Solution of Optimal Control Problems for Switched Systems

Algorithms and Applications for Hybrid Vehicles

Der Fakultät für Informatik und Elektrotechnik
der Universität Rostock
zur Erlangung des akademischen Grades eines
Doktor-Ingenieur (Dr.-Ing.)

eingereichte Dissertation

von
Herr M.Sc. Markus Sebastian Schori
aus Braunschweig
geb. am 19.02.1987 in Bonn

Datum der Einreichung: 13.04.2015

1. Gutachter: Prof. Dr.-Ing. Torsten Jeinsch[1]
2. Gutachter: Prof. Dr.-Ing. habil Dr. h.c. Bernhard Lampe[1]
3. Gutachter: Prof. Dr.-Ing. habil Ping Zhang[2]

Tag der mündlichen Prüfung: 18.12.2015

[1] Institut für Automatisierungstechnik, Universität Rostock
[2] Lehrstuhl für Automatisierungstechnik, Technische Universität Kaiserslautern

urn:nbn:de:gbv:28-diss2016-0004-8

Eidesstattliche Versicherung

Ich erkläre, dass ich die eingereichte Dissertation selbstständig und ohne fremde Hilfe verfasst, andere als die von mir angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Werken wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Markus Schori
Braunschweig, den 8. Januar 2016

Abstract

Switched systems are a subclass of hybrid systems that allow for the direct control of the switching instants and the choice of the active subsystem or mode at any time. This thesis deals with the solution of optimal control problems for switched systems. Solutions to this kind of hybrid optimal control problem include the optimal continuous control inputs as well as optimal discrete decisions. In the first part, practical algorithms for the solution are evaluated in terms of performance and robustness. The methods evaluated include existing algorithms such as dynamic programming as well as self-developed algorithms. The indirect variation of extremals appraoch is expanded for the use in switched systems. The suitability of this approach is limited, as it is highly sensitive to the guess of the initial costate. However, in some cases, the method can be fast and reliable. A method that circumvents this difficulty by combining direct and indirect approaches is proposed as well. The algorithm is rather slow but very robust and insensitive to the initial guesses of continuous control and switching sequence. Additionally, a method based on embedding is proposed for switched systems with state jumps. An interesting application of switched systems in the automotive industry are nowadays hybrid vehicles. In the second part of this thesis, hybrid vehicles are modeled as switched systems and switched optimal control problems are formulated for the optimal operation of a hybrid vehicle over a given cycle. It is then evaluated, which method is appropriate for the solution of these hybrid optimal control problems for different model-depths. A method that allows for obtaining calibration parameters for rule-based energy managements directly from hybrid optimal control theory is proposed as well as a predictive energy management that is based on the solution of a hybrid optimal control problem during the operation of the vehicle.

Hybrid Systems, Switched Systems, Optimal Control, Hybrid vehicles, Energy Management

Zusammenfassung

Geschaltete Systeme sind eine Unterklasse der hybriden Systeme, bei denen es möglich ist, die Schaltzeiten und das aktive Subsystem bzw. den aktiven Modus direkt und zu jeder Zeit zu bestimmen. Diese Arbeit behandelt die Lösung von Optimalsteuerungsproblemen für geschaltete Systeme. Die Lösung eines solchen Optimalsteuerungsproblemes beinhaltet sowohl die optimale kontinuierliche Steuerung, als auch die optimalen diskreten Entscheidungen. Im ersten Teil werden praxisnahe Algorithmen zur Lösung von Optimalsteuerungsproblemen dieser Art in Bezug auf Leistungsfähigkeit und Robustheit bewertet. Die bewerteten Methoden beinhalten dabei ebenso Methoden, die bereits aus der Literatur bekannt sind, wie neuartige Methoden. Das indirekte Schießverfahren wird auf geschaltete Systeme erweitert. Die Einsetzbarkeit dieser Methode ist jedoch eingeschränkt, da die Konvergenz des Verfahrens sehr stark von der Schätzung des Anfangswertes der Adjungierten abhängt. In einigen Fällen ist diese Methode jedoch robust und schnell. Ein Verfahren, das diese Schwachstelle umgeht, indem direkte und indirekte Lösungsverfahren kombiniert werden, wird ebenfalls vorgeschlagen. Zusätzlich wird eine Methode basierend auf der Relaxierung der diskreten Systemanteile sowie der Zustandssprünge beschrieben. Ein interessanter Anwendungsfall von geschalteten Systemen in der Automobilindustrie sind heutige Hybridfahrzeuge. Im zweiten Teil dieser Arbeit werden Hybridfahrzeuge als geschaltete Systeme modelliert und geschaltete Optimalsteuerungsprobleme für den optimalen Betrieb auf vorgegebenen Zyklen formuliert. Es wird dann bewertet, welche Algorithmen für die Lösung von hybriden Optimalsteuerungsproblemen mit unterschiedlichen Modelltiefen geeignet sind. Eine Methode, mit der Applikationsparameter für regelbasierte Betriebsstrategien direkt aus der Lösung des hybriden Optimalsteuerungsproblem erhalten werden können, wird ebenso vorgeschlagen, wie eine Betriebsstrategie, die auf der Lösung eines hybriden Optimalsteuerungsproblems im Fahrbetrieb basiert.

Hybride Systeme, Geschaltete Systeme, Optimale Steuerung, Hybridfahrzeuge, Betriebsstrategie

Acknowledgements

Without the contribution of many people, this work would not have been possible. My sincere thanks go to Dr. Thomas Böehme of the IAV Automotive Engineering for the intensive and continuous counseling in scientific matters as well as in discussions on the subject. I must also thank Professor Torsten Jeinsch and Professor Bernhard Lampe as well as Dr. Wolfgang Drewelow of the University of Rostock for their scientific guidance and the possibility to work partially at the Institute of Automation. I really enjoyed the supportive atmosphere during my time there and the invaluable advice I received. I am also deeply grateful to Matthias Schultalbers and Nick Weinhold and the IAV Automotive Engineering, Department of Mechatronic Gasoline Systems, for allowing me to perform my research work and for providing the necessary resources. Finally my thanks go to further fellows who contributed to this thesis, namely Benjamin Frank, Tobias Österwind, Martin Rucht, Frank Wobbe, Matthias Menzel, Ismail Cakar, Peter Michau and to the entire control group at IAV.

Contents

1 Introduction 1
1.1 Contribution and Previous Publications 5

2 Hybrid Systems and Hybrid Optimal Control 9
2.1 System Definition 9
2.2 Switched Systems 11
2.3 Hybrid Execution and Switching Sequence 12
2.4 Example 13
2.5 Switched Optimal Control Problems 13
2.6 Necessary Conditions for Switched Optimal Control 16
2.6.1 Further Assumptions 16
2.6.2 Conditions for Discontinuous States 17
2.6.3 Conditions for Continuous States 20
2.6.4 State Constraints 21

3 Algorithms for Solving Hybrid Optimal Control Problems 23
3.1 Overview and State-of-the-Art 23
3.1.1 Multi-Stage Decision Processes 24
3.1.2 Relaxation Methods 25
3.1.3 Two-Stage Approaches 26
3.2 Discretization 28
3.2.1 Runge-Kutta Discretization 28
3.2.2 Explicit Euler Discretization 30
3.3 Direct Solution with Fixed Switching 31
3.3.1 Nonlinear Programming 31
3.3.2 Direct Shooting 32

3.3.3 Collocation 33
3.3.4 Comparison of Direct Shooting and Collocation 33
3.3.5 Convergence Results 34
3.3.6 Recovering the Costate from a Direct Shooting 35
3.3.7 Recovering the Costate from Collocation 37
3.4 Indirect Solution with Fixed Switching 39
3.5 Algorithms for Systems with Continuous States 40
3.5.1 Embedding 40
3.5.2 Indirect Shooting 42
3.5.3 Combining Direct and Indirect Methods 44
3.6 Algorithms for Systems with Discontinuous States 49
3.6.1 Dynamic Programming 49
3.6.2 Embedding with Relaxation of the State Jump 53
3.7 Examples 59
3.7.1 Continuous State without Final State Constraints 59
3.7.2 Continuous State with Final State Constraints 62
3.7.3 Discontinuous State 64

4 Hybrid Vehicles 67
4.1 Modeling Hybrid Vehicles as Switched Systems 68
4.1.1 Quasi-Steady Model 70
4.1.2 Thermodynamic Model 75
4.2 Solution of SOCPs for Hybrid Vehicles 83
4.2.1 Problem Formulations 84
4.2.2 Evaluation of Necessary Conditions 88
4.2.3 Solving the SOCPs 90
4.3 Causal Energy Management of Hybrid Vehicles 97
4.3.1 Calculation of Look-up-Tables 98
4.3.2 Evaluation and Implementation 101
4.4 A Predictive Energy Management 105
4.4.1 Overview of the Control Strategy 106
4.4.2 Driver Model 107
4.4.3 Solving the SOCP 110
4.4.4 State Constraints 111
4.4.5 Controller and Instantaneous Controls 111

4.4.6 Implementation and Results 113

5 Conclusion and Outlook **117**

5.1 Conclusion 117

5.2 Outlook 119

A Appendix **121**

A.1 Dynamic Programming for Purely Continuous Systems 121

A.2 Smooth Function Representation 125

A.2.1 Smooth Surface Generation 125

A.2.2 Analytical Representation 127

Chapter 1

Introduction

Hybrid Systems occur naturally in many technical applications as well as in applications from natural sciences as biology or chemistry. Whenever a system has continuous control inputs, but at the same time can make discrete decisions or switch between different subsystems, the system can be modeled as a hybrid system. A simple example of a discrete decision may be the ON/OFF decision of some heating system. More complex decisions are gear choices or different modes of operation of the internal combustion engine in automotive applications. It is due to this frequent appearance in daily-life that the research interest in hybrid systems is steadily growing. In this thesis, the focus is on switched systems, a subclass of hybrid systems that switch between subsystems or modes only in response to a command. This subclass covers a great range of technical problems. Early works on hybrid systems were already performed by Witsenhausen [1966] and Seidman [1987] but especially over the last two decades, significant progress has been achieved in the field of hybrid system analysis. Among others, the fields of research include the modeling (Branicky et al. [1994]) and stability of hybrid systems (Xu [2001], Lygeros et al. [2003], Goebel et al. [2009]). However, the main focus in the literature is on optimal control. Hybrid systems are usually controlled by a continuous control law and a superimposed discrete automaton. It is often of major interest, to define a control law for the continuous control as well as for the discrete automaton, that leads to a desired behavior. Engineers and Scientists often strive for minimizing a given cost function that can resemble time, energy, money or their respective combinations. The task of finding the controls that achieve this is denominated as optimal control problem, or more specifically, as hybrid optimal control problem, when the underlying system is a hybrid system. The research efforts in this area include the theoretical analysis of hybrid optimal control problems to derive necessary conditions for optimal control (Seidman [1987], Sussmann [1999], Piccoli [1989], Shaikh [2004], Riedinger et al. [1999], Riedinger et al. [2003], Dmitruk and Kaganovich [2008], Passenberg et al. [2011]). In chapter 2, a general model description for hybrid systems and for switched systems is given and an overview of established necessary conditions for switched optimal control is given. Many approaches for solving switched optimal control problems are based on these conditions. Algorithmic development for hybrid systems was treated, among others, in the works of Hedlund and Rantzer [2002], Bengea and DeCarlo [2003], Alamir and Attia [2004], Shaikh [2004], Sager [2005] and Axelsson et al. [2008]. Even though optimal control of non-hybrid systems is well-researched and many powerful algorithms exist, the methods cannot readily be transferred to the hybrid context. Likewise, the well established methods for discrete optimization are not suitable, when the discrete decisions interact with continuous controls. The algorithmic development in this thesis concentrates on problems formulated for switched systems (switched optimal control problems), where the switching between discrete states can be triggered by external commands and no autonomous switching is present. Further, the focus is on switched optimal control problems that exhibit a strong time-dependent component, which is often the case for automotive applications due to the fact that environmental conditions can vary strongly depending on the current vehicle position. In chapter 3, existing algorithmic approaches are regarded in terms of suitability for technical problems. Self-developed algorithms are presented, that include indirect approaches (approaches that are based on necessary conditions for hybrid optimal control), direct methods (methods that reduce the continuous control problem to a nonlinear optimization problem) and combined methods. Each algorithm is suitable for problems of different complexity. The results are compared to results obtained with existing algorithms. Even though in general no implementable control law is obtained from the solution of a hybrid optimal control problem, the solution can be a first step to define controller and automaton parameters.

A practical automotive application of hybrid optimal control theory constitute nowadays hybrid vehicles. The term "hybrid" in "hybrid vehicles" does not refer to the existence of continuous and discrete dynamics but to the fact, that the system has more than one energy storage and energy converter. In most cases, the additional energy storage will be a high-voltage battery and the converter an electrical motor/generator. This adds new degrees of freedom in the control of the powertrain, such as shifting of the operation points of the converters or different drive modes. The newly gained degrees of freedom can be controlled to improve the overall system efficiency. The improvement of efficiency and the corresponding reduction of fuel consumption is a timely topic, since the reduction of crude-oil-use and green-house gas emissions is one of the major challenges of our time. Accordingly, a growing research area has been established, that focuses on optimal control of hybrid vehicles. Among many others, these works were performed by Paganelli et al. [2002], Guzzella and Sciarretta [2005], Liu and Peng [2006], Stockar et al. [2011] and Sivertsson et al. [2011]. However, the determination of optimal controls is complicated by the fact, that the system inputs include continuous controls as well as several discrete decisions and because of the strong interaction between these inputs, a separate optimization will yield inferior results than a combined approach. Consequently, a hybrid vehicle can be modeled as a hybrid system in the mathematical sense, which is seldomly regarded in the literature. The calibration process for hybrid vehicles can be a cumbersome task if no systematic calibration approach is applied. Fortunately, the progress in hybrid optimal control theory allows for the mathematical derivation of calibration parameters as well as for the development of new functional approaches for improving the vehicle's performance. In chapter 4, two models of different depth for a parallel hybrid vehicle configuration are formulated as hybrid systems. The first one includes representations of the electrical and the mechanical subsystem, whereas the second model also incorporates a detailed thermodynamic model of the internal combustion engine and the exhaust system as well as an emissions model. Switched optimal control problems are then formulated in chapter 4.2 for these models and the algorithms from chapter 3 are applied.

The solution can be helpful for the calibration process of hybrid vehicles, as it can serve as a rough guideline for a specific cycle. Yet, the step of transferring the solution into calibration parameters for the electronic control unit is non-trivial. It is shown in chapter 4.3, that look-up tables for rule-based energy-managements can be derived directly from the problem solution. The theory of hybrid optimal control also suggests new functional approaches. As such, a predictive energy management is described in chapter 4.4 that makes the solution of a switched optimal control problem amenable for online-implementation in the electronic control unit. This is only possible, when a profile of the driving route is known to the electronic control unit. A prediction based on data from modern navigation systems is made to obtain an estimation of this profile. Using hybrid optimal control theory for calibration and function development for hybrid vehicles significantly reduces the time required for a high-quality calibration and has led to significant fuel-savings, as is shown by simulations as well as real-world measurements.

1.1 Contribution and Previous Publications

To the best knowledge of the author, this thesis contains the following original contributions:

- The already established results on the relationship between Lagrange multipliers and costate trajectory for direct shooting and collocation are extended to switched systems. It is demonstrated that the approximated costate also includes an approximation of the jump condition stated in the necessary conditions for switched optimal control

- A two-stage approach that solves a continuous optimal control problem with fixed switching schedule using a direct method. The Lagrange multipliers returned by the nonlinear program solver are then used to approximate the costate and to evaluate necessary conditions of optimality. The switching sequence is altered, based on an achievable descent in the Hamiltonian function at a time instant

- The application of this method to the complex optimal control problem of optimal catalytic converter heat-up. A detailed model of a hybrid vehicle's powertrain including thermal aspects is used for the formulation of a switched optimal control problem with a high number of states. It is demonstrated that the two-stage approach can efficiently solve this demanding problem

- The general embedding principle that relaxes the discrete state to a continuous control variable is enhanced for systems with state jumps. The jump function is reformulated such that it is continuous but not continuously differentiable everywhere and hence, the switched optimal control problem with discontinuous state trajectory is transcribed to a nonsmooth nonlinear program

- A theoretical justification for the derivation of look-up tables for the energy management of hybrid vehicles by minimization of the Hamiltonian function with respect to discrete variables. The results were derived in the literature using the concept of equivalent fuel consumption minimization. In this monograph, the derivations are based on necessary conditions for optimality of a switched system and the validity of these conditions is discussed

- The efficient implementation of an indirect shooting algorithm for the solution of a switched optimal control problem that aims at the fuel-optimal operation of a hybrid vehicle over a given cycle. Based on this algorithm, a predictive energy management feasible for implementation on common electronic control units taking into account continuous as well as discrete decisions is developed and tested.

Most of the results in this thesis were published previously in peer-reviewed journal articles or conference proceedings. The following publications contain material that can also be found in this thesis. The use of the indirect shooting method as it is described in 3.5.2 for the solution of switched optimal control problems was presented in

M. Schori, T.J. Boehme, B. Frank, and M. Schultalbers. Solution of a hybrid optimal control problem for a parallel hybrid vehicle. In *7th IFAC Symposium on Advances in Automotive Control (AAC), Tokyo*, pages 109–114, 2013b.

The use of Pontryagin's Minimum Principle for hybrid systems for the calculation of ready-to-implement look-up tables (section 4.3) was published in

M. Schori, T.J. Boehme, B. Frank, and M. Schultalbers. Calibration of parallel hybrid vehicles based on hybrid optimal control theory. In *Proceedings of the 9th IFAC Symposium on Nonlinear Control Systems, Toulouse, France*, pages 475–480, 2013c.

The method combining indirect and direct methods for the solution of switched optimal control problems (section 3.5.3) was explained in the journal article

M. Schori, T.J. Boehme, U. Becker, and M. Schultalbers. Verfahren zur Lösung von hybriden Optimalsteuerungsproblemen und deren Anwendung auf den Betrieb von Hybridfahrzeugen. *at Automatisierungstechnik*, 61(12):831–840, 2013a.

The use of the combined method to solve a complex switched optimal control problem to find optimal controls for the heating process of a three-way-catalyst (section 4.2.3) is described in

M. Schori, T.J. Boehme, T. Jeinsch, and M. Schultalbers. Optimal catalytic converter heating in hybrid vehicles. In *SAE Technical Paper 2014-01-1351*, 2014b.

The embedding method with relaxation of the jump function to approximate a solution to switched optimal control problems with state jumps as it is described in section 3.6.2 was proposed in

M. Schori, T.J. Boehme, B. Frank, and M. Schultalbers. Control optimization of discontinuous hybrid systems using embedding. In *Preprints of the 12th International Workshop on Discrete Event Systems*, pages 326–331, 2014a.

An overview of the application of switched optimal control theory to hybrid vehicles and the derivation of calibration parameters as well as a first description of the predictive energy management in section 4.4 is given in

M. Schori, T.J. Boehme, B. Frank, and B. Lampe. Optimal calibration of map-based energy management for plug-in parallel hybrid configurations: a hybrid optimal control approach. *IEEE Transactions on Vehicular Technology*, 2015, accepted for publication.

A more detailed description of the predictice energy management was published in

M. Schori, T.J. Boehme, T. Jeinsch, and M. Schultalbers. A robust predictive energy management for plug-in hybrid vehicles based on hybrid optimal control theory. In *Proceedings of the American Control Conference, accepted for publication*, 2015b.

The derivation of the costate from the solution of a switched optimal control problem with fixed switching sequence with direct methods (section 3.3.6) is given in

M. Schori, T.J. Boehme, T. Jeinsch, and B. Lampe. Costate approximation from direct methods for switched systems with state jumps. In *Proceedings of the European Control Conference, accepted for publication*, 2015a.

Chapter 2

Hybrid Systems and Hybrid Optimal Control

2.1 System Definition

Hybrid systems are a class of dynamical systems that somehow incorporate behaviors typical for continuous dynamical systems and for discrete event systems. As an example may serve an electrical circuit where current and voltage can change continuously over time but can also change discontinuously when a switch is opened or closed (Goebel et al. [2009]). Both system classes have their specific ways of representation, such as discrete automata for discrete event systems and differential equations for purely continuous systems. Many physical systems exhibit both kinds of dynamical behavior and the respective model needs to incorporate both types of dynamics (Riedinger et al. [2003]).

Many different notations for describing hybrid systems are explained in the literature. A wide overview of the different modeling approaches is given in Branicky et al. [1994] and Branicky et al. [1998]. In this thesis, a general model description is used, as proposed in Riedinger et al. [1999] and Riedinger et al. [2003].

Hybrid systems are a generalization of conventional systems, that can be modeled by a single differential equation

$$\dot{x} = f(x(t), u(t), t). \tag{2.1}$$

A discrete state $q(t) \in \mathcal{Q} = \{1, \ldots, Q\}$ is added to the system description, that defines the activity of a subsystem or mode. In the literature, q is also often referred to as location. The overall system shares a common state vector $x(t) \in \mathbb{R}^n$ but the vector field, that governs the state's evolution, depends on the discrete state. For each discrete state $q \in \mathcal{Q}$, one vector field $f_q : \mathbb{R}^{n+m} \times [t_0, t_f] \to \mathbb{R}^n$ is defined. The system switches between these vector fields, depending on the current value of the discrete state $q(t)$. The overall system's differential equation can then be written as

$$\dot{x} = f_{q(t)}(x(t), u(t), t). \tag{2.2}$$

Herein, $u(t) \in \mathbb{R}^m$ is the continuous control input. In most cases, u will be subject to constraints and hence, it is assumed that u may take on values from the convex set $\mathcal{U}_q \subset \mathbb{R}^m$. Throughout the thesis, the initial time will be set as $t_0 = 0$ and the initial state $x(t_0) = x_0$ will be assumed as given. The times, where the discrete state q, and therefore the right-hand-side term in (2.2), changes, are denominated as t_j^- and t_j^+, where the former refers to the time right before a change and the latter to the time right after a change. The subscript j enumerates the number of switching and the total number of discontinuous changes in q is denoted as l. A system, starting at the initial hybrid state $(x(t_0), q(t_0))$ will then evolve according to the differential equation $\dot{x} = f_{q(t_0)}(x(t), u(t), t)$, until a switching occurs at a switching time t_1. The switching is induced by a transition in the piecewise constant function $q(t)$. This transition is modeled using the function $\Pi : \mathbb{R}^n \times \mathcal{Q} \times \mathcal{D} \times [t_0, t_f] \to \mathcal{Q}$:

$$q(t_j^+) = \Pi(x(t_j^-), q(t_j^-), d(t_j^-), t_j), \tag{2.3}$$

that captures two different kinds of characteristic behavior in hybrid systems: Autonomous switching and controlled switching. On autonomous switching, the discrete state q changes, when the state encounters a switching manifold of the form $C_{(q^-,q^+)}(x(t), t) = 0$. On controlled switching, q changes discontinuously due to a commanded change in the discrete control variable $d \in \mathcal{D} = \{1, \ldots, D\}$ (Shaikh [2004]).

Often, the state $x(t)$ is absolutely continuous. In many cases, however, the state may exhibit discontinuities, when a switching occurs. These discontinuities can be modeled with the help of Q^2 jump functions $\delta_{(q^-,q^+)} : \mathbb{R}^n \times [t_0, t_f] \to \mathbb{R}^n$:

$$x(t_j^+) = x(t_j^-) + \delta_{(q^-,q^+)}(x(t_j^-), t_j). \tag{2.4}$$

For $q^- = q^+$, $\delta_{(q^-,q^+)}(x(t_j^-), t_j) = 0$ holds. For a given time t_j and a state $x(t_j^-)$, the function values $\delta_{(q^-,q^+)}$ that resemble the height of a jump $\Delta x_j = x(t_j^+) - x(t_j^-)$ for a transition from one discrete state to another can be arranged in a table:

	1	**2**	$\cdots$	$\boldsymbol{Q}$
1	0	$\delta_{(1,2)}(x,t)$	$\cdots$	$\delta_{(1,Q)}(x,t)$
2	$\delta_{(2,1)}(x,t)$	0	$\cdots$	$\delta_{(2,Q)}(x,t)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$\boldsymbol{Q}$	$\delta_{(Q,1)}(x,t)$	$\delta_{(Q,2)}(x,t)$	$\cdots$	0

Table 2.1: Height of the jump Δx as given by the function values $\delta_{(q^-,q^+)}$ for a given state x at time t arranged in a table

A sketch of the discrete phenomena controlled switching, autonomous switching and state jump is depicted in Fig. 2.1. Controlled state jumps, where the height of the jump is a control variable, are also reported in the literature but will not be covered in this thesis (see for instance S.A. Attia [2007]).

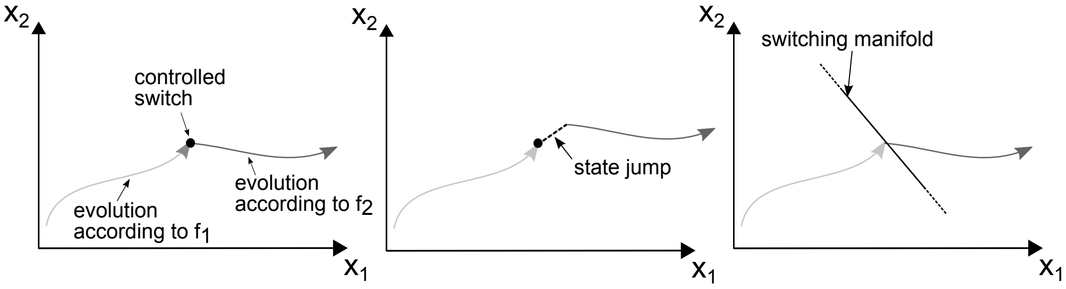

Figure 2.1: Hybrid phenomena, from left to right: Controlled switching, state jump and autonomous switching (Passenberg [2012])

2.2 Switched Systems

Of particular interest in this thesis are systems, that exhibit controlled switching only. In this case, the discrete state $q(t)$ can be chosen freely at any time from the set $\mathcal{Q}$. In some cases, a reformulation of the problem can be helpful: The Boolean vector $\sigma(t) \in \{0,1\}^Q$ is introduced that defines which discrete state q is active at time t. To assure that only one discrete state is active at any time, the constraint

$$\sum_{q=1}^{Q} \sigma_q(t) = 1, \quad \forall t \in [t_0, t_f], \tag{2.5}$$

where σ_q refers to the q-th entry in the vector σ, must be fulfilled and is added to the system description. The system's differential equation $F : \mathbb{R}^n \times \{0,1\}^Q \times \mathcal{U}_q \times [t_0, t_f] \to \mathbb{R}^n$ can then be written (Xu [2001]) as

$$\dot{x} = F(x(t), \sigma(t), u(t), t) = \sum_{q=1}^{Q} \sigma_q \cdot f_q(x(t), u(t), t). \tag{2.6}$$

With an enhanced control vector $\rho \in \mathcal{U}_q \times \{0,1\}^Q$, that contains the continuous as well as the discrete decisions,

$$\rho(t) = [u(t), \sigma(t)], \tag{2.7}$$

the system can be written in the conventional form

$$\dot{x} = F(x(t), \rho(t), t). \tag{2.8}$$

However, it should be noted, that the set of feasible controls $\rho(t)$ is no longer convex.

2.3 Hybrid Execution and Switching Sequence

Definition 2.1: An execution of a switched system is defined by the tuple $(\mathcal{T}, \mathcal{D}, \mathcal{X}, \mathcal{Z})$. Herein

- $\mathcal{T} = (t_0, t_1, \ldots, t_l, t_f)$ is a strictly increasing sequence of switching times including the initial and final time

- $\mathcal{D} = (q_0, \ldots, q_l)$ is a sequence of discrete states, such that $q(t) = q_j$, $\forall\, t \in [t_j, t_{j+1})$

- $\mathcal{X} = (\mathcal{X}_0, \ldots, \mathcal{X}_l)$ is a sequence of state trajectories that evolve according to

  $$\dot{x} = f_{q_j}(x(t), u(t), t) \tag{2.9}$$

  in the interval $[t_j, t_{j+1})$ and fulfill the state jump condition $x(t_j^+) = x(t_j^-) + \delta_{(q^-,q^+)}(x(t_j^-), t_j), j = 1, \ldots, l$

- $\mathcal{Z} = (\mathcal{Z}_0, \dots, \mathcal{Z}_l)$ is a sequence of control trajectories that satisfy $\mathcal{Z}_j \times [t_j, t_{j+1}) \rightarrow \mathcal{U}_{q_j}$

Definition 2.2: A switching sequence is defined as the tuple

$$\Theta = ((t_0, q_0)), (t_1, q_1), \dots, (t_l, q_l)), \tag{2.10}$$

where the pairs t_j and q_j are taken pairwise from $\mathcal{T}$ and $\mathcal{D}$ in Definition 2.1.

2.4 Example

Given the two vector fields $f_1(x)$ and $f_2(x)$

$$f_1(x) = \begin{bmatrix} -3 \cdot x_1 + 0.2 \cdot x_2 \\ 2 \cdot x_1 - 0.5 \cdot x_2 \end{bmatrix} \tag{2.11}$$

$$f_2(x) = \begin{bmatrix} -3 \cdot (x_1 + 1) + 0.2 \cdot (x_2 + 1) \\ 2 \cdot (x_1 + 1) - 0.5 \cdot (x_2 + 1) \end{bmatrix}, \tag{2.12}$$

a hybrid execution of $\dot{x} = f_{q(t)}(x(t))$ is depicted in Fig. 2.2. Starting at $x = [-1, -1]^T$, the state evolves to the stationary point $x = [0, 0]^T$ of f_1. At $t_1 = 10s$, the discrete state switches from $q(t_1^-) = 1$ to $q(t_1^+) = 2$ in response to a commanded switch and consequently, the active vector field switches from $f_1(x)$ to $f_2(x)$. No state-jump is in this case implied with the switching ($\delta_{(q^-, q^+)} = 0 \quad \forall q^-, q^+ \in \mathcal{Q}$) and therefore the state $x(t)$ remains continuous but clearly not continuously differentiable at t_1. After the switching, the state evolves to the stationary point $x = [-1, -1]^T$ of f_2.

