t) = F_{\text{t}}(v(t)) - F_{\text{a}}(v(t)) - F_{\text{r}}(v(t)), \tag{10.1}$$

where m_{eq} is the equivalent mass of the vehicle, $F_{\text{t}}(t)$ is the traction force, $F_{\text{a}}(t)$ is the aerodynamic drag force, and $F_{\text{r}}(t)$ is the rolling resistance force. We discretize Eq. (10.1) and approximate $\frac{\mathrm{d}v}{\mathrm{d}t}(t)$ with the explicit Euler scheme that is applied for $v(t)$. We note that this integration scheme is a simple approximation; however, it appears to be appropriate when the grid length is small. By rearranging (10.1) in terms of $F_{\text{t}}(t)$, we identify three possible operating modes for each interval $j \in [N]$:

1. $F_{\text{t}} > 0$, *traction*;

2. $F_{\text{t}} < 0$, *braking*;

3. $F_{\text{t}} = 0$, *stand-still.*

Thereafter, we collect all of the intervals that were subject to the same operating mode in the $n_p = 3$ model *phases* (see Figure 10.2), assuming disjoint subintervals $[t_{j-1}, t_j)$, $j \in [N]$, of the time horizon. Let the function

$$s : [t_0, t_f] \to [n_p], \quad s(t) = p \in [n_p], \tag{10.2}$$

map each time point t to its associated phase p.

Control variables

We introduce for $t \in [t_0, t_f]$ the integer control variable $m_{\text{c}}(t) \in [n_\omega]_0$ with $n_\omega = 7$ and the continuous control variable $u(t) \in [u_{\min}, 1]$, where $u_{\min} < 0$, models the powertrain's mixed-integer nature. The variable $m_{\text{c}}(\cdot)$ can help determine whether to operate the HEV in the *electric mode* (EM is the only power source, the ICE is turned off, and the clutch is disengaged) or in *hybrid mode* (EM and ICE are simultaneously used to power the vehicle). $m_{\text{c}}(\cdot)$ receives a value of 0 whenever the vehicle is required to operate in the electric mode. It can also take on values in the set $[n_\omega]$ when it operates in the hybrid mode with selected gears $G \in [7]$ ranging from the first to the seventh, respectively. The control variable $u(\cdot)$ allows to regulate the torque split between the ICE and the EM in each hybrid operating mode. Specifically, by varying the control u, we identify three different hybrid configurations at $t \in [t_0, t_f]$:

1. if $u(t) \in [u_{\min}, 0)$: *load point shift* (LPS). The operating point of the ICE is shifted toward higher loads and the exceeding power recharges the battery;
2. if $u(t) = 0$: *ICE mode.* ICE is the only power source and it propels the vehicle;
3. if $u(t) \in (0, 1]$: *boost.* ICE and EM can cooperate to fulfill the power requirements for the wheels.

Figure 10.3 illustrates these scenarios during the *traction/braking* phase. The control u is dropped in the *stand-still* phase since only two different scenarios are applicable: Either the ICE is turned off ($m_c(t) = 0$), or the ICE is turned on ($m_c(t) \in [n_\omega]$) so that the battery can be recharged by means of EM2 that is operated with a fixed value for the speed and torque provided by the ICE.

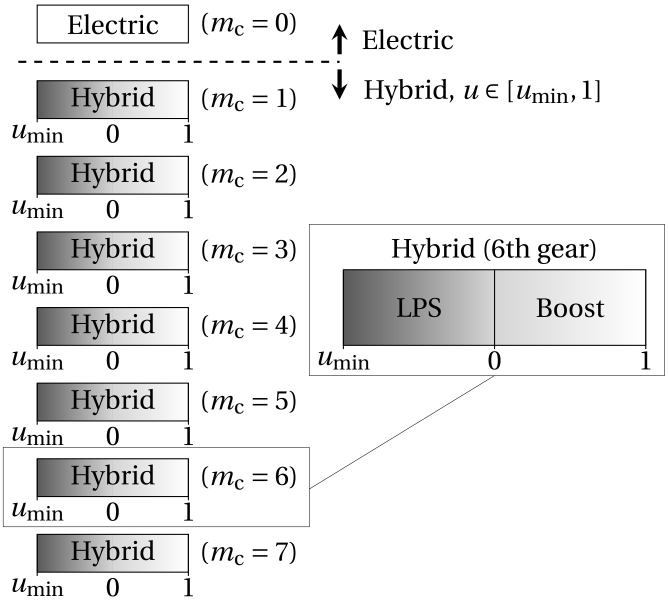


Figure 10.3: Mixed-integer control choices during *traction/braking*. During *stand-still*, u is dropped.

Differential states

We model the powertrain's dynamics with the fuel mass flow rate $\dot{m}_f(t)$, the battery state-of-charge $b_s(t)$, and the ICE cooling water temperature $T_w(t)$ as differential states. The latter is needed to account for the higher fuel consumption during the ICE heating-up transient ([256, 163]). We express the dependencies of the ODE for $t \in [t_0, t_f]$ as:

$$\begin{aligned}
\dot{m}_f(t) &= \mathrm{f}_{m_f}(s(t), m_f(t), T_w(t), u(t), m_c(t)),\\
\dot{T}_w(t) &= \mathrm{f}_{T_w}(s(t), T_w(t), m_f(t)),\\
\dot{b}_s(t) &= \mathrm{f}_{b_s}(s(t), b_s(t), u(t), m_c(t)).
\end{aligned}$$

For a detailed description of the smooth functions f(·) and the underlying ODE model we refer to the Supplemental Material 10.1.5 and to [210]. We group the differential states into vectors

$\boldsymbol{x}(\cdot)$ and their right-hand side functions into $\boldsymbol{f}(\cdot)$ as proposed in Problem 10.1.

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(s(t), \boldsymbol{x}(t), u(t), m_\mathrm{c}(t)). \tag{10.3}$$

Outer convexification

For applying the CIA decomposition, we introduce binary controls $\omega_i(t) \in \{0,1\}$, for $i \in [n_\omega]_0$, $t \in [t_0, t_f]$ that indicate the integer realization of $m_\mathrm{c}(t)$:

$$\omega_i(t) = 1, \quad \Leftrightarrow \quad m_\mathrm{c}(t) = i.$$

Let $\boldsymbol{f}_i$ denote the model function where $m_\mathrm{c}(t) = i$, for $i \in [n_\omega]_0$, holds. In this way, we reformulate Eq. (10.3) and obtain the outer convexified dynamics for $t \in [t_0, t_f]$:

$$\dot{\boldsymbol{x}}(t) = \sum_{i=0}^{n_\omega} \omega_i(t) \cdot \boldsymbol{f}_i(s(t), \boldsymbol{x}(t), u(t)), \tag{10.4a}$$

$$1 = \sum_{i=0}^{n_\omega} \omega_i(t), \tag{10.4b}$$

where (10.4b) is needed because the definition of the integer control m_c implies mutually exclusive operation modes. Thus, we apply the outer convexification approach from Section 4.1, just with the difference of an altered number of binary control functions ω_i.

Path and vanishing constraints

The state-of-charge has to fulfill path constraints in order to preserve durability and reliability of the electric buffer. The choice of these limits $b_\mathrm{min}, b_\mathrm{max} \in [0,1]$ is preference specific and is generally expressed as:

$$b_\mathrm{min} \leq b_\mathrm{s}(t) \leq b_\mathrm{max}.$$

The operating points for the ICE and EM torque and the internal current for the battery have to be within a realistic range. This restricts the choices of the continuous and binary controls. We model these restrictions by mode specific lower and upper bounds $u_{\mathrm{lb},i}, u_{\mathrm{ub},i} \in [u_\mathrm{min}, 1]$, $i \in [n_\omega]_0$, for $u(\cdot)$ and obtain vanishing constraints:

$$0 \leq \omega_i(t) \cdot (u(t) - u_{\mathrm{lb},i}), \tag{10.5a}$$

$$0 \leq \omega_i(t) \cdot (u_{\mathrm{ub},i} - u(t)). \tag{10.5b}$$

To avoid numerical issues, we relax (10.5) by replacing zero with the parameter $\varepsilon = -1\mathrm{e}^{-4}$. We chose the above indicator formulation due to its tight relaxation compared with other formulations such as the *Big M* method, please see [138] for further details.

10.1.2 Combinatorial constraints

Technical requirements in realistic scenarios imply combinatorial constraints. We already introduced the combinatorial constraint Eq. (10.4b), which ensures that exactly one mode is active for all time points. This section discusses the further restrictions that includes prefixing, MDT, and mode transition constraints. Because the combinatorial constraints appear more intuitive for the discrete setting, we define them with respect to the discretized binary control variables $\boldsymbol{w} \in \{0,1\}^{(n_\omega+1)\times N \times n_p}$. We neglect the interval length Δ_j in the constraint formulation since we apply an equidistant grid. As described in Section 4.4.3, the multiphase dynamics setting gives rise to binary variables that are also indexed regarding the phase $p \in [n_b]$ and that needs to satisfy the phase fixing constraint (4.17) as part of the multiphase (CIA) problem from Definition 4.16.

Prefixing constraints

We restrict the set of feasible gear choices to satisfy the minimum and maximum ICE speed. Since the velocity profile is known a priori, it is possible to pre-calculate the allowed gears for each interval $j \in [N]$. Therefore, we exclude some options for all phases $p \in [n_p]$:

$$w_{i,j,p} = 0, \quad \text{if gear } i \text{ is invalid at interval } j. \tag{10.6}$$

Minimum dwell time constraints

Since we deal with a problem subject to multiphase dynamics, the arising MDT constraints are a bit more general than defined before (e.g. in (4.11)-(4.12)). Problem 10.1 requires minimum up (MU) times for the electric and hybrid mode that can overlap different phases. Therefore, we introduce sets of the dwell time coupled controls

$$S_c^e := \left\{(0,p) \mid p \in [n_p]\right\}, \qquad S_c^h := \left\{(i,p) \mid i \in [n_\omega],\ p \in [n_p]\right\},$$

with the electric and hybrid specific MU times $U_{S_c^e}, U_{S_c^h} \in \mathbb{N}$. The constraints are defined for $S_c \in \{S_c^e, S_c^h\}$, $j = 2,\dots,N-1$, $l = j+1,\dots,\min\{N, j+U_{S_c}\}$ as:

$$\sum_{(i,p)\in S_c} w_{i,l,p} \geq \sum_{(i,p)\in S_c} (w_{i,j,p} - w_{i,j-1,p}). \tag{10.7}$$

Figure 10.4 illustrates an example of the MU time and mode transition constraints.

Mode transition constraints

We translate the introduced constraint class from Section 3.2.6 into the discrete setting. By mode transition restrictions we refer to the situation in which the activation of one control $w_{i_1,j,p}$ excludes some control indices i_2 from activation in the time step $j+l$, $l \geq 1$:

$$w_{i_1,j,p} = 1 \qquad \Rightarrow \qquad w_{i_2,j+l,p} = 0.$$

These restrictions are motivated by the dual-clutch gearbox at hand. This can switch one gear up or down per second, which is independent from the active phase. In practice, the driver can use the time during the electric mode to change the gear setting; however, it is limited by one gear shift per second. For all index tuples, (i_a, p_a), $i_a \in [n_\omega]$, and $p_a \in [n_p]$, representing the active mode and phase, we introduce the allowed control indices "neighborhood" for $l = 1,\dots,5$ as:

$$\mathscr{I}^A(i_a, p_a, l) := \left\{(i,p) \mid p \in [n_p],\ i = \max\{1, i_a - l\},\dots,\min\{n_\omega, i_a + l\}\right\}.$$

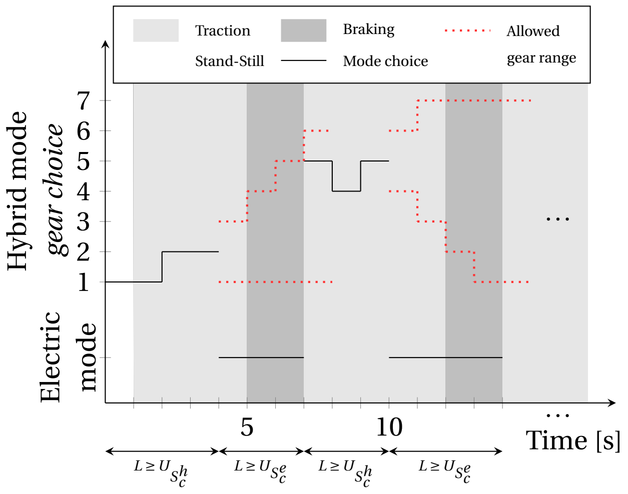


Figure 10.4: Illustration of the MU time and mode transition constraints. If activated, the hybrid, respectively electric, mode has to stay active for a duration L greater or equal to $U_{S_c^h}$, respectively $U_{S_c^e}$ even though there is no MU time for activating the individual gears. The mode transition constraint ensures that the driver can switch at most one gear up or down per second. The time during the electric phase can be used to change gears; thus, increasing the range of the allowed gears when continuing in hybrid mode (represented with the dotted lines). Both constraint classes have to be satisfied independently of the phases, which are depicted in the background.

Then, we define the constraint for all $i_a \in [n_\omega]$, $p_a \in [n_p]$, $l \in [5]$, $j = 1 + l, \dots, N$ as:

$$1 \;\geq\; w_{i_a, j-l, p_a} + \sum_{(i,p) \notin \mathcal{I}^A(i_a, p_a, l)} w_{i,j,p}. \tag{10.8}$$

10.1.3 Numerical results

We perform a case study of Problem 10.1 applied on the world harmonized light-duty vehicles test procedure (WLTP), which represents a real and challenging optimization problem due to a long time horizon and the frequent activation of vanishing constraints. Due to the combination of different constraints and the requirement not to refine the discretization, we apply the generalized CIA decomposition with multiple rounding steps from Section 4.5.3, i.e., Algorithm 4.5, for constructing feasible solutions of Problem 10.1 with promising objective value. We employ the multiphase problem (MCIA) as binary approximation problem, which has been theoretically justified in Section 4.4.3. The described combinatorial constraints from Section 10.1.2 are added to (MCIA). Before presenting result for the WLTP, we show exemplarily how the relaxed and binary controls behave as part of the rounding problem in connection with the combinatorial constraints.

Used discretization, hardware, and software

All computations were conducted on a Dell XPS15 desktop PC with an Intel Core i7-6700HQ CPU and 16 GB RAM running Ubuntu 16.04. We follow a first-discretize-then-optimize approach in the sense that we discretize the ODE with the direct collocation method and Lagrange interpolation polynomials; see [29] for more details. For the control discretization, we assume piecewise constant controls on the equidistant time grid $\mathscr{G}_N$. To parse the NLPs we used `CasADi` v3.4.5 ([9]) within the `Python` 2.7 environment, while the solution is provided by the sparse NLP solver `IPOPT` v3.12.3 ([264]), running the linear solver `MA97` from [11]. We applied *pycombina*'s branch-and-bound (BnB) algorithm 6.2 to solve the (MCIA) problems.

Exemplary (CIA) rounding step

We illustrate the functionality of the (MCIA) problem with the driving cycle from Figure 10.2. We note that $\mathscr{S}_1$ corresponds to the indices of all controls and phases, see Definition 4.23, meaning we optimize over all these variables. After solving (NLP($\mathscr{S}_1$)), we obtain the relaxed binary control values $\boldsymbol{a}$, which we depict in Figure 10.5 with the dashed lines. By observing this figure, the solution is almost of bang-bang type. We require a MU time of $U_{S_c} = 5\,\text{s}$ for both the electric and the hybrid mode. Whereas the relaxed solution satisfies already the mode transition constraint, this is not the case regarding the MU time constraints.

In the next step, we solve (MCIA($\mathscr{S}_1$)) so that we obtain binary values $\boldsymbol{w}$ that fulfill all combinatorial constraints, as depicted with the gray lines in Figure 10.5. The binary values approximate the relaxed ones quite well and the few differences are mainly due to the MU time. For instance, this can occur during the second activation of the electric mode.

Case study with the WLTP driving cycle

The velocity profile of the WLTP driving cycle is given in Figure 10.6. We solve Problem 10.1 applied to this driving cycle with Algorithm 4.5 and $n_{\text{dec}} = 2, 3$ decomposition steps. Moreover, we set $u_{\min} = -1$ as a lower bound for the torque split control. The case $n_{\text{dec}} = 2$ refers to the basic CIA decomposition, where we used a predefined gearshift profile obtained by applying a heuristic algorithm[1] and we therefore optimize only the binary choice between the electric and hybrid mode. For the algorithmic case $n_{\text{dec}} = 3$, we first optimize the gear and electric mode choices, i.e.,

$$\mathscr{S}_1 = \left\{(i, p) \mid i \in [n_\omega]_0,\ p \in [n_p]\right\}.$$

Afterwards, we fix all gear choice variables in the second NLP and CIA problem, i.e., we set

$$\mathscr{S}_2 = \left\{(i, p) \mid p \in [n_p],\ i = 0\right\}$$

to achieve optimization between the electric and hybrid mode and, complementary, we fix the gearshift pattern found in the previous step and use it as an exogenous variable in (NLP($\mathscr{S}_2$)).

To compare the proposed algorithm with a method constructing a global optimal solution, we solved Problem 10.1 also with a backward dynamic programming approach ([265]); however, we skip detailed dwell-time scenarios since this is beyond the scope of this research. We collect in Table 10.1 the values of the normalized total fuel consumption of the three approaches with varied MU times $U_{S_c^e}, U_{S_c^h}$ from one to five seconds.

The objective value increases progressively from the first to the last NLP required to solve the problem, as expected from the algorithmic approach to progressively increase the number of

[1] The heuristic gearshift strategy is speed-dependent, which reflects a normal driver's behavior. When the ICE speed is above or below a certain threshold there will be an upshift or downshift, respectively.

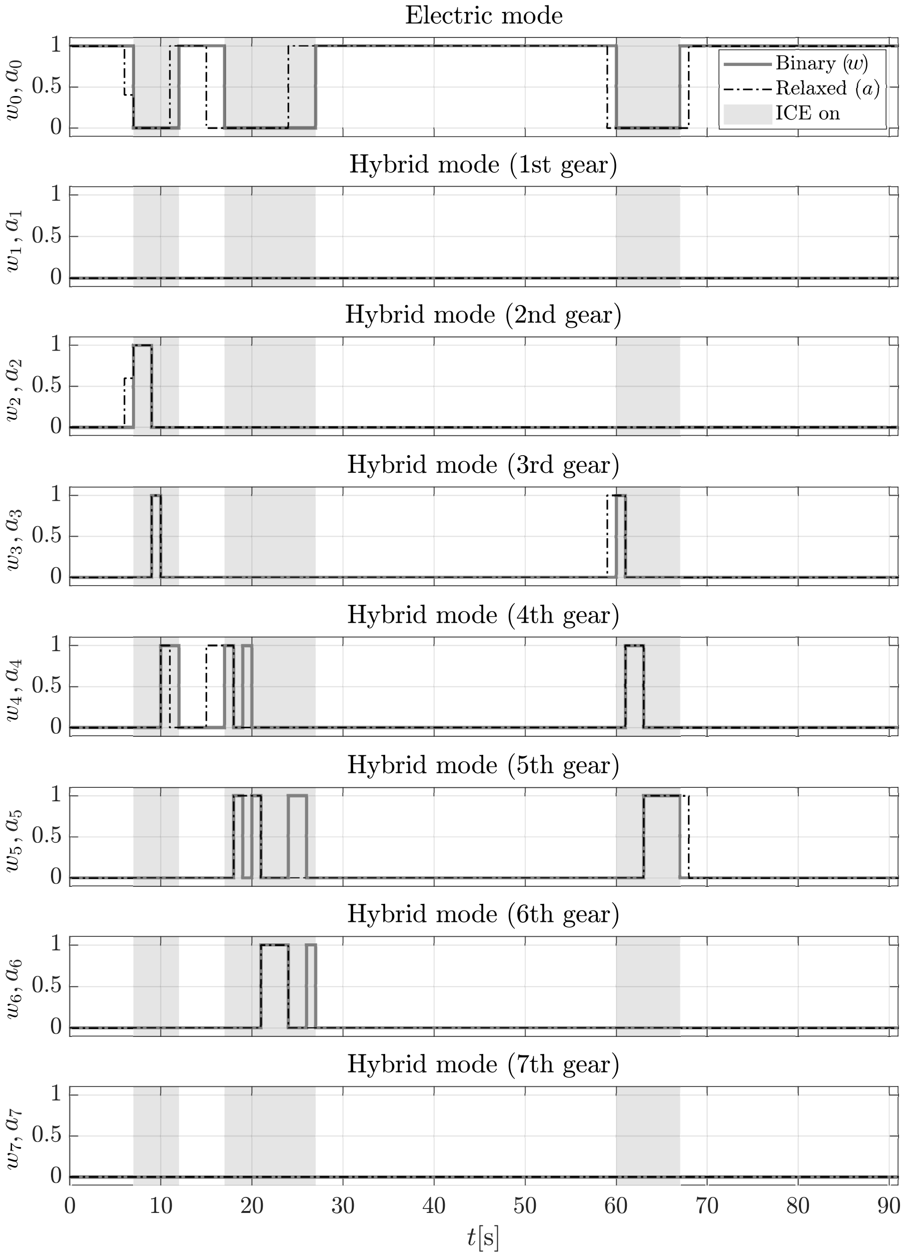


Figure 10.5: Relaxed control values $\boldsymbol{a}$ from (NLP($\mathscr{S}_1$)) and binary control values $\boldsymbol{w}$ obtained by (MCIA($\mathscr{S}_1$)) as an exemplary CIA rounding solution for the test driving circle from Figure 10.2. The gray background indicates that the ICE is turned on, i.e., the vehicle is operated in the hybrid mode.

fixed binary variables. In addition, the fuel consumption increases with increasing dwell times, which involves a substantially decreased number of switches (from 54 to 41); thus, providing a better driveability. When comparing the predefined and optimized gearshift scenarios, we obtain for the latter savings of 13.45% and 12.05% for the fuel consumption when the MU time is set to 5 s and 1 s, respectively. Note that this comes at the expense of an increased run time,

	MU time: 1 s		MU time: 5 s	
Problem	$\mathscr{C}$ [-]	Run time [s]	$\mathscr{C}$ [-]	Run time [s]
Algorithm 4.5, optimized gearshift, $n_{\text{dec}} = 3$				
(NLP($\mathscr{S}_1$))	0.8528	3190	0.8528	3190
(NLP($\mathscr{S}_2$))	0.8530	872	0.8592	1041
(NLP($\mathscr{S}_3$))	0.8534	263	0.8655	993
Algorithm 4.5, predefined gearshift, $n_{\text{dec}} = 2$				
(NLP($\mathscr{S}_1$))	0.9702	1102	0.9702	1102
(NLP($\mathscr{S}_2$))	0.9703	283	1.0000	769
Dynamic Programming, optimized gearshift				
-	0.8662	76910	-	-
Dynamic Programming, predefined gearshift				
-	0.9852	10803	-	-

Table 10.1: Comparison of the normalized objective function $\mathscr{C}$ and the run time (CPU) for the solution of Problem 10.1 obtained by dynamic programming and the NLPs from the generalized CIA decomposition Algorithm 4.5, with either optimized or predefined gear choices and varying minimum up (MU) time constraints for the WLTP driving cycle.

since a total of 4325 s instead of 1385 s is required to optimize the gearshift when the MU time is set to 1 s. We left out the run times of the CIA problems, because the tailored BnB feature of *pycombina* runs only for a few seconds.

Dynamic Programming is meant to provide globally optimal solutions. However, in a practical implementation, the solution $x(t_{j+1})$ of a forward integration on time interval $[t_j, t_{j+1}]$ is usually different from the values in the state space tabulation. One way to reduce this impact on the outcome is to use interpolation schemes or fine tabulation grids, albeit at high computational cost due to the curse of dimensionality [25]. This effect, possibly increased by using different integration schemes, is also the reason why in our implementation the objective function value of the Dynamic Programming solution has a higher objective function value than the one found by our direct optimization approach. Nevertheless, we see the similarity of the found solutions as an indication for the quality of our new approach.

Figure 10.6 presents the evolution of the state and control trajectories for both the predefined and optimized gearshift scenarios for Algorithm 4.5. It is worth noting how the variation of the state profiles between (NLP($\mathscr{S}_1$)) and (NLP($\mathscr{S}_2$)) (6th plot in Figure 10.6) is more pronounced by considering the optimization of the gearshift. This is mainly due to the enforcement of combinatorial constraints after (CIA($\mathscr{S}_1$)). The difference from (NLP($\mathscr{S}_2$)) to (NLP($\mathscr{S}_3$)) is marginal since a negligible rearrangement of the switches is needed in (CIA($\mathscr{S}_2$)).

We notice that the electric mode is always active during braking and standstill so that a large part of the kinetic energy can be recovered and in standstill the ICE is switched off, i.e., there is no recharging of the battery. On the contrary, the traction phase is characterized by both

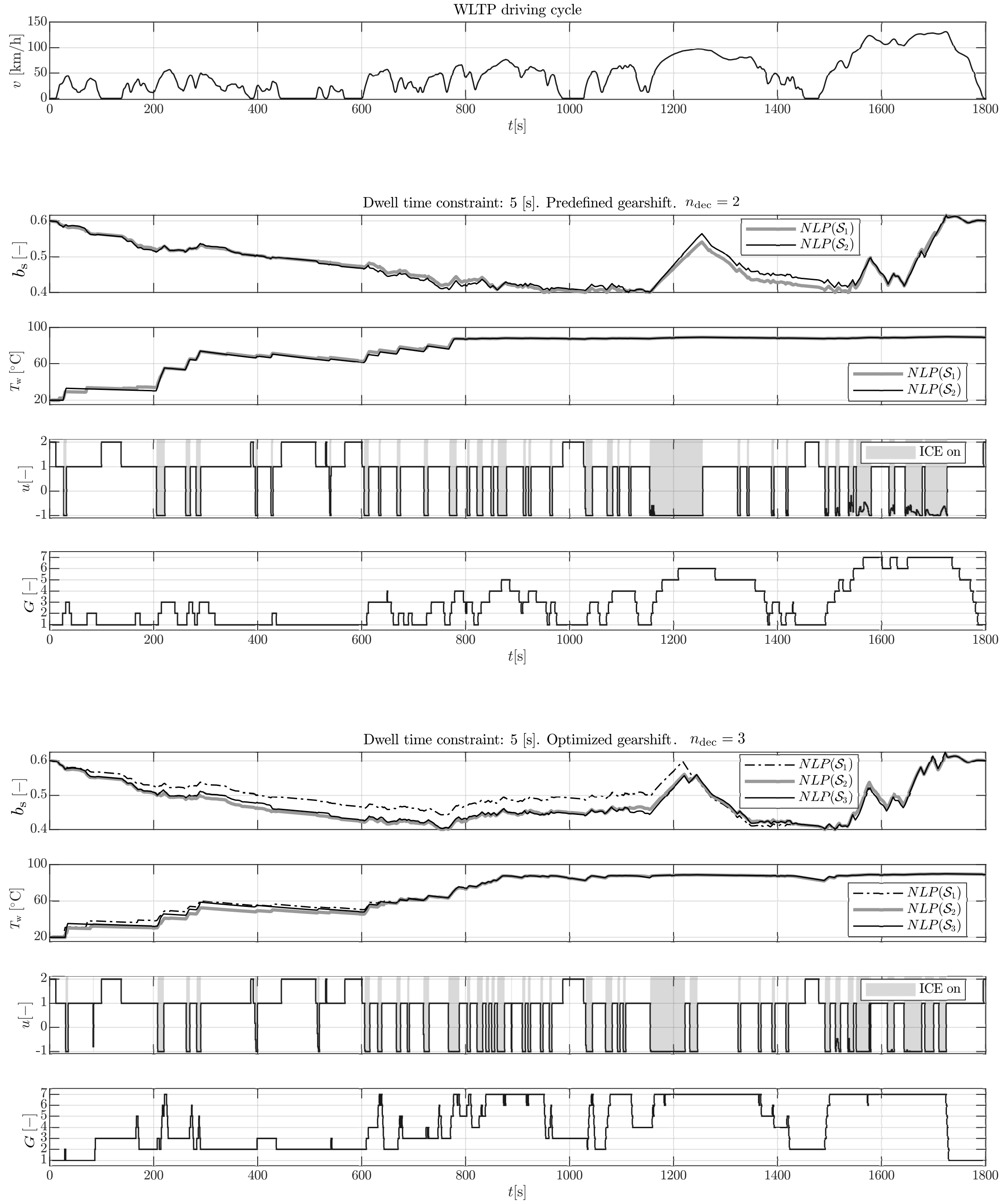


Figure 10.6: Top: velocity profile for the WLTP driving cycle. Plots 2-9: profiles of the state-of-charge for the battery $b_{\rm s}$, the ICE cooling temperature $T_{\rm w}$, the torque split u and the gear choice G of the solutions obtained with the predefined and optimized gearshift approaches of Algorithm 4.5 and with an MU time equal to five seconds. We imposed the constraints $b_{\rm s}(t_f) = 0.6$ and $0.4 \leq b_{\rm s}(t) \leq 0.7$. The first set of plots depicts the results considering predefined gearshifts, whilst the second set of plots consider optimized gearshifts.

singular arcs and bang-bang control profiles.

Finally, it can be observed in Figure 10.7 how the optimal control solutions translate into different operating points of the electric drive (ED, electric motor plus inverter) and ICE. The optimized gearshift entails a higher gear selection in comparison to the predefined gears; thus, allowing the ICE to operate with a greater torque and a lower speed within a higher efficiency region. On the other hand, the difference among the operating points of the ED is negligible since the EM is directly coupled through a constant transmission ratio to the FD shaft.

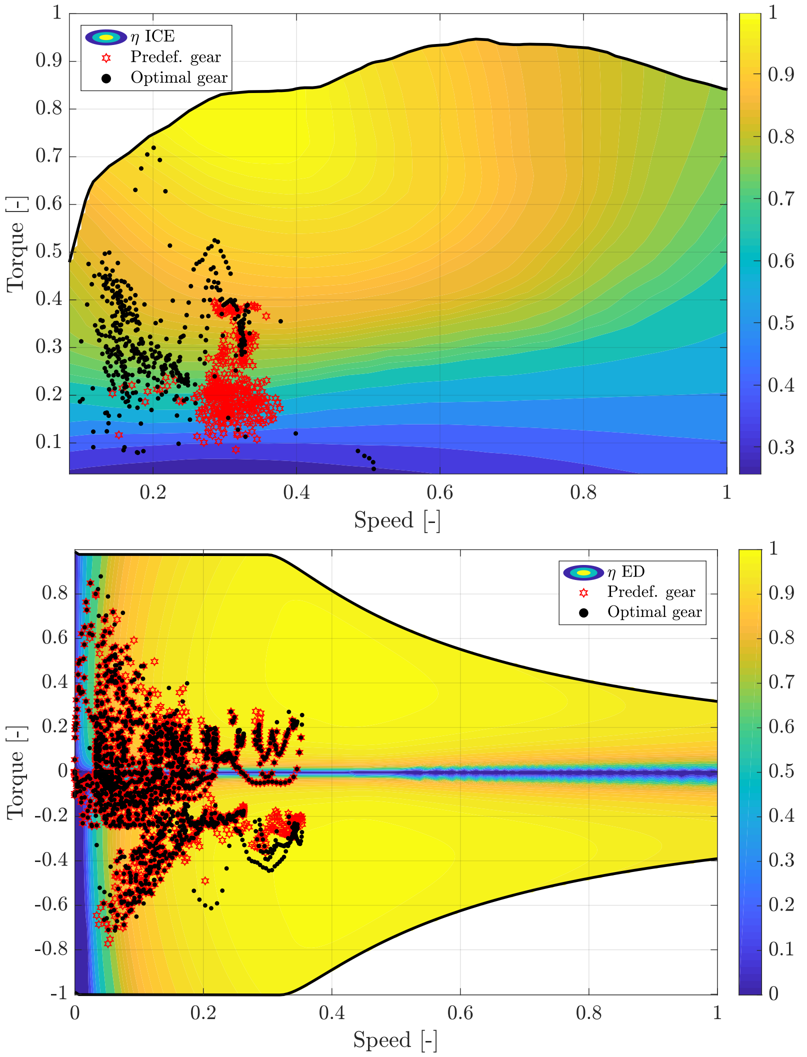


Figure 10.7: Internal combustion engine (ICE) (above) and ED (electric motor plus inverter) operating points (below) of the WLTP driving cycle for the predefined and optimized gear selection scenarios. The speed, torque, and efficiency η (right color bar) have been normalized for confidentiality reasons. The continuous black line depicts the torque limits.

10.1.4 Conclusions

This case study uses the generalized CIA decomposition with multiple NLP and rounding steps to address the solution of multiphase MIOCPs applied to the EMS for an HEV. Our algorithm is able to cope with a general problem class including multiphase, vanishing, state, and combinatorial constraints. This study showcased the effectiveness of our approach for the realistic WLTP driving cycle. It also demonstrated accurate solutions with reasonable run times, making it possible to benchmark causal controllers or objectively compare different powertrain architectures. The findings from this research can be beneficial to researchers and professionals that work in the field of HEVs. Future work in this area may address the on-line application of the proposed algorithm, more sophisticated approaches to include terminal or path constraints and other gearbox settings.

10.1.5 Supplemental material

In this Supplemental material, we present the quasi-static model of the full parallel HEV discussed in this Chapter.

Longitudinal vehicle dynamics

We consider the front and rear tires' radius as speed $v(t)$ dependent:

$$r_{\mathrm{f}}(t) = c_{\mathrm{f},1} + c_{\mathrm{f},2} \cdot v(t) + c_{\mathrm{f},3} \cdot v^2(t), \tag{10.9a}$$

$$r_{\mathrm{r}}(t) = c_{\mathrm{r},1} + c_{\mathrm{r},2} \cdot v(t) + c_{\mathrm{r},3} \cdot v^2(t), \tag{10.9b}$$

where the coefficients $c_{\mathrm{f/r},1}$, $c_{\mathrm{f/r},2}$ and $c_{\mathrm{f/r},3}$ are to be determined experimentally.
Let m_{car} be the mass of the car and m_{rot} be the equivalent mass of the moment of inertia of rotating components in the powertrain. Then, the equivalent vehicle mass is

$$m_{\mathrm{eq}} = m_{\mathrm{car}} + m_{\mathrm{rot}}. \tag{10.10}$$

The definition of the equations governing the longitudinal dynamics of the vehicle allows to evaluate the required traction torque $\mathcal{T}_{\mathrm{t}}$ at the rear wheels:

$$m_{\mathrm{eq}} \cdot \frac{\mathrm{d}v}{\mathrm{d}t}(t) = F_{\mathrm{t}}(t) - F_{\mathrm{a}}(t) - F_{\mathrm{r}}(t), \tag{10.11a}$$

$$\mathcal{T}_{\mathrm{t}}(t) = (F_{\mathrm{r}}(t) + F_{\mathrm{a}}(t) + m_{\mathrm{eq}} \cdot \frac{\mathrm{d}v}{\mathrm{d}t}(t)) \cdot r_{\mathrm{r}}(t), \tag{10.11b}$$

where F_{t} is the traction force at the wheels and F_{a}, F_{r} are the aerodynamic and rolling resistance forces, respectively, defined as

$$F_{\mathrm{a}}(t) = c_{\mathrm{ae}} \cdot v(t)^2, \tag{10.12a}$$

$$F_{\mathrm{r}}(t) = \left(c_{\mathrm{rl},0} + c_{\mathrm{rl},1} \cdot v(t)^{c_{\mathrm{rl},2}}\right) \cdot m_{\mathrm{car}} \cdot g. \tag{10.12b}$$

In the above equations, the coefficients c_{ae}, $c_{\mathrm{rl},0}$, $c_{\mathrm{rl},1}$ and $c_{\mathrm{rl},2}$ are identified experimentally and g is the gravitational acceleration.

Internal combustion engine model

The angular speed of the ICE follows from kinematic relationships and reads

$$\omega_e = \begin{cases} \dfrac{v(t)\cdot \tau_{fd}\cdot \tau_{gb,e}(G)}{r_r(t)}\cdot \Lambda & \text{if } \omega_e > \omega_{idle}, \\ \omega_{idle}\cdot \Lambda & \text{if } \omega_e \leq \omega_{idle}, \end{cases} \tag{10.13}$$

where ω_{idle} is the ICE idle speed, τ_{fd}, $\tau_{gb,e}$ the final drive and gearbox transmission ratios, and $\Lambda \in \{0,1\}$ is a binary value related to the integer variable $m_c(t)$: the clutch can either be engaged ($\Lambda = 1 \Leftrightarrow m_c(t) \in [n_\omega]$) or disengaged ($\Lambda = 0 \Leftrightarrow m_c(t) = 0$), turning off the engine whenever the clutch is open.

Furthermore, the torque request at the ICE is *phase*-dependent and described by

$$\mathscr{T}_e = \begin{cases} \dfrac{\mathscr{T}_t\cdot\Lambda\cdot(1-u)}{\tau_{fd}\cdot\eta_{fd}\cdot\tau_{gb,e}(G)\cdot\eta_{gb,e}(G)} & \text{if } traction, \\ \mathscr{F}_{e,brk}\cdot\Lambda & \text{if } braking, \\ \mathscr{T}_{rech}\cdot\Lambda & \text{if } stand-still, \end{cases} \tag{10.14}$$

where $\mathscr{F}_{e,brk} = \mathscr{F}_{e,brk}(\omega_e)$ is a look-up table implementing the ICE braking torque, η_{fd}, $\eta_{gb,e}$ the constant final drive and gearbox efficiencies, and $\mathscr{T}_{rech}$ the torque provided by the ICE whenever stand-still recharge is allowed. Finally, once the ICE's speed and torque request have been evaluated, the derivation of the fuel mass flow rate follows directly:

$$\dot{m}_f = \begin{cases} \chi(T_w)\cdot\mathscr{F}_{bsfc}\cdot\mathscr{T}_e\cdot\omega_e\cdot\Lambda & \text{if } \omega_e > \omega_{idle}, \\ \chi(T_w)\cdot \text{idle}_{cons}\cdot\Lambda & \text{if } \omega_e \leq \omega_{idle}, \end{cases} \tag{10.15}$$

where $\mathscr{F}_{bsfc} = \mathscr{F}_{bsfc}(\omega_e, \mathscr{T}_e)$ is a look-up table implementing the brake-specific fuel consumption map of the engine, idle_{cons} is the fuel mass flow rate at idle speed, and $\chi(T_w)$ is the ICE temperature-dependent coefficient accounting for the higher fuel consumption at low ICE temperature.

To derive the ICE's cooling water temperature, which is χ-dependent, we first evaluate the thermal power drawn by the refrigerant via

$$P_{loss} = \big(1 - (c_l - c_{l,rpm}\cdot\omega_e)\big)\cdot\dot{m}_f\cdot\mathscr{H}_l, \tag{10.16}$$

with c_l and $c_{l,rpm}$ coefficients to be determined experimentally and $\mathscr{H}_l$ the fuel's lower heating value.

The evaluation of the ICE's cooling water temperature dynamics is now straightforward. However, the maximum temperature that can be reached by the refrigerant is constrained at 90°C (T_m); to this end, we used a regularized Heaviside function to model the cooling process of the refrigerant without introducing state events:

$$\begin{aligned} \dot{T}_w = &\left[\frac{\pi-2}{2\pi}\cdot \text{atan}\Big(\frac{T_w - T_m}{\varepsilon}\Big)\right]\cdot\left[\frac{P_{loss} - G_c(T_w - T_0)}{C}\right], \\ &+\left[\frac{2+\pi}{2\pi}\cdot \text{atan}\Big(\frac{T_w - T_m}{\varepsilon}\Big)\right]\cdot\left[\frac{-G_{c,cool}(T_w - T_0)}{C}\right]. \end{aligned} \tag{10.17}$$

In Eq. (10.17), T_0 is the ambient temperature, ε is the coefficient used to regularize the Heaviside function ($\varepsilon = 1\mathrm{e}^{-3}$), C the thermal capacity of the ICE, and G_c, $G_\mathrm{c,cool}$ the convective heat transfer coefficients times heat-exchange area for heating-up and cooling-down phases, respectively.

Electric drive and battery model

The EM angular speed follows directly from the rear wheel's speed and reads

$$\omega_\mathrm{m} = \frac{v(t)\cdot\tau_\mathrm{fd}\cdot\tau_\mathrm{gb,m}}{r_\mathrm{r}(t)}. \tag{10.18}$$

The torque request at EM is given by this set of equations depending on the active *phase*:

$$\mathcal{T}_\mathrm{EM} = \begin{cases} \dfrac{\mathcal{T}_\mathrm{t}}{\tau_\mathrm{fd}\cdot\eta_\mathrm{fd}\cdot\tau_\mathrm{gb,m}\cdot\eta_\mathrm{gb,m}^{\mathrm{sign}(u)}}\cdot u & \text{if } traction \text{ and } (\Lambda = 1),\\ \dfrac{\mathcal{T}_\mathrm{t}}{\tau_\mathrm{fd}\cdot\eta_\mathrm{fd}\cdot\tau_\mathrm{gb,m}\cdot\eta_\mathrm{gb,m}} & \text{if } traction \text{ and } (\Lambda = 0),\\ \dfrac{\left(\mathcal{T}_\mathrm{t}-\mathcal{T}_\mathrm{e,brk,w}\cdot\Lambda\right)\cdot c_\mathrm{br}}{\tau_\mathrm{gb,m}}\cdot\eta_\mathrm{gb,m} & \text{if } braking,\\ 0 & \text{if } stand-still, \end{cases} \tag{10.19}$$

where $\mathcal{T}_\mathrm{e,brk,w}$ is the ICE braking torque evaluated at the rear wheels, $\tau_\mathrm{gb,m}$, $\eta_\mathrm{gb,m}$ the transmission ratio and efficiency of the EM gear set, and c_br is a constant coefficient required to express the amount of kinetic energy recuperated by the EM (the balance is provided by hydraulic brakes). Besides, the torque provided to EM2 through the ICE reads

$$\mathcal{T}_\mathrm{EM2} = \begin{cases} 0 & \text{if } traction,\\ 0 & \text{if } braking,\\ -\mathcal{T}_\mathrm{rech}\cdot\tau_\mathrm{pulley}\cdot\eta_\mathrm{pulley}\cdot\Lambda & \text{if } stand-still, \end{cases} \tag{10.20}$$

with τ_pulley and η_pulley being the transmission ratio and efficiency of the pulley set coupling EM2 to the ICE. Finally, we define $\mathcal{T}_\mathrm{m}$ as

$$\mathcal{T}_\mathrm{m} = \mathcal{T}_\mathrm{EM} + \mathcal{T}_\mathrm{EM2}, \tag{10.21}$$

due to mutually exclusive operation modes in each *phase*. Thus, the global electrical motor power $P_\mathrm{m,dc}$ can be defined as

$$P_\mathrm{m,dc} = \begin{cases} \dfrac{\mathcal{T}_\mathrm{m}\cdot\omega_\mathrm{e}\cdot\eta_\mathrm{m}(\omega_\mathrm{m},\mathcal{T}_\mathrm{m})}{\tau_\mathrm{pulley}} & \text{if } stand-still,\\ \dfrac{\mathcal{T}_\mathrm{m}\cdot\omega_\mathrm{m}}{\eta_\mathrm{m}^{\mathrm{sign}(\mathcal{T}_\mathrm{m})}(\omega_\mathrm{m},\mathcal{T}_\mathrm{m})} & \text{otherwise,} \end{cases} \tag{10.22}$$

where the speed- and torque- dependent efficiency η_m is provided by means of look-up tables and accounts also for the inverter losses.

The power at the terminals of the battery is given by

$$P_\mathrm{b} = P_\mathrm{m,dc} + P_\mathrm{aux}, \tag{10.23}$$

where P_aux models a phase-dependent auxiliary power flow. The current I_in can be evaluated by solving an equivalent circuit model of the battery, where R_b is the battery's internal resistance and $V_\mathrm{oc}(b_\mathrm{s})$ is the state-of-charge dependent open circuit voltage.
Thus, we obtain:

$$I_\mathrm{in} = \frac{V_\mathrm{oc}(b_\mathrm{s})}{2 \cdot R_\mathrm{b}} - \sqrt{\frac{V_\mathrm{oc}(b_\mathrm{s})^2 - 4 \cdot P_\mathrm{b} \cdot R_\mathrm{b}}{4 \cdot {R_\mathrm{b}}^2}}. \tag{10.24}$$

Finally, the state-of-charge $b_\mathrm{s}(t)$ of the battery is defined as

$$\dot{b}_\mathrm{s} = -\frac{\eta_\mathrm{b} \cdot I_\mathrm{in}}{C_\mathrm{n}}, \tag{10.25}$$

where C_n is the nominal capacity of the battery pack and η_b is its Coulombic efficiency:

$$\eta_\mathrm{b} = \begin{cases} 1 & \text{if } I_\mathrm{in} \geq 0 \\ c_{\eta_\mathrm{b}} & \text{if } I_\mathrm{in} < 0, \end{cases} \tag{10.26}$$

and c_{η_b} is evaluated experimentally.

10.2 Mixed-integer optimal pump speed control of ventricular assist devices

The main functionality of left ventricular assist devices (LVADs) is to provide mechanical circulatory blood support. They have become a well-established and successful therapy for end-stage heart failure patients with an estimated more than 5000 implanted pumps annually worldwide [63, 110]. Modern devices generate a life expectancy similar to the one of patients with a transplanted donor heart so that they are used not only as bridge-to-transplantation but also as destination therapy and in some cases even as bridge-to-recovery [152]. The heart assist devices' role is growing in recent years since there are major improvements in the long-term treatment [152]. Contemporary LVADs implement rotary continuous blood flow and are internally implanted in contrast to pulsatile and extracorporeal pumps, representing the original LVAD design, but which are bigger, less durable, and more invasive than their continuous flow counterpart [240].

These either axial- or centrifugal-flow pumps were originally designed to apply a fixed constant rotary speed. However, there is evidence that this lack of pulsatile flow can cause numerous adverse effects that include gastrointestinal bleeding [64], reduced end-organ function [197], aortic valve thrombosis, and de novo aortic insufficiency [68]. To this end, the latest generation of devices features a pulsatile mode in addition to the constant speed option that oscillates the motor rotation speed periodically for a short time before returning to the constant speed operation. Examples of these devices and modes are HeartMate 3 with the *Pulse mode*, HeartWare HVAD with the *Lavare Cycle*, and EXCOR/INCOR [116]. For further details on the devices, the medical background, therapy planning, and prognosis, we refer to the reviews [108, 158, 43, 132].

Related work. A vast amount of preclinical models for evaluating and testing LVADs via pump speed modulation have been proposed. Amacher et al. [8] reviewed a range of studies [201, 236, 254, 257] where a preselected constant, sine or square wave speed profile is assumed. Chosen parameters were adjusted for amplitude and phase shift to analyzing the effect on relevant physiological quantities. Specifically, high-speed pumping during ventricular contraction, also denoted as copulsative mode, was found to be beneficial in terms of pulsatility in the systemic arterial circulation. Counterpulsative pumping, i.e., low-speed pumping during the ventricular contraction, enhanced left ventricle (LV) unloading [190].

A preselected speed profile does not adjust to dynamic changes in the state of the cardiovascular system. For this reason, control strategies for the blood pumps were developed that take into account different physiological objectives and which were classified in the review of Bozkurt [44]. Physiological control following the Frank-Starling mechanism by pumping preload dependent has been proposed in [12, 229, 83]. Control algorithms that aim for unloading the LV were elaborated in [45, 189]. Speed regulation algorithms for generating sufficient perfusion and detecting ventricular suction [41, 77] or pulmonary oxygen gas exchange tracking [130] are other goals, and, finally, multi-objective variants exist [199].

Due to the increased necessity of LVADs for clinical use, a wide range of different methods from control engineering has been applied, such as adaptive [273, 187], robust [217], model predictive [5], fuzzy logic [57], proportional integral derivative [94], sliding mode [17], and iterative learning control [141]. We refer to [6] for a detailed review.

Contributions. This case study follows an optimal control approach since it offers a flexible framework to include and combine multiple objective and constraint functions. So far, optimal control studies based on cardiovascular system modeling appear to be very limited in the context of ventricular assist devices. [93] investigated the use of LVADs for preload manipulation maneuvers in animal trials. We build on [7], where the continuous pump speed profile is found with an optimal control algorithm based on a lumped cardiovascular system model and compared with both a constant and a sinusoidal-speed profile. In contrast, we do only numerical simulation and no verification with a mock circulation system. Our idea is to consider the cardiovascular dynamics as a system that *switches* between different phases in a single cardiac cycle, e.g., valve opening or closing, in which different dynamics apply. We use solving techniques tailored for *switched systems* to reduce the underlying system nonlinearities and leverage the computations. Within this framework, we present a novel algorithm to calculate optimal piecewise constant (pwc) pump speed modulation following the above-mentioned pulsatility modes for modern devices and concerning ventricular unloading and opening of the aortic valve. For comparison, we compute the optimal continuous and constant speed profiles. Furthermore, we consider adapting model parameters to patient-specific data with a nonlinear regression objective function to deal with a personalized model.