2.5 Switched Optimal Control Problems

A switched optimal control problem (SOCP) can be formulated as

$$\min_{u(t) \in \mathcal{U}_q, q(t) \in \mathcal{Q}} \phi(x(t_f)), \tag{2.13}$$

where $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ is a cost function of Mayer type, subject to the system description (2.2) and (2.4) and the final state constraint $\psi : \mathbb{R}^n \rightarrow \mathbb{R}^n$

$$\psi(x(t_f)) = 0. \tag{2.14}$$

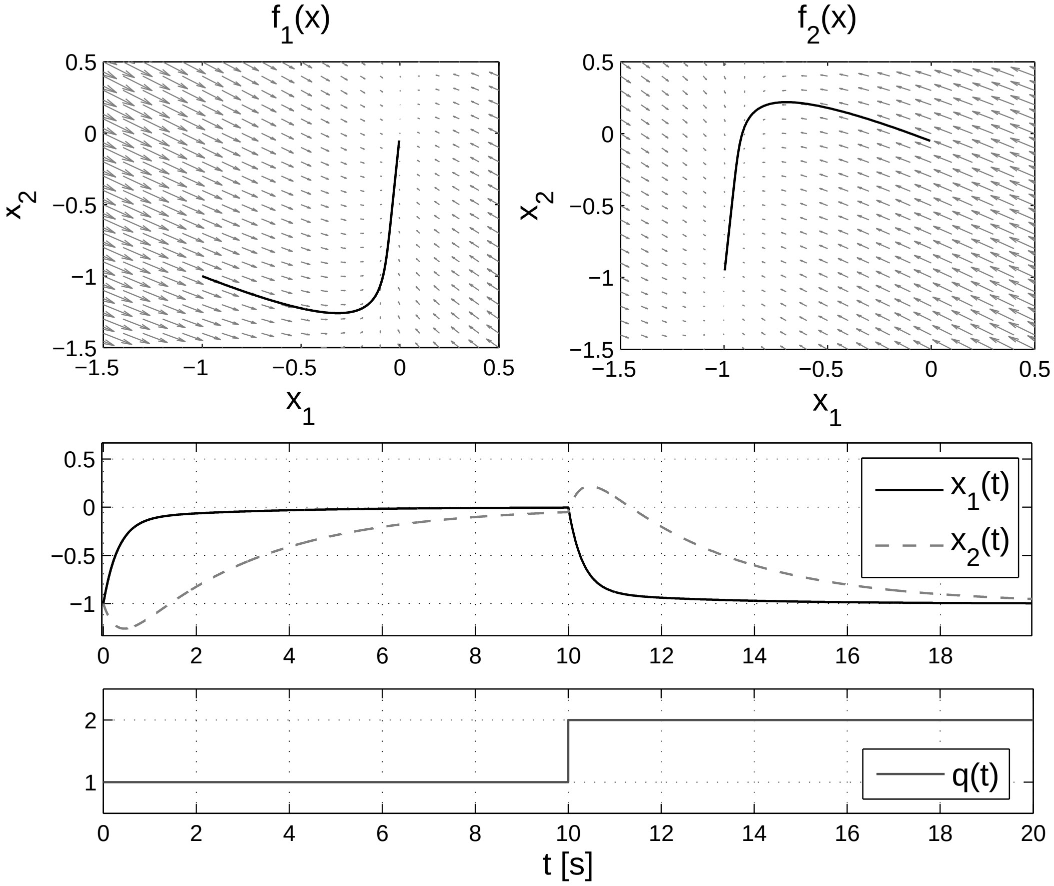

Figure 2.2: Example of a switched system

Remark 2.1: Several other forms of cost functions exist but can generally be converted to Mayer form, so that a wide range of problems can be covered with this formulation. This is exemplarily shown for a cost function of Bolza type, that consists of a final state cost $\Psi : \mathbb{R}^n \to \mathbb{R}$ and a set of Q Lagrange terms $L_q : \mathbb{R}^{n+m} \times [t_0, t_f] \to \mathbb{R}$:

$$J = \Psi(x(t_f)) + \int_{t_0}^{t_f} L_{q(t)}(x(t), u(t), t)dt. \tag{2.15}$$

By introducing an additional state x_{n+1} with the initial condition $x_{n+1}(t_0) = 0$, and therefore increasing the system's dimension by 1, this cost function type can be easily formulated as Mayer term:

$$\dot{\hat{x}} = \begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} f_{q(t)} \\ L_{q(t)} \end{bmatrix} \tag{2.16}$$

The cost function then becomes

$$\phi(\hat{x}(t_f)) = \Psi(x(t_f)) + x_{n+1}(t_f). \tag{2.17}$$

Remark 2.2: Until now, the final time t_f has been assumed to be fixed. However, if the final time is to be free, this can be achieved via the following transformation: The state vector is enhanced by a state x_{n+1}, with the time-derivative $\dot{x}_{n+1} = 0$ and the initial value $x_{n+1}(t_0) = t_f$. The enhanced state $\tilde{x}$ then evolves over the relative time interval $\tau \in [0, 1]$ according to

$$\dot{\tilde{x}} = \begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} f_{q(t)}(x(\tau), u(\tau), \tau) \cdot x_{n+1}(\tau) \\ 0 \end{bmatrix}. \tag{2.18}$$

Remark 2.3: In many technical scenarios, a switching requires a certain amount of energy and frequent switching should be avoided. This is often incorporated, by adding a switching cost term $\hat{J}$ to the cost function:

$$\hat{J} = \sum_{j=1}^{l} \Omega_{(q^-,q^+)}(x(t_j^-), t_j^-) \tag{2.19}$$

Herein Ω is a set of Q^2 mappings $\Omega_{(q^-,q^+)} : \mathbb{R}^n \times [t_0, t_f] \to \mathbb{R}$ with $\Omega_{(q^-,q^+)}(x(t_j^-), t_j^-) = 0$ for $q^- = q^+$. Switching cost can be easily incorporated into the formulation of the SOCP with Mayer term and a switched system with state jumps by adding a state x_{n+1} with

$$x_{n+1}(t_0) = 0 \tag{2.20}$$

$$\dot{x}_{n+1} = 0 \tag{2.21}$$

and varying the jump function δ as follows:

$$\hat{\delta}_{(q^-,q^+)}(x(t_j^-), t_j) = \begin{bmatrix} \delta_{(q^-,q^+)}(x(t_j^-), t_j) \\ \Omega_{(q^-,q^+)}(x(t_j^-), t_j^-) \end{bmatrix}. \tag{2.22}$$

The final state $x_{n+1}(t_f)$ is then added to the cost function.

Remark 2.4: Some solution methods require an optimal control problem to be formulated as Bolza problem without final state constraint. The final state constraint $\psi(x(t_f))$ can then included as a soft constraint in the final state penalty term $\Psi(x(t_f))$. A common way of defining Ψ is the term $\sum_{i=1}^{n} K_{\psi,i} \cdot (\psi_i(x(t_f)))^2$, where $K_{\psi,i}$ are sufficiently large penalty parameters.

2.6 Necessary Conditions for Switched Optimal Control

Formulations of Pontryagin's Minimum Principle (PMP, Pontryagin et al. [1962]), that were originally developed for the case, where no discrete variables exist, can be widely employed for hybrid systems as well. Hence, it is beneficial to define the Hamiltonian function

$$\mathcal{H}_{q(t)}(x(t), p(t), u(t), t) = p(t)^T \cdot f_{q(t)}(x(t), u(t), t), \tag{2.23}$$

where the elements in the row-vector $p(t) \in \mathbb{R}^n$ are called costates. It should be noted that due to the possibility of reformulating the system equation as in (2.6), the Hamiltonian function can also be written as

$$\mathcal{H}(x(t), p(t), u(t), \sigma(t), t) = p(t)^T \cdot F(x(t), u(t), \sigma(t), t). \tag{2.24}$$

The Hamiltonian system be defined as

$$\dot{x} = \left(\frac{\partial \mathcal{H}_{q(t)}(x(t), p(t), u(t), t)}{\partial p} \right)^T \tag{2.25}$$

$$\dot{p} = -\left(\frac{\partial \mathcal{H}_{q(t)}(x(t), p(t), u(t), t)}{\partial x} \right)^T. \tag{2.26}$$

2.6.1 Further Assumptions

Most proof techniques are based on needle variations and require some further assumptions on differentiability, continuity and boundedness. The following properties are assumed for all systems in this thesis:

- The vector fields f_q fulfill a uniform Lipschitz condition $||f_q(x_1, u, t) - f_q(x_2, u, t)|| \leq K_f \cdot ||x_1 - x_2||, x_1, x_2 \in \mathbb{R}^n, u \in \mathcal{U}_q, q \in \mathcal{Q}$ for some Lipschitz-constant $K_f < \infty$

- The input control functions $\mathcal{Z}_j : [t_j, t_{j+1}] \rightarrow \mathcal{U}_{q_j}$ are bounded and measurable and the sets of feasible controls $\mathcal{U}_q$ are convex
- There exist constants $K_\phi < \infty$ and $1 \leq \varsigma \leq \infty$ such that the cost function satisfies $|\phi(x)| \leq K_\phi \cdot (1 + ||x||^\varsigma)$
- The functions $f_q(x, u, t)$, $\phi(x)$ and $\delta_{(q^-, q^+)}(x, t)$ fulfill $f_q, \phi, \delta \in C^2$ with respect to the input arguments x, u, t

Remark 2.5: For the necessary conditions in the subsequent section to hold, the condition $f_q, \phi, \delta \in C^1$ is sufficient. However, for superlinear convergence to a local optimum of nonlinear programming methods, which are used in direct solution methods, the functions are required to be twice continuously differentiable (Broyden et al. [1973]).

Remark 2.6: In general, for hybrid systems with autonomous switching, the condition that no Zeno-behavior (see Zhang et al. [2001]) occurs in an execution of the system is demanded. However, this is not an issue for switched systems as a switching can only be triggered via an external command. Only a finite number of switching times l is allowed in this thesis.

2.6.2 Conditions for Discontinuous States

Some early results on necessary conditions for hybrid optimal control of systems with discontinuities in the state trajectory were given by Sussmann [1999]. A more readable version of most of these results is given in Riedinger et al. [2003]. A slight extension was obtained by Passenberg et al. [2011]. A proof by reformulating a hybrid optimal control problem into a conventional optimal control problem is given by Dmitruk and Kaganovich [2008].

Theorem 1: The following conditions must hold in order for the tuple (x, p, q, u) to be an optimal hybrid execution:

- The tuple (x, p, q, u) satisfies the Hamiltonian system (2.25) and (2.26)
- The minimum condition

  $$\mathcal{H}_{q(t)}(x(t), p(t), u(t), t) = \min_{u \in \mathcal{U}_q} \mathcal{H}_{q(t)}(x(t), p(t), u, t) \tag{2.27}$$

  holds for almost any $t \in [t_0, t_f]$

- At the final time t_f the transversality condition

$$p^T(t_f) = \frac{\partial \phi(x(t_f))}{\partial x(t_f)} + \mu^T \frac{\partial \psi(x(t_f))}{\partial x(t_f)}, \tag{2.28}$$

is fulfilled, where $\mu \in \mathbb{R}^n$ is a vector of Lagrange multipliers

- At a switching time t_j, the following jump conditions are satisfied

$$p^T(t_j^-) = p^T(t_j^+) \cdot \left(I + \frac{\partial \delta_{(q^-,q^+)}(x(t_j^-), t_j)}{\partial x(t_j^-)} \right) \tag{2.29}$$

$$\mathcal{H}_{q(t_j^-)}(t_j^-) = -p^T(t_j^+) \cdot \frac{\partial \delta_{(q^-,q^+)}(x(t_j^-), t_j)}{\partial t_j} + \mathcal{H}_{q(t_j^+)}(t_j^+) \tag{2.30}$$

where I is a unity matrix of dimension n.

Proof: The complete proof of the necessary conditions for a problem with final state constraints requires needle variations. The proof based on this technique can be found in the works Sussmann [1999] and Passenberg et al. [2011]. An easier way of deriving the minimization condition as well as the jump conditions of the Hamiltonian and the costate for problems of Bolza type is described by Riedinger et al. [2003] using a dynamic programming argument: Consider a hybrid optimal control problem

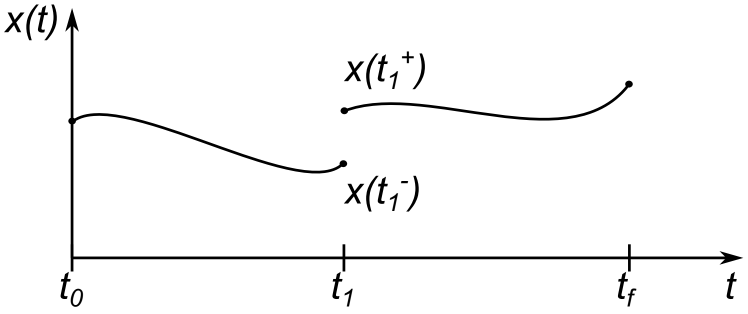

Figure 2.3: Example trajectory with one switching

$$\min_{u(t) \in \mathcal{U}_q, q(t) \in \mathcal{Q}} \int_{t_0}^{t_f} L(x(t), u(t), t) dt + \Psi(x(t_f)). \tag{2.31}$$

There exists one switching at t_1 from $q(t_1^-) = 1$ to $q(t_1^+) = 2$, as shown in Figure 2.3. Caused by the switching, a state jump from $x(t_1^-)$ to $x(t_1^+)$ with the height $\delta_{(1,2)}(x(t_1^-), t_1)$ occurs. For the purpose of abbreviation, we define

$$x(t_1^+) = x(t_1^-) + \delta_{(1,2)}(x(t_1^-), t_1) \equiv \varphi(x(t_1^-), t_1). \tag{2.32}$$

For $t \in (t_1, t_f]$, the value function V_2 gives the optimal cost for a trajectory starting at the point $(x(t), t)$:

$$V_2(x(t), t) = \min_{u(t)\in\mathcal{U}_2} \left\{ \int_t^{t_f} L(x(t), u(t), t)dt + \Psi(x(t_f)) \right\}. \tag{2.33}$$

The value function for the time range $t \in [t_0, t_1)$ can then be defined via the dynamic programming argument

$$\begin{aligned} V_1(x(t), t) &= \min_{u(t)\in\mathcal{U}_1} \left\{ \int_{t_0}^{t_1} L(x(t), u(t), t)dt + V_2(x(t_1^+), t_1) \right\} \\ &= \min_{u(t)\in\mathcal{U}_1} \left\{ \int_{t_0}^{t_1} L(x(t), u(t), t)dt + V_2(\varphi(x(t_1^-), t_1^-), t_1) \right\}. \end{aligned} \tag{2.34}$$

For $t \to t_1$, (2.34) yields

$$V_1(x(t_1^-), t_1) = V_2(\varphi(x(t_1^-), t_1), t_1) \tag{2.35}$$

and hence a new purely continuous optimal control problem can be formulated for $[t_0, t_1)$:

$$\min_{u(t)\in\mathcal{U}_1} \left\{ \int_{t_0}^{t_1} L(x(t), u(t), t)dt + V_1(x(t_1^-), t_1) \right\}. \tag{2.36}$$

From the Hamilton-Jacobi-Bellman (HJB) equations for the continuous parts $[t_0, t_1)$ and $(t_1, t_f]$, the following relations are well known (Clarke and Vinter [1987], Vinter [1988]):

$$p^T(t) = \frac{\partial V_1(x(t), t)}{\partial x(t)}, \qquad \mathcal{H}_2(t) = -\frac{\partial V_1(x(t), t)}{\partial t}, \qquad t \in [t_0, t_1) \tag{2.37}$$

$$p^T(t) = \frac{\partial V_2(x(t), t)}{\partial x(t)}, \qquad \mathcal{H}_1(t) = -\frac{\partial V_2(x(t), t)}{\partial t}, \qquad t \in (t_1, t_f]. \tag{2.38}$$

Forming the partial derivatives of $V_1(x(t_1^-), t_1)$ with respect to $x(t_1^-)$ and t_1 and using (2.35), the following conditions are obtained:

$$\begin{aligned} p^T(t_1^-) = \frac{\partial V_1(x(t_1^-), t_1)}{\partial x(t_1^-)} &= \frac{\partial V_2(\varphi(x(t_1^-), t_1)}{\partial \varphi(x(t_1^-), t_1)} \cdot \frac{\partial \varphi(x(t_1^-), t_1)}{\partial x(t_1^-)} \\ &= \frac{\partial V_2(x(t_1^+, t_1)}{\partial x(t_1^+)} \cdot \left(I + \frac{\partial \delta_{(1,2)}(x(t_1^-), t_1)}{\partial x(t_1^-)} \right) \\ &= p^T(t_1^+) \cdot \left(I + \frac{\partial \delta_{(1,2)}(x(t_1^-), t_1)}{\partial x(t_1^-)} \right) \end{aligned} \tag{2.39}$$

$$\begin{aligned}\mathcal{H}_1(t_1^-) &= -\frac{\partial V_1(x(t_1^-), t_1)}{\partial t_1^-} && (2.40)\\ &= -\frac{\partial V_2(\varphi(x(t_1^-), t_1)}{\partial \varphi(x(t_1^-), t_1)} \cdot \frac{\partial \varphi(x(t_1^-), t_1)}{\partial t_1} - \frac{\partial V_2(\varphi(x(t_1^-), t_1)}{\partial t_1}\\ &= -\frac{\partial V_2(x(t_1^+), t_1)}{\partial x(t_1^+)} \cdot \frac{\partial \delta_{(1,2)}(x(t_1^-), t_1)}{\partial t_1} - \frac{\partial V_2(x(t_1^+), t_1)}{\partial t_1} && (2.41)\\ &= -p^T(t_1^+) \cdot \frac{\partial \delta_{(1,2)}(x(t_1^-), t_1)}{\partial t_1} + \mathcal{H}_2(t_1^+).\end{aligned}$$

According to the PMP for continuous systems, the control $u(t)$ must fulfill

$$u(t) = \arg\min_{u \in \mathcal{U}_1} \mathcal{H}_1(x(t), p(t), u(t), t), \qquad \forall\, t \in [t_0, t_1^-] \tag{2.42}$$

$$u(t) = \arg\min_{u \in \mathcal{U}_1} \mathcal{H}_2(x(t), p(t), u(t), t), \qquad \forall\, t \in [t_1^+, t_f] \tag{2.43}$$

and the transversality condition (2.28) at t_f. The proof can be extended for an arbitrary finite number of switching times.

□

2.6.3 Conditions for Continuous States

Necessary conditions for switched systems without jumps in the state trajectory are derived in Riedinger et al. [1999], Xu [2001] and Shaikh [2004]:

Theorem 2: The following conditions must hold in order for the tuple (x, p, q, u) to be an optimal hybrid execution with absolutely continuous state x:

- The tuple (x, p, q, u) satisfies the Hamiltonian system (2.25) and (2.26)
- The minimum condition

  $$\mathcal{H}_{q(t)}(x(t), p(t), u(t), t) = \min_{u \in \mathcal{U}_q, q \in \mathcal{Q}} \mathcal{H}_q(x(t), p(t), u, t) \tag{2.44}$$

  holds for almost any $t \in [t_0, t_f]$
- At the final time t_f the transversality condition

  $$p^T(t_f) = \frac{\partial \phi(x(t_f))}{\partial x(t_f)} + \mu^T \frac{\partial \psi(x(t_f))}{\partial x(t_f)}, \tag{2.45}$$

  where $\mu \in \mathbb{R}^n$ is a vector of Lagrange multipliers

- At a switching time t_j, the following jump conditions are satisfied

$$p(t_j^-) = p(t_j^+) \tag{2.46}$$

$$\mathcal{H}_{q(t_j^+)}(t_j^+) = \mathcal{H}_{q(t_j^-)}(t_j^-) \tag{2.47}$$

Proof: The system can be reformulated as described in (2.6). Since the state trajectory is absolutely continuous, the continuous version of the PMP (Pontryagin et al. [1962]) can be applied. In this case, the fact is exploited that no requirements on the convexity of the set of feasible controls is made in the continuous PMP formulation (Xu [2001]).

$\square$

2.6.4 State Constraints

An overview of different necessary conditions for the case, where constrained arcs of the state variable $x(t)$ appear in the solution of a continuous optimal control problem is given by Jacobsen et al. [1972]. These conditions are in general not applicable to hybrid systems due to the possible discontinuities in the state trajectory. State constrained arcs are therefore not regarded in this thesis. Some of the algorithms however, allow for the suboptimal treatment of state constraints.

Chapter 3

Algorithms for Solving Hybrid Optimal Control Problems

3.1 Overview and State-of-the-Art

By now, no standard approach for solving hybrid optimal control problems (HOCP) exists and due to the strongly varying problem structures, that can be formulated as HOCPs, it is unlikely that a uniform approach will be developed. Most algorithms make strong requirements to the system and the problem formulation. For instance, some approaches are valid only for systems that exhibit autonomous switching only, others are suitable for problems that allow for controlled switching only. In most cases, the state will be required to be absolutely continuous. In this thesis, practical algorithms for the solution of SOCPs are regarded. Section 3.5 covers algorithms for the solution of SOCPs without state jumps, whereas section 3.6 contains algorithms for problems with state jumps. An excellent overview of existing approaches to solve HOCPs is given by Passenberg [2012] and the procedures are classified into direct methods, indirect methods and dynamic programming. Another survey on the progress in hybrid optimal control can be found in the work of Xu and Antsaklis [2003]. Many of the approaches mentioned will also appear in this thesis, however, it is preferred to use the following classification of algorithms:

1. Discretization of the continuous controls and the continuous dynamics and application of methods for the solution of multistage decision processes

2. Relaxation of the discrete controls and application of methods for the solution of purely continuous optimal control problems
3. Two-step procedures that alternate between optimizing the continuous variables and adapting the switching sequence.

3.1.1 Multi-Stage Decision Processes

The major part of algorithms, that can be assigned to the first class, is based on dynamic programming (DP) for solving the multistage decision process. Multi-stage decision processes can be described as a sequence of steps, where each step involves making a decision S, that might lead to change from the state X^k to a state X^{k+1} that is associated to a cost E. Bellman's principle of optimality (Bellman [1957]) then states that the optimal cost V at instant k with state X^k is given by

$$V(X^k, k) = \min_{S} \left\{ E(S, X^k) + V(X^{k+1}, k+1) \right\}. \tag{3.1}$$

The principle also exists in a continuous formulation, the Hamilton-Jacobi-Bellman equation. This partial differential equation can seldomly be solved but is often used for deriving necessary conditions for hybrid optimal control, as was done in section 2.6.2 or to derive algorithms for optimizing hybrid control systems (Hedlund and Rantzer [1999], Xu and Antsaklis [2000a]). Still, for a practical application of DP for hybrid systems, an appropriate discretization of the time range and the state space is required, such that the differential equation can be approximated by a difference equation and the system can be expressed as a multi-stage decision process. A practical description of the algorithm for solving optimal control problems for systems without the hybrid notion can be found in the book of Kirk [1970]. The algorithm systematically explores the state space using a backwards recursion. Because of the discrete nature of the transformed model, enhancing the algorithm for switched systems and SOCPs is straightforward. The main difference as compared to DP for conventional systems is a generalization of states and controls, such that the discrete dynamics can be included, as was acknowledged by Branicky and Mitter [1995]. DP, when applied to the solution of SOCPs has several beneficial properties, compared to other methods: Due to the exploration of the entire state space, the solution will be globally optimal with respect to the chosen discretization. Another advantage is the property, that the complexity of the algorithm grows only linearly with the number of discrete states. It is also able to incorporate switching cost, as noted by Gerdts [2012], and to take autonomous switching into account (Rungger [2011]). A practical algorithm description for solving a discretized SOCP with state jumps will be given later in this chapter. Soft state constraints can be imposed via penalty functions. On the other hand, the complexity rises exponentially with the dimension of the continuous state vector x and quickly exceeds feasible time and memory limitations. This attribute is generally called the "curse of dimensionality".

3.1.2 Relaxation Methods

Relaxation methods use a reformulation of the original problem, such that a new problem formulation has more desirable properties. This refers to the existence of the discrete variables in the system, that prevents the use of standard solution procedures. The approaches proposed by Bengea and DeCarlo [2003] and Sager [2005] share the same idea of relaxing the discrete Boolean input variable $\sigma \in \{0,1\}^Q$ to $\sigma \in [0,1]^Q$. The control set $\hat{\mathcal{V}} = \mathcal{U}_q \times [0,1]^Q$ is now a convex set if $\mathcal{U}_q$ is convex. The relaxed system $\dot{x} = F(x(t), \rho(t), t), \rho \in \hat{\mathcal{V}}$ is then purely-continuous and a conventional optimal control problem can be formulated and solved, which is in the works mentioned above performed via a direct shooting or direct collocation method. In many cases, the solution to this problem will yield a control trajectory, that is of bang-bang type with respect to the discrete control and therefore satisfies $\sigma \in \{0,1\}^Q$. In some cases, however, singular solutions may occur that do not fulfill this condition. In this case, Bengea and DeCarlo [2003] suggest, that a binary feasible trajectory can be calculated, that is arbitrarily close to the obtained state trajectory. Sager [2005] analyzes different rounding strategies to obtain a suboptimal solution. A treatment of error bounds for a rounding strategy can be found in the article of Sager et al. [2012]. A different strategy for dealing with singular solutions can be appending complementary constraints to the nonlinear program. For a system with two discrete states, the respective constraint for any time instant can be written as $\sigma_1(t) \perp \sigma_2(t)$, which means that either $\sigma_1(t)$ or $\sigma_2(t)$ is zero at time t. This constraint can be introduced as a soft constraint, by penalizing the non-fulfillment of the complementary condition, or as a hard constraint in the nonlinear program. An overview of different functions that can be used for implementing the complementary constraint is given by Leyffer [2006]. Unfortunately, these functions lead to numerical difficulties, as they are usually non-convex and in many cases locally not continuously differentiable.

3.1.3 Two-Stage Approaches

Most two-stage approaches exploit one of the two facts:

1. The optimization of the continuous controls in any interval $[t_j, t_{j+1})$ does not involve discrete variables and can therefore be solved with standard methods.

2. If for the entire time interval $[t_0, t_f]$ the discrete state trajectory $q(t)$ is fixed, again, the problem can be solved with standard methods.

Xu and Antsaklis [2000b] describe a general two-stage procedure. In the first stage, the continuous controls u are optimized. The second step involves optimizing the switching times t_j and varying the switching-order and varying the total number of switchings. No specific procedure for the variation of the switching sequence is mentioned. In a different work of the same authors (Xu and Antsaklis [2000a]), a derivative of the value function with respect to the switching time is derived and a gradient descent approach is employed to alter the switching times.

The HMP[MCS]-algorithm proposed by Shaikh [2004] is based on an initial guess of the switching times t_j and the states at the switching times $x(t_j)$. For each segment $[t_j, t_{j+1})$, a purely continuous optimal control problem is solved and the trajectories $x(t), p(t)$ are assembled for the entire time span $[t_0, t_f]$. Based on the differences in the costate and the Hamiltonian of the assembled trajectories at the switching time $p(t_j^+) - p(t_j^-)$ and $\mathcal{H}(t_j^+) - \mathcal{H}(t_j^-)$, the states $x(t_j)$ and the respective times are varied until the absolute values of the differences fulfill a lower bound condition.

In the work of Alamir and Attia [2004], a first initial guess of the discrete state q and of the control inputs u is made and the corresponding state trajectory x and costate trajectory p are calculated, such that the Hamiltonian function value can be computed for any time-instant. In the next step, optimized control inputs u and q are computed for each time instant, such that the Hamiltonian function is minimized for each time instant. A penalty factor added to the Hamiltonian function stabilizes the algorithm by reducing successive variations of the system inputs over the iterations.

The approach proposed by Nüesch et al. [2014] alternates between solving a nonlinear program for finding the continuous controls and solving a dynamic programming problem for obtaining a switching sequence. In the dynamic programming problem, the cost function is composed of the sum of the Hamiltonians at all time instants and a term for the switching cost. The costate p is assumed to be constant for the specific case regarded.

Egerstedt et al. [2006] and Axelsson et al. [2008] start with an assumed switching sequence and then optimize the length of each interval $[t_j, t_{j+1})$ without modifying the order of the sequence. Nonlinear programming is used for the first step. In a second step, the switching sequence is modified by inserting a mode, where a specifically derived derivative of the cost function with respect to the insertion of a mode does not vanish.

Another two-stage approach is the reduction of the SOCP to a mixed-integer nonlinear program (MINLP). Tree search algorithms can be applied to find the optimal discrete input variables, while for every branch of the tree, a nonlinear program has to be solved to determine the optimal continuous variables. This type of problem is $\mathcal{NP}$-hard, which means that it will most likely not be solvable in polynomial time. When SOCPs are converted to MINLPs, the number of discrete variables is in general very high and hence, this type of approach will for most cases be unfeasible. Many efforts have been made on a reduction of the search tree's size. Sager [2005] gives a wide overview of methods for solving MINLPs and their application to HOCPs.

A method for problems that have an upper bound on the number of switchings is proposed in the work of Sager et al. [2011]. An embedded optimal control problem is solved using a nonlinear programming solver. To find a switching schedule that satisfies the upper bound on the number of switchings, a mixed integer linear program is solved that minimizes a distance to the relaxed switching schedule obtained from the solution of the embedded problem but fulfills the integer conditions as well as the condition imposed on the number of switchings.

3.2 Discretization

The problems of interest in this thesis will seldomly have an analytical solution and therefore need to be treated numerically. To make the problems amenable for numerical computations, a certain discretization of the problem is required. The problem discretization will have a major effect on the quality of the solution and on the computation time and consequently, care has to be taken when choosing the discretization method as well as the discretization parameters. We restrict our considerations to explicit one-step methods of the Runge-Kutta class. The lowest-order method of this class is the explicit Euler method, which is used in the description of the algorithms due to its simplicity.

In some scenarios, the discrete approximation might fail to converge to the original continuous formulation for infinitesimal time sampling. Section 3.3.5 gives an overview of established convergence theorems, however the problem cannot fully be answered for the problems regarded in this thesis.

Remark 3.1: To allow for more compact and clearer descriptions of the algorithms in this section, the notation $f(x, q, u, t)$ is used to denote the vector field $f_q(x, u, t)$ of the hybrid system and $\delta(q^-, q^+, x, t)$ instead of $\delta_{(q^-,q^+)}(x, t)$ to define the state jump at a switching.

3.2.1 Runge-Kutta Discretization

A time grid is defined as a strictly increasing sequence of times $t^k, k = 1, \ldots, N$

$$t_0 = t^1 < t^2 < \cdots < t^N = t_f \tag{3.2}$$

that are concatenated in the discrete time vector $\bar{t}$

$$\bar{t} = [t^1, t^2, \ldots, t^N]. \tag{3.3}$$

The not necessarily equidistant sampling time interval h^k is defined as

$$h^k = t^{k+1} - t^k, \qquad k = 1, \ldots, N - 1. \tag{3.4}$$

The control is discretized via a set of $N-1$ functions $\Xi^k : \mathbb{R}^c \times [t_k, t_{k+1}) \rightarrow \mathbb{R}^m$

$$u(t) = \Xi^k(\theta^k, t), \qquad t \in [t^k, t^{k+1}) \tag{3.5}$$

that define the control at time $t \in [t^k, t^{k+1})$ depending on a parameter vector $\theta \in \mathbb{R}^c$. Piecewise constant and piecewise linear approximations schemes are described by von Stryk [1995] and an extended overview can be found in the book of Betts [2010]. Due to the piecewise constant nature of the discrete state trajectory $q(t)$, a respective piecewise constant approximation scheme is used:

$$q(t) = q^k, \qquad t \in [t^k, t^{k+1}). \tag{3.6}$$

Throughout the thesis, it is assumed that a transition in the discrete state can only take place on a sampling time t_k. The set of discretized values q^k is stored in the vector

$$\overline{q} = [q^1, q^2, \dots, q^{N-1}]. \tag{3.7}$$

Subject to the time discretization, for a given control parameter matrix

$$\overline{\theta} = [\theta^1, \theta^2, \dots, \theta^{N-1}], \tag{3.8}$$

an approximation of the state trajectory can be calculated using a numerical solver for differential equations. Runge-Kutta methods are a family of one-step-methods for the solution of initial value problems that calculate the state value's approximation for the successive sampling time according to the recursive relation

$$x^{k+1} = x^k + \Gamma_f(x^k, q^k, \theta^k, t^k, h^k) + \delta(q^k, q^{k+1}, x^k, t^k), \tag{3.9}$$

where

$$\Gamma_f(x^k, q^k, \theta^k, t^k, h^k) = h^k \cdot \sum_{j=1}^{K} a_j \cdot f_j^k, \; k = 1, \dots, N-1. \tag{3.10}$$

The subscript f in Γ_f expresses, that the function is passed as an argument to the solver Γ such that arbitrary function evaluations can be performed by the solver in the interval $[t^k, t^k + h^k)$. The parameters f_j^k are estimations of the function value f in the interior of the interval $[t^k, t^{k+1}]$ and are recursively calculated by

$$f_j^k = f(x_{i,j}, q^k, u_j, t_j) \tag{3.11}$$

$$x_{i,j} = x^k + h^k \cdot \sum_{i=1}^{K} b_{j,i} \cdot f_i^k \tag{3.12}$$

$$t_j = t^k + h^k \rho_j \tag{3.13}$$

$$u_j = \Xi^k(\theta^k, t_j). \tag{3.14}$$

The coefficients a_j, ρ_j and $b_{j,i}$ are obtained from the butcher-array of the respective Runge-Kutta method. The butcher-arrays for most methods and their respective convergence order can be found in most textbooks on numerical mathematics (for instance Dahmen and Reusken [2008] and Freund and Hoppe [2011]). In this thesis, all considerations are restricted to explicit Runge-Kutta methods that fulfill $b_{j,i} = 0$ for $i \geq j$. The set of discretized states can then be assembled in the matrix

$$\overline{x} = [x^1, x^2, \ldots, x^N]. \tag{3.15}$$

3.2.2 Explicit Euler Discretization

The most basic solver in the Runge-Kutta family is the explicit Euler method. It has the lowest computational demand but also the lowest convergence order $\mathcal{O}(h)$. Coupled with a piecewise constant control approximation, this method yields very compact definitions that will be helpful in the remainder of the thesis to keep the algorithm descriptions compact. Yet, an extension for higher order solvers is in most cases straightforward. Given the piecewise constant control approximation

$$u(t) = u^k, \qquad t \in [t^k, t^{k+1}), \tag{3.16}$$

where the elements u^k are stored in the control matrix

$$\overline{u} = [u^1, u^2, \ldots, u^{N-1}], \tag{3.17}$$

the recursion for the computation of the states is given by

$$x^{k+1} = x^k + f(x^k, q^k, u^k, t^k) \cdot h^k + \delta(q^k, q^{k+1}, x^k, t^k), \qquad k = 1, \ldots, N-1. \tag{3.18}$$

3.3 Direct Solution with Fixed Switching

Direct methods solve optimal control problems by applying an appropriate discretization scheme and hence transforming the continuous control problem into a nonlinear optimization problem (nonlinear programming problem, NLP) with a finite number of variables, assembled in the optimization vector $y = [y_1, y_2, \dots, y_{N_y}]$. This process is often referred to as a "direct transcription". Different transcription methods exist, among them direct shooting, multiple shooting and collocation methods. An early framework for a multiple-shooting algorithm is proposed by Bock and Plitt [1984]. Direct methods became more popular, due to the progress made in the development of algorithms for the solution of nonlinear optimization problems. The application of nonlinear programming methods for the solution of purely-continuous optimal control problems is well described in Betts [2010]. Direct methods can be advantageous over indirect methods due to their larger convergence area, which reduces the difficulty of initialization. They are also able to incorporate state constraints without requiring a predefined sequence of constrained and unconstrained arcs (von Stryk and Bulirsch [1992]).