Another fundamental difference between our approach to [7] and all other model-based approaches lies in the used model. Instead of applying a time-varying elastance function to represent the pressure-volume relationship in heart chambers, we base our model on the contribution of the longitudinal atrioventricular plane displacement (AVPD) to ventricular pumping, which is novel in the LVAD context. It has been established that the atrioventricular plane (AVP) behaves like a piston unit by moving back and forth in the base-apex direction, creating reciprocal volume changes between atria and ventricles [173]. Also, there is strong evidence that the magnitude of AVPD is a reliable index for heart failure diagnosis [269]. Since elastance functions cannot explain the behavior of ventricular walls and fail to simulate the interaction between the LV and an assist device [62, 258], we reuse and extend an AVPD model introduced in [175] and altered to the switched systems setting in [131]. Alternatives for replacing the elastance model are myofiber, or sarcomere mechanics approaches [143] as in the CircAdapt model [13, 172], though a great number of discontinuities and nonlinear equations limit their applicability to (gradient-based) optimization and control techniques. The presented approach is clinically applicable since the AVP motion is relatively easy to measure via noninvasive echocardiography.

Outline. The outline of this section is the following: We describe the cardiovascular and LVAD system model in Section 10.2.1 before we define constraints in Section 10.2.2, as well as the clinical data and model personalization in Section 10.2.3. Afterward, we formulate the OCP with integer restrictions in Section 10.2.4, and we define an algorithmic approach to solve it in Section 10.2.5. We present the simulation results in Section 10.2.6 and discuss the realistic and algorithmic setting with limitations in Section 10.2.7. We wrap up the case study with conclusions in Section 10.2.8. Supplemental material such as detailed model and optimization parameter values are provided in Section 10.2.9.

10.2.1 Cardiovascular system and LVAD modeling

This case study uses a lumped model of the cardiovascular system based on the left heart's representation. We combine the AVPD model as proposed and validated in [131] with an axial pump LVAD model that has been validated in [238]. The proposed model consists of nine ODEs for the pressure $P(\cdot)$ of left atrium (LA), LV, aorta (A), systemic artery (S), and venous system (V), the flow $Q(\cdot)$ in the aorta and in the LVAD as well as the velocity $v(\cdot)$ of the AVPD and its position $s(\cdot)$, where we are going to replace $(\cdot)$ with (t) in order to denote the time dependency. The cardiovascular system can be steered with the continuous control $u(\cdot)$ representing the rotary pump speed. The ODE system reads for $t \in [t_0, t_f] \subset \mathbb{R}$:

$$\begin{aligned}
\dot{P}_{\mathrm{LA}}(t) &= \frac{P_{\mathrm{V}}(t) - P_{\mathrm{LA}}(t)}{C_{\mathrm{LA}} R_V} - \frac{Q_{\mathrm{MV}}(t) - A_{\mathrm{LA}}\, v(t)}{C_{\mathrm{LA}}}, && \text{(10.27a)}\\
\dot{P}_{\mathrm{LV}}(t) &= \frac{(1 + k_{\mathrm{RAD}}) A_{\mathrm{LV}}\, v(t)}{C_{\mathrm{LV}}} + \frac{Q_{\mathrm{MV}}(t) - Q_{\mathrm{AoV}}(t) - Q_{\mathrm{LVAD}}(t)}{C_{\mathrm{LV}}}, && \text{(10.27b)}\\
\dot{P}_{\mathrm{A}}(t) &= \frac{Q_{\mathrm{AoV}}(t) + Q_{\mathrm{LVAD}}(t) - Q_A(t)}{C_A}, && \text{(10.27c)}\\
\dot{P}_{\mathrm{S}}(t) &= \frac{P_V(t) - P_S(t)}{C_S R_S} + \frac{Q_A(t)}{C_S}, && \text{(10.27d)}\\
\dot{P}_{\mathrm{V}}(t) &= \frac{P_S(t) - P_V(t)}{C_V R_S} + \frac{P_{\mathrm{LA}}(t) - P_{\mathrm{V}}(t)}{C_V R_V}, && \text{(10.27e)}\\
\dot{Q}_{\mathrm{A}}(t) &= \frac{P_A(t) - P_S(t) - R_C Q_A(t)}{L_S}, && \text{(10.27f)}\\
\dot{Q}_{\mathrm{LVAD}}(t) &= \frac{P_{\mathrm{LV}}(t) - P_A(t) - R_{\mathrm{LVAD}} Q_{\mathrm{LVAD}}(t) - \beta u(t)^2}{L_{\mathrm{LVAD}}}, && \text{(10.27g)}\\
\dot{v}(t) &= \frac{-R_{\mathrm{AVP}}\, v(t) - A_{\mathrm{LV}} P_{\mathrm{LV}}(t) + A_{\mathrm{LA}} P_{\mathrm{LA}}(t) + F_C(t)}{L_{\mathrm{AVP}}}, && \text{(10.27h)}\\
\dot{s}(t) &= v(t), && \text{(10.27i)}
\end{aligned}$$

where the default parameter values for the compliances C, resistances R, and inertances L are given in the Supplemental Material 10.2.9. The model uses the valve flows[2] defined by

$$Q_{\mathrm{MV}}(t) := \begin{cases} \frac{P_{\mathrm{LA}}(t) - P_{\mathrm{LV}}(t)}{R_M}, & \text{if } P_{\mathrm{LA}}(t) > P_{\mathrm{LV}}(t), \\ 0, & \text{else.} \end{cases} \tag{10.28}$$

$$Q_{\mathrm{AoV}}(t) := \begin{cases} \frac{P_{\mathrm{LV}}(t) - P_{\mathrm{A}}(t)}{R_{\mathrm{AoV}}}, & \text{if } P_{\mathrm{LV}}(t) > P_{\mathrm{A}}(t), \\ 0, & \text{else.} \end{cases} \tag{10.29}$$

The AVP contraction force is assumed to be a pwc function in the following sense

$$F_C(t) := \begin{cases} F_{AC}, & \text{during atrial contraction,} \\ F_{VC}, & \text{during ventricular contraction,} \\ 0, & \text{else.} \end{cases}$$

[2] We neglect valve regurgitation and set the back flow to zero. We discuss this assumption in Section 10.2.7.

We specify in Section 10.2.5 how these contraction phases are mathematically defined and skip their formal introduction here.

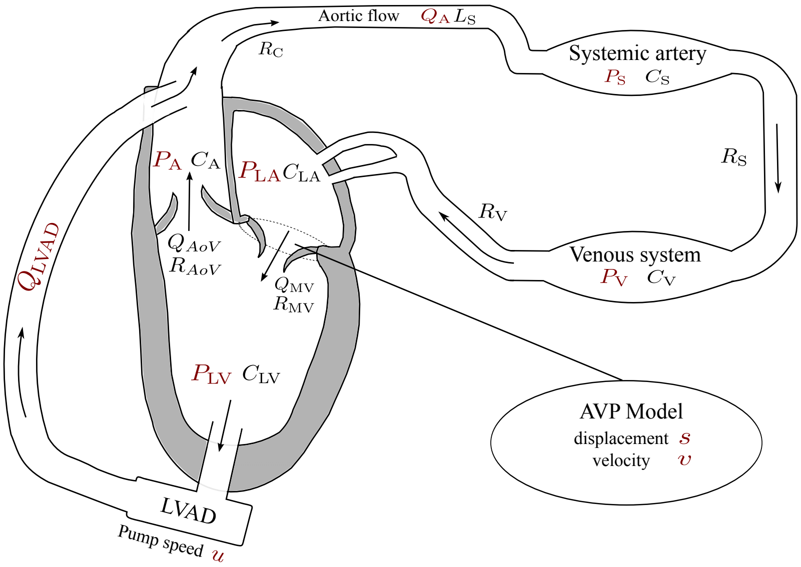


Figure 10.8: Illustration of the simplified model of the left heart, the circulatory system, and the LVAD. Differential states and the pump speed control $u(\cdot)$ are depicted in red. The cyclic flow is indicated by the arrows. The model consists of five compartments for the left atrium (LA), left ventricle (LV), aorta (A), systemic artery (S), and venous system (V), represented with the pressure functions $P(\cdot)$. These variables interact with the flows $Q(\cdot)$ in the LVAD and the aorta, while the atrioventricular interaction is modeled by the velocity $v(\cdot)$ and position $s(\cdot)$ of the atrioventricular plane displacement (AVPD). Compliance, resistance, and inertance parameters C, R, L are depicted next to the corresponding compartment.

Figure 10.8 gives a schematic overview of the lumped model of the heart and the circulatory system. In the following, we group the differential states into the vector

$$\boldsymbol{x} = [P_\mathrm{LA}, P_\mathrm{LV}, P_\mathrm{A}, P_\mathrm{S}, P_\mathrm{V}, Q_\mathrm{A}, Q_\mathrm{LVAD}, v, s]^\top$$

and write the dynamical system (10.27a)-(10.27i) as

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), u(t)), \quad \text{for } t \in [t_0, t_f]. \tag{10.30}$$

Figure 10.9 illustrates the AVPD model, where the AVP refers to the separating tissue between LV and LA that surrounds the mitral valve. During atrial contraction, the force F_C pulls the AVP towards the base and redistributes blood from the LA to the LV via the mitral valve. When it reaches the switching threshold $-S_D$, the contraction force F_C starts to work in the opposite direction, representing ventricular contraction. In this way, the AVPD leverages longitudinal pumping that results in the ejection of blood to the aorta. The ventricular contraction stops as soon as the AVP reaches the threshold S_D. A relaxation phase follows where F_C equals zero, and the AVP moves slowly to its original position. This longitudinal pumping is well described by a piston unit concept, where the piston is placed between LA and LV with constant cross-sections

A_{LA} and A_{LV}, respectively. We illustrate this piston representation at the bottom of Figure 10.9. The AVP model assumes that the radial squeezing of LV walls supports longitudinal pumping.

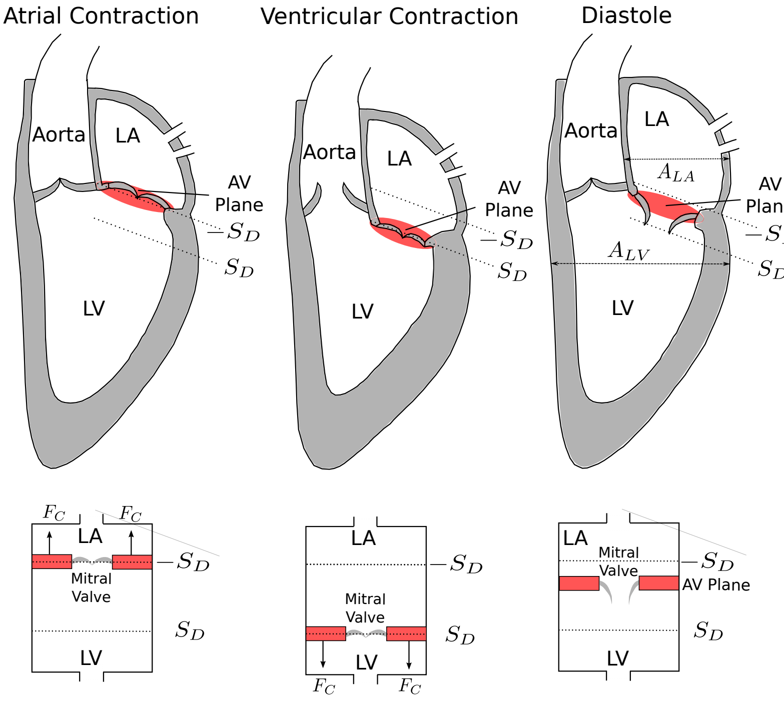


Figure 10.9: Illustration of the atrioventricular plane displacement concept for the left heart. The atrioventricular plane moves forth and back between $-S_D$ and S_D, pulled by the contraction force F_C resulting in blood redistribution from LA to the LV to the aorta. This behavior resembles a piston pump, as depicted at the bottom, where $-S_D$ and S_D mark the longitudinal displacement into basal and apical direction, respectively.

10.2.2 Physiological assumptions and constraints

This case study makes a series of assumptions, which are explained here. We list the applied constraint parameter values in Supplemental Material 10.2.9.

Dilated left heart failure

The proposed model is adapted in order to represent a typical LVAD patient candidate's heart situation [108]. This includes modeling left-sided heart failure with decreased cardiac output and dilated cardiomyopathy[3] with enlarged LA and LV. To this end, we modify certain model parameters, including increased compliance and increased cross-sectional area of LV as described in more detail in Supplemental Material 10.2.9. Besides, other parameters can be adapted to a specific patient, as explained in Section 10.2.3.

[3] Alternatively, myocardial infarction is a common case related to LVAD patients; however, it is challenging to represent scars adequately with a zero-dimensional lumped model.

Steady-state situation

We assume the cardiovascular and circulatory system is in *steady-state*, in the sense of

- there are no rapid or major changes of cardiac output and the heart cycle length,
- the system has already adapted to the LVAD implementation,
- the patient is at rest.

These assumptions justify neglecting the autoregulatory mechanisms of cardiovascular pumping, such as the systemic baroreflex feedback process and beat to beat myocardium wall strain adaptation based upon the Frank-Starling effect.

Feasible instantaneous pump speed changes

In practice, due to blood inertia, it is impossible to adjust the pump speed arbitrarily. Here we neglect blood and rotor inertia effects and assume that the pump speed can be varied without restrictions. Connected to this, blood is considered as a Newtonian fluid, and no blood rheology changes are taken into account.

Blood inflow equals outflow

In conjunction with the steady-state assumption, we require that the amount of accumulated incoming blood in the LV is equal to the accumulated amount of blood ejected out of the LV over the time horizon $[t_0, t_f]$. For this purpose, we introduce the tolerance parameter $\epsilon_{\text{flow}} > 0$ and define (with $Q_{\text{MV}}(t), Q_{\text{AoV}}(t) \geq 0$) the constraint:

$$\left| \int_{t_0}^{t_f} [Q_{\text{MV}}(t) - Q_{\text{AoV}}(t) - Q_{\text{LVAD}}(t)] \, \mathrm{d}t \right| \leq \epsilon_{\text{flow}}. \tag{10.31}$$

Periodicity of the heart cycle

The steady-state assumption implies that it is sufficient to consider only one heart cycle since there are no significant differences between several heart cycles. Thus, in this study, we fix the time horizon to the length of one heart cycle. In this way, the steady-state condition translates into a periodicity constraint denoting that the differential state and control values at the beginning of the heart cycle should be equal to the ones at the end of the cycle. In mathematical terms, this results with $\epsilon_{\text{per}} > 0$ in

$$\left| \boldsymbol{x}_i(t_f) - \boldsymbol{x}_i(t_0) \right| \leq \epsilon_{\text{per}}, \quad \text{for } i = 1, \ldots, 9. \tag{10.32}$$

Partial LVAD support

When using an LVAD in the clinical setting, a distinction is made between full and partial support. While the LV does not contribute to blood ejection with full support, the aortic valve still opens with partial support because the LV contraction force is still strong enough to pump *par-*

tially. We assume partial support, that is:

$$\int_{t_0}^{t_f} Q_{\text{AoV}}(t)\,\mathrm{d}t \geq \epsilon_{\text{partial}}, \tag{10.33}$$

with $\epsilon_{\text{partial}} > 0$.

Backflow of blood from the aorta in the LV

We want to restrict the backflow from the aorta in the LV via the LVAD. For this, we introduce the tolerance $\epsilon_{\text{back}} > 0$ and require

$$Q_{\text{LVAD}}(t) \geq -\epsilon_{\text{back}}, \qquad \text{for } t \in [t_0, t_f]. \tag{10.34}$$

Frank-Starling like control

One objective of using an LVAD is to provide sufficient perfusion to the patient's body. The Frank-Starling mechanism regulates the cardiac output physiologically, i.e., the amount of blood ejected by the LV into the aorta per minute, according to the current need. We consider this mechanism as a desirable mode of operation and seek a pump speed control policy that results in an actual cardiac output that equals approximately a desired cardiac output $V_{\text{CO}} \in \mathbb{R}_+$:

$$\left| \frac{60}{t_f - t_0} \int_{t_0}^{t_f} [Q_{\text{AoV}}(t) + Q_{\text{LVAD}}(t)]\,\mathrm{d}t - V_{\text{CO}} \right| \leq \epsilon_{\text{CO}}, \tag{10.35}$$

where $\epsilon_{\text{CO}} > 0$. We note that the actual cardiac output should not exceed the desired cardiac output up to the tolerance, since this could result in fatigue for the patient.

Variable bounds and suction prevention

We require the differential state variables and the pump speed control to be in realistic ranges. Let $\boldsymbol{x}_{\text{lb}}, \boldsymbol{x}_{\text{ub}} \in \mathbb{R}^9$, respectively $u_{\text{lb}}, u_{\text{ub}} \in \mathbb{R}$ denote appropriate lower and upper bounds. The box constraints read

$$\boldsymbol{x}_{\text{lb}} \leq \boldsymbol{x}(t) \leq \boldsymbol{x}_{\text{ub}}, \quad u_{\text{lb}} \leq u(t) \leq u_{\text{ub}}, \quad \text{for } t \in [t_0, t_f]. \tag{10.36}$$

In this way, we are able to prevent the occurrence of suction, which describes the situation of excessive pumping that may cause a collapse of the ventricle if $P_{\text{LV}}(t)$ is very low.

10.2.3 Clinical data and model personalization

This case study uses data that were obtained retrospectively from the University Hospital Magdeburg, Department of Cardiology [232]. An exemplary subject was selected who involved a dilated LV and suffered from systolic left-sided heart failure. Data collection was performed via conductance catheterization for pressure measurements and via echocardiography for other data. The subject showed in rest a heart frequency of 67 beats per minute with a cardiac output of about 3.5 liters per minute. Further hemodynamic characteristics of the selected subject are shown in Table 10.2.

Parameter	End-systolic	End-diastolic
LV volume	281 ml	228 ml
LV pressure	120 mmHg	5 mmHg
PCW pressure	28 mmHg	14 mmHg
Aortic pressure	121 mmHg	53 mmHg

Table 10.2: Measured hemodynamic data for the example subject. Pulmonary capillary wedge (PCW) pressure represents a surrogate for LA pressure.

We selected a representative cardiac cycle with the duration $h_{\text{cycle}} = 0.89$ seconds and 27 measured data points. We propose to personalize the model via a parameter estimation (PE) method. For this, we formulate an optimization problem with the model equations as constraints and a nonlinear regression term as an objective that minimizes the difference of model response values to the measured subject data. Here, we minimize the difference between measured LV pressure for selected time points and their corresponding model output values; however, this approach can also be applied to a general measured data set with more differential state types involved. We denote with $\widehat{P}_{\text{LV}}(t_i)$ the measured LV pressure at time point $t_i \in [t_0, t_f]$. We choose the parameters to be estimated as proposed in [130] with high sensitivity with respect to the LV pressure. These parameters are

$$\boldsymbol{p} = [R_{\text{AVP}}, C_{\text{LV}}, L_{\text{AVP}}, F_{\text{VC}}, F_{\text{AC}}, A_{\text{LV}}, A_{\text{LA}}, k_{\text{RAD}}, S_D]^\top.$$

We bound the parameters to be in a realistic range, i.e., $\boldsymbol{p}_{\text{lb}} \leq \boldsymbol{p} \leq \boldsymbol{p}_{\text{ub}}$, see Supplemental Material 10.2.9 for further details. The selected subject had not (yet) implanted an LVAD, so we set $Q_{\text{LVAD}}(t)$ to zero and neglect the control $u(t)$ and constraints on $Q_{\text{LVAD}}(t)$ for the PE. The parameter (point) estimation problem is defined as the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{p}} \quad & \tfrac{1}{2} \sum_{i=1}^{n_m} \left(\widehat{P}_{\text{LV}}(t_i) - P_{\text{LV}}(t_i) \right)^2 + \varphi(\boldsymbol{p}) \\ \text{s.t.} \quad & \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{p}), \quad \text{for } t \in [t_0, t_f], \\ & \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \\ & \text{constraints (10.32), (10.33), (10.36),} \end{aligned}$$

where $n_m = 27$ denotes the number of available measurements and $\boldsymbol{x}_0$ the initial values.The term $\varphi(\boldsymbol{p})$ allows incorporating a priori information of the parameters, which we here set to zero[4].

10.2.4 Optimal control problem formulation

Based on a personalized model, we are interested in an advantageous application of the LVAD for a (possible) patient. An OCP offers the framework to include generic constraints and objective functions. While we have already defined the constraints in Section 10.2.2, for the objective we reuse the multiobjective function from [8]. This objective constitutes a compromise function that aims for ventricular unloading and ensures the aortic valve's opening. Permanent

[4] Future work should consider a priori information in the form of $\varphi(\boldsymbol{p}) = \epsilon \|\boldsymbol{p} - \bar{\boldsymbol{p}}\|^2$ instead of imposing lower and upper bounds on $\boldsymbol{p}$.

closure of the aortic valve may lead to fusion of the aortic valvular cusps and a resulting thrombus formation [158]. By ventricular unloading, we refer to reducing the hydraulic work that the LV has to perform in order to provide sufficient perfusion. Let $\varrho_1 \in [0,1]$ denote a weighting parameter that facilitates to put one objective more into focus, and let ϱ_2 and ϱ_3 denote unit scaling factors, see Supplemental Material 10.2.9 for more details. Then, we introduce the objective as

$$\mathscr{C}(\boldsymbol{x}(\cdot)) := \int_{t_0}^{t_f} [\varrho_1\varrho_2 P_{\mathrm{LV}}(t)(Q_{\mathrm{AoV}}(t) + Q_{\mathrm{LVAD}}(t) - Q_{\mathrm{MV}}(t)) - (1-\varrho_1)\varrho_3 Q_{\mathrm{AoV}}(t)] \; \mathrm{d}t.$$

The first term accounts for the ventricular unloading, while the second term causes aortic valve opening via maximizing the flow through this valve. We consider the following optimization problem, where we minimize the above objective over the differential states $\boldsymbol{x}$ and the continuous control u:

$$\begin{aligned} \min_{\boldsymbol{x},u} \quad & \mathscr{C}(\boldsymbol{x}(\cdot)) \\ \text{s.t.} \quad & \text{model equations (10.30)}, \\ & \text{inflow equals outflow (10.31)}, \\ & \text{periodicity of heart cycle (10.32)}, \\ & \text{partial LVAD support (10.33)}, \\ & \text{restricted LVAD back flow (10.34)}, \\ & \text{sufficient perfusion (10.35)}, \\ & \text{variable bounds (10.36)}. \end{aligned}$$

For this optimization problem, we investigate three different scenarios regarding the pump speed control.

1. *Constant speed:* This represents the usual clinical setting and is expressed by $u(t) := u_{\mathrm{con}} \in [u_{\mathrm{lb}}, u_{\mathrm{ub}}]$ for $t \in [t_0, t_f]$.

2. *Continuous speed:* There are no restrictions on $u(\cdot)$ apart from lower and upper bounds.

3. *Pwc speed:* In order to create pulsatility, this scenario considers switching between different constant speed modes. For this, we use the indicator function notation $\chi_{[t_1,t_2]}(t)$ from Definition A.1 and we assume $u(\cdot)$ to be a step function with three different levels $u_1, u_2, u_3 \in [u_{\mathrm{lb}}, u_{\mathrm{ub}}]$:

$$u(t) := u_1\chi_{[t_0,t_1)}(t) + u_2\chi_{[t_1,t_2)}(t) + u_3\chi_{[t_2,t_3)}(t) + u_1\chi_{[t_3,t_f]}(t),$$

where t_1, t_2, t_3 are switching times to be determined[5]. We require MU times for the different speed levels because rapid changes are not feasible in a realistic setting. Let

[5] In fact, we can drop $\chi_{[t_3,t_f]}(t)$ since $\boldsymbol{x}(0)$ is free. However, the next section's algorithmic idea exploits a fixed sequence of active system phases so that we keep this term.

$C_{U_1}, C_{U_2}, C_{U_3} > 0$ denote these MU time parameters, and we introduce the constraints

$$t_1 - t_0 + t_f - t_3 \geq C_{U_1}, \quad t_2 - t_1 \geq C_{U_2}, \quad t_3 - t_2 \geq C_{U_3}.$$

In fact, the pwc scenario can be interpreted as MIOCP since u shall take values in the discrete set $\{u_1, u_2, u_3\}$. As this section is application-driven, we skip for brevity the thorough mixed-integer formulation with a discrete control v.

10.2.5 Algorithmic approach

We deal with both *explicit* and *implicit* switches, as introduced in Definition 3.4, that result in discontinuous variables for the PE and the OCP. On the one hand, the pump speed control in the pwc scenario involves explicit switches. On the other hand, the valve flows, and the contraction force induce implicit switches. While the valve switches are defined in (10.28)-(10.29), we specify the contraction force switches in the following.

Implicit switches through the contraction force

Because we consider only one heart cycle, the atrial and ventricular contraction takes place once. We assume a physiological order, that is atrial before ventricular contraction followed by a relaxation phase. Initially, let $-S_D < s(t_0) < S_D$. We further assume the following switching times exist:

$$t_{\mathrm{VC}} := \underset{t\in(t_0,t_f)}{\operatorname{argmin}}\{s(t) = -S_D\}, \qquad t_{\mathrm{R}} := \underset{t\in(t_{\mathrm{VC}},t_f)}{\operatorname{argmin}}\{s(t) = S_D\}.$$

Then, the contraction force is defined as

$$F_C(t) := \begin{cases} F_{\mathrm{AC}}, & \text{for } \; t_0 \leq t \leq t_{\mathrm{VC}}, \\ F_{\mathrm{VC}}, & \text{for } \; t_{\mathrm{VC}} < t \leq t_{\mathrm{R}}, \\ 0, & \text{for } \; t_{\mathrm{R}} < t \leq t_f. \end{cases}$$

Dividing the cardiac cycle into subphases

The periodic switching nature of the cardiac cycle model makes the solving process challenging. We need to identify when switching happens and what the successive active subsystems of $\boldsymbol{f}$ are. If we combine all possible valve positions and contraction force settings, we get 12 different subsystems. To reduce complexity, we assume a specific sequence of active subsystems for the cardiac cycle taking advantage of physiological relationships in the human heart. Thus, we divide the heart cycle into seven phases, similar to [130]. Table 10.3 and Figure 10.10 explain the phases of the ordered sequence, where the switching times are denoted with τ_i, $i = 1, \ldots, 6$.

The modes from Table 10.3 translate into the following constraints for the optimization prob-

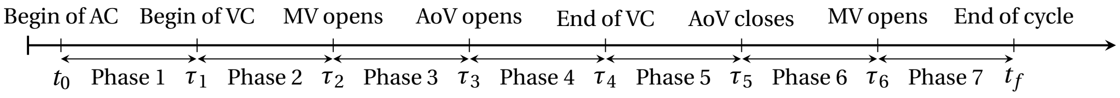


Figure 10.10: Time course of the assumed active phase sequence with switching events. The switching times τ_i are variables in the optimization problem. Atrial contraction (AC) starts before ventricular contraction (VC). Further switching events are associated with aortic valve (AoV) and mitral valve (MV) opening/closing.

Phase	F_C mode	Mitral valve	Aortic valve
1	AC	open	closed
2	VC	open	closed
3	VC	closed	closed
4	VC	closed	open
5	0	closed	open
6	0	closed	closed
7	0	open	closed

Table 10.3: Assumed sequence of active phases. For instance, in the first phase the LA contracts, the mitral valve is open and the aortic valve is closed.

lem and for $\boldsymbol{f}$:

$$\begin{array}{rlll}
\text{'AC':} & F_C(t) = F_{\text{AC}}, & \text{and} & s(t) > -S_D,\\
\text{'VC':} & F_C(t) = F_{\text{VC}}, & \text{and} & s(t) < S_D,\\
\text{'0':} & F_C(t) = 0, & & \\
\text{'MV open':} & Q_{\text{MV}}(t) = \frac{P_{\text{LA}}(t)-P_{\text{LV}}(t)}{R_M}, & \text{and} & P_{\text{LA}}(t) > P_{\text{LV}}(t),\\
\text{'MV closed':} & Q_{\text{MV}}(t) = 0, & \text{and} & P_{\text{LA}}(t) \leq P_{\text{LV}}(t),\\
\text{'AoV open':} & Q_{\text{AoV}}(t) = \frac{P_{\text{LV}}(t)-P_{\text{A}}(t)}{R_{\text{AoV}}}, & \text{and} & P_{\text{LV}}(t) > P_{\text{A}}(t),\\
\text{'AoV closed':} & Q_{\text{AoV}}(t) = 0, & \text{and} & P_{\text{LV}}(t) \leq P_{\text{A}}(t).
\end{array}$$

By fixing the sequence of active subsystems, the PE and OCP transform into multiphase problems [185], where only the switching times need to be determined. The difference to the multiphase setting of the HEV problem presented in Section 10.1 is that the time periods of the phases in the latter problem are fixed. Here, one phase ends with a switching time that depends on the above conditions, and that needs to be determined.

Switching time optimization

Considering that the OCP at hand involves both explicit and implicit switches, we note that the proposed CIA decomposition algorithms can not be directly applied. Although there are modifications of the CIA decomposition to include implicit switches [182], we suggest here applying switching time optimization [89, 185] and exploiting the reduced complexity with the derived order of phases. Thus, we come back to the methods described in Section 3.4.2 to determine the switching times $\tau_1, \ldots, \tau_6$ so that we can transform the initially discrete optimization problems into continuous ones. Using the variable time transformation, we reformulate the multiphase model equation:

$$\dot{\boldsymbol{x}}(\tau) = \begin{cases} (\tau_1 - t_0)\boldsymbol{f}_1(\tau,\cdot), & \text{if } \tau \in [0,1), \\ (\tau_i - \tau_{i-1})\boldsymbol{f}_i(\tau,\cdot), & \text{if } \tau \in [i-1,i), \text{ for } i = 2,\dots,6, \\ (t_f - \tau_6)\boldsymbol{f}_7(\tau,\cdot), & \text{if } \tau \in [6,7], \end{cases}$$

where $\boldsymbol{f}_i$ denotes the model equation for the ith phase. We notice that the phase durations enter the equation via $(\tau_i - \tau_{i-1})$ as continuous variables. At the end of each phase, the switching constraints for the contraction force and the valve flows as described in Section 10.2.5 need to be fulfilled at the transformed switching time points up to a tolerance $\epsilon_{\text{sw}} > 0$:

$$\begin{aligned} &|s(1) + S_D| \leq \epsilon_{\text{sw}}, && |P_{\text{LA}}(2) - P_{\text{LV}}(2)| \leq \epsilon_{\text{sw}}, && |P_{\text{LV}}(3) - P_{\text{A}}(3)| \leq \epsilon_{\text{sw}}, \\ &|s(4) - S_D| \leq \epsilon_{\text{sw}}, && |P_{\text{LV}}(5) - P_{\text{A}}(5)| \leq \epsilon_{\text{sw}}, && |P_{\text{LA}}(6) - P_{\text{LV}}(6)| \leq \epsilon_{\text{sw}}. \end{aligned}$$

This switching time reformulation provides the framework to solve the PE problem and the OCP with constant and continuous pump speed as an optimization problem with solely continuous variables. For the pwc pump speed modulation, we also need to find the switching times t_1, t_2, t_3 between the three different speed levels as introduced in Section 10.2.4. Here, we assume

$$\tau_2 < t_1 < \tau_3, \qquad \tau_5 < t_2 < \tau_6, \qquad \tau_6 < t_3 < t_f,$$

i.e., the first speed change occurs between mitral valve closing and aortic valve opening, the second between aortic valve closing and mitral valve opening, and the third between mitral valve opening and the end of the heart cycle. Thus, we divide the third, sixth, and seventh phases from Section 10.2.5 into two phases each such that in total nine switching times for ten phases need to be determined.

Numerical solution of optimization problems

We use direct collocation [253] to transform the continuous-time optimization problems via temporal discretization into NLPs. We apply an equidistant discretization grid with a grid length of 1 ms, and we let the control values change their values only on the grid points. The differential state trajectories are approximated with Radau collocation polynomials [31] of degree 3. We implemented the optimization problems in `Python` v3.7.5 and used `CasADi` v3.4.5 [9] to parse the resulting NLP to the solver `IPOPT` v3.12.3 [264]. For the PE problem, we applied the Gauss-Newton method [31], so that the calculation of Hessians is not required.

The model phases' lengths were extracted from pressure time series and other continuous data and used for initialization of the switching times τ_i. These phase durations were fixed for the PE problem and set variable for the OCP. We further initialized the PE problem with variable values based on a simulation with default parameter values, see Supplemental Material 10.2.9. The OCP was initialized with simulated variable values obtained with estimated parameters and constant pump speed equal to 8000 rpm.

10.2.6 Numerical results

All computational experiments were executed on a Ubuntu 18.04 workstation with 4 Intel i5-4210U CPUs (1.7 GHz) and 7.7 GB RAM.

Patient specification

Solving the PE problem from Section 10.2.3 resulted in the values

$$\boldsymbol{p}^* = [324.2, 0.6, 20.4, 4709, 900, 42, 25, 1.35, 0.5]^\top.$$

The result of the switching distance parameter, i.e., $S_D = 0.5$, is equivalent to an AVPD of only 10 mm and thus indicates a reduced ventricular function. The situation of heart failure is reflected well by the estimated parameter values. Particularly, the LV compliance is increased, the amplitude of the contraction forces F_{AC} and F_{VC} is decreased, and the parameter k_{RAD} accounting for the relative contribution of radial pumping is increased. The left plot in Figure 10.11 shows the measured data points $\widehat{P}_{\mathrm{LV}}$, together with the obtained P_{LV} from the PE solution.

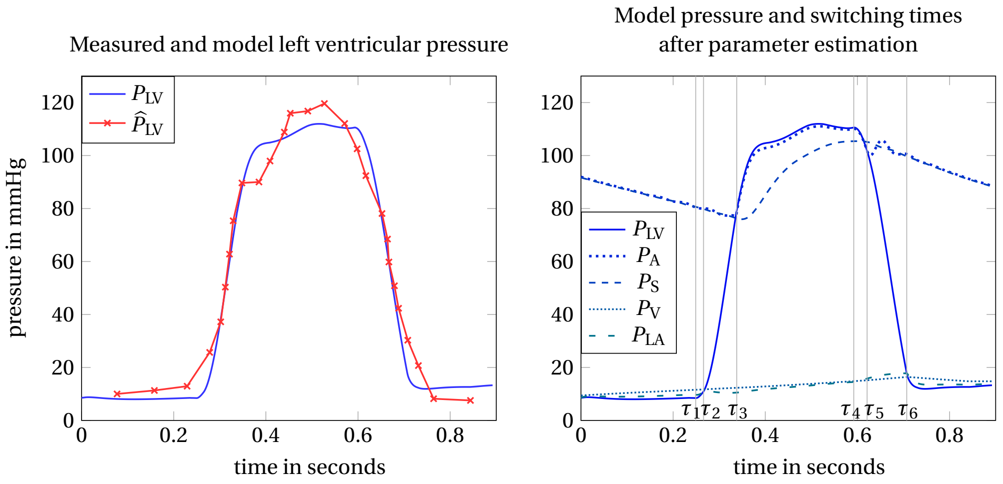


Figure 10.11: Left: Measured $\widehat{P}_{\mathrm{LV}}$ values and resulting P_{LV} trajectory based on the parameters obtained from solving the PE problem from Section 10.2.3. Right: Simulated pressure trajectories based on the parameters obtained from solving the PE problem from Section 10.2.3. The switching times τ_i are depicted with the vertical gray lines.

The model response P_{LV} reflects the measured data points, especially for the duration of ventricular contraction, while its peak is slightly underestimated. The transcripted nonlinear regression problem was solved by `IPOPT` after 230 seconds and with an objective value of 512 mmHg^2, which is equivalent to a root-mean-square deviation of 6.16 mmHg. We observed numerical instabilities when solving the PE problem. The convergence of the algorithm seems to depend heavily on the initial solution, which stresses the importance of the proposed initialization from Section 10.2.5. In addition, we have used mild termination criteria for `IPOPT` and chose a big tolerance value for the periodicity constraint (10.32). The right plot in Figure 10.11 depicts all model pressure trajectories based on the PE and the six switching times between the different model phases. We observe that the aortic pressure P_A resembles the LV pressure P_{LV} during ventricular systole and the systemic pressure P_S else, apart from some small oscillations. Likewise, P_{LA} and P_V represent similarly high pressures, although P_{LA} adopts to P_{LV} depending on the mitral valve opening.

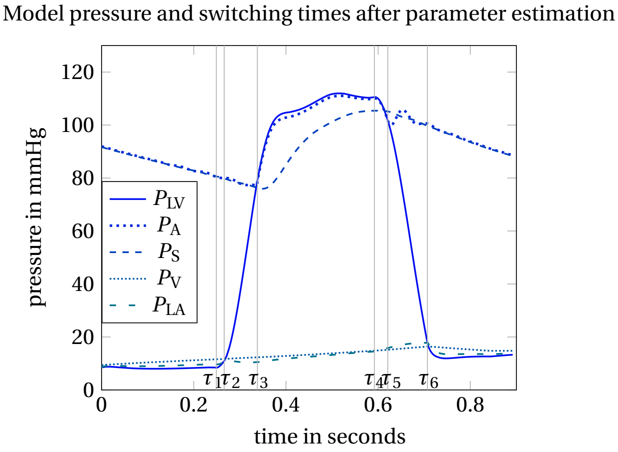


Figure 10.12: Simulated pressure functions based on the parameters obtained from solving the PE problem from Section 10.2.3. The switching times τ_i are depicted with the vertical grey lines.

Pump control policies

We solved the OCP according to the proposed algorithm from Section 10.2.5. The applied parameters for the objective and constraints such as the tolerance parameters $\epsilon_{\cdot}$, the dwell times $C_{U,\cdot}$, and lower and upper bounds on variables are listed in Section 10.2.9. Figure 10.13 shows the pressure functions for the three pump speed scenarios. The outcomes of the continuous and pwc scenario are very similar. They show an elevated peak of P_{LV} compared to the parameter estimated solution from Figure 10.11. While the rise and fall of the pressure profiles before and after the ventricular contraction is significantly steeper than with the parameter estimated solution, its duration, i.e., $\tau_4 - \tau_1$, is in a similar range due to an enforced MDT of 0.2 seconds for the ventricular contraction. Very low values occur for P_{LV} directly before τ_1, however, they are still above the threshold for suction. We notice that the cycle duration for the pwc scenario is 0.88 seconds and slightly longer than for the other two scenarios with a duration of about 0.84 seconds, as we allow a slight deviation of 50 ms from the standard cycle length. The constant speed scenario also involves an increased aortic and ventricular pressure compared to the parameter estimated solution, though their peaks are significantly lower compared with the continuous and pwc speed scenarios.

Figure 10.14 illustrates the different optimal pump speed profiles and the according results for the flows, AVPD speed, and AVPD position. We observe that the continuous speed profile provides counterpulsative pump support and the pwc speed profile approximates this profile. Due to this similar pump speed behavior, the optimal differential state trajectories resemble each other. The main portion of the flows through the LVAD and the aorta appears for constant pump speed during ventricular contraction. In contrast, with continuous or pwc speed,

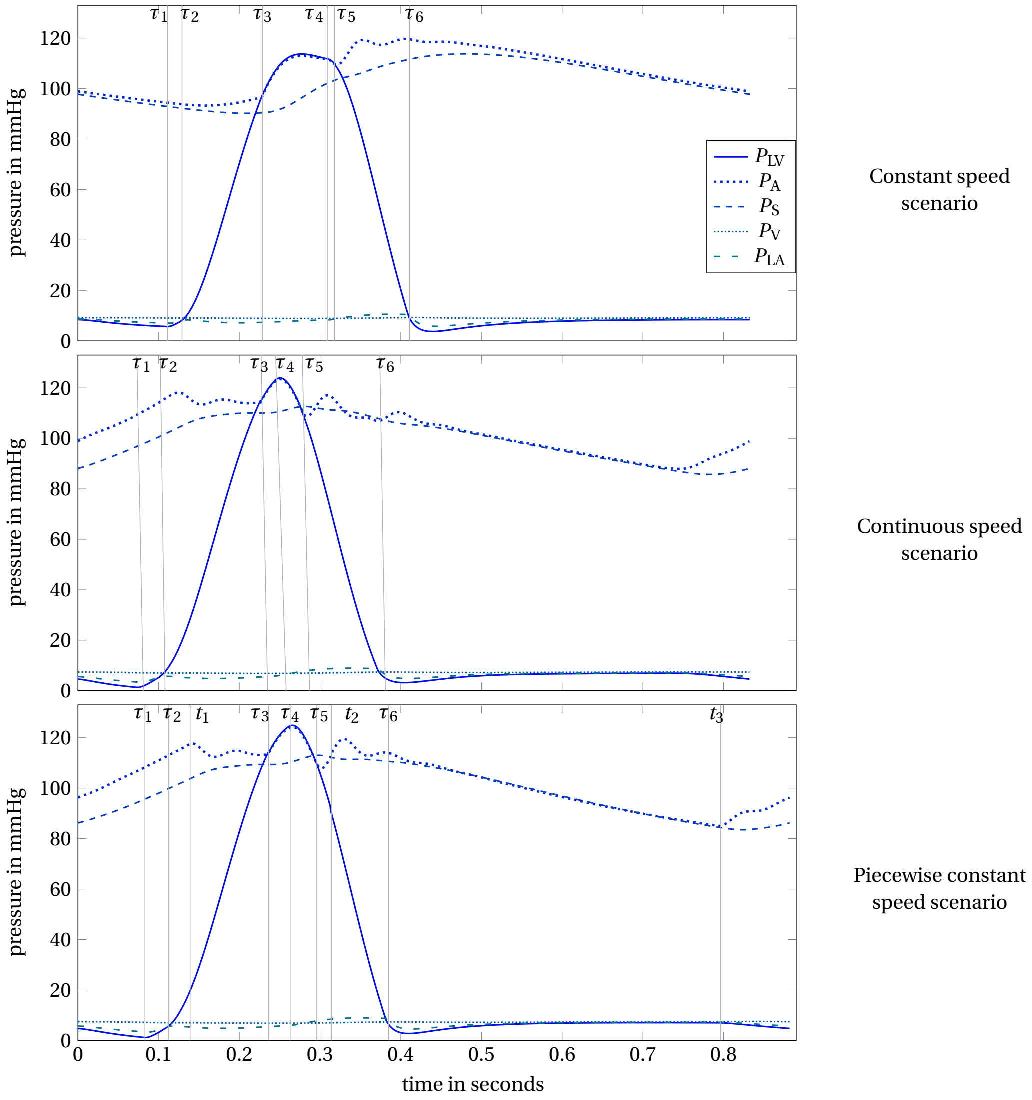


Figure 10.13: Calculated pressure differential states for the OCP solutions with different pump speed scenarios. The switching times τ_i and pwc speed changing times t_i are depicted with the vertical grey lines.

large flow values occur already during atrial contraction followed by a peak during ventricular contraction, which accounts for the remaining physiological contraction force. The flow for the continuous pump speed appears to be slightly negative around $t = 0.4$ since we relaxed the tolerance parameter in (10.34) for achieving numerical convergence. The upper right panel shows that the larger amplitudes of the pumping speed for the continuous and pwc scenario compared with constant speed translate into faster AVP movements.

Figure 10.15 summarizes the objective values and run times for the OCP solutions. Clearly, the obtained objective value with the pwc speed profile is only slightly larger than the one calculated with continuous pump speed, while the objective value with constant speed is not competitive.

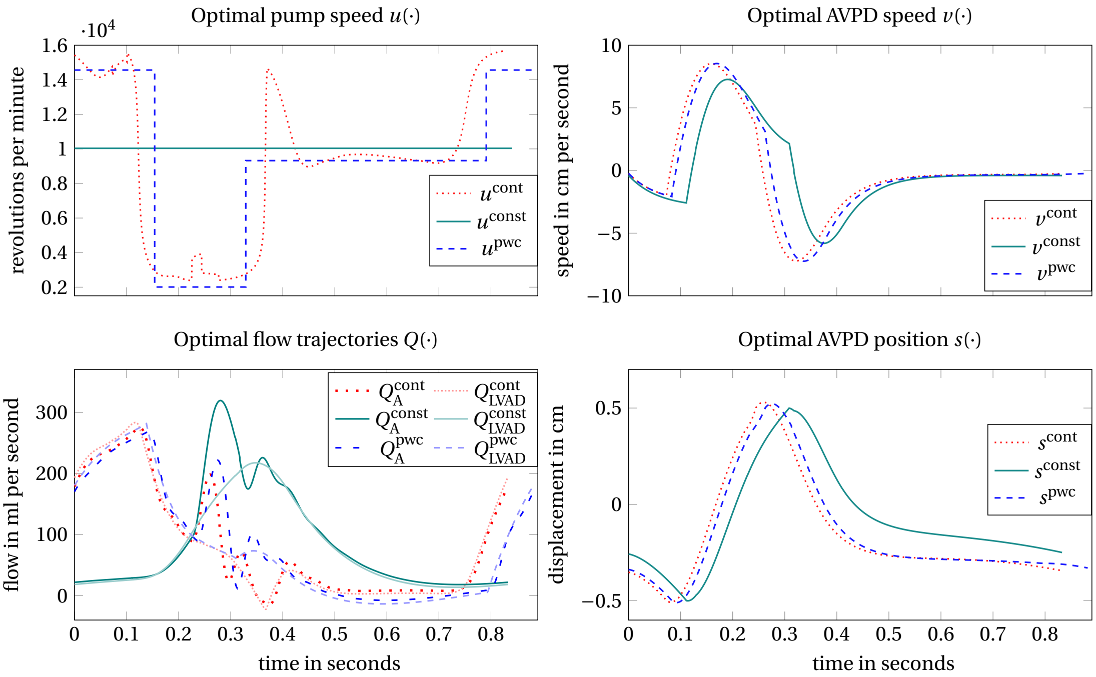


Figure 10.14: Calculated optimal pump speed u, flows Q, atrioventricular plane displacement (AVPD) speed v and position s for the OCP solutions. The superscripts *cont, const,* and *pwc* abbreviate *continuous, constant* and *piecewise constant* rotor speed.

10.2.7 Discussion

The root-mean-square error between $\widehat{P}_{\text{LV}}$ and P_{LV} is about twice as large as the one constructed via a trial method in [130]. However, the latter study did not take into account the constraints (10.32), (10.33), and (10.36) during the data fitting procedure. The performed PE should be discussed critically with respect to overfitting since we optimized nine parameters with available measurements for only one differential state. To this end, we calculated the relative standard deviation values based on the Fisher-Information matrix as defined in [134], which represents a common criterion for evaluating the quality of the results of a PE. The standard deviation values for the optimal parameter vector $\boldsymbol{p}^*$ are

$$\%\,\text{standard deviation}\,(\boldsymbol{p}^*) = [42.3, 24.5, 40.4, 35.5, 354.2, 34.6, 267.7, 49.1, 26.2]^\top.$$

The calculated standard deviation values are mostly relatively low and thus indicate a robust quality of the obtained estimation. In contrast, the values for F_{AC} and A_{LA} with about 354 % and 268 % are quite large, and we postulate that they have a minor impact on P_{LV}. Future work should focus on sensitivity analysis for choosing the right parameters to be estimated.

The LV pressure function's increased peak after using pumping assist is consistent with typical partial LVAD support [190]. The causality can be explained as follows: the LVAD continuously delivers blood to the aorta, increasing the aortic pressure. Therefore the LV pressure must be elevated for the aortic valve to open. This effect is more pronounced in the scenarios with continuous and pwc pump speed, which can be categorized as counterpulsative policies and thus facilitate the aortic pressure to oscillate less. Our finding that a counterpulsative pump-

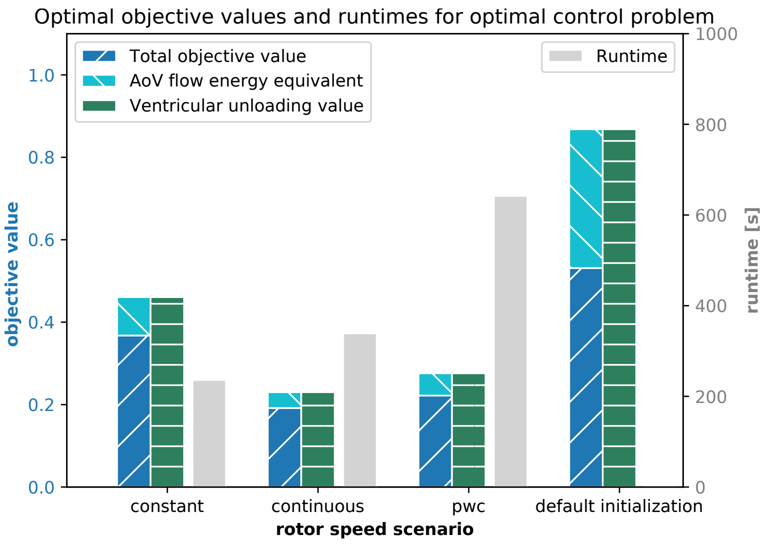


Figure 10.15: Comparison of constructed optimal objective values and run times for the OCP from Section 10.2.4. The objective values for the default initialization are obtained with $u(\cdot) = 8000\text{rpm}$. Notice that the total objective value results from subtracting the aortic valve flow value from the ventricular unloading, as defined in (10.2.4). The objective value decreases from 0.53105 after initialization to 0.36758 with constant speed. Continuous and pwc speed modulation construct even lower objective values, which amount to 0.19149 and 0.22182, respectively.

ing strategy minimizes the given objective function is in accordance with [7], where the same objective function was applied.