3.3.1 Nonlinear Programming

Definition 3.1: A nonlinear program can be formulated as

$$\min \chi(y) \tag{3.19}$$

$$c_i(y) \leq 0, i \in \mathcal{I} \tag{3.20}$$

$$c_i(y) = 0, i \in \mathcal{E}, \tag{3.21}$$

where c_i with $i \in \mathcal{E} = \{1, \dots, l_\mathcal{E}\}$ are equality constraints and c_i with $i \in \mathcal{I} = \{l_\mathcal{E} + 1, \dots, l_\mathcal{E} + l_\mathcal{I}\}$ are inequality constraints.

Nonlinear program solvers intend to iteratively find a point that satisfies a set of necessary conditions, known as Karush-Kuhn-Tucker conditions (KKT-conditions). With the definition of the Lagrangian function $\mathcal{L} : \mathbb{R}^{N_y} \times \mathbb{R}^{l_\mathcal{E}+l_\mathcal{I}} \rightarrow \mathbb{R}$

$$\mathcal{L}(y, \mu) = \chi(y) + \sum_{i=1}^{l_\mathcal{E}+l_\mathcal{I}} \mu_i c_i(y), \tag{3.22}$$

these conditions can be stated as follows:

Definition 3.2: If a point y fulfills a linear inequality constraint qualification (LICQ), meaning that the gradients $\nabla c_i(y)$ of the active constraints are linearly independent and there is a vector of Lagrange multipliers μ with the components $\mu_i, i \in \mathcal{E} \cup \mathcal{I}$ such that the following conditions hold (y, μ):

$$\nabla_y \mathcal{L}(y, \mu) = 0 \tag{3.23}$$

$$c_i(y) = 0, \qquad \forall i \in \mathcal{E} \tag{3.24}$$

$$c_i(y) \leq 0, \qquad \forall i \in \mathcal{I} \tag{3.25}$$

$$\mu_i \leq 0, \qquad \forall i \in \mathcal{I} \tag{3.26}$$

$$\mu_i c_i(y) = 0, \qquad \forall i \in \mathcal{E} \cup \mathcal{I}, \tag{3.27}$$

then y is called a KKT-Point.

The proof of the necessary conditions, summarized in the KKT-conditions, can be found in the textbooks of Gill et al. [1981] and Nocedal and Wright [2006]. Major progress in solving NLPs was made with the development of Sequential Quadratic Programming (SQP) by Wilson [1963] and further improvements were achieved among others by Powell [1987]. Today, interior point (IP) methods are steadily growing in importance. IP methods have their origin in constrained linear programming but have soon been extended to the nonlinear case. A survey of the historic developments is given by Forsgren et al. [2003]. Especially when a high number of inequality constraints is present, IP methods can outperform SQP-Methods. For a general overview of nonlinear programming and its application, we refer to Gill et al. [1981], a book including more recent developments was written by Nocedal and Wright [2006].

3.3.2 Direct Shooting

When direct shooting is applied as transcription method, the optimization vector y contains the discrete control parameters $\overline{\theta}$. In the case of the explicit Euler discretization scheme with piecewise constant control approximation, this is reduced to $y = \overline{u}$. A function evaluation of the cost function and of the final state constraints requires the knowledge of the final state. The initial state $x^1 = x_0$ and the switching sequence $\overline{q}$ as well as the time grid $\overline{t}$ are fixed boundary conditions and therefore omitted in the following equations. For a given set of parametrized controls $\overline{\theta}$, the final state x^N can be computed with the Runge-Kutta solver. The NLP can be written as

$$\min_{\overline{\theta}} \phi(x^N(\overline{\theta})) \tag{3.28}$$

$$\psi(x^N(\overline{\theta})) = 0 \tag{3.29}$$

$$c_u^k(u^k(\theta^k, k)) \leq 0, \quad k = 1, \ldots, N-1. \tag{3.30}$$

The control restraint $c_u^k \in C^2 : \mathbb{R}^m \rightarrow \mathbb{R}^m$ is chosen such that $c_u^k(u^k(\theta^k, k)) \leq 0$ when $u^k \in \mathcal{U}_{q^k},\ k = 1, \ldots, N-1$. Additional conditions on the parameters $\overline{\theta}$ as for example the condition that $u(t_k)$ be continuous and continuously differentiable, when a higher order polynomial approximation is used for Ξ, can also be imposed.

Remark 3.2: The choice of the constraints c_u^k as mentioned above does not guarantee that the control is feasible in the entire interval $[t^k, t^{k+1})$ but only on the boundary of the interval.

3.3.3 Collocation

In contrast to the direct shooting method, where only the controls are included in the optimization vector y, collocation methods add the discretized state vector $\overline{x}$ as well. To enforce the fulfillment of the differential equation constraint, the terms (3.32) and (3.33) are added as constraints in the NLP-formulation. The NLP can then be stated as

$$\min_{\overline{\theta}, \overline{x}} \phi(x^N) \tag{3.31}$$

$$x^{k+1} - x^k - \Gamma_f(x^k, \theta^k, k) - \delta(x^k, k) = 0,\ k = 1, \ldots, N-1 \tag{3.32}$$

$$x^1 - x_0 = 0 \tag{3.33}$$

$$\psi(x^N) = 0 \tag{3.34}$$

$$c_u^k(u^k(\theta^k, k)) \leq 0, \quad k = 1, \ldots, N-1. \tag{3.35}$$

3.3.4 Comparison of Direct Shooting and Collocation

The use of collocation approaches was for a long time prevented by the high number of variables in the optimization vector. In comparison to the direct shooting method, where only the control parameters are contained in this vector, the additional discrete state representation enhances the number of elements by $N \cdot n$. Looking at the numerical solution procedure, this will require more function evaluations for the gradient estimation and significantly more memory for storing the Hessian matrix. This disadvantage however can be circumvented by exploiting sparse structures of the problem that occur naturally due to the fact that many variables in the optimization vector are independent of each other. On the one hand, this can reduce the number of required function evaluations for the gradient calculation (Betts and Huffman [1999]), on the other hand, sparse matrix algebra can be applied for solving the quadratic sub-problem in the SQP-procedure. Another practical advantage of collocation methods, that is often disregarded, is the avoidance of the recursive solution of the initial value problem. Recursive function calls bring along a certain overhead, whereas the collocation approach allows much more for vectorized function evaluations and therefore avoids this type of overhead. In some cases, the gradient estimation can therefore be less costly despite the fact that the number of variables is higher. The main challenge so far remains to develop a sparse Hessian update formula. Some promising approaches exist (for instance Yamashita [2008]) but were not yet fully implemented during this thesis. Therefore, the algorithms proposed in this thesis that imply a direct solution approach are based on direct shooting methods but can easily be transferred to collocation.

3.3.5 Convergence Results

An important question is, whether the discrete approximation of a hybrid optimal control problem converges to the continuous formulation for $h^k \to 0, k = 1, \ldots, N-1$. For the specific type of systems regarded in this thesis, this question cannot fully be answered with the convergence results established in the literature so far. A detailed convergence analysis for the discrete approximation of purely continuous optimal control problems was already addressed by Mordukhovich [1978] and was enhanced later (Mordukhovich [2006]). The main result is that for optimal control problems without final state constraint, convergence can be guaranteed under mild assumptions, whereas for problems with final state constraint, an additional continuity requirement for the control is necessary. This requirement is also made in the convergence analysis of Gerdts [2012] but can be problematic, when the discrete state is relaxed to a continuous control variable. In this case a bang-bang solution is expected and therefore the control will not be continuous. A convergence rate for the approximation of a discrete initial value problem for hybrid systems with state jumps is derived in the works of Tavernini [1987] and Tavernini [2009]. This convergence rate however is only valid, when a variable time grid is used as proposed by the author. Hence, the results cannot be used for the case of a fixed time-grid as applied in this thesis. In the following, it is assumed that the discrete approximations of the formulated SOCPs converge to their continuous counterpart.

3.3.6 Recovering the Costate from a Direct Shooting

One of the major disadvantages of direct methods for the solution of the optimal control problem is the fact that the costate is not obtained from the solution in a direct manner. However, knowledge of the costate can be very helpful as it allows for the evaluation of the fulfillment of necessary conditions and in many cases, the costate can provide helpful insight into the structure of the solution. On the other hand, indirect methods provide the costate trajectory as part of the solution but may not be applicable since a good initial guess of the costate is hard to achieve. A more elegant way is the solution of the discretized optimal control problem using direct methods and therefore omitting the inclusion of the costate in the solution procedure and then recovering the costate from the solution of the problem. Methods for obtaining the costate from a direct solution were proposed by Enright and Conway [1992] and von Stryk [1995] for collocation approaches and by Büskens [1998] for a direct shooting method. The following derivation of the costate partially follows the reasoning of Büskens [1998]. To our best knowledge, systems with state jumps have not been regarded before. To allow for compact descriptions, the following abbreviations are introduced:

$$f^k \equiv f(x^k, q^k, u^k, t^k) \tag{3.36}$$

$$\delta^k \equiv \delta(q^k, q^{k+1}, x^k, t^k). \tag{3.37}$$

Using the explicit Euler procedure, for approximating a solution to the initial value problem, one step can be calculated by the recursive relation

$$x^{k+1} = x^k + f^k h^k + \delta^k. \tag{3.38}$$

The following relations can then be derived:

$$\frac{\partial x^{k+1}}{\partial x^k} = I + \frac{\partial f^k}{\partial x^k} h^k + \frac{\partial \delta^k}{\partial x^k} \tag{3.39}$$

$$\frac{\partial x^{k+1}}{\partial u^k} = \frac{\partial f^k}{\partial u^k} h^k, \tag{3.40}$$

where I is the unity matrix of dimension n. The derivative of the final state x^N with respect to any preceding state x^k and the derivative of the final state to a preceding control u^k can be obtained from the chained derivatives

$$\frac{\partial x^N}{\partial x^k} = \frac{\partial x^N}{\partial x^{N-1}} \cdots \frac{\partial x^{k+1}}{\partial x^k} \tag{3.41}$$

$$\frac{\partial x^N}{\partial u^k} = \frac{\partial x^N}{\partial x^{k+1}} \frac{\partial x^{k+1}}{\partial u^k}. \tag{3.42}$$

The Lagrangian function can be written as

$$\mathcal{L} = \phi(x^N) + \nu^T \psi(x^N) + \sum_{k=1}^{N-1} (\eta^k)^T c_u^k(u^k). \tag{3.43}$$

The final state x^N is a function of the discretized controls $\overline{u}$, and the Jacobian with respect to the controls at time instant k can be derived as

$$\frac{\partial \mathcal{L}}{\partial u^k} = \frac{\partial \mathcal{L}}{\partial x^N} \frac{\partial x^N}{\partial u^k} + \eta^k \frac{\partial c_u^k}{\partial u^k}. \tag{3.44}$$

Theorem 3: The derivative of the Lagrangian function $\mathcal{L}$ with respect to a state x^k is an approximation of the costate at time instant k:

$$(\tilde{p}^k)^T \equiv \frac{\partial \mathcal{L}}{\partial x^k}. \tag{3.45}$$

Proof: To proof this theorem, it is sufficient to show that $\tilde{p}^N$ fulfills transversality condition (2.45) and that any preceding $\tilde{p}^k$ can be calculated outgoing from $\tilde{p}^N$ by solving an initial value problem backwards using (2.26). For $\tilde{p}^N$, we have

$$(\tilde{p}^N)^T = \frac{\partial \mathcal{L}}{\partial x^N} = \frac{\partial \phi}{\partial x^N} + \nu^T \frac{\partial \psi}{\partial x^N}, \tag{3.46}$$

which is an approximation of transversality condition (2.45). If the above theorem holds, then

$$(\tilde{p}^k)^T = \frac{\partial \mathcal{L}}{\partial x^k} = \frac{\partial \mathcal{L}}{\partial x^N}\frac{\partial x^N}{\partial x^k} = (\tilde{p}^N)^T\frac{\partial x^N}{\partial x^k} \tag{3.47}$$

is valid and forming the difference between two consecutive values $\tilde{p}^{k+1}$ and $\tilde{p}^k$, the following result is obtained

$$\begin{aligned}(\tilde{p}^{k+1})^T - (\tilde{p}^k)^T &= \frac{\partial \mathcal{L}}{\partial x^N}\left(\frac{\partial x^N}{\partial x^{k+1}} - \frac{\partial x^N}{\partial x^k}\right) \\ &= \frac{\partial \mathcal{L}}{\partial x^N}\frac{\partial x^N}{\partial x^{k+1}}\left(I - \frac{\partial x^{k+1}}{\partial x^k}\right) \\ &= (\tilde{p}^{k+1})^T\left(I - \frac{\partial x^{k+1}}{\partial x^k}\right) \\ &= (\tilde{p}^{k+1})^T\left(I - \left(I + \frac{\partial f^k}{\partial x^k}h^k + \frac{\partial \delta^k}{\partial x^k}\right)\right) \\ \Rightarrow (\tilde{p}^k)^T &= (\tilde{p}^{k+1})^T + (\tilde{p}^{k+1})^T\frac{\partial f^k}{\partial x^k}h^k + (\tilde{p}^{k+1})^T\frac{\partial \delta^k}{\partial x^k},\end{aligned} \tag{3.48}$$

which clearly is a backwards explicit Euler approximation to the differential equation of the costate (2.26) including jump condition (2.29). It should be noted, that the Jacobian $\frac{\partial \delta^k}{\partial x^k}$ contains nonzero values only, if $q^{k+1} \neq q^k$.
Applying the KKT-condition (3.23) yields

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial u^k} &= \frac{\partial \mathcal{L}}{\partial x^N}\frac{\partial x^N}{\partial x^{k+1}}\frac{\partial x^{k+1}}{\partial u^k} + (\eta^k)^T\frac{\partial c_u^k}{\partial u^k} \\ &= (\tilde{p}^{k+1})^T\frac{\partial x^{k+1}}{\partial u^k} + (\eta^k)^T\frac{\partial c_u^k}{\partial u^k} \\ &= (\tilde{p}^{k+1})^T\frac{\partial f^k}{\partial u^k}h^k + (\eta^k)^T\frac{\partial c_u^k}{\partial u^k} = 0\end{aligned} \tag{3.49}$$

which is a neccessary condition for the fulfillment of Hamiltonian minimization condition (2.27).

□

3.3.7 Recovering the Costate from Collocation

In a much more direct manner, the costate can be derived when using the collocation method with an explicit Euler discretization scheme.

Proof: Defining the Lagrangian as

$$\begin{aligned}\mathcal{L} = {} & \phi(x^N) + \nu^T \psi(x^N) + \mu^T (x_0 - x^1) \\ & + \sum_{k=1}^{N-1} (\tilde{p}^{k+1})^T (x^k + f^k h^k + \delta^k - x^{k+1}) \\ & + \sum_{k=1}^{N-1} (\eta^k)^T c_u^k(u^k),\end{aligned} \tag{3.50}$$

the derivative with respect to the discretized states yields

$$\frac{\partial \mathcal{L}}{x^N} = \frac{\partial \phi}{\partial x^N} + \nu^T \frac{\partial \psi}{\partial x^N} - (\tilde{p}^N)^T \tag{3.51}$$

$$\frac{\partial \mathcal{L}}{x^1} = (\tilde{p}^2)^T \left(I + \frac{\partial f^1}{\partial x^1} h^1 + \frac{\partial \delta^1}{\partial x^1} \right) - (\tilde{p}^1)^T - \mu^T \tag{3.52}$$

and for $k = 2, \ldots, N-1$ we have

$$\frac{\partial \mathcal{L}}{x^k} = (\tilde{p}^{k+1})^T \left(I + \frac{\partial f^k}{\partial x^k} h^k + \frac{\partial \delta^k}{\partial x^k} \right) - (\tilde{p}^k)^T. \tag{3.53}$$

The derivative with respect to the controls at each time instant is given by

$$\frac{\partial \mathcal{L}}{\partial u^k} = (\tilde{p}^{k+1})^T \frac{\partial f^k}{\partial u^k} h^k + (\eta^k)^T \frac{\partial c_u^k}{\partial u^k} = 0. \tag{3.54}$$

From (3.51), again the approximation of transversality condition (2.45) is obtained with

$$(\tilde{p}^N)^T = \frac{\partial \phi}{\partial x^N} + \nu^T \frac{\partial \psi}{\partial x^N} \tag{3.55}$$

and (3.54) is, as before, a necessary condition for the fulfillment of Hamiltonian minimization condition (2.27). The difference equation for two consecutive costates is obtained from (3.53) as

$$(\tilde{p}^k)^T = (\tilde{p}^{k+1})^T + (\tilde{p}^{k+1})^T \frac{\partial f^k}{\partial x^k} h^k + (\tilde{p}^{k+1})^T \frac{\partial \delta^k}{\partial x^k}, k = 2, \ldots, N-1, \tag{3.56}$$

which equals the result obtained for the direct shooting approach.

□

Remark 3.3: The Lagrange multipliers for appending the fulfillment of difference equation (3.32) to the Lagrangian are an approximation of the costates. Consequently, when the collocation approach is used, the approximated costate can be obtained readily from the solution of the nonlinear program, since most nonlinear program solvers return the Lagrange multipliers at the solution. This is an advantage over the direct shooting method, where an additional finite differencing procedure is required (3.45), to obtain the costates.

3.4 Indirect Solution with Fixed Switching

When the switching schedule $\overline{q}$ is given, the problem of finding optimal trajectories $\overline{x}$ and $\overline{u}$ can be formulated as two-point boundary value problem that can be solved using indirect shooting methods. An early algorithm can be found in the work of Kirk [1970] and more complex problems can be solved using the algorithm developed by Oberle and Grimm [2001]. An expanded state vector is defined as follows:

$$y(t) = \begin{bmatrix} x(t) \\ p(t) \end{bmatrix}, \qquad y(t_0) = \begin{bmatrix} x(t_0) \\ p(t_0) \end{bmatrix}. \tag{3.57}$$

Between two switchings, the enhanced state obeys the differential equation law

$$\dot{y} = G(y(t), q(t), u(t), t) = \begin{bmatrix} f(x(t), q(t), u(t), t) \\ -\left(\frac{\partial \mathcal{H}(x(t), q(t), u(t), t)}{\partial x}\right)^T \end{bmatrix}. \tag{3.58}$$

At a switching time t_j, jumps in both, state trajectory as well as costate trajectory occur:

$$x(t_j^+) = x(t_j^-) + \delta_{(q(t_j^-), q(t_j^+))}(x(t_j^-), t_j) \tag{3.59}$$

$$p^T(t_j^-) = p^T(t_j^+) \cdot \left(I + \frac{\partial \delta_{(q^-, q^+)}(x(t_j^-), t_j)}{\partial x(t_j^-)} \right). \tag{3.60}$$

Discretizing the time between two consecutive switching times (t_j, t_{j+1}) appropriately, the trajectory of y can be approximated with the Runge-Kutta solver:

$$y^{k+1} = y^k + \Gamma_G(y^k, q^k, u^k, t^k, h^k). \tag{3.61}$$

Repeating this for all time intervals $j = 0, \dots, l$ and applying (3.59) and (3.60) at the switching times, a full approximation of the trajectory y in the time interval $[t_0, t_f]$ is obtained. Please note that (3.60) involves the solution of a linear system of equations. The control $u(t)$ is at every step determined by minimizing the Hamiltonian function

$$u(t) = \arg \min_{u \in \mathcal{U}_{q(t)}} \mathcal{H}(y(t), q(t), u, t). \tag{3.62}$$

In most cases, an analytical solution of the minimization problem will not be possible and therefore numerical minimization procedures have to be applied. The task is well-suited for SQP-methods.

Solving this initial value problem (IVP) will require the knowledge of the initial costate $p(t_0)$. The initial costate however will be unknown and hence needs to be found. At the final time t_f, a solution to the control problem has to satisfy transversality condition (2.28) as well as the final state constraint $\psi(x(t_f)) = 0$. These are concatenated in the vector

$$\Upsilon(y(t_f), \mu) = \begin{bmatrix} \psi(x(t_f)) \\ p(t_f) - \frac{\partial \phi(x(t_f))}{\partial x(t_f)} - \mu^T \frac{\partial \psi(x(t_f))}{\partial x(t_f)} \end{bmatrix} = 0. \tag{3.63}$$

Due to the fact that $p(t_0)$ is unknown and that $y(t_f)$ is partially constrained, the problem of finding an initial costate $p(t_0)$ that fulfills $\Upsilon(y(t_f), \mu) = 0$ is called a two-point boundary value problem. Since $y(t_f)$ is a function of $p(t_0)$, the problem can be written as $\Upsilon(p(t_0), \mu) = 0$. When this nonlinear function can be solved using an appropriate solver, the solution of the control problem is obtained as well.

Remark 3.4: The nonlinear function $\Upsilon(p(t_0), \mu)$ in the given form is of dimension $2n$ according to the size of the vector $[p(t_f), \mu]^T$ but in most cases, the dimension can be reduced by evaluating the transversality conditions (2.28) and the differential equation governing the costate evolution (2.26).

3.5 Algorithms for Systems with Continuous States

In this section, practical algorithms will be presented, that aim at approximating a solution to an SOCP defined for a system with absolutely continuous state and hence $\delta_{(q^-,q^+)} = 0 \ \forall \ q^-, q^+ \in \mathcal{Q}$. This subclass of the switched systems defined in section 2.1 benefits from much stronger necessary conditions. The most important difference is, that beneath the continuous control $u(t)$ the discrete state $q(t)$ must minimize the Hamiltonian function. Another difference is the continuity of the costate on a change of the discrete state.

3.5.1 Embedding

The embedding approach is based on the representation of the hybrid system by the differential equation (2.6) and on a relaxation of the Boolean vector

$\sigma \in \{0,1\}^Q$ to the compact set $\sigma \in [0,1]^Q$. It is assumed, that the continuous control variable may be chosen from a unique set $\mathcal{W}$ instead of separate sets $\mathcal{U}_q$ for each discrete state. In practical scenarios, this will often be the case. In other cases, a new control variable u' can be introduced as well as a set of Q functions $\Lambda_q : \mathcal{W} \to \mathcal{U}_q$ that perform a respective transformation. The vector fields $f_q(x(t), u(t), t)$ are then modified to $f_q(x(t), \Lambda_q(u'(t)), t)$. The control vector concatenates the new control u' and the switching variable σ

$$\rho(t) = [u'(t), \sigma(t)] \tag{3.64}$$

and may take on values from the compact set $\mathcal{W} \times [0,1]^Q$. The system can now be treated as a conventional system without hybrid phenomena and be solved with direct methods, such as the direct shooting method. $\sigma(t)$ is best approximated as a piecewise constant function. The problem can then be formulated as:

$$\min_{\overline{\overline{\rho}}} \phi(\hat{x}^N(\overline{\rho})) \tag{3.65}$$

$$\psi(\hat{x}^N(\overline{\rho})) = 0 \tag{3.66}$$

$$\sum_{q=1}^{Q} \sigma_q^k - 1 = 0, \quad k = 1, \ldots, N-1 \tag{3.67}$$

$$-\sigma^k \leq 0, \quad k = 1, \ldots, N-1 \tag{3.68}$$

$$\sigma^k - 1 \leq 0, \quad k = 1, \ldots, N-1 \tag{3.69}$$

$$c_\rho(\overline{\rho}) \leq 0. \tag{3.70}$$

The state $\hat{x}$ is used to illustrate that the state trajectory was obtained with a modified system description.

Definition 3.3: A discrete state trajectory $\overline{\sigma}$ is called binary feasible, if $\sigma^k \in \{0,1\}^Q, k = 1, \ldots, N-1$ and constraint (3.67) is satisfied.

If the trajectory obtained for the control σ is binary feasible, the state $\overline{\hat{x}}$ is a valid approximation for the optimal discretized state trajectory $\overline{x}$. In the case, where this condition is not fulfilled, the arcs where $\sigma^k \notin \{0,1\}^Q$ are denoted as singular arcs. An approximation for a binary feasible trajectory can then be obtained using different strategies. Sager [2005] recommends the application of a combination of rounding strategies for the switching variable

σ, switching time optimizations and a penalty term homotopy and provides detailed algorithms.

Remark 3.5: The dimension of the optimization vector $\overline{p}$ can be reduced by defining $\sigma_Q^k = 1 - \sum_{q=1}^{Q-1} \sigma_q^k$ and omitting the constraint (3.67).

Remark 3.6: As desired, σ^k will tend to the boundary of $[0,1]^Q$ for most $k = 1, \ldots, N-1$. NLP-solvers based on active-set methods will require a lot of iterations, until the optimal active set is found and will hence require much more computation time. NLP-solvers that use an interior-point approach for solving the quadratic sub-problem or use an overall interior-point approach will perform much better on embedded problems unless the active-set method uses a good warm-start method for determining the initial active-set. However, for interior-point methods, binary feasibility will only be achieved up to a certain tolerance, as the barrier-function approach will prevent an exact fulfillment.

3.5.2 Indirect Shooting

The indirect shooting algorithm in section 3.4 can easily be extended for a free switching schedule in the case of an absolutely continuous state x by using condition (2.44) as an indicator on which discrete state to use at a given time step. The idea was roughly sketched by Riedinger et al. [2005] and is justifiably not covered in detail, due to problems of convergence. Yet, for certain problems, the algorithm is still an interesting and especially efficient alternative.

The high-level task remains to find a vector of initial costates $p(t_0) = p^1$ and a vector μ that fulfill the final state condition $\Upsilon(x^N, p^N, \mu) = 0$. The main modification consists in the determination of the discrete state q^k to be applied at each time step along with the determination of the continuous control u^k at the same time-step. This is done in two stages:

1. For each $q \in \mathcal{Q}$, an optimal continuous control u_q is determined that minimizes the Hamiltonian function for the given discrete state
2. In the second stage, the discrete state q^k is chosen that minimizes the Hamiltonian function by comparing the function values for all $q \in \mathcal{Q}$ using the respective continuous controls u_q. Applying the discrete state

q^k that minimizes the Hamiltonian function, and the control $u^k = u_{q^k}$, the state x^{k+1} can be computed using an appropriate solver for the initial value problem.

The pseudo code of the functions required for the implementation of the algorithm is given below. An explicit Euler method is used for obtaining an approximation to the state trajectory. The function NONLINFUN is the nonlinear function to be solved. Depending on the input p^1, the function IVP, that solves the initial value problem, is called. In this function, the control u^k and the discrete state q^k are calculated at each time step k, as explained above. The function call HGRAD in the function IVP computes the gradient of the Hamiltonian function with respect to the state that can be approximated using finite differencing. One of the major weaknesses of most indirect methods is the necessity of estimating an initial guess of the costate's initial value. As the costate usually has no physical interpretation, this can be a hard task, especially, when the dimension n is high. The algorithm provided in the next section will cope with this challenge. If a sufficiently accurate guess of the initial costate can be made, the indirect shooting algorithm will be very fast. It also has the advantageous property that no initial estimation of the switching sequence needs to be made and that the computing time depends only linearly on the discretization N and on the number of discrete states Q. Not included in the algorithm description is the numerical method for solving the nonlinear equation NONLINFUN. A diversity of solvers is usually included in any scientific computing environment. Special care has to be taken when choosing the solver. In many cases, gradient based methods will fail due to the many sources of round-off error in the function IVP. There is also no guarantee that Υ is differentiable with respect to p^1. The application of gradient free methods for high dimensions n however is often very slow. If the dimension is $n = 1$, the nonlinear function Υ can usually reduced to a scalar function and regula-falsi methods can be applied, that iteratively partition an interval around the solution, until a lower error bound is fulfilled. These methods perform very robustly and compete with gradient based methods in terms of speed. A very efficient implementation is the Pegasus method (Dowell and Jarratt [1972]) that will converge superlinearly for many functions.

```
function NONLINFUN($p^1$, $\mu$, $x^1$, $\overline{t}$)
    $\overline{x}, \overline{u}, \overline{p}, \overline{q} \leftarrow$ IVP($p^1$, $x^1$, $\overline{t}$)
    $d \leftarrow \Upsilon(x^N, p^N, \mu)$
    return $d, \overline{x}, \overline{u}, \overline{p}, \overline{q}, \mu$
end function

function IVP($p^1$, $x^1$, $\overline{t}$)
    for $k \leftarrow 1, N-1$ do
        for all $q \in \mathcal{Q}$ do
            $u_q \leftarrow \arg\min_{u \in \mathcal{U}_q} \mathcal{H}(x^k, q, p^k, u, t^k)$
        end for
        $q^k \leftarrow \arg\min_{q \in \mathcal{Q}} \mathcal{H}(x^k, q, p^k, u_q, t^k)$
        $u^k \leftarrow u_{q^k}$
        $x^{k+1} \leftarrow x^k + f(x^k, q^k, u^k, t^k) \cdot (t^{k+1} - t^k)$
        $p^{k+1} \leftarrow p^k -$ HGRAD$(x^k, q^k, p^k, u^k, t^k) \cdot (t^{k+1} - t^k)$
    end for
    return $\overline{x}, \overline{u}, \overline{p}, \overline{q}$
end function
```

Algorithm 1: Nonlinear function to be solved for the two-point boundary value problem

3.5.3 Combining Direct and Indirect Methods

As has been mentioned in the previous section, making an initial guess of the costate can be difficult for many practical problems and the use of the indirect shooting algorithm can be impaired by this difficulty. One of the beneficial characteristics of direct methods is that for a predefined switching sequence, the optimal control problem can be solved efficiently and robustly. No costate is required in the solution procedure but the costate can be recovered from the solution after the termination of the algorithm, as has been shown in section 3.3.6. With an approximation of the costate at hand, the necessary conditions for an optimal hybrid execution can be reviewed and the switching sequence be altered, based on an appropriately chosen criterion.

A valid criterion is the largest gradient with respect to the insertion of a new mode q. Assume $\Theta = ((t_0, q_0), (t_1, q_1), (t_2, q_2))$ to be an optimal switching sequence with respect to the switching times t_j (Figure 3.1). Consider the insertion of an additional mode q^* for a time interval Δt centered at time t^* with $t_1 < t^* - \frac{1}{2}\Delta t$ and $t^* + \frac{1}{2}\Delta t < t_2$, such that the new switching schedule is written $\hat{\Theta} = ((t_0, q_0), (t_1, q_1), (t^* - \Delta t, q^*), (t^* + \Delta t, q_1), (t_2, q_2))$. The insertion of the new mode may yield an additional reduction of the cost function. Egerstedt et al. [2006] and Axelsson et al. [2008] provide a gradient of the cost function with respect to the length of the time interval as follows:

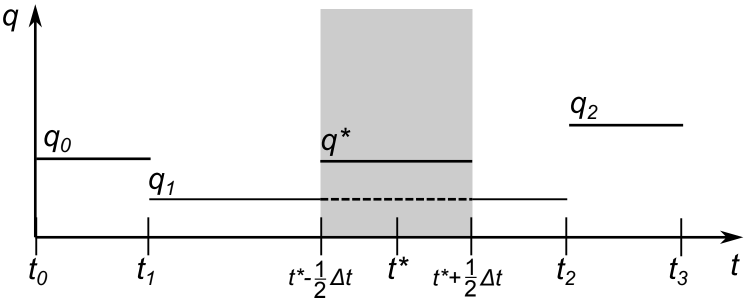

Figure 3.1: Insertion of the mode q^* in the interval $[t^* - \frac{1}{2}\Delta t, t^* + \frac{1}{2}\Delta t]$

$$\begin{aligned}\frac{\partial \phi}{\partial \Delta t} &= p^T(t^*) \cdot (f_{q^*}(x(t^*), u(t^*), t^*) - f_{q_1}(x(t^*), u(t^*), t^*)) \\ &= \mathcal{H}_{q^*}(x(t^*), p(t^*), u(t^*), t^*) - \mathcal{H}_{q_1}(x(t^*), p(t^*), u(t^*), t^*). \qquad (3.71)\end{aligned}$$

The derivation of the gradient is based on a derivative of the cost functional with respect to a switching time. This type of gradient was also derived by Xu and Antsaklis [2000a] and Kamgarpour and Tomlin [2012]. The insertion of a mode q^* may yield a decrease in the cost function value, when $\frac{\partial \phi}{\partial \Delta t} < 0$. It is therefore desirable to alter the switching sequence, where the difference (3.71) has the lowest negative value. Based on this observation, the following two-stage algorithm is introduced:

1. In the first stage, a direct method is applied for finding the optimal continuous control inputs for an initial guess of the switching sequence. Once the NLP-solver terminates, a discrete state $\overline{x}$, the controls $\overline{u}$ as well as a set of Lagrange multipliers γ are obtained for the assumed switching

sequence $\overline{q}$. With the results from section 3.3.6, an approximation of the costate $\overline{p}$ is calculated

2. In the second stage, the switching schedule $\overline{q}$ is altered, based on the evaluation of the necessary conditions. The schedule is altered at the instant k, where the largest descent in the Hamiltonian function can be achieved by altering q at this instant.