In practice, the LV pressure increases during diastole, and there is an atrial kick at the end of ventricular diastole. This seems to be not captured in the model since the end-diastolic LV pressure seems to be reducing, see Figure 10.11. Also, the difference between P_{LV} and P_{A} appears to be over 10 mmHg, while natural pressure gradients are in the 2-3 mmHg range, suggesting that the valve resistance is too high.

After applying the LVAD support, we observed shortened atrial contraction phases, equivalent to shortened ventricular filling phases. We interpret this behavior as a result of maximizing the ventricular unloading. This implies the LA pumps against less resistance and reaches the maximum contraction state more quickly, which is represented by the AVP reaching the switching distance $-S_D$. In this way, the AVPD model can realistically capture interactions of an LVAD and the cardiac system, as elaborated in [62, 258]. The continuous and pwc pump speed scenarios have more degrees of freedom compared with the constant speed scenario and thus improve the diastolic function even more, as depicted by the shortened atrial contraction phases.

Connection to clinical application

LVADs work in an online environment where the system state can change rapidly, especially the heart rate and blood volume shift. Currently, there are no sensors available that provide long-term measure signals of the presented differential states [44]. Despite these aspects and our restricting assumptions (see next subsection), we claim that optimized speed profiles from offline computations may provide superior performance (see Figure 10.15) and could be considered in the following way.

1. The optimal control framework can be used to benchmark a whole range of speed profiles, in particular modern pwc speed profiles, that result from different objective functions, models, and constraints.

2. These evaluations can be carried out on a patient-specific basis. As proof of concept, we demonstrated for one patient that the cardiovascular model can be efficiently altered to represent the patient's LV pressure function. This approach can be extended to include time-series measurements of the aortic and LA pressure (via pulmonary capillary wedge pressure and conductance catheterization), the flows at the mitral and aortic valve, and the AVP speed and displacement (all via Doppler echocardiography). Overall, at least five differential states could be used for model personalization. The data used so far were measured invasively. In contrast, in routine clinical investigations, echocardiography can be used to measure and use non-invasive data concerning approximated pressure-volume loops and time series of valve flows.

3. Offline computations of an optimal speed profile for different situations, e.g., rest, exercise, or rhythm disturbances such as atrial fibrillation, could be done beforehand and used in an online setting assuming information about the system status is available.

4. The presented algorithmic idea can be extended to model predictive control. In particular, the possibility to incorporate MDT constraints to avoid rapid speed changes paves the way for a realistic extension to the online setting.

Limitations

Simplified Model
We applied a lumped model that simplifies the heart and the cardiovascular system by neglecting the right heart, the pulmonary system, valve regurgitation, and spatial interactions between compartments. Dilated heart failure is very commonly associated with valve regurgitation so that our assumption to neglect it should be seen as critical. According to the Frank-Starling law, the passive LV compliance C_{LV} is set to be constant but may change instantaneously. The way we model the LVAD and its interaction with the heart is also highly simplified. Moreover, neurological feedback processes such as the baroreflex are not captured.

Assumptions
In reality, the heart rate and thus the duration of the cycle is very variable, especially through exertion or sport. In most cases of patients requiring LVAD therapy, the stable heart rhythm is a scarce phenomenon. The rhythm disturbances are rather the dominant pattern in the individuals suffering from heart failure. Therefore the steady-state assumption must be viewed critically. Furthermore, we neglect rotor and blood inertia so that the rotor speed can be controlled arbitrarily. Nevertheless, our framework may be suitable as a general starting point for twofold development as part of future work. First, we may be able to base the optimization process on critical parameters for clinical availability. Second, the early detection of atrial fibrillation as the most common rhythm disturbance in heart failure could be implemented to

switch the working regime of the pump into different mode [227].

Measured data
We included in this study only one subject and measured data for only one differential state. Future work shall address numerical tests with several patients and with additional measured states.

Control approach
Other ventricular work measures, such as the pressure-volume area, could be applied as an objective function. This study assumes a canonical order of the active phases for the valves and the contraction force. This order might not always be correct in practice, so that future work should consider optimization with implicit switches, but without fixing the order of active phases. Besides, we optimize over one heart cycle, whereas the pulsatility speed mode profiles of some LVADs such as HeartMate 3 last for more than one heart cycle.

Pump flow rates
The realistic range of LVAD pump flow rate is between 2 and 10 liters per minute, where the lower bound is due to the risk of thrombosis. The computed pump flow rates fall below or exceed this range on some time points but are on average over the whole heart cycle in the realistic range. Moreover, the almost instantaneous flow rate changes from almost zero up to 18 liters per minute will not happen in practice.

Switched systems framework applicability

The developed multiphase algorithm is applicable to other models and settings. For example, the OCP can also be interpreted on a cardiac model with time-varying elastance function as a switched system, with the valves still representing the implicit switches and changes of the constant pump speed representing controllable switches. Analogously, the framework can be beneficial for PE of cardiac models without LVAD application, but with different scope, e.g., cardiac resynchronization therapy.

There are similar devices as an LVAD available or under development, for which an OCP could be solved efficiently with the switched systems framework. For instance, total artificial hearts such as RealHeart [250] and Carmat [241] or intra-aortic blood pumps [84] also involve discrete system changes induced by piston pumps (RealHeart), controlled valves (Carmat), or pulsatility rotor speed modes (intra-aortic blood pump). Finally, the next generation LVADs may include more advanced control features that can lead to different control modes to switch on/off. The TORVAD device [96] falls into this category and works with two magnetic pistons within a torus generating pulsatile flow.

10.2.8 Conclusion

We have proposed a switched systems algorithm for the optimal control of LVADs that can calculate optimal constant, pwc, or continuous pump speed profiles. As proof of concept, we showed that this algorithm can be used to adapt a cardiovascular model to patient-specific data and benchmark simulations of personalized LVAD control policies. The importance of achieving hemodynamic optimization in LVAD patients is highlighted by a significantly lower rate of hospital readmissions [132] and could benefit from in silico analysis, such as the presented speed profile evaluations. Moreover, we have demonstrated realistic simulations of a model based on AVPD instead of using the widespread time-varying elastance model and examined thereby the heart to LVAD interactions. Future work may test the algorithm on more patient data, more realistic conditions such as exercise or rhythm disturbance, and model extensions. The proposed algorithm could also help to evaluate pulsatile speed modulation modes of modern devices such as HeartMate 3 or HeartWare HVAD. Moreover, the problem could be formulated and solved with dropped assumptions on the order of phases and with more discrete choices.

10.2.9 Supplemental material

Model parameter values

The resistance and inertance of the LVAD is captured by the parameters R_{LVAD} and L_{LVAD}. We use the values from [238] as shown in Table 10.4 and given by

$$L_{\mathrm{LVAD}} = L_i + L_o + 0.02177, \qquad R_{\mathrm{LVAD}} = R_i + R_o + 0.1707.$$

We altered the model from [130] to the situation of left heart failure by increasing the left atrial and ventricular compliance, increasing the systemic resistance, decreasing the length of AVPD, increasing the cross-section of LA and LV to account for a dilated heart, and decreasing the contraction force F_C. The specific parameter values are defined in Table 10.4.

Optimization parameter values

The following lower and upper bounds for the variables of the optimization problems were applied.

$$\begin{aligned}
\boldsymbol{p} &= [R_{\mathrm{AVP}}, C_{\mathrm{LV}}, L_{\mathrm{AVP}}, F_{\mathrm{VC}}, F_{\mathrm{AC}}, A_{\mathrm{LV}}, A_{\mathrm{LA}}, k_{\mathrm{RAD}}, S_D]^\top,\\
\boldsymbol{p}_{\mathrm{lb}} &= [100, 0.45, 15, 4500, 900, 30, 15, 0.5, 0.2]^\top,\\
\boldsymbol{p}_{\mathrm{ub}} &= [600, 5, 60, 7000, 1500, 42, 25, 1.35, 0.6]^\top,\\
\boldsymbol{x} &= [P_{\mathrm{LA}}, P_{\mathrm{LV}}, P_{\mathrm{A}}, P_{\mathrm{S}}, P_{\mathrm{V}}, Q_{\mathrm{A}}, Q_{\mathrm{LVAD}}, v, s]^\top,\\
\boldsymbol{x}_{\mathrm{lb}} &= [0, 0, 40, 0, 0, -100, -60, -15, -1.2]^\top,\\
\boldsymbol{x}_{\mathrm{ub}} &= [60, 150, 160, 160, 150, 1000, 300, 15, 1.2]^\top,\\
\boldsymbol{u}_{\mathrm{lb}} &= 2000, \quad \boldsymbol{u}_{\mathrm{ub}} = 16000.
\end{aligned}$$

We used as initial differential state value for the PE problem:

$$\boldsymbol{x}_0 = [8, 8, 90, 90, 8, 10, 0, 0, -0.2]^\top.$$

Furthermore, the following parameters for constraints were applied:

$$\begin{aligned}
&\epsilon_{\mathrm{flow}} = 10 \text{ ml/s}, \quad \epsilon_{\mathrm{back}} = 40 \text{ ml/s}, \qquad \epsilon_{\mathrm{CO}} = 80 \text{ ml}, \qquad V_{\mathrm{CO}} = 5000 \text{ ml},\\
&\epsilon_{\mathrm{sw}} = 0.001, \qquad \epsilon_{\mathrm{per}} = 0.09, \qquad \epsilon_{\mathrm{partial}} = 0.01 \text{ ml}.
\end{aligned}$$

Parameter	Description	Value
C_{LA}	Left atrial compliance	0.6 ml/mmHg
C_{LV}	Left ventricular compliance	0.45 ml/mmHg
C_{A}	Aortic compliance	0.08 ml/mmHg
C_{S}	Systemic arterial compliance	1.2 ml/mmHg
C_{V}	Venous compliance	50 ml/mmHg
R_{M}	Mitral valve resistance	0.005 mmHg s/ml
R_{AoV}	Aortic valve resistance	0.005 mmHg s/ml
R_{S}	Systemic arterial resistance	1.1 mmHg s/ml
R_{C}	Characteristic resistance	0.05 mmHg s/ml
R_{i}	LVAD inlet resistance	0.0677 mmHg s/ml
R_{o}	LVAD outlet resistance	0.0677 mmHg s/ml
R_{AVP}	Damping of AVP	300 mmHg cm s
L_{S}	Arterial inertance	0.001 mmHg s^2/ml
L_{i}	Inlet inertance of LVAD	0.0127 mmHg s^2/ml
L_{o}	Outlet inertance of LVAD	0.0127 mmHg s^2/ml
L_{AVP}	Inertia of AVP	30 mmHg cm s^2
β	Pump-to-pressure coefficient	-9.9025e-7
A_{LA}	Left atrial cross-section	25 cm^2
A_{LV}	Left ventricular cross-section	50 cm^2
F_{AC}	Left atrial contraction force	1000 mmHg cm^2
F_{VC}	LV contraction force	5000 mmHg cm^2
k_{RAD}	Radial function coefficient	1.2
S_{D}	Switching threshold for AVP	0.4 cm

Table 10.4: Default parameter values for the cardiovascular, circulatory system, and LVAD model.

We permitted a deviation of 50 ms for the model heart cycle from the subject's measured cycle length. The MU times for the pwc pump speed levels were

$$C_{U_1} = 0.1s, \quad C_{U_2} = 0.1s, \quad C_{U_3} = 0.1s.$$

We also enforced upper bounds $\boldsymbol{\tau}_{\mathrm{ub}}$ on the seven phase durations

$$\boldsymbol{\tau}_{\mathrm{ub}} = [0.3, 0.2, 0.2, 0.2, 0.2, 0.3, 0.5]^\top \text{seconds}.$$

The objective parameters are

$$\varrho_1 = 0.5, \quad \varrho_2 = 0.000133\mathrm{J} \ / \ (\mathrm{mmHg\ ml}), \quad \varrho_3 = 0.01\mathrm{J\ s/ml}.$$

We used the following `IPOPT` setting: *'tol': 1e-6, 'constr_viol_tol': 1e-6, 'compl_inf_tol': 1e-6, 'dual_inf_tol': 1e-6.*

Chapter 11

Concluding remarks and outlook

We end this thesis by providing conclusions about the derived algorithmic framework and provide possible directions for future research.

11.1 Conclusions

This thesis extended and generalized the CIA decomposition algorithm for the error-controlled solution of MIOCPs. The extension was achieved for different MILP formulations of the rounding problem with information from the NLP, whose solutions can be further improved by recombination. Furthermore, we proposed decomposing the solution process into multiple rounding and NLP steps to gradually increase the number of fixed binary variables. Another extension addressed the inclusion of path constraints and multiphase dynamics into the (CIA) rounding problem.

In Chapter 5, we recapitulated results that establish approximation bounds on the constructed binary control solution of the decomposition algorithm with respect to its relaxed solution. The results imply that the optimal solution of (BOCP) can be arbitrarily well approximated by refining the discretization grid and applying the CIA decomposition until the desired solution quality has been achieved. We proved associated results with respect to the algorithmic generalizations. Although arbitrarily good solution quality can also be achieved for the proposed algorithmic CIA extensions by grid refinement, the extensions are specially designed to construct the best possible solution of a fixed grid. Moreover, in practical implementations, the grid is often fixed due to model assumptions, or grid refinement is undesirable due to the practical setting, such as a vast problem size.

Another observation from the real-world modeling of MIOCPs is the common need for combinatorial constraints, such as minimum dwell time or limiting switching constraints. To this end, Chapter 6 presented different algorithms for solving the (CIA)-problem with and without combinatorial conditions. We introduced fast rounding algorithms such as STO-BnB, DNFR, DSUR, and AMDR and examined their properties in terms of the worst-case integral deviation gap of the constructed binary solution. The integral deviation gap was further investigated in Chapter 7, where we proved the tight bound $\frac{2n_\omega-3}{2n_\omega-2}\bar{\Delta}$ for the (CIA) problem. From this, we derived a tight bound for MU time constraints, and we discussed bounds for the minimum down (MD) time-constrained case. If the discrete total variation (TV) of the binary control is required to be bounded by $\sigma_{\max}$, we demonstrated the tight bound $\frac{N+\sigma_{\max}+1}{3+2\sigma_{\max}}\bar{\Delta}$ for $n_\omega = 2$ and equidistant discretization and also discussed more general cases.

The developed algorithms were implemented in the open-source package `pycombina`. Numerical results from several test problems together with the real-world case studies involving electric vehicles and cardiology highlighted the advantageous properties of the proposed algorithmic framework in terms of solution quality and run time.

The CIA decomposition framework is simple to use for the solution process of a generic MIOC problem class since only the relaxed problem, a rounding problem, and the problem with fixed binary variables have to be solved. Efficient NLP solvers are available for the first and third problems, while the MILP rounding problem can be solved with `pycombina`, allowing combinatorial constraints to be added intuitively. Numerical experiments show that the CIA decomposition performs notably well in terms of the objective quality for problems with small combinatorial requirements since its deviation from the relaxed solution is negligible. Furthermore, the decomposition algorithm works much faster than mixed-integer nonlinear program (MINLP) solvers such as `Bonmin`, while still offering qualitatively similar solutions. Last but not least, the algorithmic framework is generic in the sense that it is also applicable for problems with implicit switches [182] or partial differential equation (PDE) constraints [176], which were not the focus of this thesis.

11.2 Future work

The research results from this thesis can be extended in several directions. The algorithms for solving CIA problems under combinatorial constraints in Chapter 6 could be further numerically compared with each other. In particular, a benchmark study on BnB, STO-BnB, a recent MILP solver, the shortest path algorithm [28], and the penalty alternating direction method [102] would be appealing. Application of the fast rounding methods DNFR, DSUR, and AMDR could leverage the mixed-integer solution process of the model predictive control (MPC) context. Future research will most likely improve MINLP solvers, so their development is also attractive for the solution of MIOCPs.

This thesis provided methods to incorporate information from the NLP into the (CIA) problem step. Additional work is necessary to derive further tailored MILP formulations for specific MIOCP classes and to develop efficient algorithms for these cases. Instead of applying the (CIA) rounding problem, an interesting idea is to construct a second-order Taylor approximation modification of the binary controls at the optimal solution of ($\mathrm{NLP}_{\mathrm{rel}}$). This results in an mixed-integer quadratic program (MIQP) problem based on a Gauss-Newton type linear-quadratic expansion, whose numerical performance should be investigated.

In the case of PDE-restricted MIOCPs, some of the time-coupled combinatorial constraints presented in this thesis make little sense. Since these problems are solved within the CIA decomposition framework by employing space-filling curves, MDT constraints would, for example, cause a series of mesh cells successively connected by the space-filling curve to all assume the same binary value. Nevertheless, other combinatorial conditions, like the total maximum up time constraints from Section 3.2.5, may be compatible, and their application in the PDE context should therefore be further investigated.

Another potential research direction is the application of machine/deep learning to solve MIOCPs and improve the CIA decomposition. For instance, machine learning could be used to learn the "best" or most suitable (CIA) problem version, respectively rounding method, for specific MIOC problem classes. For problems with time-critical solution requirements, such as in the MPC context, an optimal solution could be learned in advance for various default situations, and it could then be used as a starting value in the solution process.

The software package `pycombina` could be further developed to enhance the applicability of the proposed algorithms. It would also be interesting to compare the CIA decomposition with algorithms from the switched systems community [260, 204]. In this context, it is relevant to promote the algorithm and mitigate parallel developments, such as what happened with the embedding transformation technique.

Finally, there are also open research questions regarding the application problems presented. For example, the efficient use of energy will remain an important topic in the future, and therefore, an extension of the algorithm presented for the online control of HEVs will be useful. Due to high patient variability, the control of LVADs in cardiology is associated with more uncertainty than control approaches for automotives. This is one reason why optimal control techniques have thus far found little impact in the field of cardiology. Overall, there is still much pioneering work to be done in control for cardiology, which means that there is also great potential for improvement. Specifically related to the LVAD study, the proposed algorithms could be tested with more patient data and a mock circulatory loop system. The algorithms should also be generalized to deal with several heart cycles and an arbitrary order of system phases.

Appendix A

Definitions of function spaces

This appendix chapter reviews function spaces that are referred to in this thesis. We assume that the reader is familiar with Lebesgue measure and integration theory. We refer to the textbook [3] for more details.

Definition A.1 (Indicator function χ)
Let $\Omega_2 \subseteq \Omega_1 \subseteq \mathbb{R}^n$ be subsets. The indicator function $\chi_{\Omega_2} : \Omega_1 \to \{0,1\}$ is defined as

$$\chi_{\Omega_2}(x) := \begin{cases} 1, & \textit{if } x \in \Omega_2, \\ 0, & \textit{else.} \end{cases}$$

Definition A.2 (k-times continuously differentiable functions, C^k)
Consider an interval $\mathscr{I} \subseteq \mathbb{R}$ and $k \in \mathbb{N}$. Let the function space of all k-times continuously differentiable functions $C^k(\mathscr{I}, \mathbb{R})$ defined by its contained functions $f : \mathscr{I} \to \mathbb{R}$ which are assumed to be continuous on $\mathscr{I}$ and so are all their derivatives $f^{(i)}(\cdot)$ of order $i \le k$ assumed to be continuous.

We note that the above definition can be directly extended to multidimensional functions that depend on more than one variable by using partial derivatives.

Definition A.3 (Lebesgue spaces L^p)
Consider a non-empty set $\Omega \subseteq \mathbb{R}^n$ and $1 \le p \le \infty$. Let $f : \Omega \to \mathbb{R}$ a measurable function such that $|f|^p$ is integrable. We identify in the Lebesgue space $L^p(\Omega, \mathbb{R}^n)$ functions f that are equal almost everywhere in Ω, i.e., $L^p(\Omega, \mathbb{R}^n)$ is defined as the space of all equivalence classes of the functions f. We equip this space with the norm

$$\|f\|_{L^p} := \begin{cases} \sqrt[p]{\int_\Omega |f(x)|^p \, dx}, & \textit{if } 1 \le p < \infty, \\ \operatorname{ess\,sup}_{x \in \Omega} |f(x)|, & \textit{if } p = \infty. \end{cases}$$

In contrast to the quotient space L^p, we denote with $\mathscr{L}^p(\Omega, \mathbb{R}^n)$ the space of measurable functions f, where $|f|^p$ is integrable, without neglecting their values on sets of measure zero.

For defining Sobolev spaces, we introduce the multi-index notation for partial derivatives. Let $\boldsymbol{\alpha} \in \mathbb{N}^n$ denote such a multi-index, with $|\boldsymbol{\alpha}| := \sum_{i=1}^n \alpha_i \le k \in \mathbb{N}$. If a function f is k-times differentiable, we set for its partial derivatives

$$\partial^{\boldsymbol{\alpha}} f := \frac{\partial^{|\boldsymbol{\alpha}|} f}{\partial x_1^{\alpha_1} \partial x_2^{\alpha_2} \cdots \partial x_n^{\alpha_n}}.$$

Definition A.4 (Sobolev spaces $W^{k,p}$)
Consider a non-empty set $\Omega \subseteq \mathbb{R}^n$ and $1 \le p \le \infty$. The Sobolev *space $W^{k,p}(\Omega, \mathbb{R})$ consists of all functions of $L^p(\Omega, \mathbb{R})$ that admit all partial derivatives of order at most $k \in \mathbb{N}$:*

$$W^{k,p}(\Omega, \mathbb{R}) := \{f \in L^p(\Omega, \mathbb{R}) \mid \partial^{\boldsymbol{\alpha}} f \in L^p(\Omega, \mathbb{R}) \textit{ for all } 0 \le |\boldsymbol{\alpha}| \le k.\}.$$

We equip the space $W^{k,p}(\Omega, \mathbb{R})$ *with the following* Sobolev-*norm:*

$$\|f\|_{W^{k,p}} := \begin{cases} \sqrt[p]{\sum_{|\boldsymbol{\alpha}|\leq k} \|\partial^{\boldsymbol{\alpha}} f\|_{L^p}}, & \textit{if } 1 \leq p < \infty, \\ \max_{|\boldsymbol{\alpha}|\leq k} \|\partial^{\boldsymbol{\alpha}} f\|_{L^\infty}, & \textit{if } p = \infty. \end{cases}$$

Bibliography

[1] P. Abichandani, H. Benson, and M. Kam. Multi-vehicle path coordination under communication constraints. In *American Control Conference*, pages 650–656, 2008. doi: 10.1109/ACC.2008.4586566.

[2] W. Achtziger and C. Kanzow. Mathematical programs with vanishing constraints: optimality conditions and constraint qualifications. *Mathematical Programming Series A*, 114:69–99, 2008. doi: 10.1007/s10107-006-0083-3.

[3] R. Adams and J. Fournier. *Sobolev Spaces*, volume 140 of *Pure and Applied Mathematics (Amsterdam)*. Elsevier/Academic Press, Amsterdam, second edition, 2003.

[4] U. Ali and M. Egerstedt. Optimal control of switched dynamical systems under dwell time constraints. In *53rd IEEE Conference on Decision and Control*, pages 4673–4678. IEEE, 2014.

[5] A.-H. H. AlOmari, F. Javed, A. V. Savkin, E. Lim, R. F. Salamonsen, D. G. Mason, and N. H. Lovell. Non-invasive measurements based model predictive control of pulsatile flow in an implantable rotary blood pump for heart failure patients. In *2011 19th Mediterranean Conference on Control & Automation (MED)*, pages 491–496. IEEE, 2011.

[6] A.-H. H. AlOmari, A. V. Savkin, M. Stevens, D. G. Mason, D. L. Timms, R. F. Salamonsen, and N. H. Lovell. Developments in control systems for rotary left ventricular assist devices for heart failure patients: a review. *Physiological measurement*, 34(1):R1, 2012.

[7] R. Amacher, J. Asprion, G. Ochsner, H. Tevaearai, M. J. Wilhelm, A. Plass, A. Amstutz, S. Vandenberghe, and M. S. Daners. Numerical optimal control of turbo dynamic ventricular assist devices. *bioengineering*, 1(1):22–46, 2014.

[8] R. Amacher, G. Ochsner, and M. Schmid Daners. Synchronized pulsatile speed control of turbodynamic left ventricular assist devices: review and prospects. *Artificial organs*, 38 (10):867–875, 2014.

[9] J. A. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl. Casadi: A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1):1–36, 2019.

[10] 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, 3 1998.

[11] M. Arioli, I. Duff, N. Gould, J. Hogg, J. Scott, and H. Thorne. The HSL Mathematical Software Library. http://www.hsl.rl.ac.uk/, 2007.

[12] A. Arndt, P. Nüsser, and B. Lampe. Fully autonomous preload-sensitive control of implantable rotary blood pumps. *Artificial organs*, 34(9):726–735, 2010.

[13] T. Arts, T. Delhaas, P. Bovendeerd, X. Verbeek, and F. W. Prinzen. Adaptation to mechanical load determines shape and properties of heart and circulation: the circadapt model. *American Journal of Physiology-Heart and Circulatory Physiology*, 288(4):H1943–H1954, 2005.

[14] U. Ascher and L. Petzold. *Computer Methods for Ordinary Differential Equations and Differential–Algebraic Equations.* SIAM, Philadelphia, 1998.

[15] U. Ascher, R. Mattheij, and R. Russell. *Numerical Solution of Boundary Value Problems for Differential Equations.* Prentice Hall, Engelwood Cliffs, NJ, 1988.

[16] H. Axelsson, Y. Wardi, M. Egerstedt, and E. Verriest. Gradient descent approach to optimal mode scheduling in hybrid dynamical systems. *Journal of Optimization Theory and Applications*, 136(2):167–186, 2008.

[17] M. A. Bakouri, R. F. Salamonsen, A. V. Savkin, A.-H. H. AlOmari, E. Lim, and N. H. Lovell. A sliding mode-based starling-like controller for implantable rotary blood pumps. *Artificial organs*, 38(7):587–593, 2014.

[18] V. Bär. Ein Kollokationsverfahren zur numerischen Lösung allgemeiner Mehrpunktrandwertaufgaben mit Schalt– und Sprungbedingungen mit Anwendungen in der optimalen Steuerung und der Parameteridentifizierung. Diploma thesis, Rheinische Friedrich–Wilhelms–Universität zu Bonn, 1983.

[19] L. D. Beal, D. C. Hill, R. A. Martin, and J. D. Hedengren. Gekko optimization suite. *Processes*, 6(8):106, 2018.

[20] R. Bellman. *Dynamic Programming.* University Press, Princeton, N.J., 6th edition, 1957. ISBN 0-486-42809-5 (paperback).

[21] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. Mixed-Integer Nonlinear Optimization. In A. Iserles, editor, *Acta Numerica*, volume 22, pages 1–131. Cambridge University Press, 2013. doi: 10.1017/S0962492913000032.

[22] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, 1999.

[23] A. Bemporad, A. Giua, and C. Seatzu. A master-slave algorithm for the optimal control of continuous-time switched affine systems. In *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, volume 2, pages 1976–1981. IEEE, 2002.

[24] S. Bengea and R. DeCarlo. Optimal control of switching systems. *Automatica*, 41(1): 11–27, 2005. doi: 10.1016/j.automatica.2004.08.003.

[25] D. Bertsekas. *Dynamic programming and optimal control, Volume 1.* Athena Scientific, Belmont, Mass., 3. ed. edition, 2005. ISBN 1-886529-26-4 ; 978-1-886529-26-7.

[26] D. Bertsekas. *Dynamic programming and optimal control, Volume 2.* Athena Scientific, Belmont, Mass., 4. ed. edition, 2012.

[27] F. Bestehorn, C. Hansknecht, C. Kirches, and P. Manns. A switching cost aware rounding method for relaxations of mixed-integer optimal control problems. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 7134–7139. IEEE, 2019.

[28] F. Bestehorn, C. Hansknecht, C. Kirches, and P. Manns. Mixed-integer optimal control problems with switching costs: A shortest path approach. *Mathematical Programming, Series B*, pages 1–32, 2020.

[29] J. Betts. *Practical Methods for Optimal Control Using Nonlinear Programming.* SIAM, Philadelphia, 2001.

[30] L. Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers & Chemical Engineering*, 8:243–248, 1984.

[31] L. Biegler. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes.* Series on Optimization. SIAM, 2010.

[32] H. Bock. Numerische Behandlung von zustandsbeschränkten und Chebyshev-Steuerungsproblemen. Technical Report R106/81/11, Carl Cranz Gesellschaft, Heidelberg, 1981.

[33] H. Bock and R. 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.

[34] H. Bock and K. 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.

[35] H. G. Bock, C. Kirches, A. Meyer, and A. Potschka. Numerical solution of optimal control problems with explicit and implicit switches. *Optimization Methods and Software*, 33(3): 450–474, 2018.

[36] P. Boggs and J. Tolle. Sequential Quadratic Programming. *Acta Numerica*, 4:1–51, 1995.

[37] T. J. Böhme and B. Frank. *Hybrid Systems, Optimal Control and Hybrid Vehicles.* Springer, 2017.

[38] P. Bonami, A. Olivares, M. Soler, and E. Staffetti. Multiphase mixed-integer optimal control approach to aircraft trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 36(5):1267–1277, 2013.

[39] F. Borrelli, M. Baotić, A. Bemporad, and M. Morari. Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica*, 41(10): 1709–1721, 2005.

[40] F. Borrelli, A. Bemporad, and M. Morari. *Predictive control for linear and hybrid systems.* Cambridge University Press, 2017.

[41] J. R. Boston, J. F. Antaki, and M. A. Simaan. Hierarchical control of heart-assist devices. *IEEE Robotics & Automation Magazine*, 10(1):54–64, 2003.

[42] J. Bouchat. Reinforcement learning for the optimal control of hybrid systems. Master's thesis, UC Louvain, 2019.

[43] S. Bouchez, Y. Van Belleghem, F. De Somer, M. De Pauw, R. Stroobandt, and P. Wouters. Haemodynamic management of patients with left ventricular assist devices using echocardiography: the essentials. *European Heart Journal-Cardiovascular Imaging*, 20 (4):373–382, 2019.

[44] S. Bozkurt. Physiologic outcome of varying speed rotary blood pump support algorithms: a review study. *Australasian physical & engineering sciences in medicine*, 39(1):13–28, 2016.

[45] S. Bozkurt and S. Bozkurt. In-silico evaluation of left ventricular unloading under varying speed continuous flow left ventricular assist device support. *Biocybernetics and Biomedical Engineering*, 37(3):373–387, 2017.

[46] A. Bryson and Y.-C. Ho. *Applied Optimal Control*. Wiley, New York, 1975.

[47] A. Bürger. *Nonlinear mixed-integer model predictive control of renewable energy systems*. PhD thesis, Albert-Ludwigs-Universität Freiburg, 2020.

[48] A. Bürger, C. Zeile, Altmann-Dieses, S. A., Sager, and M. Diehl. An algorithm for mixed-integer optimal control of solar thermal climate systems with MPC-capable runtime. In *2018 European Control Conference (ECC)*, pages 1379–1385. IEEE, 2018.

[49] A. Bürger, C. Zeile, Altmann-Dieses, S. A., Sager, and M. Diehl. Design, implementation and simulation of an MPC algorithm for switched nonlinear systems under combinatorial constraints. *Journal of Process Control*, 81:15–30, 2019.

[50] A. Bürger, C. Zeile, M. Hahn, A. Altmann-Dieses, S. Sager, and M. Diehl. pycombina: An open-source tool for solving combinatorial approximation problems arising in mixed-integer optimal control. In *Proceedings of the IFAC World Congress*, 2020. accepted.

[51] M. Burger, M. Gerdts, S. Göttlich, and M. Herty. Dynamic programming approach for discrete-valued time discrete optimal control problems with dwell time constraints. In *IFIP Conference on System Modeling and Optimization*, pages 159–168. Springer, 2015.

[52] J. Burgschweiger, B. Gnädig, and M. Steinbach. Nonlinear Programming Techniques for Operative Planning in Large Drinking Water Networks. *The Open Applied Mathematics Journal*, 3:1–16, 2009.

[53] C. Büskens and H. Maurer. SQP-methods for solving optimal control problems with control and state constraints: adjoint variables, sensitivity analysis and real-time control. *Journal of Computational and Applied Mathematics*, 120:85–108, 2000.

[54] 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.

[55] A. Cauligi, P. Culbertson, B. Stellato, D. Bertsimas, M. Schwager, and M. Pavone. Learning mixed-integer convex optimization strategies for robot planning and control. *arXiv preprint arXiv:2004.03736*, 2020.

[56] Y. Chen and M. Lazar. An efficient MPC algorithm for switched nonlinear systems with minimum dwell time constraints. *arXiv preprint arXiv:2002.09658*, 2020. (visited on 2020-06-27).

[57] S. Choi, J. R. Boston, and J. F. Antaki. Hemodynamic controller for left ventricular assist device based on pulsatility ratio. *Artificial organs*, 31(2):114–125, 2007.

[58] T. Christof and A. Löbel. PORTA – POlyhedron Representation Transformation Algorithm. `http://www.zib.de/Optimization/Software/Porta/`. PORTA Homepage.

[59] F. J. Christophersen, M. Baotić, and M. Morari. Optimal control of piecewise affine systems: A dynamic programming approach. In *Control and Observer Design for Nonlinear Finite and Infinite Dimensional Systems*, pages 183–198. Springer, 2005.

[60] M. Claeys, J. Daafouz, and D. Henrion. Modal occupation measures and LMI relaxations for nonlinear switched systems control. *Automatica*, 64:143–154, 2016.

[61] M. Conforti, G. Cornuéjols, and G. Zambelli. Extended formulations in combinatorial optimization. *4OR*, 8(1):1–48, 2010.

[62] L. G. Cox, S. Loerakker, M. C. Rutten, B. A. De Mol, and F. N. Van De Vosse. A mathematical model to evaluate control strategies for mechanical circulatory support. *Artificial organs*, 33(8):593–603, 2009.

[63] M. G. Crespo-Leiro, M. Metra, L. H. Lund, D. Milicic, M. R. Costanzo, G. Filippatos, F. Gustafsson, S. Tsui, E. Barge-Caballero, N. De Jonge, et al. Advanced heart failure: a position statement of the heart failure association of the european society of cardiology. *European journal of heart failure*, 20(11):1505–1535, 2018.

[64] S. Crow, R. John, A. Boyle, S. Shumway, K. Liao, M. Colvin-Adams, C. Toninato, E. Missov, M. Pritzker, C. Martin, et al. Gastrointestinal bleeding rates in recipients of nonpulsatile and pulsatile left ventricular assist devices. *The Journal of thoracic and cardiovascular surgery*, 137(1):208–215, 2009.

[65] J. Daafouz, P. Riedinger, and C. Iung. Stability analysis and control synthesis for switched systems: a switched lyapunov function approach. *IEEE transactions on automatic control*, 47(11):1883–1887, 2002.

[66] R. Davoudi and S. M. Hosseini. A semidefinite programming approach for polynomial switched optimal control problems. *Optimal Control Applications and Methods*, 40(4): 626–646, 2019.

[67] A. De Marchi. On the mixed-integer linear-quadratic optimal control with switching cost. *IEEE Control Systems Letters*, 3(4):990–995, 2019.

[68] S. V. Deo, V. Sharma, Y. H. Cho, I. K. Shah, and S. J. Park. De novo aortic insufficiency during long-term support on a left ventricular assist device: a systematic review and meta-analysis. *ASAIO journal*, 60(2):183–188, 2014.

[69] M. Diehl, D. Leineweber, and A. Schäfer. MUSCOD-II Users' Manual. IWR-Preprint 2001-25, Universität Heidelberg, 2001.

[70] A. I. Doban and M. Lazar. A switched systems approach to cancer therapy. In *2015 European Control Conference (ECC)*, pages 2718–2724. IEEE, 2015.

[71] J. Duan, Z. Yi, D. Shi, C. Lin, X. Lu, and Z. Wang. Reinforcement-learning-based optimal control of hybrid energy storage systems in hybrid AC–DC microgrids. *IEEE Transactions on Industrial Informatics*, 15(9):5355–5364, 2019.

[72] M. Duran and I. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36(3):307–339, 1986.

[73] S. Ebbesen, M. Salazar, P. Elbert, C. Bussi, and C. H. Onder. Time-optimal control strategies for a hybrid electric race car. *IEEE Transactions on Control Systems Technology*, 26 (1):233–247, 2018.

[74] M. Egerstedt, Y. Wardi, and F. Delmotte. Optimal Control of Switching Times in Switched Dynamical Systems. In *Proceedings of the 42nd IEEE Concference of Decision and Control*, 2003.

[75] M. Egerstedt, Y. Wardi, and H. Axelsson. Transition-time optimization for switched-mode dynamical systems. *IEEE Transactions on Automatic Control*, 51:110–115, 2006.

[76] P. Elbert, M. Widmer, H.-J. Gisler, and C. Onder. Stochastic dynamic programming for the energy management of a serial hybrid electric bus. *International Journal of Vehicle Design*, 69(1-4):88–112, 2015.

[77] G. Faragallah, Y. Wang, E. Divo, and M. A. Simaan. A new current-based control model of the combined cardiovascular and rotary left ventricular assist device. In *Proceedings of the 2011 American Control Conference*, pages 4775–4780. IEEE, 2011.

[78] T. Faulwasser and A. Murray. Turnpike properties in discrete-time mixed-integer optimal control. *IEEE Control Systems Letters*, 4(3):704–709, 2020.

[79] R. Fourer, D. Gay, and B. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Duxbury Press, 2002.

[80] G. Franze, W. Lucia, and F. Tedesco. Command governor for constrained switched systems with scheduled model transition dwell times. *International Journal of Robust and Nonlinear Control*, 27(18):4949–4967, 2017.

[81] D. Frick, A. Domahidi, and M. Morari. Embedded optimization for mixed logical dynamical systems. *Computers & Chemical Engineering*, 72:21–33, 2015.

[82] 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.

[83] N. R. Gaddum, M. Stevens, E. Lim, J. Fraser, N. Lovell, D. Mason, D. Timms, and R. Salamonsen. Starling-like flow control of a left ventricular assist device: in vitro validation. *Artificial organs*, 38(3):E46–E56, 2014.

[84] B. Gao, Y. Chang, Y. Xuan, Y. Zeng, and Y. Liu. The hemodynamic effect of the support mode for the intra-aorta pump on the cardiovascular system. *Artificial organs*, 37(2): 157–165, 2013.

[85] D. W. Gao, C. Mi, and A. Emadi. Modeling and simulation of electric and hybrid vehicles. In *Proceedings of the IEEE*, volume 95, pages 729–745. IEEE, 2007.

[86] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, New York, 1979.

[87] D. Garmatter, M. Porcelli, F. Rinaldi, and M. Stoll. Improved penalty algorithm for mixed integer PDE constrained optimization (MIPDECO) problems. *arXiv preprint arXiv:1907.06462*, 2019.

[88] 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.

[89] M. Gerdts. A variable time transformation method for mixed-integer optimal control problems. *Optimal Control Applications and Methods*, 27(3):169–182, 2006.

[90] M. Gerdts. *Optimal Control of Ordinary Differential Equations and Differential-Algebraic Equations.* Habilitation, University of Bayreuth, 2006.

[91] M. Gerdts. *Optimal Control of ODEs and DAEs.* De Gruyter, 2012.

[92] M. Gerdts and S. Sager. Mixed-Integer DAE Optimal Control Problems: Necessary conditions and bounds. In L. Biegler, S. Campbell, and V. Mehrmann, editors, *Control and Optimization with Differential-Algebraic Constraints*, pages 189–212. SIAM, 2012.

[93] J. Gesenhues, M. Hein, M. Habigt, M. Mechelinck, T. Albin, and D. Abel. Nonlinear object-oriented modeling based optimal control of the heart: performing precise preload manipulation maneuvers using a ventricular assist device. In *2016 European Control Conference (ECC)*, pages 2108–2114. IEEE, 2016.

[94] G. A. Giridharan and M. Skliar. Nonlinear controller for ventricular assist devices. *Artificial organs*, 26(11):980–984, 2002.

[95] R. Goebel, R. G. Sanfelice, and A. R. Teel. Hybrid dynamical systems. *IEEE control systems magazine*, 29(2):28–93, 2009.

[96] J. R. Gohean. *Hierarchical control of a two-piston toroidal blood pump.* PhD thesis, University of Texas at Austin, 2019.

[97] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Professional, 1989.

[98] H. Gonzalez, R. Vasudevan, M. Kamgarpour, S. S. Sastry, R. Bajcsy, and C. Tomlin. A numerical method for the optimal control of switched systems. In *49th IEEE Conference on Decision and Control (CDC)*, pages 7519–7526. IEEE, 2010.

[99] H. Gonzalez, R. Vasudevan, M. Kamgarpour, S. S. Sastry, R. Bajcsy, and C. J. Tomlin. A descent algorithm for the optimal control of constrained nonlinear switched dynamical systems. In *Proceedings of the 13th ACM international conference on Hybrid systems: computation and control*, pages 51–60, 2010.

[100] 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.

[101] S. Göttlich, A. Potschka, and U. Ziegler. Partial outer convexification for traffic light optimization in road networks. *SIAM Journal on Scientific Computing*, 39(1):B53–B75, 2017.

[102] S. Göttlich, F. M. Hante, A. Potschka, and L. Schewe. Penalty alternating direction methods for mixed-integer optimal control with combinatorial constraints. *arXiv preprint arXiv:1905.13554*, 2019.

[103] S. Göttlich, A. Potschka, and C. Teuber. A partial outer convexification approach to control transmission lines. *Computational Optimization and Applications*, 72(2):431–456, 2019.

[104] M. Gräber, C. Kirches, H. G. Bock, J. P. Schlöder, W. Tegethoff, and J. Köhler. Determining the optimum cyclic operation of adsorption chillers by a direct method for periodic optimal control. *International Journal of Refrigeration*, 34(4):902–913, 2011.

[105] R. L. Graham, E. L. Lawler, J. K. Lenstra, and A. R. Kan. Optimization and approximation in deterministic sequencing and scheduling: a survey. In *Annals of discrete mathematics*, volume 5, pages 287–326. Elsevier, 1979.

[106] S. Gros and M. Diehl. *Numerical Optimal Control (DRAFT)*. 2019. `https://www.syscop.de/files/2019ss/NOC/book-NOCSE.pdf`, (accessed 2020-8-10).

[107] 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.

[108] M. Guglin. What did we learn about VADs in 2018? *The VAD Journal*, 2019.

[109] I. Gurobi Optimization. Gurobi Optimizer Reference Manual, 2015. `http://www.gurobi.com`.

[110] F. Gustafsson and J. G. Rogers. Left ventricular assist device therapy in advanced heart failure: patient selection and outcomes. *European journal of heart failure*, 19(5):595–602, 2017.

[111] L. Guzzella and A. Sciarretta. *Vehicle propulsion systems*. Springer, Berlin, 3 edition, 2013.

[112] O. Habeck, M. E. Pfetsch, and S. Ulbrich. Global optimization of mixed-integer ode constrained network problems using the example of stationary gas transport. *SIAM Journal on Optimization*, 29(4):2949–2985, 2019.

[113] M. Hahn and S. Sager. Combinatorial integral approximation for mixed-integer pde-constrained optimization problems. Technical report, 2018. ANL Preprint ANL/MCS-P9037-0118.

[114] M. Hahn, C. Kirches, P. Manns, S. Sager, and C. Zeile. Decomposition and approximation for PDE-constrained mixed-integer optimal control. In M. H. et al., editor, *SPP1962 Special Issue*. Birkhäuser, 2019. (accepted).

[115] M. Hahn, S. Leyffer, and S. Sager. Binary optimal control by trust-region steepest descent. *submitted for peer review*, 2020. `http://www.optimization-online.org/DB_HTML/2020/01/7589.html`, (accessed 2020-07-11).

[116] J. Han and D. R. Trumble. Cardiac assist devices: early concepts, current technologies, and future innovations. *Bioengineering*, 6(1):18, 2019.

[117] F. Hante and S. Sager. Relaxation Methods for Mixed-Integer Optimal Control of Partial Differential Equations. *Computational Optimization and Applications*, 55(1):197–225, 2013.

[118] F. M. Hante. Relaxation methods for hyperbolic PDE mixed-integer optimal control problems. *Optimal Control Applications and Methods*, 38(6):1103–1110, 2017. ISSN 1099-1514. doi: 10.1002/oca.2315. oca.2315.

[119] F. M. Hante. Relaxation methods for hyperbolic pde mixed-integer optimal control problems. *Optimal Control Applications and Methods*, 38(6):1103–1110, 2017.

[120] R. Hartl, S. Sethi, and R. Vickson. A survey of the Maximum Principles for optimal control problems with state constraints. *SIAM Review*, 37:181–218, 1995.

[121] J. D. Hedengren, R. A. Shishavan, K. M. Powell, and T. F. Edgar. Nonlinear modeling, estimation and predictive control in APMonitor. *Computers & Chemical Engineering*, 70: 133 – 148, 2014. ISSN 0098-1354. doi: http://dx.doi.org/10.1016/j.compchemeng.2014.04.013. Manfred Morari Special Issue.

[122] S. Hedlund and A. Rantzer. Convex Dynamic Programming for Hybrid Systems. *IEEE Transactions on Automatic Control*, 47(9):1536–1540, Sept. 2002.

[123] H. Hermes and J. Lasalle. *Functional analysis and time optimal control*, volume 56 of *Mathematics in science and engineering*. Academic Press, New York and London, 1969.

[124] J. P. Hespanha and A. S. Morse. Stability of switched systems with average dwell-time. In *Proceedings of the 38th IEEE conference on decision and control (Cat. No. 99CH36304)*, volume 3, pages 2655–2660. IEEE, 1999.

[125] P. Hespanhol, R. Quirynen, and S. Di Cairano. A structure exploiting branch-and-bound algorithm for mixed-integer model predictive control. In *2019 18th European Control Conference (ECC)*, pages 2763–2768. IEEE, 2019.

[126] A. Heydari. Optimal switching with minimum dwell time constraint. *Journal of the Franklin Institute*, 354(11):4498–4518, 2017.

[127] G. Hicks and W. Ray. Approximation methods for optimal control systems. *Can. J. Chem. Engng.*, 49:522–528, 1971.

[128] T. Hoheisel. *Mathematical Programs with Vanishing Constraints.* PhD thesis, Julius–Maximilians–Universität Würzburg, July 2009.

[129] W. Horn. Some simple scheduling algorithms. *Naval Research Logistics Quarterly*, 21(1): 177–185, 1974.

[130] F. Huang, Z. Gou, and Y. Fu. Preliminary evaluation of a predictive controller for a rotary blood pump based on pulmonary oxygen gas exchange. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 233(2):267–278, 2019.

[131] H. Huang, Z. Shu, B. Song, L. Ji, and N. Zhu. Modeling left ventricular dynamics using a switched system approach based on a modified atrioventricular piston unit. *Medical engineering & physics*, 63:42–49, 2019.

[132] T. Imamura, B. Chung, A. Nguyen, G. Sayer, and N. Uriel. Clinical implications of hemodynamic assessment during left ventricular assist device therapy. *Journal of cardiology*, 71(4):352–358, 2018.

[133] W. Jakob, J. Rhinelander, and D. Moldovan. pybind11 – seamless operability between C++11 and Python, 2016. Last accessed October 29, 2019.

[134] D. Janka, S. Körkel, and H. G. Bock. Direct multiple shooting for nonlinear optimum experimental design. In *Multiple Shooting and Time Domain Decomposition Methods*, pages 115–141. Springer, 2015.

[135] M. Jung. *Relaxations and Approximations for Mixed-Integer Optimal Control.* PhD thesis, University Heidelberg, 2013.

[136] M. Jung, C. Kirches, and S. Sager. On Perspective Functions and Vanishing Constraints in Mixed-Integer Nonlinear Optimal Control. In M. Jünger and G. Reinelt, editors, *Facets of Combinatorial Optimization – Festschrift for Martin Grötschel*, pages 387–417. Springer Berlin Heidelberg, 2013.

[137] M. Jung, G. Reinelt, and S. Sager. The Lagrangian Relaxation for the Combinatorial Integral Approximation Problem. *Optimization Methods and Software*, 30(1):54–80, 2015.

[138] M. N. Jung, C. Kirches, S. Sager, and S. Sass. Computational approaches for mixed integer optimal control problems with indicator constraints. *Vietnam Journal of Mathematics*, 46:1023–1051, 2018. doi: https://doi.org/10.1007/s10013-018-0313-z.

[139] A. A. Kahloul and A. Sakly. Hybrid approach for constrained optimal control of nonlinear switched systems. *Journal of Control, Automation and Electrical Systems*, pages 1–9, 2020.

[140] Y. Kawajiri and L. Biegler. A Nonlinear Programming Superstructure for Optimal Dynamic Operations of Simulated Moving Bed Processes. *I&EC Research*, 45(25):8503–8513, 2006.

[141] M. Ketelhut, S. Stemmler, J. Gesenhues, M. Hein, and D. Abel. Iterative learning control of ventricular assist devices with variable cycle durations. *Control Engineering Practice*, 83:33–44, 2019.