The algorithm alternates between these two stages, until a termination criterion is fulfilled. As such criterion may serve $\Delta H_{max} > \epsilon$, where ΔH_{max} is the largest difference between the current Hamiltonian at one of the time instants k and any other Hamiltonian calculated for a different q at the same instant.

Remark 3.7: Other termination criteria might be necessary and can include a maximum number of iterations. Due to the fact that the costate $\overline{p}$ only is an approximation, numerical errors may prevent the fulfillment of the termination criterion $\Delta H_{max} > \epsilon$.

The overall algorithm consists of three functions. The main function COMBMETHOD accepts an initial guess of $\overline{q}$ and uses an NLP-solver to find optimized control inputs $\overline{u}$ for this switching sequence. The solver iteratively calls the function IVP to calculate the state trajectory such that the cost function and constraints can be evaluated. It should be noted that the function IVP differs from the function of the same denominator in the previous section in that it does not approximate the costate trajectory. It is also more flexible, as it can be started at arbitrary instants k for a given initial value x^k. This will allow for a faster approximation of the costate in the function COSTATE. After the completion of the nonlinear-optimization, the state vector $\overline{x}$ is computed for the optimized controls and the given switching sequence. As a side-product the Lagrange-multipliers γ are also returned by the NLP-solver. The function COSTATE then calculates an approximation of the costate vector $\overline{p}$ using one-sided finite differencing to evaluate equation (3.45).

Remark 3.8: A more efficient way would be to calculate p^N only and then use (2.26) to calculate the remaining values p^k (Büskens [1998]). However this would require more coding, whereas this functional approach offers great flexibility. The for-loops are also parallelizable such that the required time-span is usually acceptable.

Once $\overline{p}$ is returned to the main function COMBMETHOD, the minimum Hamiltonian function values H_q^k can be computed for every time instant k and for every discrete state q and are compared against the current Hamiltonian function values H_{act}^k. The combination of indices k and q are determined that yield the largest decrease in the Hamiltonian function at a time instant k. The switching schedule is then altered respectively. These steps are repeated until the termination criterion is fulfilled. The overall algorithm is rather slow, since the switching schedule is altered at one instant k only on each iteration. Numerical experiments, where more than one instant were modified during each iteration have worked in some cases but have failed to converge in others and hence this approach cannot be generally recommended.

```
function COMBMETHOD
    Define $\epsilon$, $\overline{q}_{init}$, $\overline{u}_{init}$, $x^1$, $\overline{t}$
    $\overline{q} \leftarrow \overline{q}_{init}$
    $\overline{u} \leftarrow \overline{u}_{init}$
    $\Delta H_{max} \leftarrow \infty$
    while $\Delta H_{max} > \epsilon$ do
        $\overline{u}, \gamma \leftarrow$ NLP($\overline{q}$, $\overline{u}$, $\overline{t}$)
        $\overline{x} \leftarrow$ IVP($\overline{q}$, $\overline{u}$, $\overline{t}$, $x^1$, 1)
        $\overline{p} \leftarrow$ COSTATE($\overline{x}$, $\overline{q}$, $\overline{u}$, $\overline{t}$, $\gamma$)
        for $k \leftarrow 1, N-1$ do
            for all $q \in \mathcal{Q}$ do
                $H_q^k \leftarrow \min_{u \in \mathcal{U}_q} \mathcal{H}(x^k, q, p^k, u, t^k)$
                $u_q^k \leftarrow \arg\min_{u \in \mathcal{U}_q} \mathcal{H}(x^k, q, p^k, u, t^k)$
            end for
            $H_{act}^k \leftarrow H_{q^k}^k$
        end for
        $i, j \leftarrow \max_{q,k}(H_{act}^k - H_q^k)$
        $\overline{q}^j \leftarrow i$
        $\overline{u}^j \leftarrow u_i^j$
        $\Delta H_{max} \leftarrow H_{act}^j - H_i^j$
    end while
    return $\overline{x}, \overline{u}, \overline{p}, \overline{q}$
end function

function IVP($\overline{q}$, $\overline{u}$, $\overline{t}$, $x^s$, $s$)
    for $k \leftarrow s, N-1$ do
        $x^{k+1} \leftarrow x^k + f(x^k, q^k, u^k, t^k) \cdot (t^{k+1} - t^k)$
    end for
    return $\overline{x}$
end function

function COSTATE($\overline{x}$, $\overline{q}$, $\overline{u}$, $\overline{t}$, $\gamma$)
    $\mathcal{L} \leftarrow \phi(x^N) + \gamma^T \psi(x^N)$
    for $k \leftarrow 1, N$ do
        for $l \leftarrow 1, n$ do
            $x^+ \leftarrow x^k$
            $x_l^+ \leftarrow x_l^+ + \epsilon$
            $\tilde{x} \leftarrow \textsc{ivp}(\overline{q},\, \overline{u},\, \bar{t},\, x^+,\, k)$
            $\mathcal{L}^+ \leftarrow \phi(\tilde{x}^N) + \gamma^T \psi(\tilde{x}^N)$
            $p_l^k \leftarrow (\mathcal{L}^+ - \mathcal{L})/\epsilon$
        end for
    end for
    return $\overline{p}$
end function
```

Algorithm 2: Algorithm combining indirect and direct methods

3.6 Algorithms for Systems with Discontinuous States

3.6.1 Dynamic Programming

The major challenge in optimal control of switched systems is the occurrence of discrete variables. Many algorithms for the solution of optimal control problems, as they will be presented in the next chapters, use gradient information to compute an optimal trajectory. Dynamic programming however searches a discrete space for an optimal solution and uses the Bellman principle to reduce the dimension of the problem. Expanding this discrete search space by an additional discrete state, as it is present in switched systems is rather straighforward. The general idea of applying DP for the solution of hybrid optimal control problems was formulated by Branicky and Mitter [1995] and an algorithm for problems with switching cost was given by Gerdts [2012]. An introduction to DP for purely continuous systems is given in Appendix A.1.

In order to formulate the algorithm for switched DP, the problem formulation has to be altered as Bolza problem without final state constraint

$$\min_{u(t)\in\mathcal{U}_q, q(t)\in\mathcal{Q}} \int_{t_0}^{t_f} L(x(t), u(t), t)dt + \Psi(x(t_f)) \tag{3.72}$$

$$\tag{3.73}$$

as described in Remark 2.4. Similar formulations are used by Gerdts [2012]. The major difference to the DP algorithm for purely continuous system is a generalization of the state vector z and of the control vector w, which are enhanced as follows:

$$z = \begin{bmatrix} q \\ x \end{bmatrix}, \qquad w = \begin{bmatrix} \Delta q \\ u \end{bmatrix} \tag{3.74}$$

The new control input $\Delta q = q^{k+1} - q^k$ allows for controlling the switching and may take on values from the set $\mathcal{B}_q \equiv \{\Delta q : q + \Delta q \in \mathcal{Q}\}$ and consequently, the admissible concatenated controls are contained in the set $\mathcal{W}_q = \mathcal{U}_q \times \mathcal{B}_q$. Let the discrete state equation be given by

$$z^{k+1} = z^k + g(z^k, w^k, t^k) \tag{3.75}$$

$$\tag{3.76}$$

with

$$g(z^k, w^k, t^k) = \begin{bmatrix} \Delta q \\ f(z^k, w^k, t^k) \cdot h^k + \delta(\Delta q, z^k, t^k) \end{bmatrix}. \tag{3.77}$$

The value function to be minimized is defined as

$$V(x^1, t^1) = \Psi(z^N) + \sum_{k=1}^{N-1} L(z^k, w^k, t^k) \cdot h^k, \tag{3.78}$$

so that the discrete Bellman equation can be written as

$$\begin{aligned} V(z^k) &= \min_{w^k} \left\{ L(z^k, w^k, t^k) \cdot h^k + V(z^{k+1}) \right\} \\ &= \min_{w^k} \left\{ L(z^k, w^k, t^k) \cdot h^k + V(g(z^k, w^k, t^k)) \right\}. \end{aligned} \tag{3.79}$$

To outline the DP algorithm in a compact form, some definitions need to be made in advance:

$\mathbb{G}_x$ is a finite grid of the state space

$\mathbb{G}_z = \mathcal{Q} \times \mathbb{G}_x$ is a finite grid of the assembled state

$\mathbb{G}_w(q)$ is a grid of controls of the set of feasible controls $\mathcal{W}_q$.

(3.80)

The main function HYBDYNPROG has the task of storing the cost of an optimal trajectory from a given state z^k at instant k to the instant N. The respective cost as well as the extended controls w are stored for each generalized state z^k on the grid $\mathbb{G}_z$. This is done via a backwards recursion: The cost at instant $k = N$ is calculated by evaluating the final state penalty term Ψ for every grid point in $\mathbb{G}_z$. Moving backwards from $N-1$ to 1, for every state on $\mathbb{G}_z$, all the controls in $\mathbb{G}_w(q)$ are applied and the states z^+ obtained are calculated. An optimal policy is then chosen, that minimizes (3.79). To approximate $V(z^{k+1})$, an interpolation of the cost for the consecutive instant $k+1$ is performed. The optimal policy w is then stored in W and the optimal cost in V. Once the Matrices V and W are completely filled, an optimal trajectory, starting from the initial state z_0 can be recovered by interpolating the optimal policy from the matrix W. The policy is applied and the next state z^{k+1} calculated, for which the same actions are repeated. These steps are performed in the function DPTRAJECTORY.

Remark 3.9: In practice, the inner loop in the function HYBDYNPROG can be avoided by vectorization or can be accelerated via parallelization, which will yield a significant performance gain of the algorithm.

Remark 3.10: The computational demand grows linearly with the discretization rate of each continuous state. To reduce the range, in which the state needs to be discretized, it is helpful to compute the reachable set of states in advance. A method for computing this set for hybrid systems is described by Althoff et al. [2010].

As noted in the introduction of this chapter, DP suffers from the so called curse of dimensionality. The size of the grid $\mathbb{G}_z$ grows exponentially with the dimension n, which will effect both, memory use and computation time exponentially. The advantages of the dynamic programming solution, whenever applicable, are the global optimality with respect to the chosen discretization and the fact that memory demand and computing time depend only linearly on the number of discrete states Q.

```
function HYBDYNPROG
    Define $\mathbb{G}_z$, $\mathbb{G}_w(q)$, $\bar{t}$
    $V^N \leftarrow \Psi(\mathbb{G}_z)$
    for $k \leftarrow N-1, -1, 1$ do
        for all $z_i \in \mathbb{G}_z$ do
            for all $w_j \in \mathbb{G}_w(q_i)$ do
                $z^+ \leftarrow z_i + g(z_i, w_j, t^k)$
                $C_j \leftarrow L(z_i, w_j, t^k) \cdot h^k +$ INTP($\mathbb{G}_z$, $V^{k+1}$, $z^+$)
            end for
            $j^* \leftarrow \arg\min_j C_j$
            $V_i^k \leftarrow C_{j^*}$
            $W_i^k \leftarrow w_{j^*}$
        end for
    end for
    return $V, W$
end function

function DPTRAJECTORY($V$, $W$, $\mathbb{G}_z$, $x^1$, $\bar{t}$)
    $q^1 \leftarrow \arg\min_q$ INTP($\mathbb{G}_z$, $V^1$, $[q \quad x^1]^T$)
    $z^1 \leftarrow [q^1 \quad x^1]^T$
    for $k \leftarrow 1, N-1$ do
        $w^k \leftarrow$ INTP($\mathbb{G}_z$, $W^k$, $z^k$)
        $z^{k+1} \leftarrow z^k + g(z^k, w^k, t^k)$
    end for
    return $\overline{w}, \overline{z}$
end function
```

Algorithm 3: Dynamic programming algorithm

Remark 3.11: Due to the interpolation in the function DPTRAJECTORY, the discrete state may take on values $q \notin \mathbb{Z}$ and hence $q \notin \mathcal{Q}$. In this case, the discrete state needs to be rounded to the nearest $q \in \mathcal{Q}$.

3.6.2 Embedding with Relaxation of the State Jump

The main idea of the embedding method described in section 3.5.1 is the relaxation of the discrete states to obtain a purely continuous approximation of the SOCP. This approach cannot directly be transferred to the case where state jumps are present. The reason for this is that the function $\delta_{(q^-,q^+)}$ has a proper definition only on the non-convex set $\mathcal{Q} \times \mathcal{Q} \times \mathbb{R}^n \times [t_0, t_f]$.

Definition 3.4: A switching is called binary feasible from q^- to q^+, $\quad q^- \neq q^+$ if the following conditions are satisfied

$$q^-, q^+ \in \mathcal{Q} \tag{3.81}$$

$$\sigma_q^k = \begin{cases} 0, & q \in \mathcal{Q} \setminus q^- \\ 1, & q = q^- \end{cases} \tag{3.82}$$

$$\sigma_q^{k+1} = \begin{cases} 0, & q \in \mathcal{Q} \setminus q^+ \\ 1, & q = q^+ \end{cases} \tag{3.83}$$

Our aim is to formulate a relaxed initial value problem as follows:

$$\hat{x}^{k+1} = \hat{x}^k + \sum_{q=1}^{Q} \sigma_q \cdot f(\hat{x}^k, q, u^k, t^k) \cdot h^k + \hat{\delta}(\sigma^k, \sigma^{k+1}, \hat{x}^k, t^k), \tag{3.84}$$

Herein, $\hat{\delta}$ is a convexified jump function with the following properties:

- $\hat{\delta}(\sigma^k, \sigma^{k+1}, \hat{x}^k, t^k) = 0$, if $\sigma^k = \sigma^{k+1}$
- $\hat{\delta}(\sigma^k, \sigma^{k+1}, \hat{x}^k, t^k) = \delta_{(q^-,q^+)}(x^k, t^k)$, if a binary feasible switching from q^- to q^+ occurs
- $\hat{\delta}$ is defined on the convex set $[0,1]^Q \times [0,1]^Q \times \mathbb{R}^n \times [t_0, t_f]$

The first property demands that no jump occurs, when two consecutive vectors σ are equal and hence there is no switching that could lead to a jump. The second property requires the relaxed jump function to take on the same values as the original jump function for a binary feasible switching. The third property assures that σ_q^k can take on values from the convex set $[0,1]$ during the numerical optimization. Defining

$$\Delta^k = \sigma^{k+1} - \sigma^k \tag{3.85}$$

$$\Delta^+ = \max(0, \Delta^k) \tag{3.86}$$

$$\Delta^- = \max(0, -\Delta^k), \tag{3.87}$$

and as $||\Delta^+||_1$ the 1-norm of the vector Δ^k, a function $\hat{\delta}$ that fulfills the above stated requirements can be given by

$$\hat{\delta}(\Delta^k, x^k, t^k) = \begin{cases} \sum_{i=1,j=1}^{Q} \frac{\Delta_i^+ \cdot \Delta_j^-}{||\Delta^+||_1} \cdot \delta_{(i,j)}(x^k, t^k), & ||\Delta^+||_1 > 0 \\ 0, & ||\Delta^+||_1 = 0, \end{cases} \tag{3.88}$$

Remark 3.12: If $\sum_{q=1}^{Q} \sigma_q^k = 1$ and $\sum_{q=1}^{Q} \sigma_q^{k+1} = 1$, then $\sum_{q=1}^{Q} \Delta_q^k = 0$ holds.

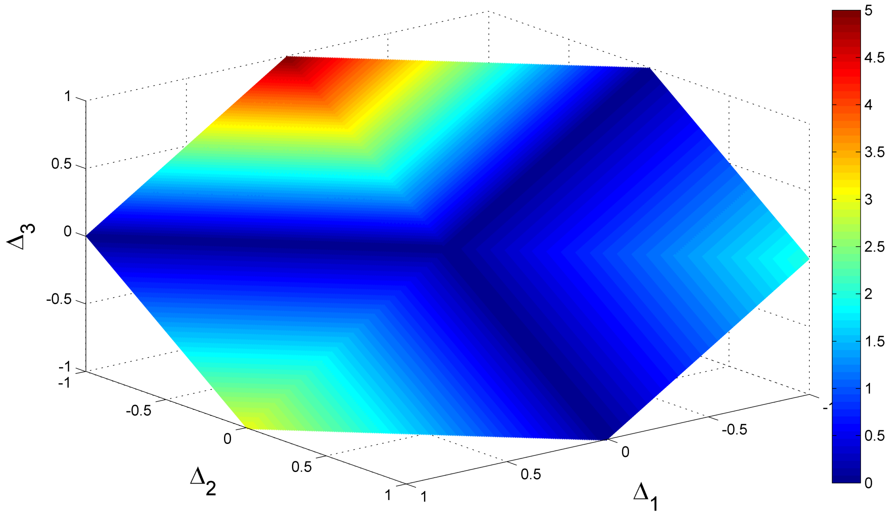

Figure 3.2: Exemplary jump function $\hat{\delta}$ for a system with three subsystems on the surface $\Delta_1 + \Delta_2 + \Delta_3 = 0$

Figure 3.2 depicts the values of the relaxed jump-function $\hat{\delta}$ on the surface $\Delta_1 + \Delta_2 + \Delta_3 = 0$ for a system with $n = 1$ and $Q = 3$ and the original jump-functions

$$\begin{aligned} \delta_{(1,2)} &= 2 \qquad & \delta_{(1,3)} &= 0 \\ \delta_{(2,1)} &= 0 \qquad & \delta_{(2,3)} &= 5 \\ \delta_{(3,1)} &= 3 \qquad & \delta_{(3,2)} &= 0. \end{aligned} \tag{3.89}$$

A switching from $q^k = 2$ to $q^{k+1} = 3$ corresponds to a switching from $\sigma^k = [0 \quad 1 \quad 0]^T$ to $\sigma^{k+1} = [0 \quad 0 \quad 1]^T$. For $\Delta^k = \sigma^{k+1} - \sigma^k = [0 \quad -1 \quad 1]$, the height of the jump $\delta_{(2,3)} = 5$ is exactly represented.

An embedded optimal control problem can then be defined as in (3.65)-(3.70) and using (3.84) to solve the initial value problem for a given parameter set

$\overline{\rho}$. The problem is passed to an appropriately chosen NLP-solver. However, the solver needs to be carefully selected. The function $\hat{\delta}$ is continuous and piecewise linear in Δ_i and therefore not continuously differentiable at $\Delta_q = 0, i \in \{1, \ldots, Q\}$ as can be noted from the partial derivatives

$$\frac{\partial(\Delta_i^+)}{\partial \Delta_j} = \begin{cases} 0, & i \neq j \\ 0, & i = j, \Delta_i < 0 \\ 1, & i = j, \Delta_i > 0. \end{cases} \tag{3.90}$$

$$\frac{\partial(\Delta_i^-)}{\partial \Delta_j} = \begin{cases} 0, & i \neq j \\ 0, & i = j, \Delta_i > 0 \\ -1, & i = j, \Delta_i < 0 \end{cases} \tag{3.91}$$

$$\frac{\partial(||\Delta^+||_1)}{\partial \Delta_i} = \begin{cases} -1, & \Delta_i < 0 \\ 1, & \Delta_i > 0. \end{cases} \tag{3.92}$$

A piecewise linear approximation was chosen to promote binary feasible solutions of the resulting nonlinear program.

Definition 3.5: The set of sub-gradients is the convex hull of the limits of gradients of a function evaluated at sequences that converge towards the point of non-differentiability (Lemaréchal [1989]).

The sets of sub-gradients for Δ^+, Δ^- and $||\Delta^+||_1$ are given by

$$\frac{\partial(\Delta_i^+)}{\partial \Delta_j} \in [0, 1], \quad i = j, \Delta_i = 0 \tag{3.93}$$

$$\frac{\partial(\Delta_i^-)}{\partial \Delta_j} \in [-1, 0], \quad i = j, \Delta_i = 0 \tag{3.94}$$

$$\frac{\partial(||\Delta^+||_1)}{\partial \Delta_i} \in [-1, 1], \quad i = j, \Delta_i = 0. \tag{3.95}$$

Optimization methods commonly used for smooth nonlinear programs may fail when the objective function or the constraints are not continuously differentiable everywhere. This failure can have several reasons, among them (Lemaréchal [1989], Sagastizábal [1997]):

- The objective function is for smooth methods usually approximated by a linear or quadratic model. These models are undefined when encountering points where the gradient does not exist and close to these points, the model can be a poor approximation

- Stopping tests, that implement the norm of the gradient at a current iterate as stopping criterion cannot be used since the gradient might not exist at an optimal point
- Calculating derivatives by finite differencing may yield poor approximations of the sub-gradient and should therefore be avoided on non-smooth problems.

Even though no proof of convergence exists, quasi-Newton methods have in many cases shown to perform well on non-smooth problems. A wide overview of literature on the successful use of these methods via numerical experiments is given in Lewis and Overton [2012], among them Lukšan and Vlček [1999]. However, the procedure for determining the gradients might need to be adapted as well due to the undefined gradients where kinks occur. As has been mentioned above, finite-differencing may yield an approximation of the gradient that is outside of the set of sub-gradients. Automatic differentiation (Rall [1981]) in combination with a black-box-method, that heuristically defines the gradient from the set of sub-gradients where a function is not continuously differentiable, can be applied to avoid this situation (Lemaréchal [1989]).

In many cases, binary unfeasible arcs will exist in the solution returned by the NLP-solver. In this case, a method for obtaining a (suboptimal) solution that fulfills the binary constraint needs to be applied. Several methods exist in the literature for solving mixed integer programs that use a relaxed problem as helper problem, such as branch and bound or cutting planes (for an overview, see Grossmann and Kravanja [1993]). In most cases, where the mixed integer program is the result of a transcription from a hybrid system, the dimension of the integer vector $\overline{\sigma}$ is rather high. Searching the entire tree is of combinatorial complexity and therefore rarely possible. We therefore apply a heuristic rounding strategy to obtain a suboptimal solution for the mixed integer program.

The tuples $\mathcal{J}$ and $\mathcal{A}$ are introduced that define constraints to be respected, when solving the NLP. The tuple $\mathcal{J}$ contains index-pairs $(k, q), k = 1, \dots, N-1, q \in \mathcal{Q}$ and the tuple $\mathcal{A}$ contains values a from the set $\{0, 1\}$. For each pair of elements from $\mathcal{J}$ and $\mathcal{A}$, the constraint

$$\sigma_q^k = a \tag{3.96}$$

is additionally imposed. Initially, both tuples are empty and the NLP is solved. The tuples are then extended in two steps: The first step is a rounding step that rounds all $\sigma_q^k > 1 - \varepsilon$ for $0 < \varepsilon \ll 1$ to 1 and all $\sigma_q^k < \varepsilon$ to 0. The rounded values are then imposed as additional constraints. In the second step, the index pair $(q, k) \notin \mathcal{J}$ is found that has the largest distance $|0.5 - \sigma_q^k|$. For this index pair, σ_q^k is also rounded to zero or one, depending on which value is closer. The second step can be repeated r times. Choosing $r \in \mathbb{N}$ higher will speed up the algorithm but may yield inferior results. The same is valid for the value of ε. With the newly defined constraints, the NLP is then solved again. The algorithm stops, when all σ_q^k are constrained. This strategy is a greedy algorithm (Turau [2009]), as it does not revise the elements in $\mathcal{J}$ and $\mathcal{A}$ once they are part of the sets. Greedy algorithms are rather local in nature but in return often much faster than algorithms that seek globally optimal solutions and often they provide satisfying results. A subsequent switching time optimization could be used to refine the solution. The pseudo-code of this algorithm can be written as follows:

```
function ROUNDOPT
    define $r \in \mathbb{N}$ and $0 < \varepsilon \ll 1$
    $\mathcal{J}, \mathcal{A} \leftarrow \emptyset$
    while not all $(k, q) \in \mathcal{J}$ do
        $\overline{u}, \overline{\sigma} \leftarrow$ NLP($\mathcal{J}$, $\mathcal{A}$)
        for all $k = 1, \ldots, N-1, q \in \mathcal{Q}$ do
            if $\sigma_q^k \geq (1-\varepsilon)$ and $(q, k) \notin \mathcal{J}$ then
                $\mathcal{J} \leftarrow \mathcal{J} \cup (k, q)$
                $\mathcal{A} \leftarrow \mathcal{A} \cup 1$
            else if $\sigma_q^k \leq \varepsilon$ and $(q, k) \notin \mathcal{J}$ then
                $\mathcal{J} \leftarrow \mathcal{J} \cup (k, q)$
                $\mathcal{A} \leftarrow \mathcal{A} \cup 0$
            end if
        end for
        for $l \leftarrow 1, r$ do
            $i, j = \max_{(q,k) \notin \mathcal{J}} |0.5 - \sigma_q^k|$
            $\mathcal{J} \leftarrow \mathcal{J} \cup (i, j)$
            if $\sigma_i^j > 0.5$ then
                $\mathcal{A} \leftarrow \mathcal{A} \cup 1$
            else
                $\mathcal{A} \leftarrow \mathcal{A} \cup 0$
            end if
        end for
    end while
    return $\overline{u}, \overline{\sigma}$
end function
```

Algorithm 4: Greedy rounding strategy

3.7 Examples

3.7.1 Continuous State without Final State Constraints

The following example was used by Xu and Antsaklis [2004] and was also applied by Shaikh [2004] to demonstrate the algorithms proposed in the respective works. The system description in it's original form is given by the two vector fields

$$f_1 = \begin{bmatrix} 0.6 & 1.2 \\ -0.8 & 3.4 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u, \quad f_2 = \begin{bmatrix} 4 & 3 \\ -1 & 0 \end{bmatrix} x + \begin{bmatrix} 2 \\ -1 \end{bmatrix} u. \tag{3.97}$$

The optimal control problem consists in finding a hybrid execution that minimizes the functional

$$J = \frac{1}{2} \cdot (x_1(t_f) - 4)^2 + \frac{1}{2} \cdot (x_2(t_f) - 2)^2 + \frac{1}{2} \cdot \int_{t_0}^{t_f} (x_2(t) - 2)^2 + u^2(t) dt \tag{3.98}$$

over the time interval $t \in [0, 2]$. The initial state is $x(t_0) = [0 \quad 2]^T$. To comply with the scheme in this thesis, an additional state is introduced to incorporate the Lagrangian term of the functional above, as has been described in section 2.5. The system can then be written as

$$f_1 = \begin{bmatrix} 0.6x_1 + 1.2x_2 + u \\ -0.8x_1 + 3.4x_2 + u \\ (x_2 - 2)^2 + u^2 \end{bmatrix}, \quad f_2 = \begin{bmatrix} 4x_1 + 3x_2 + 2u \\ -x_1 - u \\ (x_2 - 2)^2 + u^2 \end{bmatrix} \tag{3.99}$$

and the cost function as

$$\phi(x(t_f)) = \frac{1}{2} \cdot (x_1(t_f) - 4)^2 + \frac{1}{2} \cdot (x_2(t_f) - 2)^2 + \frac{1}{2} \cdot x_3(t_f). \tag{3.100}$$

The cost function value obtained by Xu and Antsaklis [2004] and Shaikh [2004] is $J \approx 9.766$. The algorithms applied to the same problem in this thesis find a very different trajectory that yields a lower value of the cost function $\phi(x(t_f)) = J$ as it can be seen in Table 3.1. As time discretization of $N = 200$ was chosen, which yields a discrete time interval of $h^k = h = 0.01s$. As discretization method, the explicit Euler discretization scheme was employed. The combined method finds the lowest value of the cost function, followed by the indirect shooting algorithm and the embedding method. By evaluating the transversality condition (2.45) for the given system, the costate p_3 is found to be $p_3(t_f) = p_3(t) = 0.5$. Therefore, only the initial costates $p_1(t_0)$ and $p_2(t_0)$ are variables to the nonlinear equation solver. For these, the condition

$$\Upsilon(p(t_0)) = \begin{bmatrix} p_1(t_f) - (x_1(t_f) - 4) \\ p_2(t_f) - (x_2(t_f) - 2) \end{bmatrix} = 0 \tag{3.101}$$

must be fulfilled. Solving this nonlinear equation is the main problem in the indirect shooting algorithm. As initial guess for the costate $p(t_0) = [1 \quad 5 \quad 0.5]^T$ was estimated, which is close to the value obtained $p(t_0) \approx [1.036 \quad 4.74 \quad 0.5]$. The convergence region of the initial guess is rather limited. For $p(t_0) = [3 \quad 7 \quad 0.5]$, the algorithm does not converge to a solution that fulfills a lower error bound $\Upsilon^T\Upsilon < 10^-5$. Even for the starting values close to the final values, a residual of $\Upsilon^T\Upsilon = 0.57288$ remains, which explains the inferior value of the cost function, compared to the much more complex combined method. As solver for the nonlinear equation, Matlab's global newton-type method FSOLVE was used and the golden section search method FMINBND was implemented to find the minimum of the Hamiltonian function.

The embedding method yields the weakest result in terms of the cost function value achieved and the trajectories $x(t)$ deviate visibly from the trajectories obtained with the other methods. However it performs very stable and finds a good solution in feasible time. Especially the requirements on the initial guess of both, control $\overline{u}$ and $\overline{\sigma}$ are rather low in this case. The control guess was set to $u^k = -2, k = 1, \ldots, N-1$, the switching variable to $\sigma^k = [0.5 \quad 0.5]^T, k = 1, \ldots, N-1$. As solver for the resulting nonlinear optimization problem, the Matlab function FMINCON was utilized with the implemented SQP-option.

The best cost function value is obtained by the combined method, yet this method requires by far the most computation time, due to the fact, that the switching sequence is altered at only one time-instant at each iteration of the main procedure. As initial guess for the switching function, $q^k = 2, k = 1, \ldots, 100, q^k = 1, k = 101, \ldots, 200$ was used. The evolution of the switching function over the iterations can be seen for a similar example in the next section in Figure 3.4.