[142] A. Khakimova, A. Kusatayeva, A. Shamshimova, D. Sharipova, A. Bemporad, Y. Familiant, A. Shintemirov, V. Ten, and M. Rubagotti. Optimal energy management of a small-size building via hybrid model predictive control. *Energy and Buildings*, 140:1–8, 2017.

[143] E.-j. Kim and M. Capoccia. Synergistic model of cardiac function with a heart assist device. *Bioengineering*, 7(1):1, 2020.

[144] N. Kim, S. Cha, and H. Peng. Optimal control of hybrid electric vehicles based on pontryagin's minimum principle. *IEEE Transactions on Control Systems Technology*, 19(5): 1279–1287, 2011.

[145] 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/.

[146] C. Kirches, S. Sager, H. Bock, and J. Schlöder. Time-optimal control of automobile test drives with gear shifts. *Optimal Control Applications and Methods*, 31(2):137–153, March/April 2010.

[147] C. Kirches, H. Bock, and S. Leyffer. Modeling Mixed-Integer Constrained Optimal Control Problems in AMPL. In F. Breitenecker and I. Troch, editors, *Proceedings of MATHMOD 2012, Vienna, February 15–17, 2012*, 2012. ARGESIM Report No. S38.

[148] C. Kirches, H. Bock, J. Schlöder, and S. Sager. Mixed-integer NMPC for predictive cruise control of heavy-duty trucks. In *European Control Conference*, pages 4118–4123, Zurich, Switzerland, July 17-19 2013.

[149] C. Kirches, F. Lenders, and P. Manns. Approximation properties and tight bounds for constrained mixed-integer optimal control. *Optimization Online*, April 2016. (submitted to SIAM Journal on Control and Optimization).

[150] C. Kirches, E. Kostina, A. Meyer, and M. Schlöder. Numerical solution of optimal control problems with switches, switching costs and jumps. Technical report, March 2019. (visited on 2020-07-03).

[151] D. E. Kirk. *Optimal control theory: an introduction*. Courier Corporation, 2004.

[152] J. K. Kirklin, D. C. Naftel, F. D. Pagani, R. L. Kormos, L. W. Stevenson, E. D. Blume, S. L. Myers, M. A. Miller, J. T. Baldwin, and J. B. Young. Seventh intermacs annual report: 15,000 patients and counting. *The Journal of Heart and Lung Transplantation*, 34(12): 1495–1504, 2015.

[153] T. Koch, B. Hiller, M. E. Pfetsch, and L. Schewe, editors. *Evaluating Gas Network Capacities.* SIAM-MOS series on Optimization. SIAM, 2015.

[154] B. Korte and J. Vygen. *Combinatorial Optimization.* Springer Verlag, Berlin Heidelberg New York, 3rd edition, 2006. ISBN 3-540-25684-9 (hardcover).

[155] D. Kress, M. Barketau, and E. Pesch. Single-machine batch scheduling to minimize the total setup cost in the presence of deadlines. *Journal of Scheduling*, 21(6):595–606, 2018.

[156] M. Kutta. Beitrag zur näherungsweisen Integration totaler Differentialgleichungen. *Zeitschrift für Mathematik und Physik*, 46:435–453, 1901.

[157] M. Kvasnica, P. Grieder, M. Baotić, and M. Morari. Multi-parametric toolbox (MPT). In *International Workshop on Hybrid Systems: Computation and Control*, pages 448–462. Springer, 2004.

[158] I. D. Laoutaris. Restoring pulsatility and peakVO2 in the era of continuous flow, fixed pump speed, left ventricular assist devices:a hypothesis of pump's or patient's speed? *European journal of preventive cardiology*, 26(17):1806–1815, 2019.

[159] D. Lebiedz, S. Sager, H. 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.

[160] J. Lee and S. Leyffer. *Mixed integer nonlinear programming*, volume 154. Springer Science & Business Media, 2011.

[161] J. Lee, J. Leung, and F. Margot. Min-up/min-down polytopes. *Discrete Optimization*, 1: 77–85, 2004.

[162] F. Lenders. *Numerical methods for mixed-integer optimal control with combinatorial constraints.* PhD thesis, Ruprecht-Karls-Universität Heidelberg, February 2018.

[163] J. Lescot, A. Sciarretta, Y. Chamaillard, and A. Charlet. On the integration of optimal energy management and thermal management of hybrid electric vehicles. In *2010 IEEE Vehicle Power and Propulsion Conference*, pages 1–6. IEEE, 2010.

[164] S. Leyffer, D. K. P. Cay, and B. van Bloemen Waanders. Mixed-integer PDE-constrained optimization. In L. Liberti, S. Sager, and A. Wiegele, editors, *Mixed-integer Nonlinear Optimization: A Hatchery for Modern Mathematics*, volume 46 of *Oberwolfach Reports*, pages 2738–2740. 2015.

[165] D. Liberzon and A. S. Morse. Basic problems in stability and design of switched systems. *IEEE control systems magazine*, 19(5):59–70, 1999.

[166] P. Lilienthal, M. Tetschke, E. Schalk, T. Fischer, and S. Sager. Optimized and personalized phlebotomy schedules for patients suffering from polycythemia vera. *Frontiers in Physiology*, 2020. accepted.

[167] D. Limebeer, G. Perantoni, and A. Rao. Optimal control of formula one car energy recovery systems. *International Journal of Control*, 87(10):2065–2080, 2014.

[168] H. Lin and P. J. Antsaklis. Stability and stabilizability of switched linear systems: a survey of recent results. *IEEE Transactions on Automatic control*, 54(2):308–322, 2009.

[169] E. Lindelöf. Sur l'application des méthodes d'approximations successives à l'étude des intégrales réelles des équations différentielles ordinaires. *Journal de Mathématiques Pures et Appliquées*, 10:117–128, 1894.

[170] A. Locatelli. *Optimal control – an introduction.* Birkhäuser, Basel Boston Berlin, 2001.

[171] W. Lu, P. Zhu, and S. Ferrari. A hybrid-adaptive dynamic programming approach for the model-free control of nonlinear switched systems. *IEEE Transactions on Automatic Control*, 61(10):3203–3208, 2015.

[172] J. Lumens, T. Delhaas, B. Kirn, and T. Arts. Three-wall segment (triseg) model describing mechanics and hemodynamics of ventricular interaction. *Annals of biomedical engineering*, 37(11):2234–2255, 2009.

[173] S. Lundbäck. Cardiac pumping and function of the ventricular septum. *Acta physiologica scandinavica. Supplementum*, 127(550), 1986.

[174] A. Maitland and J. McPhee. Fast NMPC with mixed-integer controls using quasi-translations. volume 51, pages 343–348. Elsevier, 2018.

[175] E. Maksuti, A. Bjällmark, and M. Broomé. Modelling the heart with the atrioventricular plane as a piston unit. *Medical engineering & physics*, 37(1):87–92, 2015.

[176] P. Manns. *Approximation Properties of Sum-Up Rounding.* Dissertation, Technische Universität Carolo-Wilhelmina zu Braunschweig, 2019.

[177] P. Manns and C. Kirches. Improved regularity assumptions for partial outer convexification of mixed-integer pde-constrained optimization problems. *ESAIM: Control, Optimisation and Calculus of Variations*, 2019.

[178] P. Manns and C. Kirches. Multi-dimensional sum-up rounding using hilbert curve iterates. In *Proceedings in Applied Mathematics and Mechanics*, 2019. (accepted).

[179] P. Manns, C. Kirches, and F. Lenders. Approximation properties of sum-up rounding in the presence of vanishing constraints. *Mathematics and Computations*, 2020.

[180] T. Marcucci and R. Tedrake. Warm start of mixed-integer programs for model predictive control of hybrid systems. *arXiv preprint arXiv:1910.08251*, 2019.

[181] S. Mehrotra. On the Implementation of a Primal-Dual Interior Point Method. *SIAM Journal on Optimization*, 2(4):575–601, 1992.

[182] A. Meyer. *Numerical solution of optimal control problems with explicit and implicit switches.* PhD thesis, Ruprecht-Karls-Universität Heidelberg, January 2020.

[183] R. T. Meyer, M. Zefran, and R. A. DeCarlo. A comparison of the embedding method with multiparametric programming, mixed-integer programming, gradient-descent, and hybrid minimum principle-based methods. *IEEE Transactions on Control Systems Technology*, 22(5):1784–1800, 2014.

[184] R. T. Meyer, R. A. DeCarlo, and S. Pekarek. Hybrid model predictive power management of a battery-supercapacitor electric vehicle. *Asian Journal of Control*, 18(1):150–165, 2016.

[185] K. Mombaur. *Stability Optimization of Open-loop Controlled Walking Robots*. PhD thesis, Universität Heidelberg, 2001.

[186] V. V. Naik. *Mixed-integer quadratic programming algorithms for embedded control and estimation*. PhD thesis, IMT School for Advanced Studies Lucca, 2018.

[187] B. C. Ng, P. A. Smith, F. Nestler, D. Timms, W. E. Cohn, and E. Lim. Application of adaptive starling-like controller to total artificial heart using dual rotary blood pumps. *Annals of biomedical engineering*, 45(3):567–579, 2017.

[188] H. J. Oberle and H. J. Pesch. Numerical Treatment of Delay Differential Equations by Hermite Interpolation. *Numerische Mathematik*, 37:235–255, 1981.

[189] G. Ochsner, R. Amacher, M. J. Wilhelm, S. Vandenberghe, H. Tevaearai, A. Plass, A. Amstutz, V. Falk, and M. Schmid Daners. A physiological controller for turbodynamic ventricular assist devices based on a measurement of the left ventricular volume. *Artificial organs*, 38(7):527–538, 2014.

[190] D. Ogawa, S. Kobayashi, K. Yamazaki, T. Motomura, T. Nishimura, J. Shimamura, T. Tsukiya, T. Mizuno, Y. Takewa, and E. Tatsumi. Mathematical evaluation of cardiac beat synchronization control used for a rotary blood pump. *Journal of Artificial Organs*, 22(4):276–285, 2019.

[191] J. Oldenburg, W. Marquardt, D. Heinz, and D. Leineweber. Mixed Logic Dynamic Optimization Applied to Batch Distillation Process Design. *AIChE Journal*, 49(11):2900–2917, 2003.

[192] A. Pakniyat and P. E. Caines. On the hybrid minimum principle: The hamiltonian and adjoint boundary conditions. *IEEE Transactions on Automatic Control*, 2020.

[193] K. Palagachev. *Mixed-Integer Optimal Control and Bilevel Optimization: Vanishing Constraints and Scheduling Tasks*. PhD thesis, Universität der Bundeswehr München, 2017.

[194] K. Palagachev and M. Gerdts. Mathematical programs with blocks of vanishing constraints arising in discretized mixed-integer optimal control problems. *Set-Valued and Variational Analysis*, 23(1):149–167, 2015.

[195] J. Pantoja and D. Q. Mayne. Sequential quadratic programming algorithm for discrete optimal control problems with control inequality constraints. *International Journal on Control*, 53:823–836, 1991.

[196] B. Passenberg. *Theory and algorithms for indirect methods in optimal control of hybrid systems*. PhD thesis, Technische Universität München, 2012.

[197] H. Patel, R. Madanieh, C. E. Kosmas, S. K. Vatti, and T. J. Vittorio. Complications of continuous-flow mechanical circulatory support devices. *Clinical Medicine Insights: Cardiology*, 9:CMC–S19708, 2015.

[198] S. Peitz and S. Klus. Koopman operator-based model reduction for switched-system control of PDEs. *Automatica*, 106:184–191, 2019.

[199] A. Petrou, M. Monn, M. Meboldt, and M. S. Daners. A novel multi-objective physiological control system for rotary left ventricular assist devices. *Annals of biomedical engineering*, 45(12):2899–2910, 2017.

[200] C. Picard. Mémoire sur la théorie des équations aux dérivées partielles et la méthode des approximations successives. *Journal de Mathématiques Pures et Appliquées*, 6:145–210, 1890.

[201] T. Pirbodaghi, S. Axiak, A. Weber, T. Gempp, and S. Vandenberghe. Pulsatile control of rotary blood pumps: does the modulation waveform matter? *The Journal of thoracic and cardiovascular surgery*, 144(4):970–977, 2012.

[202] L. Pontryagin, V. Boltyanski, R. Gamkrelidze, and E. Miscenko. *The Mathematical Theory of Optimal Processes*. Wiley, Chichester, 1962.

[203] D. Rajan, S. Takriti, et al. Minimum up/down polytopes of the unit commitment problem with start-up costs. 2005.

[204] R. B. Ramanarayan Vasudevan, Humberto Gonzalez and S. S. Sastry. Consistent approximations for the optimal control of constrained switched systems—part 2: An implementable algorithm. *SIAM Journal on Control and Optimization*, 51:4484–4503, 2013.

[205] J. R. Rice. *Numerical Methods in Software and Analysis*. Elsevier, 2014.

[206] A. Richards and J. How. Mixed-integer programming for control. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 2676–2683. IEEE, 2005.

[207] R. M. Rieck. *Discrete Controls and Constraints in Optimal Control Problems*. PhD thesis, Technische Universität München, 2017.

[208] M. Ringkamp, S. Ober-Blöbaum, and S. Leyendecker. On the time transformation of mixed integer optimal control problems using a consistent fixed integer control function. *Mathematical Programming*, 161(1):551–581, 2017. ISSN 1436-4646. doi: 10.1007/s10107-016-1023-5.

[209] G. Rizzoni and S. Onori. Energy management of hybrid electric vehicles: 15 years of development at the Ohio State University. In *IFP Energies nouvelles International Conference: IFAC Workshop on Engine and Powertrain Control, Simulation and Modeling*, volume 70, pages 41–54, 2015.

[210] N. Robuschi. *Mixed-integer optimal control methods for the energy management of hybrid electric vehicles*. PhD thesis, Politecnico di Milano, Italy, 2019.

[211] N. Robuschi, M. Salazar, P. Duhr, F. Braghin, and C. H. Onder. Minimum-fuel engine on/off control for the energy management of a hybrid electric vehicle via iterative linear programming. *IFAC-PapersOnLine*, 52(5):134–140, 2019.

[212] N. Robuschi, C. Zeile, S. Sager, and F. Braghin. Multiphase mixed-integer nonlinear optimal control of hybrid electric vehicles. *Automatica*, 123:109325, 2021. doi: https://doi.org/10.1016/j.automatica.2020.109325.

[213] J. Roll, A. Bemporad, and L. Ljung. Identification of piecewise affine systems via mixed-integer programming. *Automatica*, 40(1):37–50, 2004.

[214] W. Rudin. *Real and Complex Analysis*. McGraw-Hill, 1966.

[215] F. Rüffler. *Control and Optimization for Switched Systems of Evolution Equations*. PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2019.

[216] C. D. T. Runge. Über die numerische Auflösung von Differentialgleichungen. *Mathematische Annalen*, 46(2):167–178, 1895.

[217] D. Rüschen, S. Opitz, P. von Platen, L. Korn, S. Leonhardt, and M. Walter. Robust physiological control of rotary blood pumps for heart failure therapy. *at-Automatisierungstechnik*, 66(9):767–779, 2018.

[218] S. Sager. *Numerical methods for mixed–integer optimal control problems*. PhD thesis, Universität Heidelberg, 2006.

[219] S. Sager. Reformulations and Algorithms for the Optimization of Switching Decisions in Nonlinear Optimal Control. *Journal of Process Control*, 19(8):1238–1247, 2009.

[220] S. Sager. On the Integration of Optimization Approaches for Mixed-Integer Nonlinear Optimal Control. University of Heidelberg, August 2011. Habilitation.

[221] S. Sager. A benchmark library of mixed-integer optimal control problems. In J. Lee and S. Leyffer, editors, *Mixed Integer Nonlinear Programming*, pages 631–670. Springer, 2012.

[222] S. Sager and C. Zeile. On mixed-integer optimal control with constrained total variation of the integer control. *Computational Optimization and Applications*, 2020. doi: 10.1007/s10589-020-00244-5.

[223] S. Sager, G. Reinelt, and H. Bock. Direct Methods With Maximal Lower Bound for Mixed-Integer Optimal Control Problems. *Mathematical Programming*, 118(1):109–149, 2009.

[224] S. Sager, M. Jung, and C. Kirches. Combinatorial Integral Approximation. *Mathematical Methods of Operations Research*, 73(3):363–380, 2011. doi: 10.1007/s00186-011-0355-4.

[225] S. Sager, H. Bock, and M. Diehl. The Integer Approximation Error in Mixed-Integer Optimal Control. *Mathematical Programming A*, 133(1–2):1–23, 2012.

[226] S. Sager, M. Claeys, and F. Messine. Efficient upper and lower bounds for global mixed-integer optimal control. *Journal of Global Optimization*, 61(4):721–743, 2015. doi: 10.1007/s10898-014-0156-4.

[227] S. Sager, F. Bernhardt, F. Kehrle, M. Merkert, A. Potschka, B. Meder, H. Katus, and E. Scholz. Expert-enhanced machine learning for cardiac arrhythmia classification. *Preprint (Optimization Online), submitted to Artificial Intelligence in Medicine*, 2020. submitted.

[228] M. Sakly, A. Sakly, N. Majdoub, and M. Benrejeb. Optimization of switching instants for optimal control of linear switched systems based on genetic algorithms. *IFAC Proceedings Volumes*, 42(19):249–253, 2009.

[229] R. F. Salamonsen, E. Lim, N. Gaddum, A.-H. H. AlOmari, S. D. Gregory, M. Stevens, D. G. Mason, J. F. Fraser, D. Timms, M. K. Karunanithi, et al. Theoretical foundations of a starling-like controller for rotary blood pumps. *Artificial organs*, 36(9):787–796, 2012.

[230] M. Salazar, P. Elbert, S. Ebbesen, C. Bussi, and C. H. Onder. Time-optimal control policy for a hybrid electric race car. *IEEE Transactions on Control Systems Technology*, 25(6): 1921–1934, 2017.

[231] R. Sargent and G. Sullivan. The development of an efficient optimal control package. In J. Stoer, editor, *Proceedings of the 8th IFIP Conference on Optimization Techniques (1977), Part 2*, Heidelberg, 1978. Springer.

[232] A. Schmeisser, T. Rauwolf, A. Ghanem, T. Groscheck, D. Adolf, F. Grothues, K. Fischbach, O. Kosiek, C. Huth, S. Kropf, et al. Right heart function interacts with left ventricular remodeling after crt: A pressure volume loop study. *International journal of cardiology*, 268:156–161, 2018.

[233] M. Schori. *Solution of optimal control problems for switched systems. Algorithms and applications for hybrid vehicles.* PhD thesis, PhD thesis, Universität Rostock, 2015.

[234] A. Sciarretta, G. De Nunzio, and L. L. Ojeda. Optimal ecodriving control: Energy-efficient driving of road vehicles as an optimal control problem. *Control Systems, IEEE*, 35(5): 71–90, 2015.

[235] M. Shaikh and P. Caines. On the hybrid optimal control problem: Theory and Algorithms. *IEEE Transactions on Automatic Control*, 52:1587–1603, 2007.

[236] Y. Shi, P. V. Lawford, and D. R. Hose. Numerical modeling of hemodynamics with pulsatile impeller pump support. *Annals of biomedical engineering*, 38(8):2621–2634, 2010.

[237] J. Sieber and B. Krauskopf. Complex balancing motions of an inverted pendulum subject to delayed feedback control. *Physica D*, 197:332–345, 2004.

[238] M. A. Simaan, A. Ferreira, S. Chen, J. F. Antaki, and D. G. Galati. A dynamical state space representation and performance analysis of a feedback-controlled rotary left ventricular assist device. *IEEE Transactions on Control Systems Technology*, 17(1):15–28, 2008.

[239] M. Sirvent. *Incorporating Differential Equations into Mixed-Integer Programming for Gas Transport Optimization.* PhD thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg, 2018.

[240] M. S. Slaughter, J. G. Rogers, C. A. Milano, S. D. Russell, J. V. Conte, D. Feldman, B. Sun, A. J. Tatooles, R. M. Delgado III, J. W. Long, et al. Advanced heart failure treated with continuous-flow left ventricular assist device. *New England Journal of Medicine*, 361(23): 2241–2251, 2009.

[241] D. M. Smadja, S. Susen, A. Rauch, B. Cholley, C. Latrémouille, D. Duveau, L. Zilberstein, D. Méléard, M.-F. Boughenou, E. Van Belle, et al. The carmat bioprosthetic total artificial heart is associated with early hemostatic recovery and no acquired von willebrand syndrome in calves. *Journal of cardiothoracic and vascular anesthesia*, 31(5):1595–1602, 2017.

[242] 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.

[243] E. D. Sontag. Interconnected automata and linear systems: A theoretical framework in discrete-time. In *International Hybrid Systems Workshop*, pages 436–448. Springer, 1995.

[244] B. Stellato. *Mixed-integer optimal control of fast dynamical systems.* PhD thesis, University of Oxford, 2017.

[245] B. Stellato, S. Ober-Blöbaum, and P. J. Goulart. Second-order switching time optimization for switched dynamical systems. *IEEE Transaction on Automatic Control*, 62(10):5407–5414, 2017. doi: 10.1109/TAC.2017.2697681.

[246] J. Stoer. *Numerische Mathematik 1.* Springer Verlag, Berlin Heidelberg New York, 8th edition, 1999. ISBN 3-540-66154-9.

[247] O. Stryk. *Numerische Lösung optimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der adjungierten Variablen.* PhD thesis, TU Munich, 1995.

[248] Z. Sun, S. S. Ge, and T. H. Lee. Controllability and reachability criteria for switched linear systems. *Automatica*, 38(5):775–786, 2002.

[249] H. Sussmann. A maximum principle for hybrid optimal control problems. In *Conference proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, 1999.

[250] Z. Szabó, J. Holm, A. Najar, G. Hellers, I. Pieper, and H. Ahn. Scandinavian real heart (SRH) 11 implantation as total artificial heart (TAH) – experimental update. *J Clin Exp Cardiolog*, 9(578):2, 2018.

[251] N. Tauchnitz. *Das Pontrjaginsche Maximumprinzip für eine Klasse hybrider Steuerungsprobleme mit Zustandsbeschränkungen und seine Anwendung.* PhD thesis, BTU Cottbus, 2010.

[252] S. Trenn. Switched differential algebraic equations. In *Dynamics and Control of Switched Electronic Systems*, pages 189–216. Springer, 2012.

[253] T. Tsang, D. Himmelblau, and T. Edgar. Optimal control via collocation and non-linear programming. *International Journal on Control*, 21:763–768, 1975.

[254] A. Umeki, T. Nishimura, M. Ando, Y. Takewa, K. Yamazaki, S. Kyo, M. Ono, T. Tsukiya, T. Mizuno, Y. Taenaka, et al. Alteration of LV end-diastolic volume by controlling the power of the continuous-flow LVAD, so it is synchronized with cardiac beat: development

of a native heart load control system (NHLCS). *Journal of Artificial Organs*, 15(2):128–133, 2012.

[255] K. Uthaichana, S. Bengea, R. DeCarlo, S. Pekarek, and M. Zefran. Hybrid model predictive control tracking of a sawtooth driving profile for an HEV. In *American Control Conference, 2008*, pages 967–974, 2008.

[256] K. van Berkel, W. Klemm, T. Hofman, B. Vroemen, and M. Steinbuch. Optimal control of a mechanical hybrid powertrain with cold-start conditions. *IEEE Transactions on Vehicular Technology*, 63(4):1555–1566, 2014.

[257] S. Vandenberghe, P. Segers, J. F. Antaki, B. Meyns, and P. R. Verdonck. Hemodynamic modes of ventricular assist with a rotary blood pump: continuous, pulsatile, and failure. *Asaio Journal*, 51(6):711–718, 2005.

[258] S. Vandenberghe, P. Segers, P. Steendijk, B. Meyns, R. A. Dion, J. F. Antaki, and P. Verdonck. Modeling ventricular function during cardiac assist: Does time-varying elastance work? *ASAIO journal*, 52(1):4–8, 2006.

[259] R. Vanderbei. LOQO: An interior point code for quadratic programming. *Optimization Methods and Software*, 11(1–4):451–484, 1999. doi: 10.1080/10556789908805759.

[260] R. Vasudevan, H. Gonzalez, R. Bajcsy, and S. S. Sastry. Consistent approximations for the optimal control of constrained switched systems—part 1: A conceptual algorithm. *SIAM Journal on Control and Optimization*, 51(6):4463–4483, 2013.

[261] I. Vierhaus and R. Gottwald. SD-SCIP – system dynamics scip: A SCIP plug-in for solving system dynamics optimization problems. `http://sdscip.zib.de/`, 2017. (visited on 2020-07-10).

[262] V. Volterra. Variazioni e fluttuazioni del numero d'individui in specie animali conviventi. *Mem. R. Accad. Naz. dei Lincei.*, VI-2, 1926.

[263] A. Wächter and L. Biegler. Line Search Filter Methods for Nonlinear Programming: Motivation and Global Convergence. *SIAM Journal on Computing*, 16(1):1–31, 2005.

[264] A. Wächter and L. Biegler. On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. *Mathematical Programming*, 106(1): 25–57, 2006.

[265] X. Wang, H. He, F. Sun, and J. Zhang. Application study on the dynamic programming algorithm for energy management of plug-in hybrid electric vehicles. *Energies*, 8:3225–3244, 2015.

[266] Y. Wardi, M. Egerstedt, and M. U. Qureshi. Hamiltonian-based algorithm for relaxed optimal control. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7222–7227. IEEE, 2016.

[267] S. Wei, Y. Zou, F. Sun, and O. Christopher. A pseudospectral method for solving optimal control problem of a hybrid tracked vehicle. *Applied Energy*, 194:588–595, 2017.

[268] M. Willem. *Minimax theorems*, volume 24. Springer Science & Business Media, 1997.

[269] R. Willenheimer, C. Cline, L. Erhardt, and B. Israelsson. Left ventricular atrioventricular plane displacement: an echocardiographic technique for rapid assessment of prognosis in heart failure. *Heart*, 78(3):230–236, 1997.

[270] L. Wolsey and G. Nemhauser. *Integer and Combinatorial Optimization*. Number ISBN 0-471-35943-2. Wiley, Chichester, 1999.

[271] X. Wu, Q. Liu, K. Zhang, M. Cheng, and X. Xin. Optimal switching control for drug therapy process in cancer chemotherapy. *European Journal of Control*, 42:49–58, 2018.

[272] X. Wu, K. Zhang, and M. Cheng. Optimal control of constrained switched systems and application to electrical vehicle energy management. *Nonlinear Analysis: Hybrid Systems*, 30:171–188, 2018.

[273] Y. Wu, P. E. Allaire, G. Tao, and D. Olsen. Modeling, estimation, and control of human circulatory system with a left ventricular assist device. *IEEE transactions on control systems technology*, 15(4):754–767, 2007.

[274] X. Xu and P. J. Antsaklis. A dynamic programming approach for optimal control of switched systems. In *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No. 00CH37187)*, volume 2, pages 1822–1827. IEEE, 2000.

[275] X. Xu and P. J. Antsaklis. Optimal control of switched systems: new results and open problems. In *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*, volume 4, pages 2683–2687. IEEE, 2000.

[276] M. F. Y. Löhr, M. Klauco and M. Mönnigmann. Machine learning assisted solutions of mixed integer MPC on embedded platforms. In *IFAC World Congress 2020, Berlin (accepted)*, 2020.

[277] J. Yu and M. Anitescu. Multidimensional sum-up rounding for integer programming in optimal experimental design. *Mathematical Programming*, pages 1–40, 2019.

[278] C. Zeile, E. Scholz, and S. Sager. A simplified 2D heart model of the Wolff-Parkinson-White syndrome. In *Proceedings of the Foundations of Systems Biology in Engineering (FOSBE) Conference*, volume 49, pages 26–31. Magdeburg, Germany, Elsevier, 2016.

[279] C. Zeile, T. Rauwolf, A. Schmeisser, T. Weber, and S. Sager. The influence of right ventricular afterload in cardiac resynchronization therapy: A circadapt study. In *Computing in Cardiology 2017 -PapersOnLine Proceedings*, 2017.

[280] C. Zeile, T. Weber, and S. Sager. Combinatorial integral approximation decompositions for mixed-integer optimal control. Technical report, 2018. (preprint available under `http://www.optimization-online.org/DB_HTML/2018/02/6472.html`).

[281] C. Zeile, T. Rauwolf, A. Schmeisser, J. Mizerski, R. C. Braun-Dullaeus, and S. Sager. A personalized switched systems approach for the optimal control of ventricular assist devices based on atrioventricular plane displacement. *IEEE Transactions on Biomedical Engineering*, 2020. submitted.

[282] C. Zeile, N. Robuschi, and S. Sager. Mixed-integer optimal control under minimum dwell time constraints. *Mathematical Programming*, pages 1–42, 2020. doi: https://doi.org/10.1007/s10107-020-01533-x.

[283] J. Zhai, T. Niu, J. Ye, and E. Feng. Optimal control of nonlinear switched system with mixed constraints and its parallel optimization algorithm. *Nonlinear Analysis: Hybrid Systems*, 25:21–40, 2017.

[284] P. Zhao, S. Mohan, and R. Vasudevan. Optimal control for nonlinear hybrid systems via convex relaxations. *arXiv preprint arXiv:1702.04310*, 2017.

[285] F. Zhu and P. J. Antsaklis. Optimal control of hybrid switched systems: A brief survey. *Discrete Event Dynamic Systems*, 25(3):345–364, 2015.

Notation and nomenclature

Here, we provide an overview of the notation and nomenclature used in this thesis.

We use the blackboard symbols $\mathbb{N}, \mathbb{Z}, \mathbb{R}$ to refer to the set of natural numbers excluding zero, the set of all integer numbers, and the set of real numbers, respectively. In this thesis, vectors are usually described with boldface letters, e.g., $\boldsymbol{v} \in \mathbb{R}^n$. We use mostly uppercase calligraphic style for sets, e.g., $\mathscr{A}$, and in most cases uppercase bold letters for matrices, e.g., $\boldsymbol{A} \in \mathbb{R}^{m\times n}$, but write their entries with Latin letters, e.g., $a_{i,j} \in \mathbb{R}$. The ith row and jth column of a matrix $\boldsymbol{A}$ are denoted by $\boldsymbol{a}_{i,\cdot}$ and $\boldsymbol{a}_{\cdot,j}$, respectively. We write $\boldsymbol{A}^\top$ to indicate the transposed matrix $\boldsymbol{A}$.

Let $[n] := \{1,\dots,n\}, [n]_0 := \{0\} \cup [n]$, for $n \in \mathbb{N}$. We use Gauss' bracket notation, i.e., $\lfloor x \rfloor := \max\{k \in \mathbb{Z} \mid k \le x\}$, $x \in \mathbb{R}$, and analogously for $\lceil x \rceil$. We indicate by $\lceil x \rceil_{0.5}$ the rounding up of $x \in \mathbb{R}$ to the next multiple of 0.5:

$$\lceil x \rceil_{0.5} := \min\{y \mid y = n \cdot 0.5,\ n \in \mathbb{N}, y \ge x\}.$$

We omit a thorough definition of computational complexity classes P and NP and refer to standard literature for further details [86]. We use these concepts to guide expectations of the difficulty of problems.

We write "for a.a. $t \in \mathscr{T}$" to abbreviate for all $t \in \mathscr{T} \subset \mathbb{R}$, except on a set of measure zero. Moreover, we write *control* to abbreviate the control realization $\omega_i(\cdot), i \in [n]$, of a control function $\omega(\cdot) = (\omega_1(\cdot),\dots,\omega_n(\cdot))^\mathsf{T}$.

We use the notation $\frac{\mathrm{d}f}{\mathrm{d}x}$ to denote the FRÉCHET derivative, which is a generalization of derivatives to Banach spaces, for a FRÉCHET differentiable function $f : D \subseteq X \to Y$ between normed spaces X, Y. By $\frac{\partial f}{\partial x}$, we denote the partial derivative of f with respect to x. We associate the independent variable of a trajectory $\boldsymbol{x} : \mathscr{T} \subseteq \mathbb{R} \to \mathbb{R}^{n_x}$ with the time t and write the derivative of $\boldsymbol{x}$ with respect to the time as $\dot{\boldsymbol{x}} := \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t}$. We denote the time derivative of a function φ with a one-dimensional codomain by φ'.

In the following, we list the commonly used symbols, variables, and parameters of this thesis.

List of Symbols

$\square$	End of a proof
$\lightning$	Contradiction symbol
$\lceil \cdot \rceil$	Component-wise mapping of a real number to the next largest integer value
$\lfloor \cdot \rfloor$	Component-wise mapping of a real number to the next smallest integer value
$\lceil \cdot \rceil_{0.5}$	Component-wise mapping of a real number to the next largest multiple of 0.5
$\lvert \cdot \rvert$	Component-wise mapping of a real number to the absolute value
$\lVert \cdot \rVert$	(Unspecified) norm of a vector
$\lvert\lVert \cdot \rVert\rvert$	(Unspecified) norm of a matrix
$\{\,\}$	Set delimiters
$\cup$	Set–theoretic union (“unified with”)
$\cap$	Set–theoretic intersection (“intersected with”)
$\subseteq, \subset$	Subset of a set (“is a (proper) subset of”)
$\in, \notin$	Set membership (“is (not) an element of”)
$\emptyset$	The empty set
$\forall$	Universal quantifier (“for all”)
$\exists$	Existential quantifier (“there exist(s)”)
$\vee$	Logical inclusive disjunction (“OR”)
$\wedge$	Logical conjunction disjunction (“AND”)
$\mathbf{0}_{n,m}, \mathbf{0}$	n–by–m matrix of zeros, vector/matrix of zeros with unspecified size
$\mathbf{1}_{n,m}, \mathbf{1}$	n–by–m matrix of ones, vector/matrix of ones with unspecified size

Blackboard Symbols

$\mathbb{N}, \mathbb{N}_0$	Set of natural numbers excluding (including) zero
$\mathbb{Z}$	Set of integer numbers
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	Space of n-vectors with elements from the set $\mathbb{R}$
$\mathbb{R}^{m \times n}$	Space of $m \times n$–matrices with elements from the set $\mathbb{R}$

Calligraphic Symbols

$\mathscr{A}$	Feasible function space for relaxed binary control functions
$\mathscr{A}_N$	Feasible discretized function space for relaxed binary controls
$\mathscr{C}$	Objective cost function of the OCP
$\mathscr{F}, \mathscr{F}_1, \mathscr{F}_2$	Feasibility sets for the extended formulated problem in Section 6.4
$\mathscr{G}_N$	Discretization grid
$\widetilde{\mathscr{G}}_N$	Auxiliary grid for finding the initial active control ((CIA-TV) context)
$\mathscr{H}$	Hamiltonian function
$\mathscr{I}(P)$	Set of all problem instances of problem class (P)
$\mathscr{I}_i^A$	Set of allowed modes that can be activated directly after mode i has been active
$\mathscr{I}_b^D$	Down time forbidden control index set
$\mathscr{I}_k$	Index set for blocking constraints
$\mathscr{I}_j^{\mathrm{SUR}}$	Index set of down time forbidden controls on interval j as part of DSUR.

$\mathscr{I}_{k+1}(C_1)$	Index set of intervals affected by the MDT span C_1
$\mathscr{I}_{\text{path}}$	Index set of path-constrained intervals
$\mathscr{I}_l^{\text{sing}}$	The lth singular arc interval index set
$\mathscr{I}_s^f$	Set of feasible time point indices for switch s
$\mathscr{I}_b$	Index set of intervals for dwell time block b
$\mathscr{I}_k^{\text{SUR}}$	Index set of MDT induced intervals from interval k on for DSUR.
$\tilde{\mathscr{I}}$	Index set of down time-forbidden intervals in the proof of Theorem 7.2
$\mathscr{I}^{i_1}$	Index set of intervals defining $\boldsymbol{a}_{i_1,\cdot}$ in the proof of Theorem 7.6
$\mathscr{L}_b$	Length of dwell time block b
$\overline{\mathscr{L}}, \underline{\mathscr{L}}$	Maximum and minimum over all dwell time block lengths
$\mathscr{N}_j(i)$	Next interval j on which control mode i becomes forced
$\mathscr{P}$	Set of all partitions of the interval $\mathscr{T}$
$\mathscr{S}^v$	Set of all switching time points for integer control $\boldsymbol{v}$
$\mathscr{S}_j$	Index set of binary control variables to be optimized in the jth step of the generalized CIA decomposition
$\mathscr{T}$	Time horizon of an ODE or OCP
$\mathscr{T}_{\text{arc}}$	Time interval of a singular arc
$\mathscr{T}_i^A$	Auxiliary domain in the proof of Proposition 7.5
$\mathscr{U}$	Feasible function space of the control function $\boldsymbol{u}$
$\mathscr{V}$	Feasible function space of the integer control function $\boldsymbol{v}$

Greek Symbols

$\boldsymbol{\alpha}$	Relaxed vector-valued binary control function
β	Penalty factor of the penalty term Φ_{pen} for reducing switches
$\beta_{i,j}$	Auxiliary variables for formulation of the total variation constraints
$\gamma_{i,j}$	Forward control deviation for control i and interval j
$\Gamma_{i,b}$	Forward control deviation for control i and block b
δ	Tolerance or distance parameter
δ_j	Length of the jth activation block (AMDR context)
Δ_j	Length of the jth discretization interval
$\bar{\Delta}, \underline{\Delta}$	Maximum and minimum grid length parameters
$\boldsymbol{\epsilon}$	Vector-valued tolerance parameter
ϵ, ϵ_b	Tolerance or distance parameter
$\zeta_{i,j}$	Auxiliary variables for formulation of the 1-norm as part of MILPs
η	Activation duration between two switches in (STO-CIA)
θ	Objective term of the (CIA) problem
$\theta_{i,j}$	Accumulated control deviation for control mode i and interval j
$\theta(\boldsymbol{w})$	Integer deviation error for binary control $\boldsymbol{w}$
$\theta^{\max}$	Maximum optimal objective value of all (CIA) problem instances

θ^*_{CIA}	Optimal objective value of the (CIA) problem with unspecified norm
θ^*_{SCIA}	Optimal objective value of the (SCIA) problem with unspecified norm
$\theta^*_{\lambda\text{CIA}}$	Optimal objective value of the (λCIA) problem
$\bar{\theta}$	Integer deviation rounding threshold parameter for MDR and AMDR
$\Theta_{i,b}$	Accumulated control deviation for control i and block b
$\boldsymbol{\iota}$	State-dependent condition function for implicit switches
$\kappa_{s,j}$	Extended formulation: Indicator variable for the sth switch on jth grid point
$\boldsymbol{\lambda}$	Adjoint or costate variable function
$\tilde{\lambda}$	Evaluated dual variables of the ODE constraints
μ_i	Auxiliary function in the proof of Theorem 5.1 for the ith control mode
$\nu(\cdot,\cdot)$	Weighted accumulated infeasibility over all intervals of a given state trajectory
Ξ	Switching signal vector of switching times and active modes
ξ_k	Extended formulation: Indicator variable of the kth switch
$\pi(i,k)$	Mapping that indicates whether the ith mode is active in the kth position of Π
Π	Sequence of active controls
$\sigma_{i,\max}$	Number of allowed switches for the ith control mode
$\sigma_{\max}$	Total number of allowed switches
$\sigma_{i,j}$	Auxiliary variables for formulation of the total variation constraints
$\sigma(\boldsymbol{w})$	Number of switches of control function $\boldsymbol{w}$
τ	Time variable in the time transformation setting and integrals.
τ_j	Interval of jth switch (AMDR context)
Φ	Mayer term function as part of the OCP objective
Φ_{pen}	Objective penalty term for reducing switches
Φ^{rec}	Objective value for the (BOCP) for singular arc block heuristic
χ_D	Parameter indicating the presence of MD time constraints
Ψ	Lagrange term function as part of the objective of the control problem
$\boldsymbol{\omega}$	Vector-valued binary control function
Ω	Feasible function space for binary control functions
Ω_{comb}	Feasible binary control function space including combinatorial constraints
Ω_N	Feasible discretized function space for binary control functions with N intervals
Ω^p	Feasible function space for multiphase binary control functions
Ω^b_a	Set of admissible controls for block b
Ω^b_f	Set of future-forced controls for block b

Roman Symbols

$\boldsymbol{a}$	Discretized vector-valued relaxed binary control function represented as a matrix
$\boldsymbol{A}$	Unspecified matrix
b	Index for dwell time blocks
B_i	Activation block for control i (AMDR context)
$\boldsymbol{B}$	Unspecified matrix
$\boldsymbol{c}$	Path constraint function
c_i	Constant parameter as part of the numerical experiments
c_i	Mode-dependent child node in the BnB algorithm

C_1	Minimum dwell time span
C_2	Rounding threshold factor parameter for DNFR
$C_B, \hat{C}_B$	Constants for bounding $\boldsymbol{f}$
C_D	Mode-independent MD time span
$C_{i,D}$	MD time span for control mode i
$C_{i,M}$	Maximum dwell time span for control mode i
C_U	Mode-independent MU time span
$C_{i,U}$	MU time span for control mode i
C_{ub}	Upper bound in path constraint
$C_{i,\max}$	Total maximum up time span for control mode i
C_j	Scheduling context: Completion time of job j
$C(n_\omega)$	Constant that depends on n_ω
$\tilde{C}_1$	Constant that is used to establish bounds on (CIA-TV)
$\boldsymbol{d}$	Mode-dependent mixed control-state constraint function
d	Scheduling context: Due time or deadline
d	BnB context: Depth of the corresponding node in the tree
$db_{i,k}$	Deadline of a block that begins with the kth activation for control i
f	Scheduling context: Job family
$\boldsymbol{f}$	Right-hand side (model) function of the ODE
$\tilde{\boldsymbol{f}}$	Evaluated discretized model function
F^{rec}	Recombination mapping
F_C, F_c	Switching cost functions
$F_{b_1,b}$	Index set of controls activated after dwell time block b_1 that becomes forced until block b
$\boldsymbol{g}_j$	Constraint function for the jth interval
g	Objective function for an extended formulated problem in Section 6.4
i	Index, usually referring to control mode i
i_b^D	Down time forbidden control index
j	Index, usually referring to interval j. Scheduling context: Job
J	Cost-to-go function from the Hamilton-Jacobi-Bellman equation
k	Index; Numerical results section: Parameter
$k_l^{\mathrm{start}}, k_l^{\mathrm{end}}$	First and last interval index of the lth singular arc
$k_l^{(\mathrm{init})}$	Last activation for block $B_{i,l}^{(\mathrm{init})}$ (AMDR context)
K	Decision parameter value for decision problems
l	Index, usually referring to the lth interval
l_b	Last interval index for dwell time block b
L	Constant for the Grönwall lemma
LB	Lower bound variable
m	Index, in Chapter 4 refers to specific MILPs
$m_{i,j}$	Entry in the ith row and jth column of a matrix
M_D	Rounded up number of minimum down time intervals (proof of Theorem 7.2)
n	Node in BnB algorithms
n	Scheduling context: Number of jobs
n_f	Scheduling context: Number of jobs of the fth job family
n_I	Auxiliary term in the proof of Theorem 7.6
no	Variable representing the choice of norm
p	Index, usually referring to phase p; Scheduling context: Processing time

$\boldsymbol{p}_j^c$	Coefficient vector of the collocation polynomial for the jth interval
$\boldsymbol{P}$	Partition of $\mathscr{T}$
P_i	AMDR context: Partition of all activations for control i
P_j^c	Collocation polynomial for the jth discretization interval
Q	Node queue for BnB
r	Auxiliary index; Scheduling context: Release time
$\boldsymbol{R}(\cdot,\cdot)$	Remainder term function of a first-order Taylor approximation
R	Auxiliary term in the proofs in Chapter 7
$\boldsymbol{s}$	Switching function
s	Extended formulation context: Index for the sth switch
s_j	Shooting variable for the jth discretization interval
S	Set of realizations for the switching function $\boldsymbol{s}$
S_{b_1,b_2}	Set of active controls between dwell time blocks b_1 and b_2
$\tilde{S}^{\text{CIA}}, S^{\text{CIA}}$	Set of CIA problems
$\tilde{S}^{\text{CIA}}, S^{\text{REC}}$	Set of recombination mappings
$SV(\boldsymbol{v},\tilde{\mathscr{T}})$	Switching variation of integer control $\boldsymbol{v}$ on interval $\tilde{\mathscr{T}}$
t	Time variable
$\tilde{t}$	Auxiliary term in the proof of Proposition 7.5
t_0, t_f	Start and end times of the time horizon
T	BnB context: Time point index of the last activation
$T_{\max}$	Scheduling context: Maximum tardiness over all jobs
TOL	Tolerance parameter
TSC,SC	Scheduling context: (Total) setup costs
$TV(\boldsymbol{v})$	Total variation of integer control $\boldsymbol{v}$
$\boldsymbol{u}$	Vector-valued continuous control function
UB	Upper bound variable
$\boldsymbol{v}$	Vector-valued integer or binary control function
V	Discrete image space of the integer control function $\boldsymbol{v}$
$\boldsymbol{w}$	Discretized vector-valued binary control function represented as a matrix
$\boldsymbol{x}$	Differential state trajectory/function
x	Optimization variable of the extended formulated problem in Section 6.4
$\boldsymbol{y}$	Alternative differential state trajectory representation
z, z_1, z_2	Integrable functions in Section 5.1; Extended formulation: Switching variable

Dimensions and cardinalities

M	Number of discretization intervals for the differential states (in Section 9.1)
n_b	Number of dwell time blocks
n_c	Number of path constraint functions c_i
n_{CIA}	Number of different CIA problem formulations
n_d	Number of mode-dependent mixed control-state constraint functions d_i
n_{dec}	Number of decompositions applied in the generalized CIA algorithm
n_i	Number of possible activations for control mode i
n_{ivl}	Number of unfixed intervals in the complexity reduction Algorithm 6.1

n_p	Number of control problem phases
n_s	Dimension of switching function realization $\boldsymbol{s}_i$
n_{sing}	Number of singular arcs
n_u	Number of continuous control functions
n_v	Dimension of integer control realization $\boldsymbol{v}_i$
n_{x}	Number of differential states
n_ι	Number of switching function realizations
n_ω	Number of binary control functions or number of integer control realizations
n_σ	Number of switches or switching events
$nb_{i,\min}^{(\text{init})}$	Cardinality of the partition $P_{i,\min}^{(\text{init})}$ for control i
N	Number of (control) discretization intervals
$\tilde{N}$	Number of intervals of $\widetilde{\mathscr{G}}_N$

HEV application

b_{s}	Battery state-of-charge
$F_{\text{a}}, F_{\text{r}}, F_{\text{t}}$	Aerodynamic, rolling, and traction force
G	Gear choice
m_{c}	Mode control function
m_{eq}	Equivalent mass of the vehicle
m_{f}	Fuel mass flow
S_c^e, S_c^h	Set of the dwell time-coupled control indices
T_{w}	ICE cooling water temperature
$U_{S_c^e}, U_{S_c^h}$	MU times for electric and hybrid mode
$v(t)$	Velocity at time t
η	Energy conversion efficiency

LVAD application

$A_\cdot$	Cross-section parameter of specific compartment “·”
$C_\cdot$	Compliance parameter of specific compartment “·”
$F_\cdot$	Contraction force during phase “·”
k_{RAD}	Radial function coefficient of the contraction force

$L_{\cdot}$	Inertance parameter of specific compartment "·"
n_m	Number of available measurement time points
$\boldsymbol{p}$	Vector of parameters to be estimated for patient specification
$P_{\cdot}$	Pressure state function of specific compartment "·"
$Q_{\cdot}$	Flow state function of specific compartment "·"
$R_{\cdot}$	Resistance parameter of specific compartment "·"
s	Position function of the AVP
S_D	Switching threshold parameter for the AVPD
t_1, t_2, t_3	Controlled switching time points
v	Velocity function of the AVP
β	Pump-to-pressure coefficient
ϱ_i	Scaling or weighting parameters
τ_i	Switching time points of implicit switches
φ	A priori information function for the PE

List of Figures

List of Tables

List of Acronyms

AMDR	adaptive maximum dwell rounding
AVP	atrioventricular plane
AVPD	atrioventricular plane displacement
BnB	branch-and-bound
CIA	combinatorial integral approximation
DAE	differential-algebraic equation
DNFR	dwell time next-forced rounding
DSUR	dwell time sum-up rounding
EM	electric motor
EM2	second electric motor
EMS	energy management strategy
FD	final drive
HEV	hybrid electric vehicle
ICE	internal combustion engine
IVP	initial value problem
LA	left atrium
LV	left ventricle
LVAD	left ventricular assist device
MD	minimum down
MDR	maximum dwell rounding
MDT	minimum dwell time
MILP	mixed-integer linear program
MINLP	mixed-integer nonlinear program
MIOC	mixed-integer optimal control
MIOCP	mixed-integer optimal control problem
MIQP	mixed-integer quadratic program
MPC	model predictive control
MU	minimum up
NFR	next-forced rounding
NLP	nonlinear program
OCP	optimal control problem
ODE	ordinary differential equation
PDE	partial differential equation
PE	parameter estimation
pwc	piecewise constant
SUR	sum-up rounding
TV	total variation
WLTP	world harmonized light-duty vehicles test procedure