	$\phi(x(t_f))$	$t_{comp}[s]$
Indirect shooting	5.1419	64
Embedding	5.5711	296
Combined method	5.1270	839

Table 3.1: Comparison of cost function value and computation time

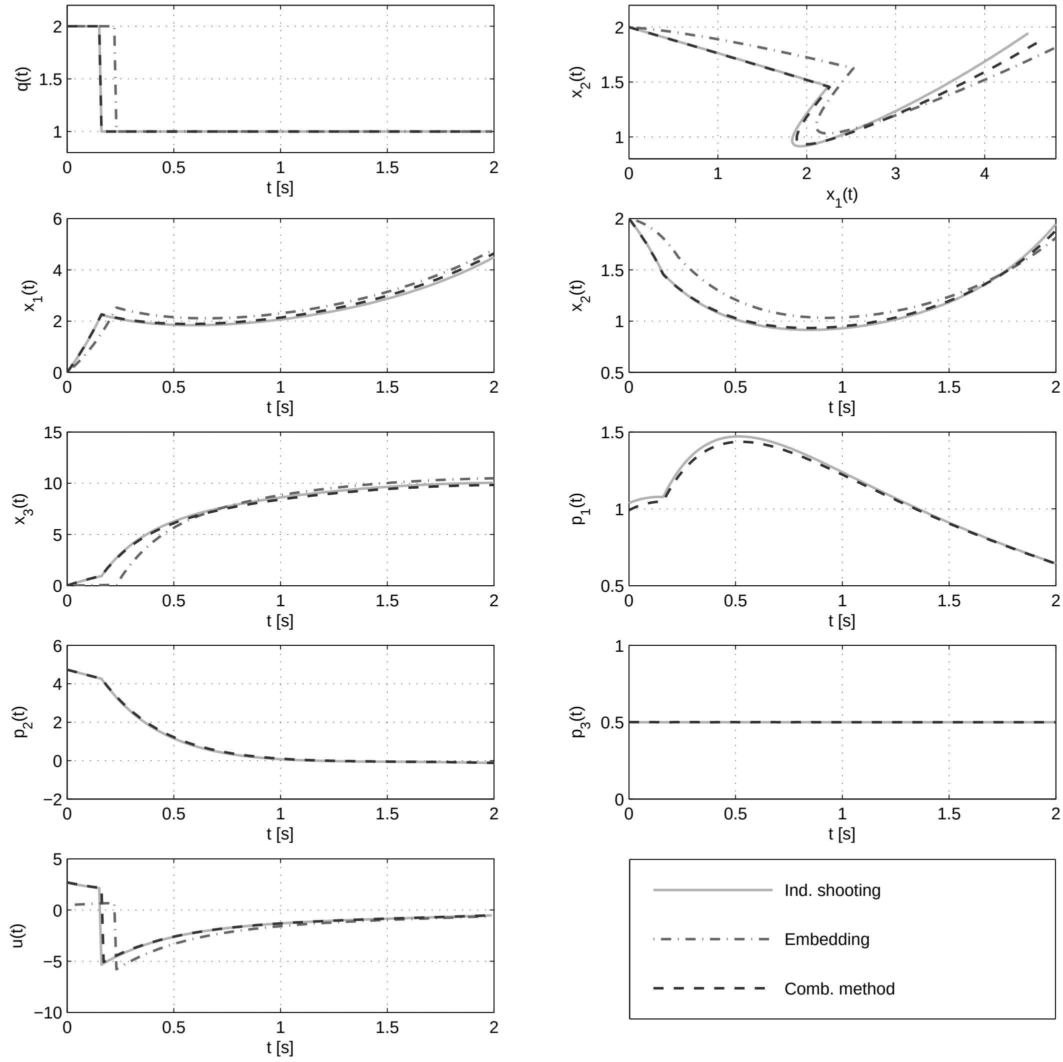

Figure 3.3: Control and state trajectories obtained for Example 1 with the indirect shooting method, embedding and the combined method

3.7.2 Continuous State with Final State Constraints

The SOCP in the previous section does not include any final state constraints. To test the algorithms on a problem with final state constraints, the soft constraints for the states $x_1(t_f)$ and $x_2(t_f)$ are removed and replaced by the final state constraints

$$\psi_1(x(t_f)) = x_1(t_f) - 4 = 0 \tag{3.102}$$

$$\psi_2(x(t_f)) = x_2(t_f) - 2 = 0. \tag{3.103}$$

The cost function is altered to

$$\phi(x(t_f)) = \frac{1}{2} \cdot x_3(t_f). \tag{3.104}$$

The nonlinear equation to solve for the indirect shooting algorithm becomes

$$\Upsilon(p(t_0)) = \begin{bmatrix} x_1(t_f) - 4 \\ x_2(t_f) - 2 \end{bmatrix} = 0. \tag{3.105}$$

As has been observed in the very similar previous problem, the indirect shooting algorithm does not fully converge to a solution that fulfills $\Upsilon^T \Upsilon < 10^{-5}$ but a small residual remains. In this case, this does not only lead to an inferior solution but to a non-fulfillment of the imposed constraints. The deviations from the final state constraints can be seen in Table 3.2. Due to the non-fulfillment of these constraints, the cost function results of the indirect shooting algorithm cannot be directly compared to the very similar results of the embedding approach and the combined method. Both of the latter methods obtain a solution that fulfills the final state constraints up to a very narrow tolerance. The difference in the cost function value is smaller than in the previous example but again, the combined method performs better than the embedding method. Looking at the computation time, embedding is again much faster than the combined method. The numerical settings, such as starting values where necessary are entirely set as in the previous example.

	$\phi(x(t_f))$	$t_{comp}[s]$	$\psi_1(x(t_f))$	$\psi_2(x(t_f))$
Indirect shooting	5.3052	32	0.0673	-0.0346
Embedding	5.3797	274	$0.16 \cdot 10^{-12}$	$0.34 \cdot 10^{-12}$
Combined method	5.3555	1099	$0.26 \cdot 10^{-9}$	$-0.1 \cdot 10^{-9}$

Table 3.2: Comparison of cost function value, computation time and fulfillment of the final state constraint

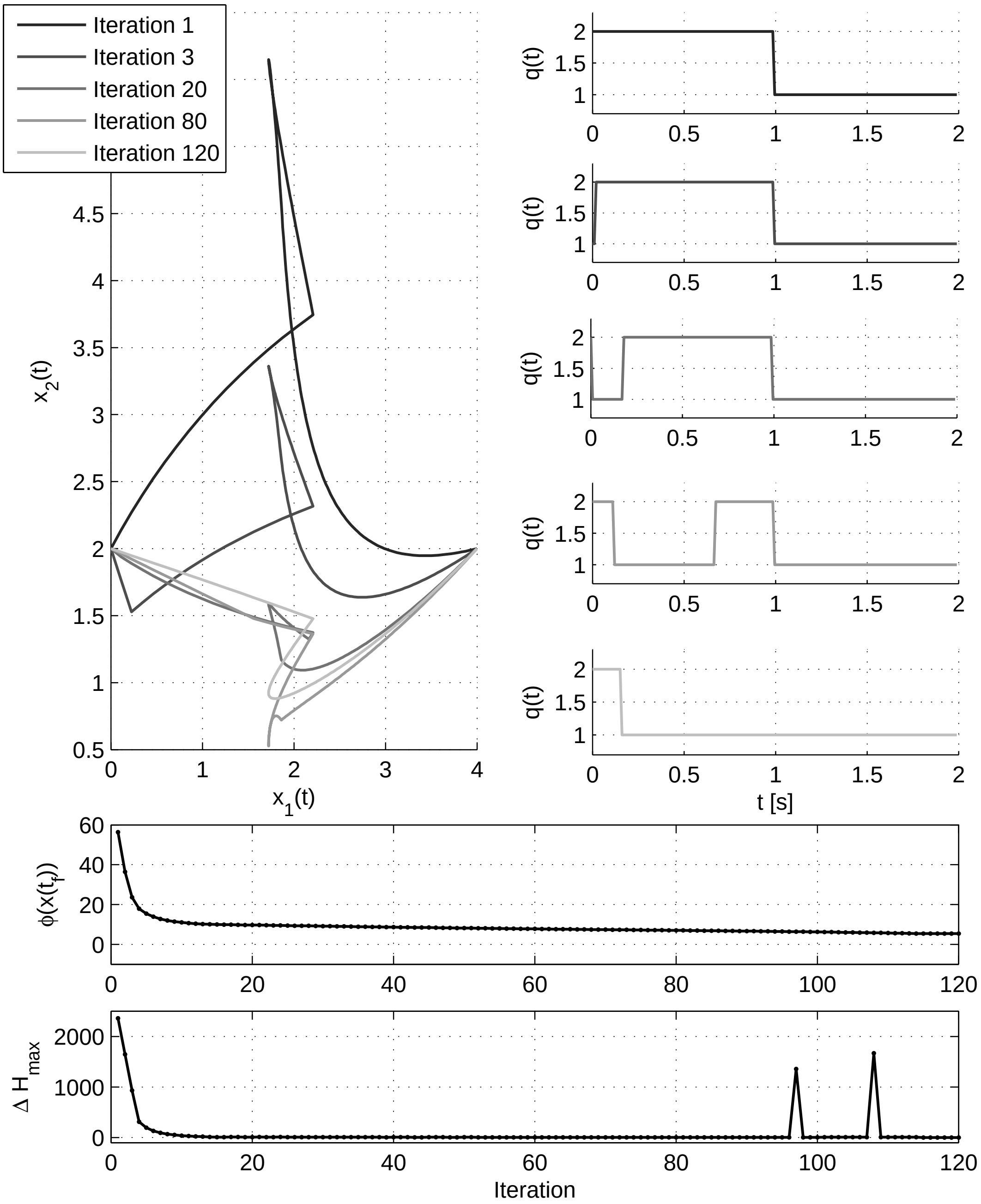

Figure 3.4: Discrete state $q(t)$ and state space $x_1(t), x_2(t)$ at different iterations of the combined method

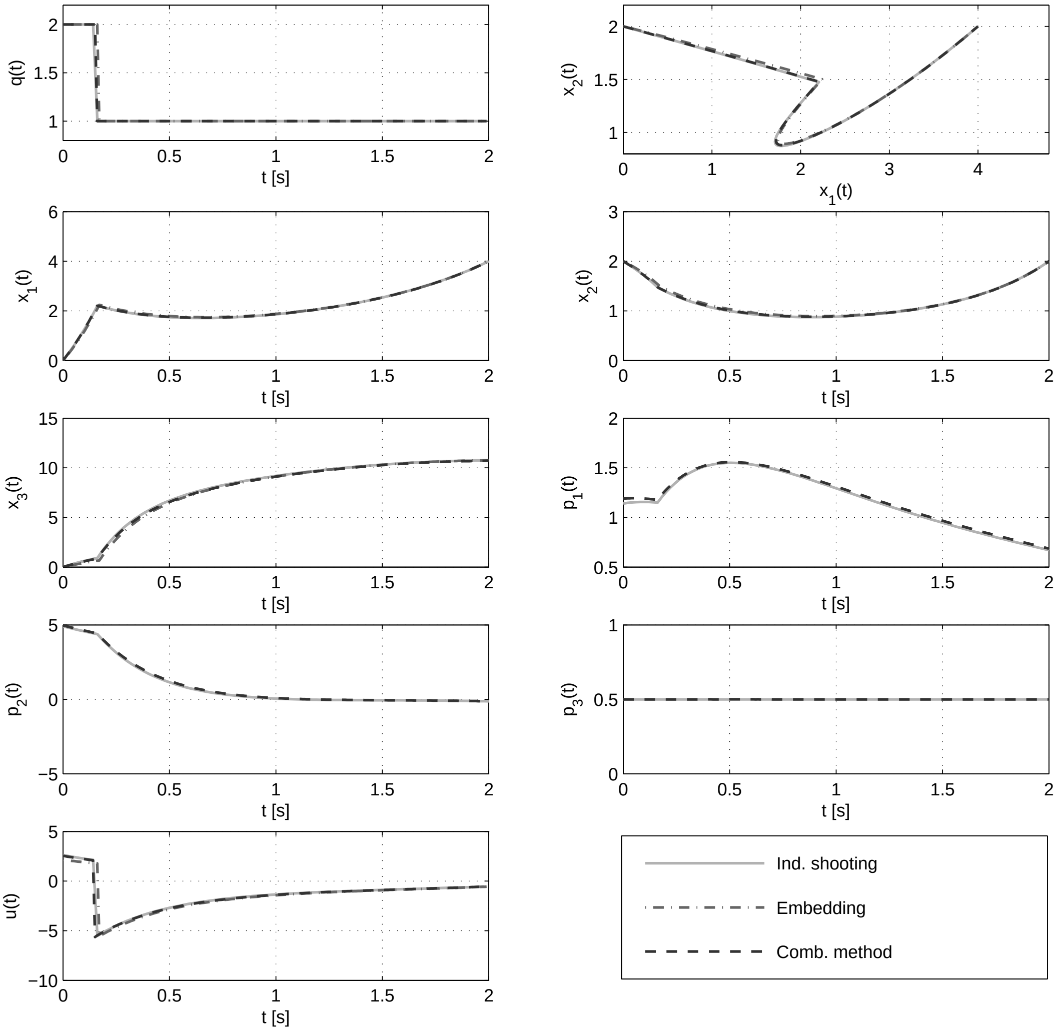

Figure 3.5: Control and state trajectories obtained for Example 2 with the indirect shooting method, embedding and the combined method

3.7.3 Discontinuous State

In the following example (introduced by Siburian [2004] and used also by Gerdts [2012]), no continuous control is present. The system is only influenced by discrete decisions. The second state exhibits jumps whenever the discrete state changes. As in the previous section, the problem was reformulated to fit into the hybrid system description used in this thesis. The system description is given by the three vector fields

$$f_1 = \begin{bmatrix} 1 \\ \left(\sin\left(\frac{\pi}{2}t\right) - x_1\right)^2 \end{bmatrix}, f_2 = \begin{bmatrix} -1 \\ \left(\sin\left(\frac{\pi}{2}t\right) - x_1\right)^2 \end{bmatrix}, f_3 = \begin{bmatrix} 2t \\ \left(\sin\left(\frac{\pi}{2}t\right) - x_1\right)^2 \end{bmatrix}, \tag{3.106}$$

and the jump function

$$\delta_{(q^-,q^+)} = \begin{cases} [0 \quad 0.0001]^T, & q^- \neq q^+ \\ [0 \quad 0]^T, & q^- = q^+. \end{cases} \tag{3.107}$$

The cost function to be minimized is

$$\phi(x(t_f)) = x_1^2(t_f) + x_2(t_f). \tag{3.108}$$

As expected, the dynamic programming algorithm obtains a slightly better solution, which is the global optimum with respect to the chosen discretization. In this example, the state x_1 is discretized in the interval $[-3, 3]$ with 4000 steps. The second state does not need to be discretized as it is a Lagrangian state and therefore only accumulates the running cost. The embedding with relaxation of the discontinuity yields a slightly inferior cost function value due to the fact that it deviates from the optimal switching sequence at $t \approx 1\,\mathrm{s}$ but it requires only a third of the computation time. This difference will be much stronger, when a higher number of continuous states is present in the problem. The solution obtained after the first nonlinear programming iteration is binary unfeasible and therefore the greedy rounding scheme is applied. Overall, 27 rounding iterations are needed, until a binary feasible solution is achieved. The evolution of the switching variable σ at different iterations is depicted in Figure 3.7.

	$\phi(x(t_f))$	$t_{comp}[s]$
Dynamic Programming	0.032286	1514
Embedding	0.032889	421

Table 3.3: Comparison of cost function value and computation time

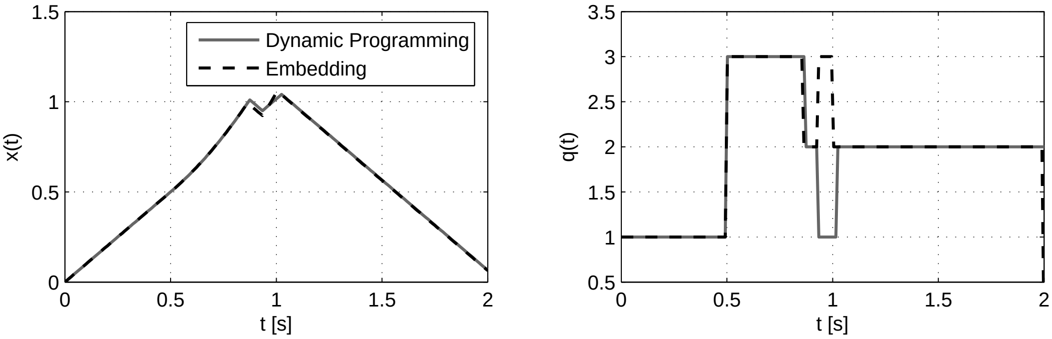

Figure 3.6: Control and state trajectories obtained for Example 3 with dynamic programming and embedding

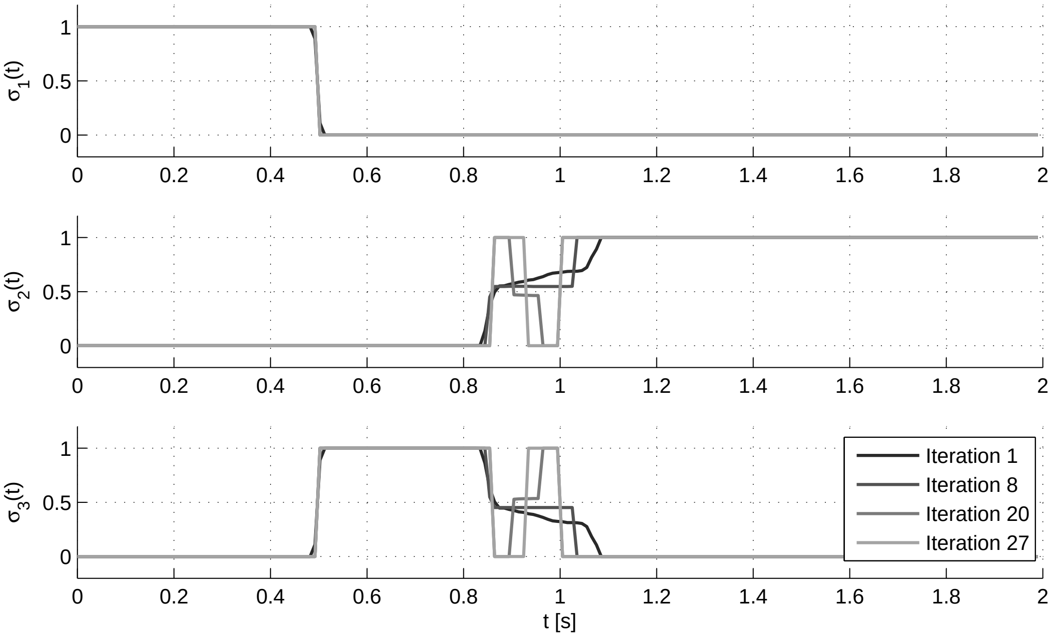

Figure 3.7: $\sigma(t)$ for different iterations of the rounding scheme

Chapter 4

Hybrid Vehicles

Increasing prices for crude oil, growing environmental concerns as well as stronger legislative requirements have caused a strong need for the development of new powertrain architectures for reducing the emissions of carbon-dioxide and other combustion products that are harmful to the environment and/or to the human health. One of those developments constitute hybrid vehicles, that have attained considerable progress over the last years. In this type of powertrain architecture, an additional energy source and an additional converter are added to the conventional powertrain with fuel tank and internal combustion engine (ICE). In most cases, the secondary energy source will be a high voltage battery and the additional converter be at least one electrical motor/generator (MG). The vehicle is then called a hybrid electric vehicle (HEV). If the vehicle can be recharged externally, using the local power grid, the vehicle can be referred to as a plug-in hybrid vehicle (PHEV). Several different layouts exist that differ in how the energy from the thermal path and the electrical path is coupled (see for instance Hofmann [2010] for an overview). The additional complexity of the powertrain brings along new degrees of freedom, that can be used to improve the overall efficiency of the powertrain and hence can be described as additional control inputs for the system. These degrees of freedom comprise continuous controls as well as discrete decisions and therefore the overall system constitutes a hybrid system and should be modeled respectively. It should be pointed out, that the term "hybrid" in "hybrid vehicle" does not necessarily refer to the existence of discrete phenomena but to the fact that at least two energy storages and converters exist. The hybrid powertrain is usually controlled by a controller structure that consists of several layers. The upper-layer control structure that is responsible for determining the desired power-split, the drive mode and the desired gear in order to improve the overall efficiency is called energy management. Because of the wide range of possible parameters, defining such an energy management can be a cumbersome task. Compared to heuristic approaches, the use of analytical methods, as for example optimal control theory, can significantly reduce this burden and yield much better results. In section 4.1, models of a hybrid vehicle will be described. Based on these models, SOCPs with different boundary conditions will be formulated in section 4.2 and the efficient application of the algorithms explained in the previous chapter will be demonstrated. In section 4.3, it will be shown how the existing hybrid optimal control theory can be used to calculate lookup-tables for a rule-based energy management automatically from a SOCP-solution. The theory can also be used to develop new functional approaches for further reducing emissions. In chapter 4.4, a predictive energy management will be detailed that solves a hybrid optimal control problem online in the vehicle.

4.1 Modeling Hybrid Vehicles as Switched Systems

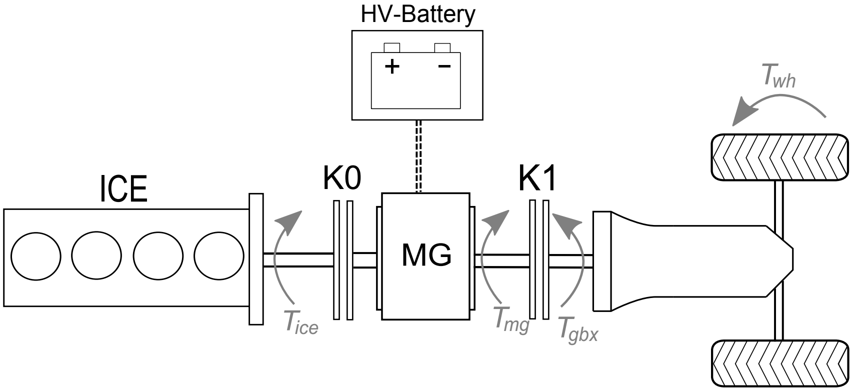

Figure 4.1: Sketch of a parallel hybrid powertrain configuration with clutches K0 and K1 and an automatic gearbox

During the operation of hybrid-vehicles, several continuous decisions as well as discrete decisions can be made to obtain a desired operation set-point. Continuous controls may for instance include a large set of parameters of the internal combustion engine, such as ignition angle, throttle, crank-shaft positions and many others. Discrete decisions can imply the gear choice of an automatic transmission, different clutch-states as well as several discrete parameters in the operation of the ICE, such as the activation of the charge-motion-valve. A model, containing all these decisions and their effect on the system would be very expensive in terms of computation time and would require a large set of information that is usually not easily available in early stages of the automotive calibration process. It is therefore advisable, to carefully select the control parameters and states needed in a model and to define the required depth of the model. The basic operation parameters of the ICE will in most cases be well defined, before the calibration process of the energy management for hybrid vehicles begins. These parameters can consequently be assumed as given and the required model dimension is significantly reduced. In this section, two models for hybrid vehicles will be described. Both models are quasi-steady in terms of some dynamics, as for instance the vehicle-motion. The velocity of the vehicle will hence not be assumed as a state but as a time-dependent boundary-condition. This helps to further reduce the dimension of the model. The first model is based on the quasi-steady torque and speed relations in a hybrid vehicle and uses the torque-split between ICE and MG as continuous control input and a drive-mode (hybrid or pure electric) as well as the gear selection as discrete control input. This model allows for the optimization of these parameters under the assumption that the vehicle is warmed-up. Yet, this can not always be assumed. Especially over the rather short cycles applied for homologation purposes, the thermodynamic influence especially on the ICE cannot be entirely disregarded, since the heat-up process amounts for a significant part of the entire drive-cycle. Additionally, some constraints apply that largely depend on thermodynamic conditions. Especially noxious emissions are strongly limited by the diverse legislations and cannot be disregarded in the calibration process. The second model includes the most important thermodynamic states and has the ignition angle as additional control variable to control the heat-up of the three-way catalytic converter (TWC).

4.1.1 Quasi-Steady Model

The following model is described for a parallel hybrid vehicle architecture with a six-speed automatic gearbox, as it can be seen in Figure 4.1. Using the model for different powertrain architectures, however, is straightforward. The model is based on a back-wards approach, meaning that a drive-cycle is assumed in advance and the torque and speed required at the wheel are calculated based on the predefined drive-cycle. Very similar models were derived by Wei [2004], Guzzella and Sciarretta [2007], de Jager et al. [2013] and others. The proposed models have in common the quasi-steady relationships between the interacting torques and a simple short circuit analogy for modeling the electrical subsystem. The highly nonlinear efficiencies or related information such as fuel mass flow or electrical power, are stored differently. Willans-line methods are are often used (Wei [2004]). In this thesis, we prefer the storage of such highly nonlinear functions as blended splines or as tensor-product spline. A method for defining these functions is given in Appendix A.2.

Based on the given trajectories $v(t) > 0\ \forall t$ and $s(t)$, where $v(t)$ is the vehicle velocity and $e(t)$ is the road inclination angle, the required wheel-torque $T_{wh}(t)$ and wheel-speed ω_{wh} can be calculated as

$$T_{wh}(t) = T_{drag}(t) + T_{incl}(t) + T_{acc}(t) \tag{4.1}$$

$$T_{drag}(t) = c_0 + c_1 \cdot v(t) + c_2 \cdot v^2(t) \tag{4.2}$$

$$T_{incl}(t) = m \cdot g \cdot e(t) \tag{4.3}$$

$$T_{acc}(t) = m \cdot \dot{v}(t) \tag{4.4}$$

$$\omega_{wh}(t) = \frac{v(t)}{r_{wh}}. \tag{4.5}$$

The terms T_{drag}, T_{incl} and T_{acc} refer to the torque required to compensate for the drag caused by rolling-resistance, aerodynamic drag, frictional losses and others, the torque due to road inclination and the torque required for a given acceleration, respectively. The parameters m and r_{wh} represent the vehicle mass and the wheel-radius. The drag coefficients c_0, c_1 and c_2 are usually determined during a coast-down experiment by fitting the polynomial to the measured velocity-profile. In automotive practice, it is more common to store the rotational speeds as revolutions per minute instead. The respective transformation can be easily performed as follows:

$$n = \omega \cdot \frac{60\,\mathrm{s}}{2\pi \cdot 1\,\mathrm{min}} \tag{4.6}$$

The gearbox input torque $T_{gbx}(t)$ and the gearbox input speed ω_{gbx} are then determined using the gear ratio ι_{gbx} and adding the frictional gearbox losses $T_{gbx,loss}$, given by a lookup-table. Both depend on the selected gear at time t, $\kappa(t) \in \{1, 2, \ldots, 6\}$:

$$T_{gbx}(t) = \frac{T_{wh}(t)}{\iota_{gbx}(\kappa(t))} + T_{gbx,loss}(\kappa(t), T_{wh}(t), \omega_{wh}(t)) \tag{4.7}$$

$$\omega_{gbx}(t) = \omega_{wh}(t) \cdot \iota_{gbx}(\kappa(t)). \tag{4.8}$$

At any time t, the gearbox input torque must be provided in sum by ICE and MG:

$$T_{gbx}(t) = T_{mg}(t) + T_{ice}(t). \tag{4.9}$$

The given powertrain allows for pure electric driving by opening the clutch K0 and therefore separating the ICE from the active powertrain. Drag losses of the ICE are avoided in electric drive mode. The active drive mode at time t can be identified by a discrete decision $\zeta \in \{0, 1\}$:

$$\zeta(t) = \begin{cases} 0, & \text{electric drive mode} \\ 1, & \text{hybrid drive mode.} \end{cases} \tag{4.10}$$

The speeds of MG and ICE can then be determined as

$$\omega_{mg}(t) = \omega_{gbx}(t) \tag{4.11}$$

$$\omega_{ice,q}(t) = \begin{cases} 0, & \zeta = 0 \\ \omega_{gbx}(t), & \zeta = 1. \end{cases} \tag{4.12}$$

The torque-split between MG and ICE in (4.9) provides a continuous degree of freedom that can be used as control input to the system. The definition of the control $u(t)$ is herein somewhat arbitrary and hence the definition

$$u(t) = T_{ice}(t) \tag{4.13}$$

is made. The set of feasible controls can then be defined as

$$\mathcal{U}_\zeta = \begin{cases} \{0\}, & \zeta = 0 \\ [T_{ice,min}(\omega_{ice}), T_{ice,max}(\omega_{ice})], & \zeta = 1. \end{cases} \tag{4.14}$$

The discrete state $q(t)$ implies the decisions $\zeta(t)$ and $\kappa(t)$. The definition

$$q(t) = 6 \cdot \zeta(t) + \kappa(t) \tag{4.15}$$

assigns a unique value $q(t) \in \{1, 2, \ldots, 12\}$ to each possible combination of drive mode and gear selection. The fuel volume flow $\dot{\beta}$ of the internal combustion engine depends on the engine speed, torque and the brake specific fuel consumption $bsfc$ for a given engine operating point:

$$\dot{\beta} = \gamma \cdot T_{ice}(u(t)) \cdot \omega_{ice,q(t)}(t) \cdot bsfc\left(T_{ice}(u(t)), \omega_{ice,q(t)}(t)\right). \tag{4.16}$$

The constant γ includes several natural constants and the $bsfc$ is here given by a mapping that is twice continuously differentiable with respect to the continuous control variable. In many cases in this thesis, it will be necessary to store measured information as a smooth function, due to the requirement of differentiability for many optimization algorithms. In Appendix A.2, a method for obtaining such functions based on a set of measured points will be explained. The electrical system can be modeled using a simple circuit analogy with internal resistance, as it can be seen in Figure 4.2. The battery power P_{bat} is the sum of the electrical power of the MG and the power P_{aux} required to supply the auxiliary devices in the vehicle that additionally require some amount of electrical energy:

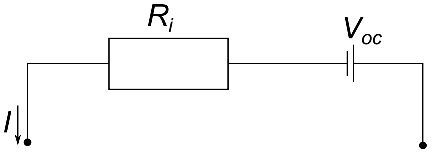

Figure 4.2: Open circuit analogy applied as battery model

$$P_{bat,q(t)}(u(t), t) = -P_{mg,q(t)}(u(t), t) - P_{aux}(t). \tag{4.17}$$

Herein the electrical power of the MG is again given by a twice differentiable function $P_{mg}(t)(T_{mg}, \omega_{mg})$. Considering the losses caused by the battery's internal resistance R_i, the sum of powers in the electrical circuit yields

$$P_{bat,q(t)}(u(t), t) - R_i(\xi(t)) \cdot I^2(t) = V_{oc}(\xi(t)) \cdot I, \tag{4.18}$$

where V_{oc} is the open circuit voltage of the battery, whose dependence of the state of charge ξ is given by a smooth function. Solving this equation for I yields

$$I_{q(t)}(\cdot) = \frac{-V_{oc}(\xi(t)) + \sqrt{V_{oc}^2(\xi(t)) + 4R_i \cdot P_{bat,q(t)}(u(t),t)}}{2R_i(\xi(t))}. \tag{4.19}$$

The internal resistance's dependence on ξ can for many modern battery types be disregarded. Only for very low values of ξ, the internal resistance rises quickly but in the remaining range, the resistance is nearly constant. The time derivative of the state of charge of the high voltage battery $\xi(t)$ is proportional to the battery current I:

$$\dot{\xi} = \frac{1}{Q_{bat}} \cdot I_{q(t)}(\xi(t), u(t), t). \tag{4.20}$$

Q_{bat} denotes the total capacity of the battery. For further reference in this thesis, the overall hybrid system model is denoted as

$$\mathcal{M}_1 : \dot{x} = \begin{bmatrix} \dot{\beta} \\ \dot{\xi} \end{bmatrix} = f_{q(t)}(x(t), u(t), t). \tag{4.21}$$

For the consideration of a vehicle's energy management, a start of the ICE can for most vehicles without significant loss of accuracy be assumed to be executed within one time-instant. An engine-start will require some additional torque of ICE and/or MG whereas an engine-stop is usually performed by cutting of fuel and therefore no additional energy is needed for the stop, nor is any kinetic energy recuperated. The additional energy for the start is modeled with the jump-function

$$\delta_{(\zeta^-,\zeta^+)} = \begin{cases} [\Delta\beta \quad \Delta\xi]^T, & \zeta^- = 0 \wedge \zeta^+ = 1 \\ 0, & \text{otherwise,} \end{cases} \tag{4.22}$$

where $\Delta\beta$ and $\Delta\xi$ are the respective measured jumps that result from an engine start.

A plot of the deviations of the model from the respective measurements is depicted in Fig. 4.3. The validation was performed for the mechanical subsystem by providing the velocity of three drive cycles (FTP, US06 and NEFZ) and comparing the measured gearbox input torque for a specified gear-sequence with the torque calculated by the model. For validation of the electrical subsystem, the MG-torque and MG-speed were fed to the model and the resulting ξ-trajectory is compared with a measured trajectory. It should be pointed out, that ξ can actually not be measured but is estimated by the battery management system based on an observer. Both, the mechanical subsystem as well as the electrical subsystem can be modeled with sufficient accuracy. An observable deviation between measured and calculated final state of charge can be noticed for the US06 cycle. Due to model errors, a solution obtained from an optimal control problem with a final state constraint must not necessarily fulfill this final state constraint in a measurement.

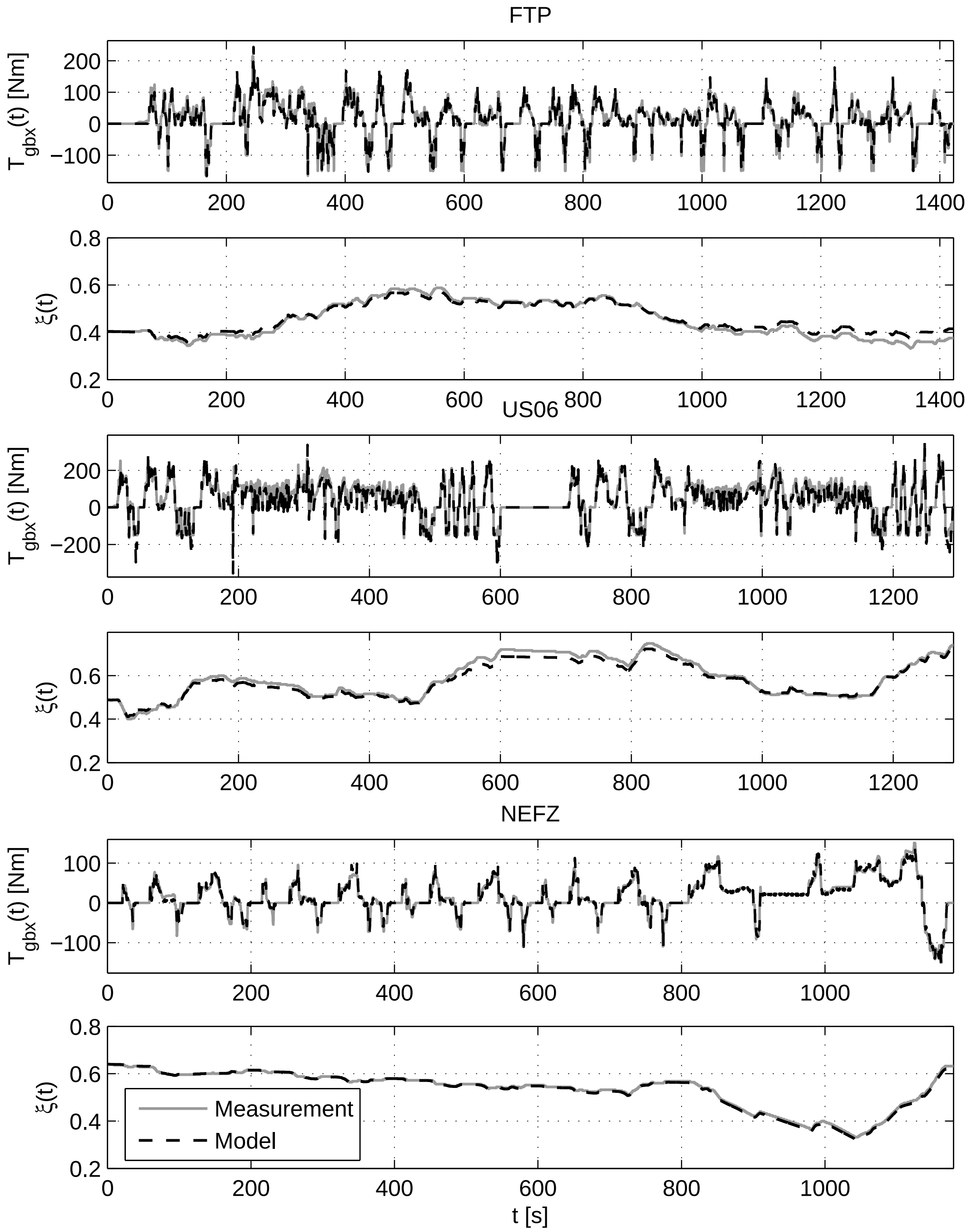

Figure 4.3: Validation of the models for the mechanical and electrical subsystem for three different drive cycles.

4.1.2 Thermodynamic Model

The model described in the previous section is only valid for a heated-up engine with cooling-water temperatures of well above 330 K. The heating-process itself cannot be further investigated using this simplified model. Therefore, in this section, a much more detailed model is described that also incorporates a thermodynamic model. In the beginning of any drive cycle, certain attention needs to be paid to heating the TWC in the exhaust system since only above a given temperature threshold, the TWC operates with acceptable efficiency. A common measure to achieve a quick heat-up is the retardation of the ignition-angle, which leads to higher exhaust enthalpies at the cost of lower combustion efficiencies. To allow for very late ignition angles, a homogen-split (HSP) injection scheme is commonly used (Basshuysen [2013]), as opposed to the standard injection scheme (STD). Additional degrees of freedom during the TWC-heating come from the states of the clutches K0 and K1. In contrast to the model in the previous section, the ICE is not necessarily switched off, when K0 is open but the engine can be operated in idling mode. The overall discrete decisions are summarized in Table 4.1.

q	K0	K1	ICE onoff	Injection scheme
1	Closed	Open	On	HSP
2	Open	Closed	On	HSP
3	Closed	Closed	On	HSP
4	Open	Closed	Off	None
5	Closed	Closed	On	STD

Table 4.1: Discrete decisions made at any time t for a given value $q(t)$

The modes for $q \in \{1, 2, 3\}$ are modes designed specifically for the TWC-heating, $q = 4$ is the pure electric drive mode and $q = 5$ denotes the conventional hybrid drive mode. To reduce the already very high model complexity, the gearbox is not regarded in the system description. Instead, measured trajectories of n_{gbx} and T_{gbx} serve as time-dependent boundary conditions. For the engine speed

$$\omega_{ice}(t) = \begin{cases} 0, & q(t) \in \{4\} \\ \omega_{idle}, & q(t) \in \{1, 2\} \\ \omega_{gbx}(t), & q(t) \in \{3, 5\} \end{cases} \tag{4.23}$$

applies and for the MG-speed

$$\omega_{mg}(t) = \begin{cases} \omega_{idle}, & q(t) \in \{1\} \\ \omega_{gbx}(t), & q(t) \in \{2,3,4,5\}. \end{cases} \tag{4.24}$$

Again, the gearbox input torque has to be supplied in sum by ICE and MG and hence the condition

$$T_{gbx}(t) = T_{ice}(t) + T_{mg}(t) \tag{4.25}$$

holds. The vector of continuous control inputs comprises the relative cylinder charge m_{cyl} and the ignition angle α:

$$u(t) = \begin{bmatrix} m_{cyl}(t) \\ \alpha(t) \end{bmatrix} \tag{4.26}$$

The convex sets of feasible controls are defined as

$$\mathcal{U}_q = \left\{ u \,\middle|\, \begin{bmatrix} m_{cyl,min}(\omega_{ice}) \\ \alpha_{min,q}(m_{cyl}, \omega_{ice}) \end{bmatrix} \leq u \leq \begin{bmatrix} m_{cyl,max}(\omega_{ice}) \\ \alpha_{max,q}(m_{cyl}, \omega_{ice}) \end{bmatrix} \right\}. \tag{4.27}$$

For HSP injection, the ignition angle can usually be retarded much more, such that the set of feasible controls is larger for $q \in \{1,2,3\}$. For electric drive mode $\mathcal{U}_q = \{0\}$ applies. Based on these control variables, the engine output torque T_{ice} is formed as follows: An optimal ignition angle is given by the smooth functions g_1 and g_2, depending on whether HSP or standard injection is active. Throughout this section, smooth mappings will be denoted as g_i.

$$\alpha_{opt}(t) = \begin{cases} g_1(m_{cyl}(t), \omega_{ice}(t)), & q(t) \in \{1,2,3\} \\ g_2(m_{cyl}(t), \omega_{ice}(t)), & q(t) \in \{5\} \end{cases} \tag{4.28}$$

Applying the optimal ignition angle would yield a theoretically optimal torque

$$T_{\alpha_{opt}}(t) = g_3(m_{cyl}(t), \omega_{ice}(t)). \tag{4.29}$$

Deviating from the optimal ignition angle leads to a decrease in combustion efficiency

$$\eta_{d\alpha}(t) = g_4(\alpha_{opt}(t) - \alpha(t)) \tag{4.30}$$

which in turn reduces the inner engine torque

$$T_{\alpha}(t) = T_{\alpha_{opt}}(t) \cdot \eta_{d\alpha}(t). \tag{4.31}$$

The engine output torque is then obtained by subtracting the temperature-dependent frictional torque T_l

$$T_{ice}(t) = T_{\alpha}(t) - T_l(\vartheta_{cw}). \tag{4.32}$$

Herein, the cooling water temperature ϑ_{cw} is used to express the temperature-dependence. In general the oil-temperature would be a better measure to express the internal friction loss but this would require the introduction of an additional state. Assuming a constant air-fuel-ratio λ, the fuel volume flow is proportional to the air-mass-flow $\dot{m}_{air}$ passing the cylinder.

$$\dot{\beta} = \frac{1}{\lambda} \cdot \gamma \cdot \dot{m}_{air}(m_{cyl}, \omega_{ice}) \tag{4.33}$$

Again, γ is a product of different natural constants. The electrical subsystem is modeled completely analogously to the preceding section with the simple circuit model. The thermodynamics of the system are modeled using a system of three temperature states

$$\vartheta(t) = \begin{bmatrix} \vartheta_{cw}(t) \\ \vartheta_{cyl}(t) \\ \vartheta_{twc}(t) \end{bmatrix} \tag{4.34}$$

describing the temperatures of the cooling water, cylinder and manifold and TWC. The intermediate state ϑ_{cyl} is a state that combines the wall-temperatures of several elements in the exhaust system, as for instance cylinder, outlet valve and manifold. Modeling each pipe-element in the exhaust system with a separate state would again lead to a very high system dimension, which is to be avoided. The raw exhaust temperature is given by a mapping

$$\vartheta_{exh}(t) = g_4(m_{cyl}(t), \omega_{ice}(t)). \tag{4.35}$$

Retarding the ignition angle α leads to an increase of the exhaust temperature and the respective correction factors obtained from measurements are given by g_5 for HSP and by g_6 for standard injection:

$$\vartheta_{exh,corr}(t) = \begin{cases} \vartheta_{exh}(t) \cdot g_5(m_{cyl}(t), \omega_{ice}(t)), & q(t) \in \{1,2,3\} \\ \vartheta_{exh}(t) \cdot g_6(m_{cyl}(t), \omega_{ice}(t)), & q(t) \in \{5\}. \end{cases} \tag{4.36}$$

Due to temperature losses to cylinder wall, exhaust valves and exhaust manifold wall, the gas temperature in the manifold is reduced to

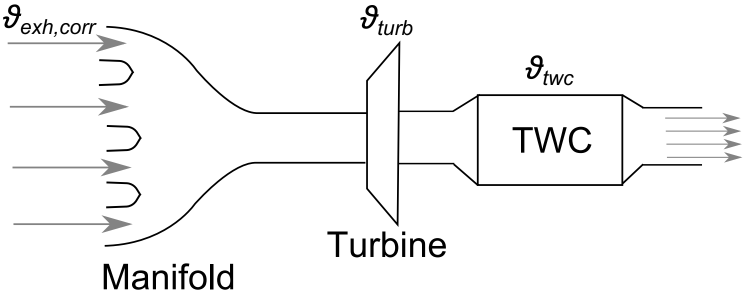

Figure 4.4: Sketch of the elements in the exhaustsystem regarded in the model

$$\vartheta_{man}(t) = \vartheta_{exh,corr}(t) - p_1 \cdot (\vartheta_{exh,corr}(t) - \vartheta_{cyl}(t)) \tag{4.37}$$

and the evolution of the temperature state ϑ_{cyl} is governed by the differential equation

$$\dot{\vartheta}_{cyl} = p_2 \cdot (\vartheta_{exh,corr}(t) - \vartheta_{cyl}(t)) - p_3 \cdot (\vartheta_{cyl}(t) - \vartheta_{cw}(t)). \tag{4.38}$$

Here and in the following, the parameters p_i include heat capacities, heat transfer coefficients and natural constants. Further, the gas temperature in the exhaust system is reduced in the turbine and the reduction is described by

$$\vartheta_{turb} = g_7(\dot{m}_{exh}, \vartheta_{man}) \cdot \vartheta_{man}. \tag{4.39}$$

It is assumed that the injected fuel has only a minor effect on the exhaust mass flow and therefore $\dot{m}_{exh} = \dot{m}_{air}$ holds. The catalytic converter temperature can finally be modeled by the differential equation

$$\dot{\vartheta}_{twc}(t) = p_4 \cdot (\vartheta_{turb}(t) - \vartheta_{twc}(t)) - p_5 \cdot (\vartheta_{twc}(t) - \vartheta_{amb}). \tag{4.40}$$

Exothermic reactions caused by unburnt hydrocarbon also play a role in the temperature increase, when the TWC-light-off temperature is reached. As we are mostly interested in the heating behavior, before the light-off temperature is reached, exothermic reactions are not considered. The time derivative of the cooling water temperature is given by the energy flow balance

$$\vartheta_{cw}(t) = p_7 \cdot (P_{fuel}(t) - P_{ice}(t) - \dot{\tau}_{exh}(t) - \dot{\tau}_{amb}(t)), \tag{4.41}$$

where P_{fuel}, $\dot{\tau}_{exh}$, $\dot{\tau}_{amb}$ denote the energy flows due to combustion, losses to the exhaust and to the environment, respectively. P_{ice} is the mechanical power of the ICE:

$$P_{fuel}(t) = H_l \cdot \dot{m}_{fuel}(t) \tag{4.42}$$

$$\dot{\tau}_{exh}(t) = p_8 \cdot \dot{m}_{exh}(t) \cdot \vartheta_{man}(t) \tag{4.43}$$

$$\dot{\tau}_{amb}(t) = p_9 \cdot (\vartheta_{cw}(t) - \vartheta_{amb}(t)) \tag{4.44}$$

$$P_{ice}(t) = T_{ice}(t) \cdot \omega_{ice}(t). \tag{4.45}$$

H_l denotes the lower heating value of the fuel and $\dot{m}_{fuel}$ the fuel mass flow. Noxious emissions modeling has been a growing research area for many years. The calculation times of most detailed emission models however, are still too high for optimization purposes. On the other hand, it is well known that quasi-steady map-based models do not provide sufficient accuracy to achieve quantitatively reliable results (Silva et al. [2006]). As a consequence, artificial states Z_i are introduced that resemble the emission components at least qualitatively. For every emission component $i \in \{1, \ldots, E\}$, a state governed by the differential equation

$$\dot{Z}_i = \dot{m}_{e,i}(m_{cyl}(t), \omega_{ice}(t), q(t)) \cdot (1 - \eta_{conv}(\vartheta_{twc}(t))) \tag{4.46}$$

with the initial state $Z_i(t_0) = 0$ is added to the system description. The function $\dot{m}_{e,i}(m_{cyl}(t), \omega_{ice}(t), q(t))$ is a map of raw emissions that may additionally depend on the injection scheme applied and η_{conv} is the temperature dependent conversion efficiency of the TWC. The conversion efficiency can be approximated by an arcus tangens function as follows (Kum et al. [2011]):

$$\eta_{conv}(\vartheta_{twc}) = \frac{1}{\pi} \cdot \left(\arctan\left(\frac{\vartheta_{twc} - \vartheta_{lo}}{s_1} \right) + \frac{\pi}{2} \right). \tag{4.47}$$

The parameters ϑ_{lo} and s_1 are the light-off temperature and a fitting parameter, respectively. The average conversion efficiency of the emission component i in the interval $[t_0, t]$ can be expressed by

$$\eta_{twc}(t) = 1 - \frac{Z_i(t)}{m_{e,i}(t)}. \tag{4.48}$$

The overall model is then given by the hybrid system

$$\mathcal{M}_2 : \dot{x} = \begin{bmatrix} \dot{\beta} \\ \dot{\xi} \\ \dot{\vartheta}_{cw} \\ \dot{\vartheta}_{cyl} \\ \dot{\vartheta}_{twc} \\ \dot{Z}_1 \\ \vdots \\ \dot{Z}_E \\ \dot{m}_1 \\ \vdots \\ \dot{m}_E \end{bmatrix} = f_{q(t)}(x(t), u(t), t). \tag{4.49}$$

Figure 4.5 shows a comparison of the temperature trajectories ϑ_{twc} and ϑ_{cw} with the respective measured trajectories obtained for measured inputs of $q(t), m_{cyl}(t)$ and $\alpha(t)$. The heat-up procedure can be modeled with high accuracy. At higher temperatures $\vartheta_{twc} \gg 550\,\mathrm{K}$, the exothermic reactions in the catalytic converter have a significant impact on the TWC-temperature. The modeling of these chemical reactions however is rather difficult. Since the heat-up process itself up to this temperature is of the biggest interest, the model is still sufficiently reliable.

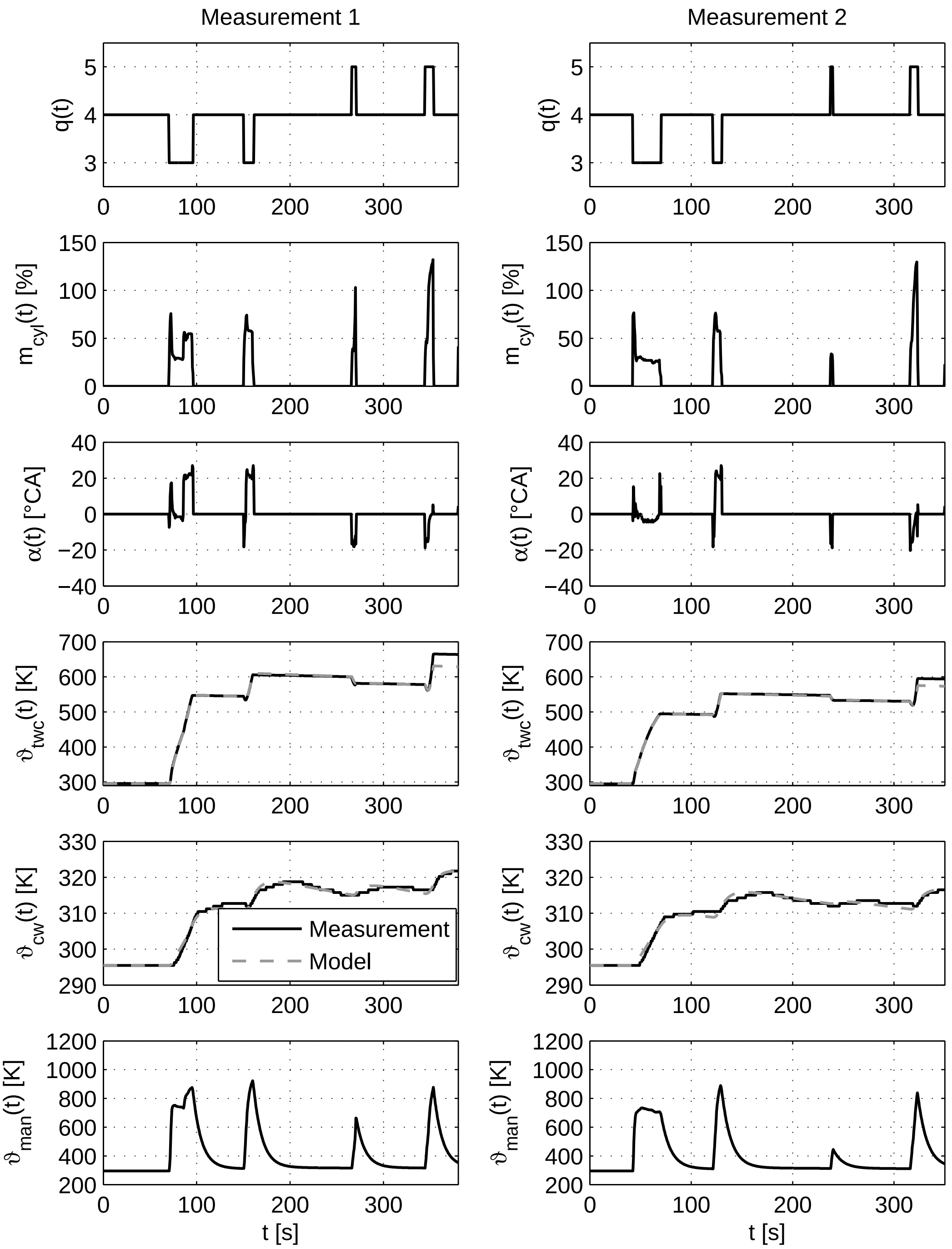

Figure 4.5: Comparison of the results of the thermodynamic model compared to two measurements

4.2 Solution of SOCPs for Hybrid Vehicles

The aim of an energy management system for hybrid vehicles is to control the degrees of freedom present in a given architecture, to minimize a certain cost function. In the majority of the cases, the cost function will resemble the fuel consumption over a drive cycle. Additional constraints can be present that shall impose limits on noxious emissions or on the state of charge at the end of the drive cycle. The energy management determines the set-point values for each element of the powertrain that can be controlled up to some degree. A lower-layer control structure is responsible for realizing these set-point values. Computational methods have become inalienable in the calibration process due to the high number of calibration parameters available in today's electronic control units. A first step towards the definition of an energy management system can be the solution of a respectively formulated optimal control problem. One large field of research is the use of optimal control theory to define such energy managements. The use of Pontryagin's Minimum Principle to solve an optimal control problem was proposed by Rousseau et al. [2007], Serrao and Rizzoni [2008], Kim et al. [2009], Stockar et al. [2011], Kim et al. [2011] and others. These works share the common approach to model the operation of a hybrid vehicle with a model similar to the quasi-steady model in section 4.1.1 and reduce the problem to finding the initial value of the costate. In Serrao and Rizzoni [2008], battery aging is additionally included in the problem formulation. A closely related field of research are Equivalent Fuel Consumption Minimization Strategies (ECMS), initially proposed by Paganelli et al. [2002] and enhanced by Sciarretta et al. [2004], Chen and Salman [2005], Musardo et al. [2005], among others. The strategy is based on the instantaneous minimization of a sum of fuel mass flow and weighted electrical power. This sum equals the Hamiltonian function for the PMP, if the model is set up correspondingly. The weighting factor for the electrical power is called equivalence factor and is usually assumed to be constant. In the Hamiltonian function, the costate can be interpreted as weighting factor. It will be shown in section 4.2.2 that the costate can be assumed to be constant without significant loss of accuracy. The indirect shooting method can be applied to determine the equivalence factor analogously to the initial costate value. The relationship between ECMS and the PMP was described by Kim et al. [2011]. Dynamic programming is also widely applied to solve the problem for the quasi-steady model, often to compare the results of ECMS/PMP with the global optimum with respect to a chosen discretization (Karbowski et al. [2006], Rousseau et al. [2007], de Jager et al. [2013]). In the work of Kum et al. [2011], dynamic programming is used to solve a much more complex optimal control problem considering thermodynamic and emission constraints. Discrete phenomena have attained less attention in the literature on energy management. The use of embedding for solving a system with two modes is encouraged by Uthaichana et al. [2008]. Gear changes as well as engine starts and their respective costs are included in the two-stage algorithm proposed by Nüesch et al. [2014]. The development of algorithms for the solution of SOCPs in the recent years however, allows for the efficient solution of many problems. The choice of the applied algorithm must be made specifically for each model, as will be shown in this section.

4.2.1 Problem Formulations

Given a vehicle configuration and a corresponding model, the optimal operation of a HEV over a given drive cycle can be formulated as SOCP. If a solution to this problem can be found, the solution will contain the optimal continuous controls, the optimal switching sequence and the optimal state trajectories over the time interval $[t_0, t_f]$. At first glance, the solution will be cycle specific and involve a feed-forward control only. An exception is dynamic programming, where the value function $V(x^k, t^k)$ and the control function $W(x^k, t^k)$ contain the optimal cost and controls for any state on the grid $\mathbb{G}_z$ at time instant t^k. It will later be shown that even though the problem was solved for one specific cycle, the solution can still be used for the general calibration of hybrid vehicles and for the development of predictive energy managements. Different drive cycles are used worldwide to asses the fulfillment of legislative requirements of a specific vehicle. Thus, the optimal operation of HEVs over these drive cycles is of particular interest. In Europe, the New European Drive Cycle (NEDC) is used for this purpose, whereas in the USA the Federal Test Procedure (FTP) is applied. Each of these drive-cycles is defined by a given trajectory $v(t), t \in [0, t_{cyc}]$, where t_{cyc} denotes the total length of the drive-cycle. When the vehicle to be tested possesses a high-capacity electric energy storage, it is usually demanded, that the amount of energy stored in the storage be balanced over the entire cycle:

$$\xi(0) - \xi(t_{cyc}) \equiv 0. \tag{4.50}$$

Therefore this condition is regarded as a final state constraint in most of the problem formulations. Four different SOCPs are defined in this section with

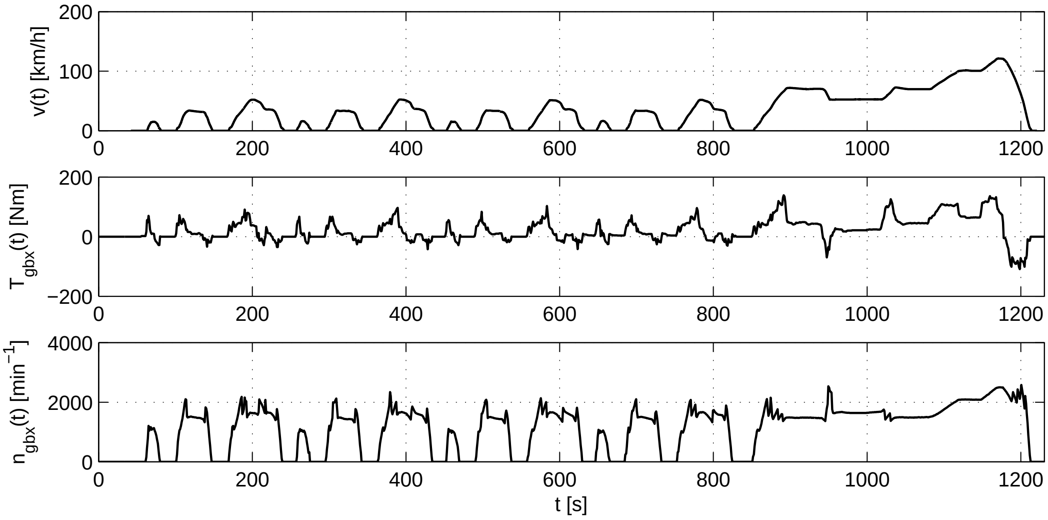

Figure 4.6: New European Drive Cycle velocity profile and measured gearbox input torque and input shaft speed for a given gear sequence

varying model complexity. Problems $\mathcal{P}_1$-$\mathcal{P}_3$ are based on the NEDC drive cycle. A solution to problem $\mathcal{P}_1$ contains the optimal drive-mode decision $\zeta(t)$ and the optimal torque-split between MG and ICE. The gear selection $\kappa(t)$ is performed before the optimization via a predefined strategy. State-jumps are not considered in $\mathcal{P}_1$.

$$\mathcal{P}_1 := \begin{cases} \min_{\zeta(t)\in\{0,1\},u(t)\in\mathcal{U}_q} \beta(t_f) \\ \text{subject to } \mathcal{M}_1 \\ \psi(x(t_f)) = 0.5 - \xi(t_f) = 0 \\ \xi(t_0) = 0.5 \\ \beta(t_0) = 0 \\ \delta_{(\zeta^-,\zeta^+)} = 0 \ \forall\ \zeta^-, \zeta^+ \in \{0,1\}. \end{cases} \tag{4.51}$$

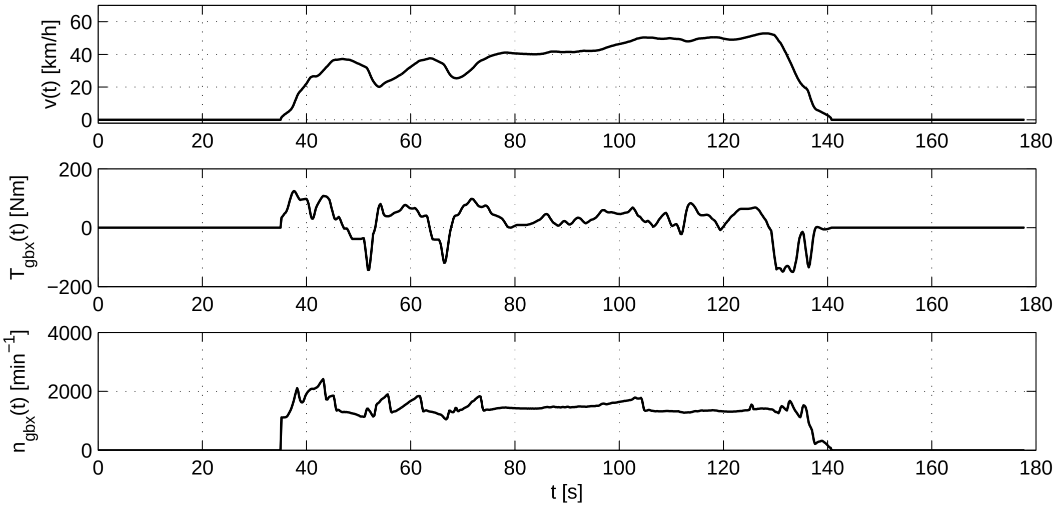

Figure 4.7: First hill of the Federal Test Procedure: Velocity profile and measured gearbox input torque and input shaft speed for a given gear sequence

Problem $\mathcal{P}_2$ additionally aims at finding the optimal gear selection $\kappa(t)$. The gear selection found will often be rather unrealistic, due to many additional constraints, such as driving comfort, that are hard to account for in a mathematical model. Problem $\mathcal{P}_2$ is therefore of rather academic interest and has only minor practical relevance. As in $\mathcal{P}_1$ state-jumps are disregarded.

$$\mathcal{P}_2 := \begin{cases} \min_{\zeta\in\{0,1\},\kappa\in\{1,\ldots,6\},u(t)\in\mathcal{U}_q} \beta(t_f) \\ \text{subject to } \mathcal{M}_1 \\ \psi(x(t_f)) = 0.5 - \xi(t_f) = 0 \\ \xi(t_0) = 0.5 \\ \beta(t_0) = 0 \\ \delta_{(\zeta^-,\zeta^+)} = 0 \ \forall\ \zeta^-,\zeta^+ \in \{0,1\}. \end{cases} \tag{4.52}$$

Problem $\mathcal{P}_3$ is similar to $\mathcal{P}_1$ except for the fact that a state jump is considered, as the ICE starts. An engine start will here lead to a decrease in the state of charge ξ and to an increase in the fuel consumption β:

$$\mathcal{P}_3 := \begin{cases} \min_{\zeta(t)\in\{0,1\},u(t)\in\mathcal{U}_q} \beta(t_f) \\ \text{subject to } \mathcal{M}_1 \\ \psi(x(t_f)) = 0.5 - \xi(t_f) = 0 \\ \xi(t_0) = 0.5 \\ \beta(t_0) = 0 \\ \delta_{(\zeta^-,\zeta^+)} \text{ from (4.22).} \end{cases} \tag{4.53}$$

Problem $\mathcal{P}_4$ uses the much more complex model $\mathcal{M}_2$ and is formulated over the first phase of the FTP cycle as it is shown in Fig. 4.7. The aim is to find a strategy for optimally heating up the engine and the exhaust system. The boundary conditions for the SOCP are formulated differently. It is imposed that the ICE be heated up to at least 353 K at time t_f. Not the entire driving cycle is regarded but t_f is chosen such that $0 < t_f < t_{cyc}$. It is therefore not necessary, to demand a fixed value $\xi(t_f)$ as ξ needs to be balanced at time t_{cyc} but not for $t \in (0, t_f)$. However, if the state of charge $\xi(t_f)$ has a low value, more charging has to be done over the remaining cycle with $t \in (t_f, t_{cyc}]$. The function $B_{rem}(\xi(t_f) - \xi(t_{cyc}))$ gives the fuel consumption for the remaining cycle, depending on the difference between $\xi(t_f)$ and the target value $\xi(t_{cyc})$. The function is shown in Fig. 4.8. Since it is demanded that the ICE be heated up sufficiently at t_f, the function can be easily set up by solving a SOCP for the time range $(t_f, t_{cyc}]$ with the simpler model $\mathcal{M}_1$. This is done on a grid of $\xi(t_f)$. Between the grid values, a cubic spline interpolation is performed. The cost function is then defined as

$$B(x(t_f)) = \beta(t_f) + B_{rem}(\xi(t_f)). \tag{4.54}$$

One emission component is regarded, which is resembled by the state $Z(t)$ and an upper bound on $Z(t_f)$ is imposed. State-jumps are not considered:

$$\mathcal{P}_4 := \begin{cases} \min_{q(t)\in\{1,\ldots,5\},u(t)\in\mathcal{U}_q} B(x(t_f)) \\ \text{subject to } \mathcal{M}_2 \\ \psi_1(x(t_f)) = 353\,\mathrm{K} - \vartheta_{cw}(t_f) \leq 0 \\ \psi_2(x(t_f)) = Z(t_f) - Z_{max} \leq 0 \\ \xi(t_0) = 0.345 \\ \beta(t_0) = 0 \\ \vartheta_{cw}(t_0) = \vartheta_{cyl}(t_0) = \vartheta_{twc} = 293\,\mathrm{K}. \end{cases} \tag{4.55}$$

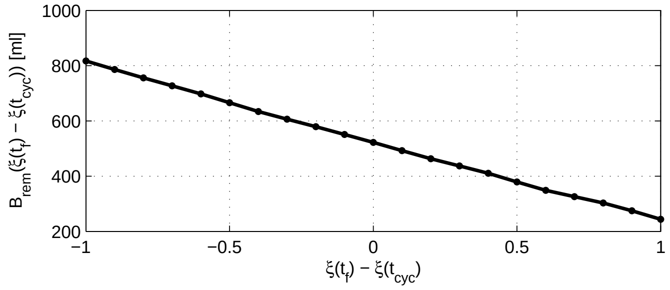

Figure 4.8: Function $B_{rem}(\xi(t_f) - \xi(t_{cyc}))$ that provides the optimal fuel consumption for the remaining drive cycle $t \in (t_f, t_{cyc}]$, depending on the difference $\xi(t_f) - \xi(t_{cyc})$

4.2.2 Evaluation of Necessary Conditions

Evaluating the necessary conditions is helpful as it can provide useful insight into the problem structure and hence facilitate the choice of an appropriate algorithm for the solution of the SOCP. Especially for the problems $\mathcal{P}_1$-$\mathcal{P}_3$, the necessary conditions yield some interesting results that can be used to simplify the problem. The Hamiltonian function for the model $\mathcal{M}_1$ is defined as

$$\begin{aligned} \mathcal{H}_{q(t)}(x(t), p(t), u(t), t) &= p^T(t) \cdot f_{q(t)}(x(t), u(t), t) \\ &= p_1(t) \cdot \dot{\beta}_{q(t)}(u(t), t) + p_2(t) \cdot \dot{\xi}_{q(t)}(x(t), u(t), t). \end{aligned} \tag{4.56}$$

Evaluating the transversality condition, the time derivative of the costates results to

$$\dot{p}_1 = -\frac{\partial \mathcal{H}_{q(t)}(x(t), p(t), u(t), t)}{\partial \beta(t)} = 0 \tag{4.57}$$

$$\dot{p}_2 = -\frac{\partial \mathcal{H}_{q(t)}(x(t), p(t), u(t), t)}{\partial \xi(t)} = -\frac{1}{Q_{bat}} \frac{\partial I_{q(t)}(x(t), u(t), t)}{\partial \xi(t)}. \tag{4.58}$$

Hence, the first costate remains constant. The total variation over a given interval $[t_0, t_f]$ of the second costate will also be small, since the last term in the second costate's time derivative yields

$$\frac{\partial I_{q(t)}(x(t), u(t), t)}{\partial \xi(t)} = \frac{\partial I_{q(t)}(x(t), u(t), t)}{\partial V_{oc}(t)} \cdot \frac{dV_{oc}(\xi(t))}{d\xi(t)}. \tag{4.59}$$

For modern batteries, the last term in this equation takes on very small values, caused by the minor dependence of the open-circuit voltage of the state of charge. Due to the fact that $\frac{\partial \delta}{\partial x} = 0$ and $\frac{\partial \delta}{\partial t} = 0$, the Hamiltonian as well as the costate are continuous on a switching:

$$p(t_j^+) = p(t_j^-) \tag{4.60}$$

$$\mathcal{H}_{q(t_j^+)}(t_j^+) = \mathcal{H}_{q(t_j^-)}(t_j^-). \tag{4.61}$$

The final value of the first costate can then be calculated as

$$p_1(t_f) = \frac{\partial \left(\phi(x(t_f) + \mu^T \psi(x(t_f))\right)}{\partial \beta(t_f)} = 1. \tag{4.62}$$

Thus the first costate can be disregarded in the Hamiltonian and with the definition

$$p(t) \equiv p_2(t), \tag{4.63}$$

$\mathcal{H}$ can be written as

$$\mathcal{H}_{q(t)}(x(t), p(t), u(t), t) = \dot{\beta}_{q(t)}(u(t), t) + p(t) \cdot \dot{\xi}_{q(t)}(x(t), u(t), t). \tag{4.64}$$

If state jumps are not considered ($\delta_{(\zeta^-, \zeta^+)} = 0 \ \forall \ \zeta^-, \zeta^+ \in \{0, 1\}$), the Hamiltonian needs to be minimized by both continuous control $u(t)$ and discrete state $q(t)$ at almost any time t. If state jumps may occur, this is not a necessary condition. In this case the Hamiltonian needs to be minimized by the continuous control only at almost any t.

4.2.3 Solving the SOCPs

Embedding and indirect shooting are both appropriate methods for solving the optimal control problem $\mathcal{P}_1$. The indirect shooting method strongly benefits from the fact, that there is only one costate and that this costate remains nearly constant over the entire time-interval. The burden of finding an initial guess $p(t_0)$ that is in acceptable proximity to the true initial value, that leads to fulfillment of the final state constraint (4.51), is reduced, since for different vehicle configurations and different cycles, the costate is in similar ranges. Consequently the indirect shooting algorithm is a very efficient and easy to apply solution method for problem $\mathcal{P}_1$. Nevertheless, special care has to be taken, when minimizing the Hamiltonian function. The Hamiltonian may exhibit multiple local minima. One way, to reduce the chance of converging to a local minimum in the solution procedure is to calculate the Hamiltonian function value on a rough grid first and then use the lowest value as an initial value for the minimization procedure. The equation $\Upsilon(p(t_0)) = 0$ has only one variable and can consequently be solved efficiently using regula-falsi methods. In this case, the Pegasus method (Dowell and Jarratt [1972]) was applied. Embedding attains comparable results in terms of the cost function value but requires significantly more computation time, as can be seen in Table 4.2. For all methods and all problems $\mathcal{P}_1$-$\mathcal{P}_4$, a time discretization with $h_k = h = 0.5\,\mathrm{s}$ was used. DP can also be employed to $\mathcal{P}_1$ and yields nearly equal results as the indirect shooting methods but requires much more computation time. In this case, DP should only be employed to validate a solution obtained with the indirect shooting approach. The state space of ξ was discretized with 10000 elements in the interval $[0.4, 0.6]$.

The solutions obtained with all methods exhibit frequent switching, due to the neglected jumps in fuel consumption and state of charge caused by a switching. Even if the jumps in fuel and electrical energy storage are of minor importance, this is an unrealistic scenario as frequent engine starts will deteriorate the driving comfort and augment the engine wear. Nevertheless, if the indirect shooting method is applied, a realistic estimator of the initial costate is obtained that can be useful for the calibration of hybrid vehicles, as will be shown in section 4.3.

	$\phi(x(t_f))$[ml]	t_{comp}[s]
Indirect shooting	496.9	29.39
Embedding	497.9	1899.4
Dynamic programming	496.9	4386.5

Table 4.2: Comparison of cost function value and computation time for $\mathcal{P}_1$

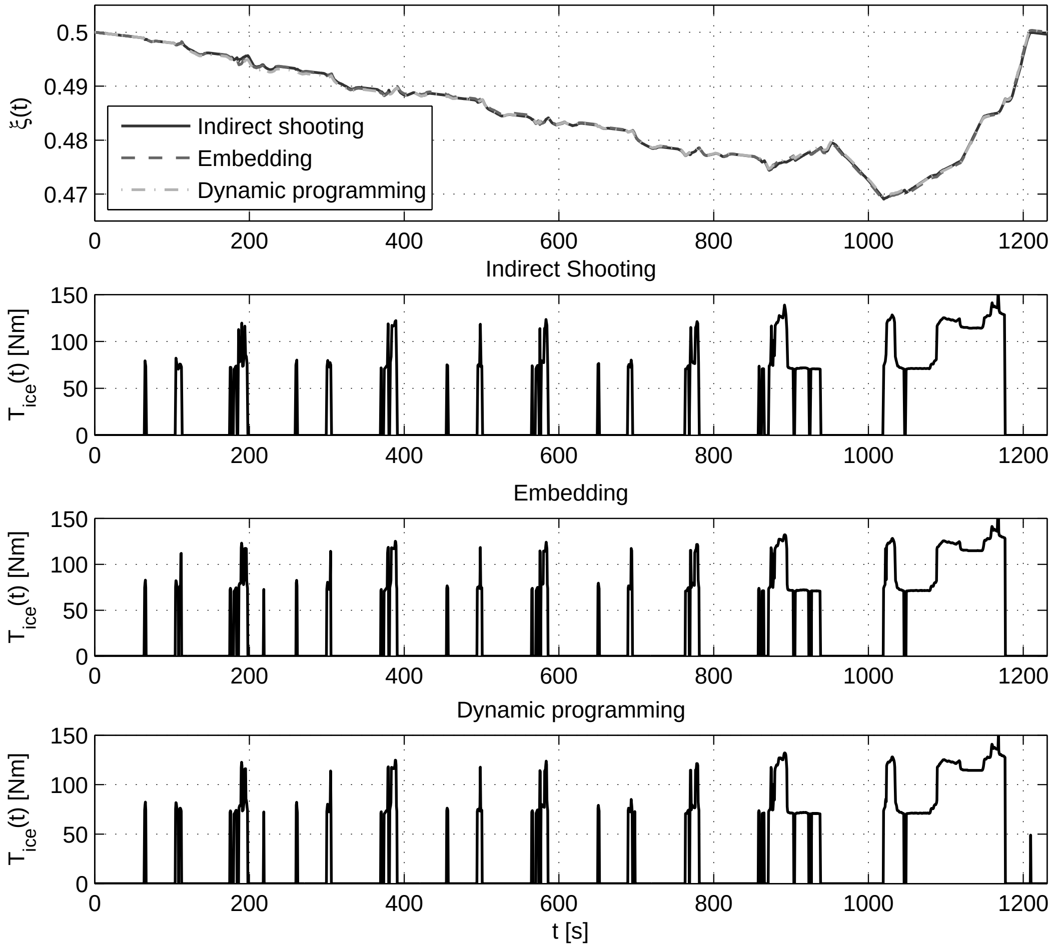

Figure 4.9: Solution of problem $\mathcal{P}_1$ obtained with the indirect shooting method and the embedding method

Solving $\mathcal{P}_2$ with embedding is complicated by the high dimension of the discrete state vector $Q = 12$, which leads to a very large size of the optimization vector and to a high probability of singular arcs in the solution, where the solution is binary unfeasible. Indirect shooting however, can be applied reliably. The computation time increases only linearly with the number of discrete states Q. The solution of $\mathcal{P}_2$ is shown in Figure 4.10. It was obtained in a computation time of $t_{comp} = 150\,\mathrm{s}$ and a cost function value of $\phi(x(t_f)) = 496.1\,\mathrm{ml}$ was achieved. It can be noted that there are very frequent gear changes and that often the highest gear possible is selected during the optimization. There are many more constraints, such as driving comfort, that cannot be accounted for with the proposed model and are therefore not included in the optimization.

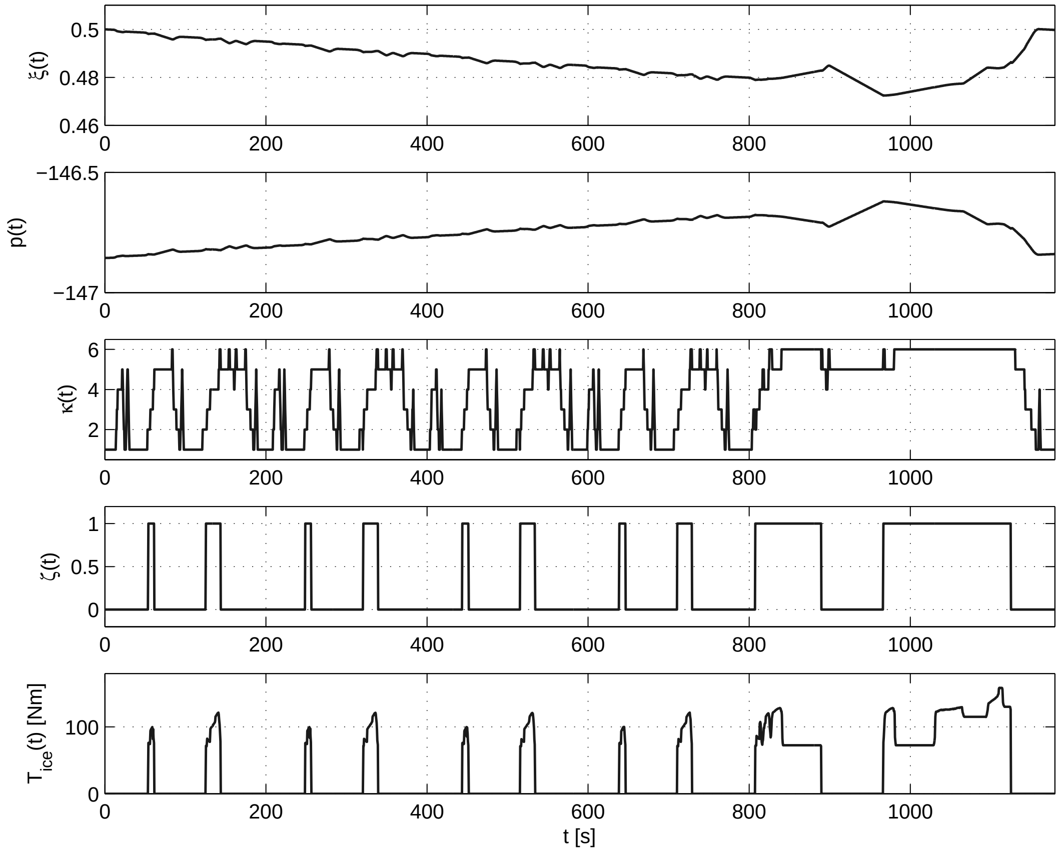

Figure 4.10: Solution of problem $\mathcal{P}_2$ obtained with the indirect shooting method

Due to the discontinuities in the state trajectory on a switching, problem $\mathcal{P}_3$ cannot be solved using methods for continuous states as the indirect shooting method, the embedding approach or the combined method. DP is an adequate solution method for $\mathcal{P}_3$, as the dimension of the continuous state vector can be reduced to 1. The state β is a Lagrangian state and hence does not need to be gridded. Consequently, the global optimum with respect to a chosen discretization of the state ξ can be found by DP in a feasible amount of time. Embedding with relaxation of the state jump and the greedy rounding scheme also yields comparable results but this method has in this case a slightly longer computation time. Yet, it has more potential for reducing the computation time by transcribing the problem as a collocation problem instead of a direct shooting problem and by exploiting the sparse problem structure. Additionally, the computation time can be further reduced, if automatic differentiation is used instead of finite differencing, to calculate the gradients at each iteration of the NLP-solver.

A comparison of the results can be seen in Fig. 4.11. In the solution obtained by the embedding approach, the third ICE start is omitted. To compensate for this, the ICE torques are chosen higher, whenever the ICE is started. Except for the omitted start, the remaining switching sequence is identical. The difference in the fuel consumption is within acceptable bounds. Figure 4.12 depicts the evolution of the relaxed switching variable σ_1, where $\sigma_1 = 1$ represents hybrid drive mode, over the iterations of the greedy rounding scheme. After iteration 40, σ_1 takes on values $\sigma_1 \approx 0.49$ at $t \approx 580\,\text{s}$. This demonstrates that the switching is in this case nearly singular, meaning that the decision of an engine start at this point has only a minor effect on the total fuel consumption. The height of the jump at an engine start was assumed to be $[\Delta\beta \quad \Delta\xi]^T = [0.65\,\text{ml} \quad -0.00013]^T$. The state discretization for the dynamic programming method was chosen as in $\mathcal{P}_1$.

	$\phi(x(t_f))$[ml]	t_{comp}[s]
Dynamic programming	505.75	9922
Embedding	506.02	11563

Table 4.3: Comparison of cost function value and computation time for $\mathcal{P}_3$

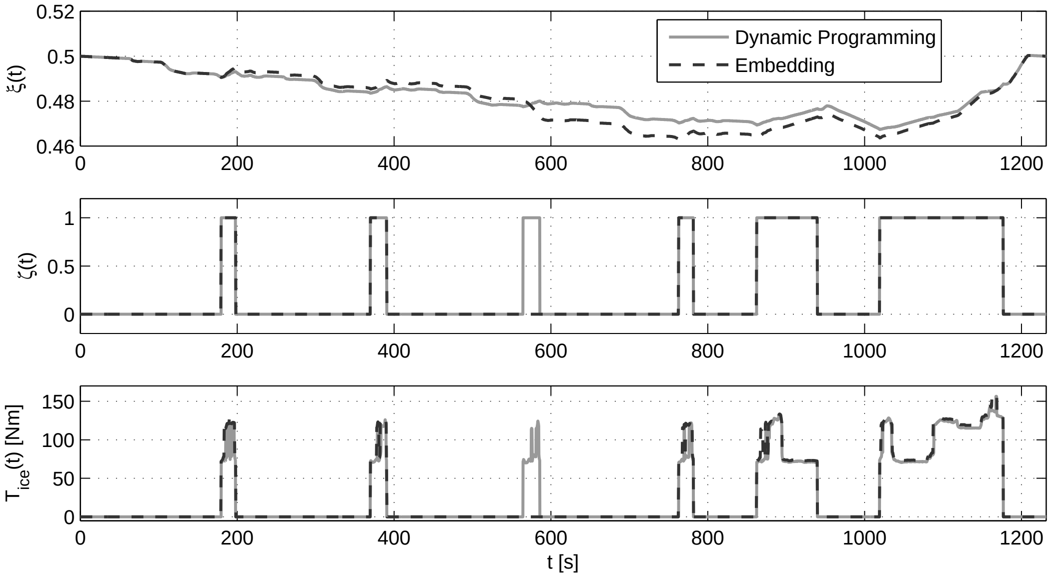

Figure 4.11: Comparison of the results for $\mathcal{P}_3$ obtained with dynamic programming and the embedding with relaxed jump function approach

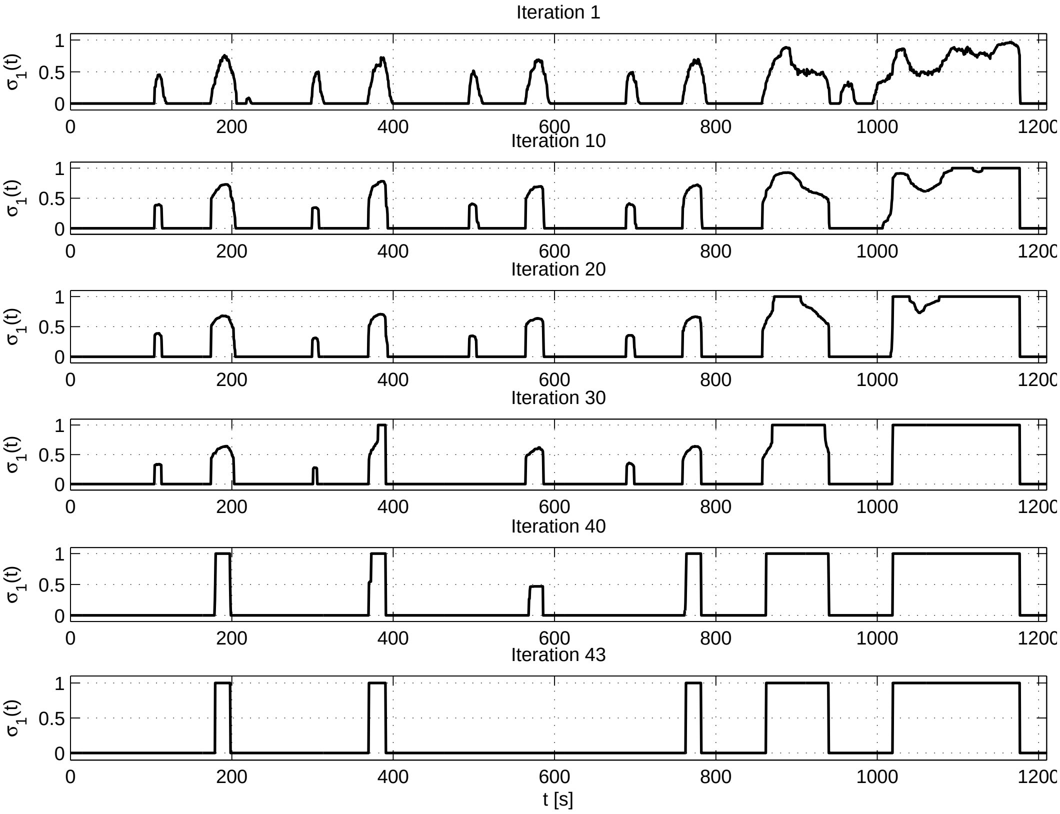

Figure 4.12: Evolution of the relaxed switching variable σ_1 over the iterations of the greedy rounding scheme

Problem $\mathcal{P}_4$ is a much more complex problem than $\mathcal{P}_1$-$\mathcal{P}_3$ due to its high dimension of the continuous state vector as well as the high number of discrete states. Indirect shooting is unlikely to yield a solution because of its low convergence region. It is also a difficult task to find a guess of the initial costate values that will converge to solution satisfying the boundary conditions. Applying embedding to the problem will lead to a large optimization vector due to the high number of discrete states. Using the combined method, a solution could be found in an acceptable time range.

To evaluate the effect of different bounds on the artificial emission component Z, optimizations were performed with three different upper bounds $Z_{max,1} = Z_{ref}$, $Z_{max,2} = 1.1 \cdot Z_{ref}$, $Z_{max,3} = 1.2 \cdot Z_{ref}$ to see, how a less stringent constraint on emissions affects the continuous controls as well as the switching function. The results can be seen in Fig. 4.13. It can be noted that a lower upper bound leads to an only slightly decreased time-span used for TWC-heating but to a significant retardation of the ignition angle, which causes a faster TWC-heating. TWC-heating is mostly performed during idle with the K0 closed and the MG connected ($q(t) = 1$). As soon as a TWC-temperature with a good conversion efficiency is attained, the ignition angles tend to the lower bound to increase combustion efficiency. The base ignition angle was set as lower limit. The late ignition angles also lead to lower values of ξ at the end of the catalytic heating process, since a lower output torque is provided by the ICE. During TWC-heating, the exhaust mass flow is constrained, as high mass flows decrease the catalytic converter efficiency. In the solution, the relative cylinder charge tends to the respective bound.

The approximated costates are also depicted in Figure 4.13. The costates offer in this case a physical interpretation. Whenever the costate for a respective state is low, increasing the corresponding state at this point is more beneficial than at other times. Increasing the TWC-temperature ϑ_{twc} early in the cycle is recommended to quickly achieve good conversion efficiencies. Once the light-off temperature is reached, a further increase of the temperature has hardly any effect. The costate tends to zero, as soon as the light-off temperature is exceeded.

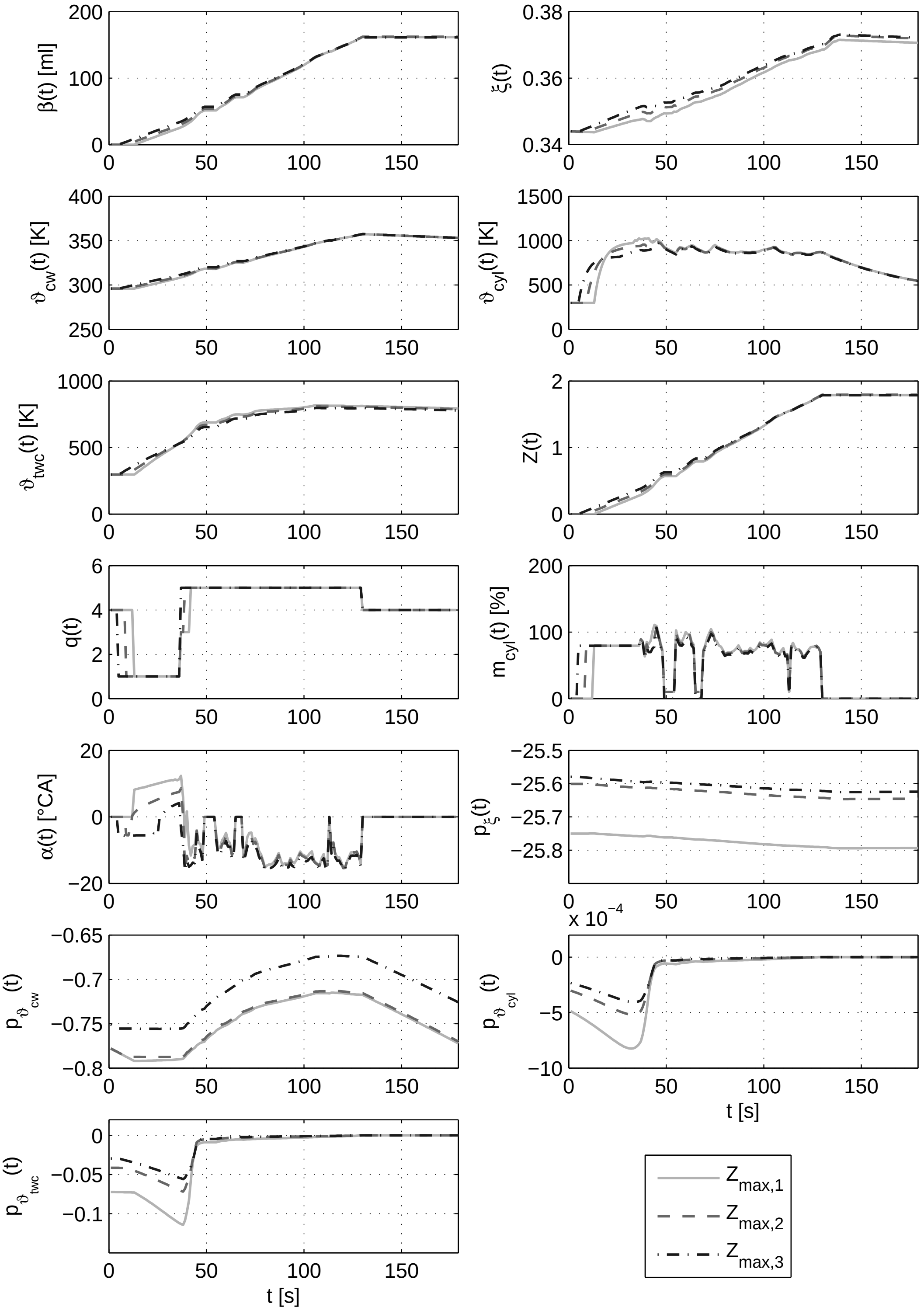

Figure 4.13: Results and approximated costates obtained for $\mathcal{P}_4$ with different bounds for the artificial emission state Z. The corresponding states of each depicted costate are written as subscript

4.3 Causal Energy Management of Hybrid Vehicles

The solution of optimal control problems is only a first step in the definition of an energy management for hybrid vehicles. The solution of SOCPs online during the vehicle operation is usually prevented by the very limited computational performance available in today's electronic control units (ECUs). Especially direct methods are difficult to implement due to the high memory requirements needed for storing the Jacobian and Hessian matrices. Most energy management systems for hybrid vehicles still rely on rule-based control strategies because of the ease of implementation and the low computational demand. Energy managements that only use the current state of the vehicle, consisting of all values that can be measured or estimated at a certain time t to determine the set-values for the powertrain, are called causal. If estimated future information is also used to define the current set-values, the energy management is called predictive. The literature provides a wealth of possible ways to implement causal energy managements. In the work of Lin et al. [2001], the results of a cycle-specific solution obtained by dynamic programming are used for the manual definition of rules. A very similar approach was chosen by Karbowski et al. [2006]. A rule based energy management based on a fuzzy logic controller was proposed by Schouten et al. [2002].

Over the last decade, research in this area has more and more focused on analytical methods. The most attention has been paid to the PMP and the related ECMS. The use of these approaches is appealing due to the fact that the high-dimensional optimization problem is reduced to solving a nonlinear equation and hence to finding a feasible costate value. The problem is additionally simplified by assuming the costate to be constant. An evaluation of the effect of this assumption will be given later in this section. If the costate is known, the optimal controls that lead to a desired final state of charge for a specific drive cycle are entirely defined, if only continuous controls are regarded. The problem then remains to define a costate value without knowledge of the drive cycle. Kim et al. [2009] suggest to select the costate based on heuristically defined values that represent the drive pattern. A learning procedure that corrects the costate value, when a lower or upper bound for the state of charge is hit, is described in the work of Chen and Salman [2005]. A study on the relationship between different road-type events and the costate value that leads to a charge-sustaining operation of the HEV was performed by Gong et al. [2011].

4.3.1 Calculation of Look-up-Tables

With a known costate, the optimal continuous controls can be found by minimization of the Hamiltonian function at each time instant via a numerical procedure. Despite the fact, that the minimization problem will for most powertrain architectures involve only a single variable, the numerical procedure is not ready-to-implement on today's ECUs. In this section, we derive, how the look-up tables (LUTs) storing the optimal continuous controls as well as optimal discrete controls can be automatically generated. The idea of storing the minimum of the Hamiltonian function with respect to the continuous control was also proposed by Chen and Salman [2005] in the context of ECMS. Also with the ECMS-formulations, it was extended to the gear-choice by Sivertsson et al. [2011] but without theoretical justification. In Fig. 4.14, a sketch of an LUT-based energy management is depicted. The reference values $\hat{\kappa}$, $\hat{\zeta}$ and $\hat{T}_{mg}$, supplied to the lower level controller structure are herein determined by LUTs. Based on the current driving condition that is represented by the torque-demand at the wheel and the current wheel-speed, a gear recommendation is made based on a LUT. If this gear is actually used will depend on many more conditions. These rather practical aspects are not regarded in this thesis. With a gearbox model, the gearbox input torque and speed can then be calculated and based on these values a recommendation for the drive-mode is again obtained from a LUT. If hybrid drive mode is selected, the torque-split between MG and ICE finally needs to be determined. The respective MG-torque is stored in a table.

These LUTs can be obtained from the solution of the SOCP in a straightforward manner. As has been demonstrated in section 4.2.2, the costate variation over the entire cycle will in general be negligible, when the variation of ξ is small and it will also be continuous on a switching (see (4.60)). But also for larger variations of ξ, $p(t)$ will remain nearly constant. A constant costate can also be assumed in the solution procedure of the indirect shooting method.

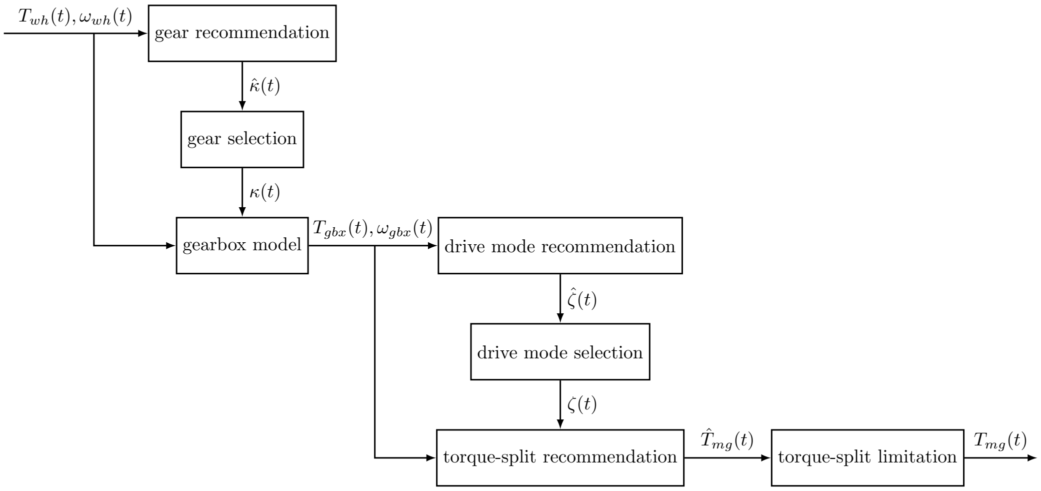

Figure 4.14: LUTs defining the MG-torque in hybrid drive mode for different costate values

The minor effect of this assumption is demonstrated in Figure 4.15, where the evolution of $\xi(t)$ and $p(t)$ is compared with and without the constant costate assumption for two different vehicles over a longer drive cycle and a large difference between $\xi(t_0)$ and $\xi(t_f)$. The effect on the fuel consumption is hardly noticeable as is shown in Table 4.4. A constant costate is widely assumed in the literature for purely continuous optimal control problems (Guzzella and Sciarretta [2005], Kim et al. [2011]).

	fuel consumption [ml] for varying p	fuel consumption [ml] for constant p
Vehicle 1	869.1	870.3
Vehicle 2	1271.8	1276.1

Table 4.4: Effect of the constant costate assumption on the fuel consumption

Consequently, an important result of the solution to problem $\mathcal{P}_1$ is the value of the costate $p(t) \approx p(t_0)$. Once the costate is known, the value of the Hamiltonian function only depends on the gearbox input torque T_{gbx}, the gearbox input shaft speed ω_{gbx}, and the engine torque T_{ice}:

$$\mathcal{H}(p, T_{ice}, \omega_{gbx}, T_{gbx}). \tag{4.65}$$

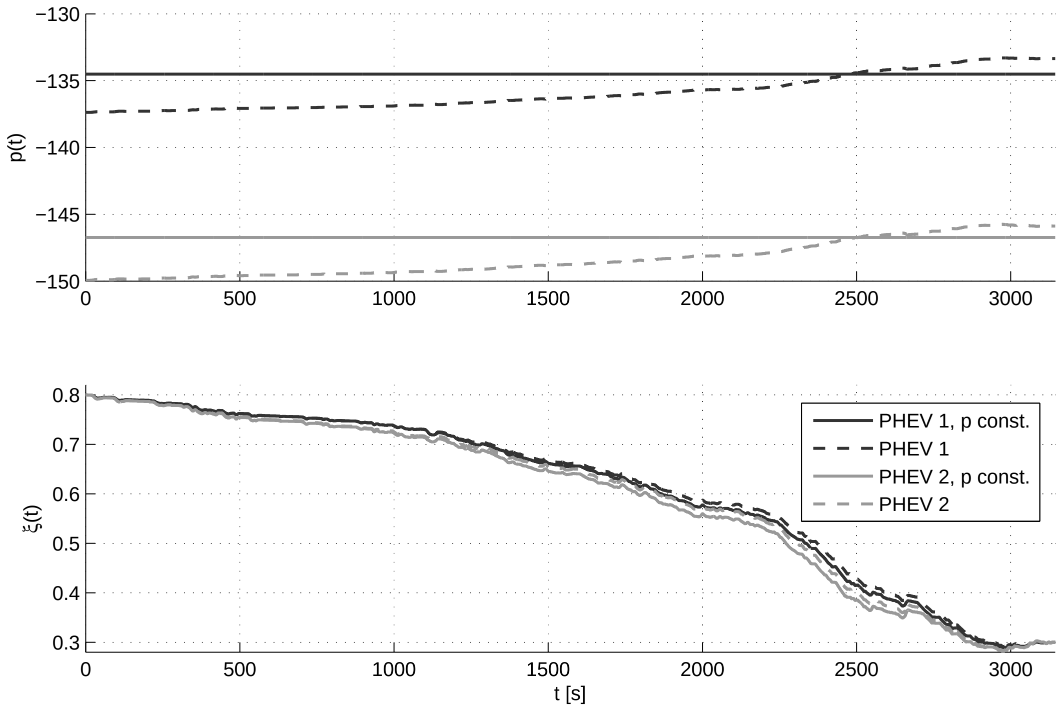

Figure 4.15: Comparison of $p(t)$ and $\xi(t)$ with and without constant costate assumption

LUTs can be calculated by minimization of the Hamiltonian function with respect to each continuous or discrete argument. The minor dependence of the battery current I on the state of charge ξ can be neglected to reduce the number of dimensions of the respective LUTs. It is shown in Sivertsson et al. [2011], that saving the controls on a discrete grid has only minor effects on the fuel consumption. Consequently, a LUT as a function of instantaneous engine speed and engine torque during engine load shifting ($\zeta(t) = 1$) can be generated for suboptimal values of $\hat{T}_{ice}(\omega_{gbx}, T_{gbx})$ by calculating

$$\hat{T}_{ice}(\omega_{gbx}, T_{gbx}) \equiv \arg \min_{u(t)\in\mathcal{U}} \mathcal{H}(p, u, \omega_{gbx}, T_{gbx}, p) \tag{4.66}$$

on a grid of (p_{gbx}, T_{gbx}). In the next step, the LUT for the suboptimal choice of the drive mode $\zeta(t)$ can be calculated by

$$\hat{\zeta}(\omega_{gbx}, T_{gbx}) \equiv \arg \min_{\zeta(t)\in\{0,1\}} \mathcal{H}(p, T_{ice}, \omega_{gbx}, T_{gbx}) \tag{4.67}$$

on the same grid. In the definition above, for $\zeta = 1$, T_{ice} is determined from the LUT $\hat{T}_{ice}(\omega_{gbx}, T_{gbx})$, otherwise $T_{ice} = 0$ applies.

Remark 4.1: From observations it appears that for each combination ω_{gbx} and p, there is a torque threshold $T_{start}(\omega_{gbx}, p)$ such that $\mathcal{H}(p, \hat{T}_{ice}(\omega_{gbx}, T_{gbx}), \zeta = 1, \omega_{gbx}, T_{gbx})$ has a lower value than $\mathcal{H}(p, T_{ice} = 0, \zeta = 0, \omega_{gbx}, T_{gbx})$ if and only if $T_{gbx} > T_{start}(\omega_{gbx}, p)$. The recommended drive mode can then be described via a speed-depending torque threshold map $\hat{T}_{start}$. When the driver demand T_{gbx} exceeds $T_{start}(p, \omega_{gbx})$, an ICE-start is demanded.

For both drive modes, recommended gears κ can be calculated over a grid of (T_{wh}, ω_{wh}) as follows:

$$\hat{\kappa}(T_{wh}, \omega_{wh}) \equiv \arg\min_{\kappa \in K} \mathcal{H}(p, \kappa, T_{ice}, \omega_{wh}, T_{wh}). \tag{4.68}$$

Again, for $\zeta = 1$, $T_{ice}(\omega_{gbx}, T_{gbx}) = \hat{T}_{ice}(\omega_{gbx}, T_{gbx})$, and for $\zeta = 0$, $T_{ice} = 0$ applies. Figure 4.16 depicts LUTs for the suboptimal determination of the torque-split in hybrid drive mode. Instead of saving $\hat{T}_{ice}$, it is more common in automotive practice to store the MG-torque $\hat{T}_{mg}$. This transformation can be easily done using (4.9). Figure 4.17 depicts the start-torque thresholds for different costate values and Figure 4.18 shows LUTs with recommended gears $\hat{\kappa}$ for a given costate value.

4.3.2 Evaluation and Implementation

Equations (4.67) and (4.68) can only be justified by necessary conditions of optimality when state jumps are disregarded, since for the case where discontinuities occur in the state trajectory, the minimization of the Hamiltonian function by the discrete inputs is not a necessary condition (cf. (2.27) and (2.44)). However, practical experience has shown that the LUTs derived from these equations can be used in combination with hystereses and delay times to obtain results close to the results calculated offline using hybrid dynamic programming. The hysteresis and delay parameters can be found by applying gradient free optimization methods. The LUT $\hat{T}_{ice}(\cdot)$ can usually be implemented in a PHEV without further modification. This map will lead to a charge sustaining behavior only for specific cycles. Therefore several maps are calculated for different values of the costate that need to be stored in the ECU. Modern PHEVs can leave the choice of an operating mode to the driver. A general depleting mode that intends to minimize the fuel used over an unknown route until the next recharge facility is available can be implemented,

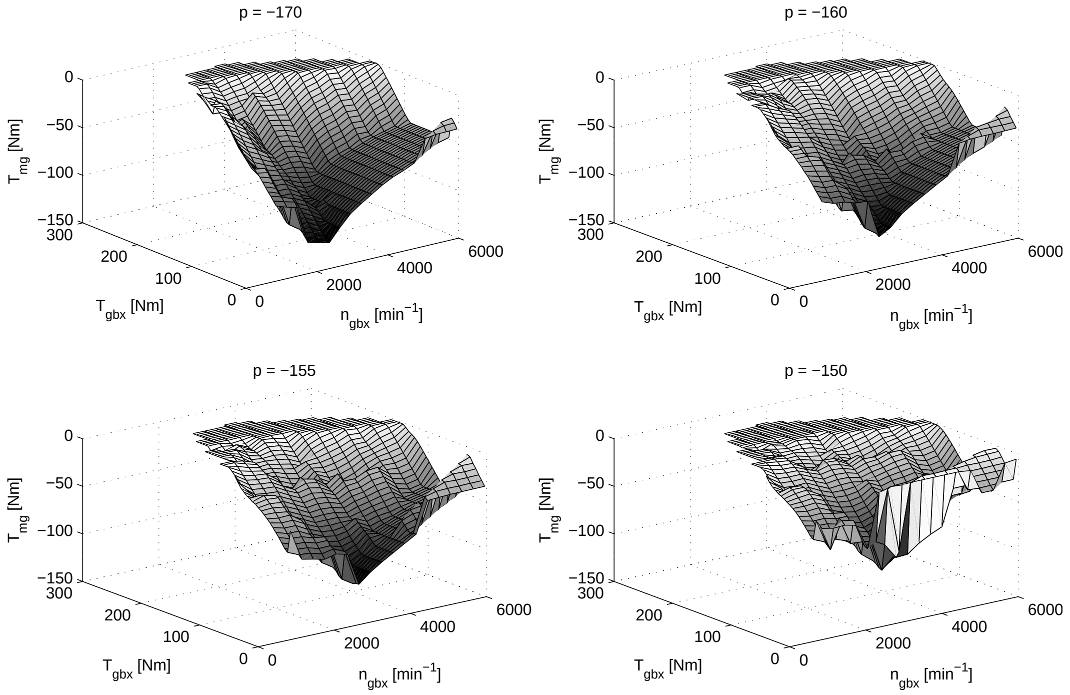

Figure 4.16: LUTs defining the MG-torque in hybrid drive mode for different costate values

by using a constant costate obtained from the solution of an SOCP over a representative cycle and a low value for $\xi(t_f)$ as the boundary condition. To reliably deplete the battery to its minimum state of charge, a predictive energy management strategy is necessary as it will be described in the next section. To determine the recommended gears, additional constraints such as limitations due to driving comfort apply. These factors (e.g. human perception, debounce hysteresis, and so forth) are hard to account for in a mathematical solution. As a consequence, the gear recommendations cannot always be followed. In this case, it has shown to be helpful to evaluate the effect of deviating from the recommended solution. If the values from the LUTs are used, (4.61) holds and the Hamiltonian is continuous during a change in the piecewise constant switching function as, for instance, during a transition from one drive mode to another or at gear changes. When deviating from the recommended transitions, a difference in the Hamiltonian

$$\Delta\mathcal{H} = \mathcal{H}(t_j^+) - \mathcal{H}(t_j^-) \tag{4.69}$$

occurs. The meaning of this difference is twofold: On the one hand, it consti-

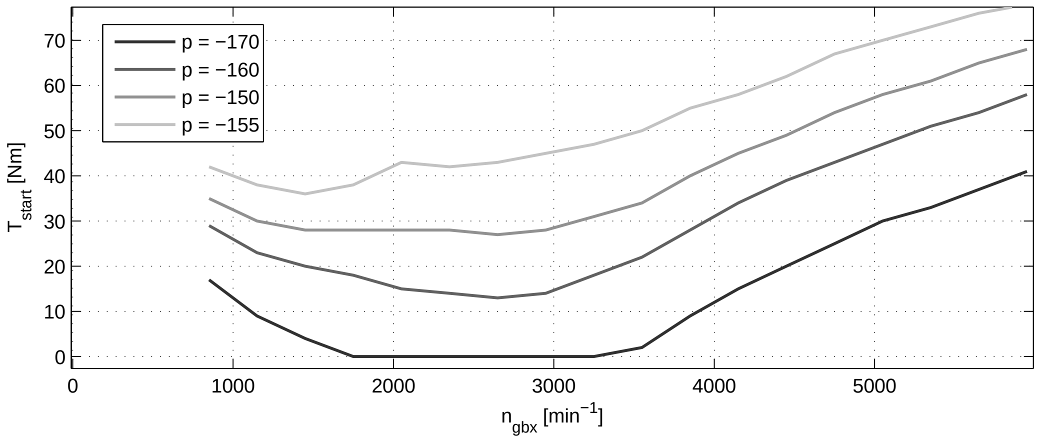

Figure 4.17: LUT defining the torque-threshold for an ICE-start

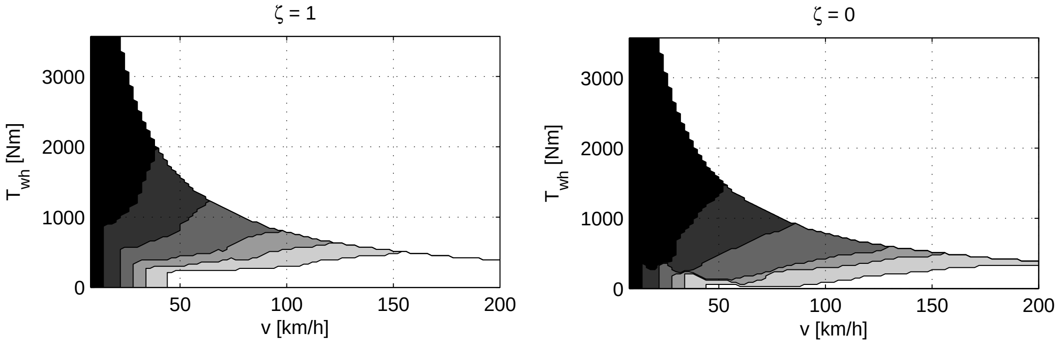

Figure 4.18: LUTs defining the optimal gear selection for a given driver request

tutes a deviation from the optimality conditions. On the other hand, with the interpretation of the Hamiltonian as the weighted sum of battery current and fuel mass flow, it is indicated that a control with lower value of this weighted sum exists, but cannot be used because of some unknown constraint. The value of $\Delta\mathcal{H}$ can be depicted over gearbox input torque and speed for a given costate. Figure 4.19 shows the absolute value $|\Delta\mathcal{H}|$ depending on n_{gbx} and T_{gbx} between electric and hybrid drive mode. The recommended switching is where the difference vanishes. If this recommended switching cannot be followed, Figure 4.19 allows for an evaluation of the effects. A deviation from the recommended switching is more acceptable when the value of $|\Delta\mathcal{H}|$ is low, which is more the case for lower engine speeds than for higher speeds.

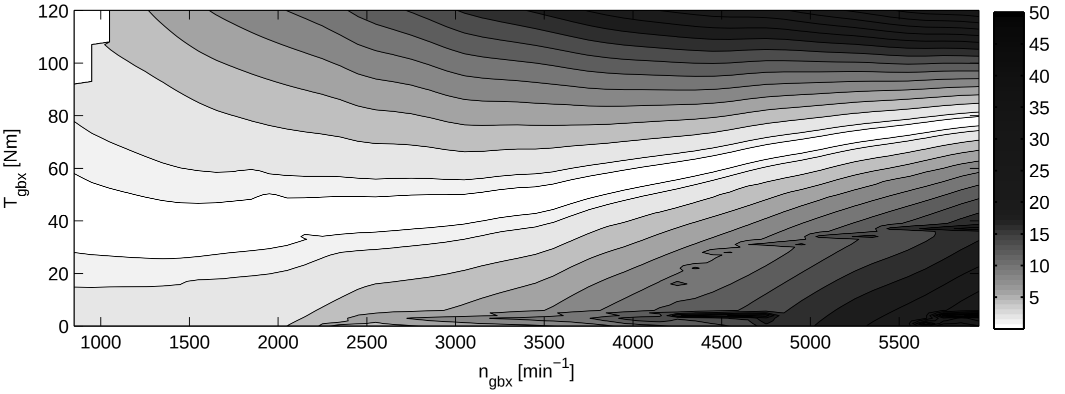

Figure 4.19: Difference $|\Delta\mathcal{H}|$ for electric and hybrid drive mode and $p = -150$. Deviating from the recommended start torque is more acceptable at low engine speeds

4.4 A Predictive Energy Management

The solution of optimal control problems is on the one hand usually prevented by the limited computing capacity of the ECU. On the other hand, the solution is based on a predefined drive cycle given by the trajectories of velocity $v(t)$ and road slope $e(t)$, which are generally unknown on real driving missions. It is however possible, to predict the trajectories based on the information provided by an intelligent traffic system (ITS) as it is included in many recent navigation systems (Ress et al.). The developments in this area are continuously growing such that more and more additional road information will be known to the ECU. With the help of this finite set of information, an estimation of the trajectories can be made and these estimations can be used for solving the OCP. The existence of discrete phenomena makes the problems harder to solve. However, it was shown in section 4.2, that the indirect shooting method is highly efficient on the problems $\mathcal{P}_1$ and $\mathcal{P}_2$. At the same time, as shown in section 4.3, the optimal controls for a given costate and a given driving situation can easily be stored in LUTs without significant loss of accuracy. These facts make the implementation of a control strategy that is based on the solution of a SOCP online in the vehicle possible.

In the literature on hybrid vehicle control, model-predictive control is often used for a rather short prediction horizon, as in the works of Back [2005] and Borhan et al. [2009]. It appears, however, that strategies, based on PMP or ECMS, grow here in importance as well. Many approaches based on finding the costate for a purely continuous OCP are explained in the literature. A general framework for the adaptive control using ECMS by periodically updating the costate, based on a prediction of the driving profile, is described in Musardo et al. [2005]. In Lee et al. [2012], the velocity profile is predicted, and using the average requested wheel-power, the costate is chosen from a table. In the work of Kermani et al. [2012], a model predictive controller is implemented to determine the costate over a predicted driving profile. A predictive energy management that also incorporates discrete decisions is proposed by Johannesson et al. [2009]. Herein, the optimization of the clutch-states and the optimization of the continuous controls are performed in two stages. Dynamic programming and approximate dynamic programming are employed for the optimizations.

The predictive energy management in this thesis uses an indirect shooting method to solve an SOCP similar to problem $\mathcal{P}_1$, involving the torque-split as continuous control input and the drive-mode as discrete decision. The gear-selection strategy is not regarded in the SOCP but is defined according to the heuristic gear-selection strategy, as it is implemented in the vehicle.

The strategy is implemented as a depleting strategy and is employed when the target destination provides a charging facility and the total driving distance exceeds the electrical range for the current state of charge. In this case the entire electrical energy can be depleted but the ICE has to be started several times, to prevent the battery from falling below its minimum value before the target destination is reached. It is the task of the energy management system, to define when the ICE is started and to define the torque split. However, the predictive energy management can be generalized to any situation, where a certain target value of ξ has to be attained over a predictable cycle.

4.4.1 Overview of the Control Strategy

As depicted in Fig. 4.20, the predictive control strategy consists of the following elements:

- The ITS provides information on speed limit, slope and obstacles that might require a stopping of the car
- A driver-model uses this set of information to calculate a predicted velocity profile over the given distance. A vehicle-model then calculates the required gearbox input torque and angular velocity
- A reference trajectory $\xi_{ref}(s)$ for the battery's state of charge and for the costate $p_{ref}(s)$ is then calculated by solving an SOCP
- Depending on the deviation of $\xi(t)$ to the predicted value $\xi_{ref}(s(t))$, an offset $\Delta p(t)$ is calculated by a PI-control law to assure that the reference trajectory is being followed
- Based on the current value $p(t) = p_{ref}(s(t)) + \Delta p(t)$ and the current driving condition, the instantaneous controls (drive-mode and torque-split) are determined

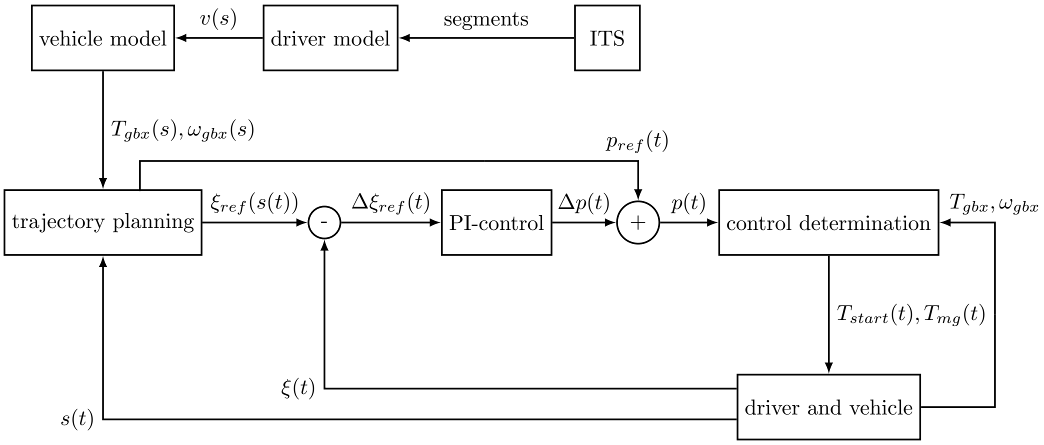

Figure 4.20: Control strategy for the predictive energy management

4.4.2 Driver Model

Since dynamic memory allocation is in general not supported in today's ECUs, the reference trajectory is calculated over a grid of constant size N in the spatial domain

$$0 = s^1 < s^2 < \cdots < s^N = s_{dist}, \tag{4.70}$$

where the total length of the driving cycle s_{dist} is obtained from the navigation system. The position instances $s^k, \quad k = 1, \ldots, N$ are stored in the vector

$$\overline{s} = [s^1, s^2, \ldots, s^N]. \tag{4.71}$$

The grid is chosen to be equidistant and hence

$$\Delta s = s^{k+1} - s^k = const. \tag{4.72}$$

applies. Task of the driver model is to generate a realistic velocity profile $v(s^k)$ from the information provided by the navigation system. A Markov chain model as proposed by Gong et al. [2011] is in general desirable, as it is able to reflect the statistical behavior in the real profile. However, the expenditure for the determination of the respective transition matrices is quite high and the matrices would require a high amount of storage in the ECU. Dynamic programming is used by Lee et al. [2012] to minimize a weighted sum of driving energy, time and acceleration to define the velocity profile.

Dynamic programming, even for only one state, would require a major part of the ECU's capacity. Therefore, we use an approach proposed by Treiber and Kesting [2010] that was developed for the purpose of traffic analysis. The model uses a set of differential equations that imitates the driver's behavior in certain driving scenarios and is originally described in the time domain. These equations can be transferred to the spatial domain by dividing the right-hand side of the differential equation by v. This becomes clear, when considering the following transformation:

$$\frac{dv}{ds} = \frac{dv}{dt} \cdot \frac{dt}{ds} = \frac{dv}{dt} \cdot \frac{1}{v}. \tag{4.73}$$

Depending on the current driving situation d, where $d = 1$ indicates acceleration, $d = 2$ deceleration to a lower speed limit and $d = 3$ deceleration to standstill, the velocity's spatial derivative is given by

$$\frac{dv}{ds} = \begin{cases} \frac{e}{v} \cdot (1 - (\frac{v}{v_{lim}})^{\delta}), & d = 1 \\ -\frac{e}{v} \cdot (1 - (\frac{v_{lim}}{v})^{\delta}), & d = 2 \\ \frac{1}{h \cdot v} \cdot (\frac{v^2}{2 \cdot s_{rem}})^2, & d = 3. \end{cases} \tag{4.74}$$

The driving situation d is determined by a discrete automaton, that chooses the respective scenario based on the distance to the next speed limit or obstacle, that might require a vehicle stop. The constants e and h represent typical accelerations and decelerations, respectively, and the constant δ determines, when the acceleration is reduced when approaching the target speed v_{lim}. In the time domain, this model was also applied by Boehme et al. [2013]. As can be noticed from measurements, the velocity often exhibits oscillations around the speed limit. To account for these oscillations, the sum of l cosines with different amplitudes A_i, angular velocities ω_i and phase shifts ϕ_i is added to the speed limit v_{sign}

$$v_{lim} = v_{sign} + \sum_{i=1}^{l} A_i(v_{sign}) \cdot \cos(\omega_i \cdot t^k + \phi_i). \tag{4.75}$$

The amplitudes, frequencies and phase shifts can be identified from measurements via Fourier analysis. With equation (4.74), an IVP can be solved using the explicit Euler approach as follows:

$$v^{k+1} = \max(\epsilon, v^k + \frac{dv}{ds} \cdot \Delta s). \tag{4.76}$$

The constant ϵ is a lower bound for the speed. This is necessary, since (4.74) is not defined for $v = 0$. Knowing velocity v and position s, the corresponding values for the time t and acceleration a can be approximated by

$$t^{k+1} = t^k + \frac{\Delta s}{v^k} \tag{4.77}$$

$$a^k = \frac{v^{k+1} - v^k}{t^{k+1} - t^k}. \tag{4.78}$$

The first $N-1$ values of v^k, t^k and a^k are then stored in the vectors

$$\overline{v} = [v^1, v^2, \ldots, v^{N-1}] \tag{4.79}$$

$$\overline{t} = [t^1, t^2, \ldots, t^{N-1}] \tag{4.80}$$

$$\overline{a} = [a^1, a^2, \ldots, a^{N-1}]. \tag{4.81}$$

In general, some discrepancy exists between the predicted and the measured velocity profile. Especially in dense traffic situations, the actual velocity will be lower than the predicted velocity. Yet, the robust controller design can cope with these inaccuracies. This will be demonstrated later in this chapter by providing the results of detailed robustness tests.

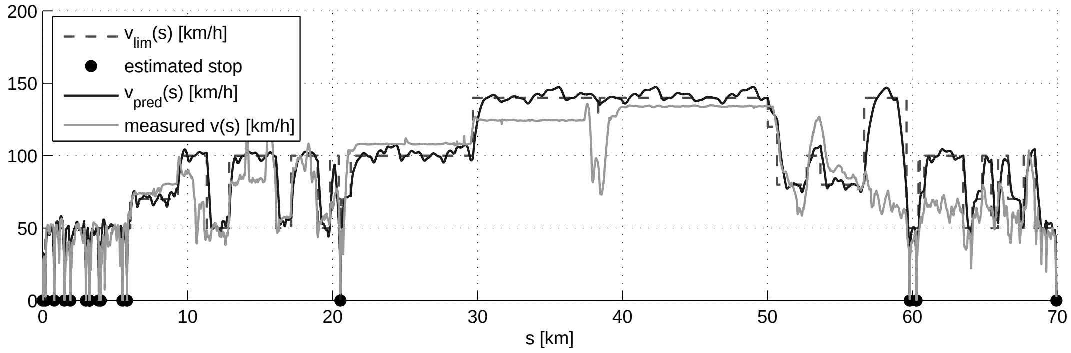

Figure 4.21: Estimated velocity profile over a given drive cycle

In discretized form, the SOCP can be formulated as

$$\min_{\zeta,u} \beta^N \tag{4.82}$$

subject to the constraints

$$\zeta^k \in \{0,1\} \tag{4.83}$$

$$u^k \in \mathcal{U}^k \tag{4.84}$$

$$\xi_{min} \leq \xi^k \leq \xi_{max} \tag{4.85}$$

$$\psi = \xi^N - \xi_{min} = 0, \tag{4.86}$$

where ξ_{min}, ξ_{max} are bounds for the state of charge. ξ_{min} also is the desired target for ξ at the end of the trip.

4.4.3 Solving the SOCP

The SOCP is solved by applying the indirect shooting approach. The most expensive operation in this algorithm is the minimization of the Hamiltonian, to find the optimal continuous control input. It is therefore beneficial, to perform this operation offline for a given vehicle and to store the information in the LUTs $\hat{T}_{mg}(p, T_{gbx}, \omega_{gbx})$ and $\hat{T}_{start}(p, \omega_{gbx})$. This can be done by calculating the LUTs described in section 4.3 for a range of values of the costate. Thus, the operation of finding the minimum of the Hamiltonian function is reduced to a computationally cheap interpolation. The mode-sequence $\overline{\zeta}$ is then defined as follows:

$$\zeta^k = \begin{cases} 1, & \omega_{gbx}^k \geq \hat{T}_{start}(p, \omega_{gbx}^k) \\ 0, & \omega_{gbx}^k < \hat{T}_{start}(p, \omega_{gbx}^k) \end{cases} \quad k = 1, \ldots, N-1. \tag{4.87}$$

Once the IVP has been solved, function (4.86) can be evaluated and the initial guess of p^1 can be improved. The SOCP is therefore reduced to solving the scalar nonlinear equation

$$\psi(p^1) = \xi^N(p^1) - \xi_{min} = 0. \tag{4.88}$$

As in the offline solution, the Pegasus method is used and performs efficiently and robustly on this equation.

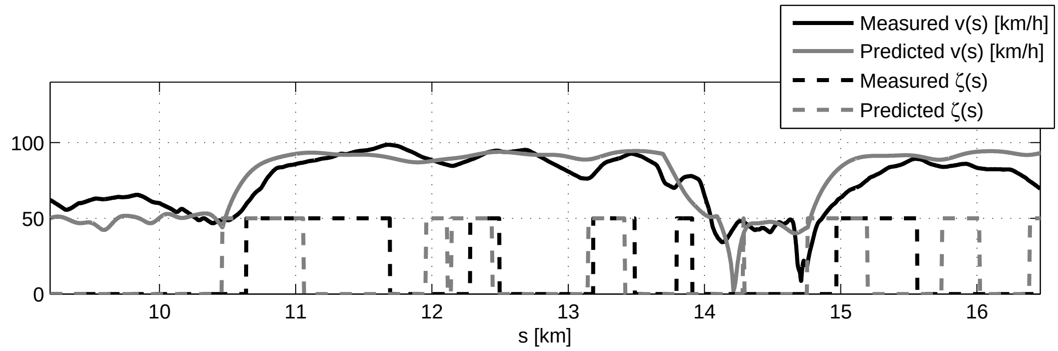

Figure 4.22: Predicted velocity and optimal engine start sequence with respect to the prediction. Deviations to the measured trajectory of v and the measured start sequence can be noticed

4.4.4 State Constraints

When the discrete solution-trajectory $\overline{\xi}$ contains an arc for which

$$\xi^k < \xi_{min} \tag{4.89}$$

applies, an interior point-condition is added to the optimal control problem, as proposed in de Jager et al. [2013]. The spatial instant $s^{k,min}$ that has the lowest value ξ^k is identified and the interior-point condition

$$\psi_1 = \xi^{k,min} - \xi_{min} = 0 \tag{4.90}$$

is added to the SOCP formulation. The SOCP is then resolved, first solved over the interval $[s^1, s^{k,min}]$ with (4.90) as final state constraint and then over the interval $[s^{k,min}, s^N]$ with ξ_{min} as initial value and (4.86) as final state constraint. The procedure is depicted in Fig. 4.23. The black trajectory contains an arc that falls below a lower bound $\xi_{min} = 0.2$. The spatial instant $s^{k,min}$ is identified that has the lowest value $\xi^{k,min}$. An interior point condition is then inserted that requires this point to be on the lower state bound

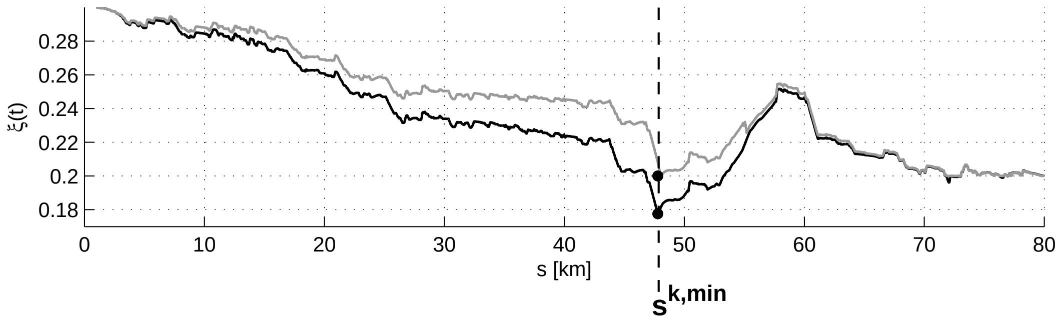

Figure 4.23: Procedure for obtaining a suboptimal trajectory fulfilling the state constraint

4.4.5 Controller and Instantaneous Controls

Several uncertainties in the trajectory planning make a controller inalienable, among them model error and error due to discretization. The most influential uncertainty, however, is the driver behavior, which can be only roughly predicted. The costate p can well be used as control variable, as it has a direct influence on the torque-split as well as on the start torque. A lower value of the costate will lead to earlier engine starts and higher load torques. A simple PI-controller can fulfill the task of disturbance rejection. Small deviations from the planned ξ-trajectory are acceptable. Thus, the gains of proportional path and integral path are kept rather low, which has shown to be advantageous for the fuel consumption. The instantaneous controls can be determined, using the look-up tables $\hat{T}_{mg}$ and $\hat{T}_{start}$ defined in the previous section with the corrected costate

$$p(t) = p_{ref}(s(t)) + \Delta p(t) \tag{4.91}$$

and the current driving condition $T_{gbx}(t), \omega_{gbx}(t)$. Since the value $\xi(t)$ is only estimated in the real-world vehicle and the estimation is corrected from time to time, jumps in the trajectory may occur. Due to the low controller gains, this will not cause instabilities.

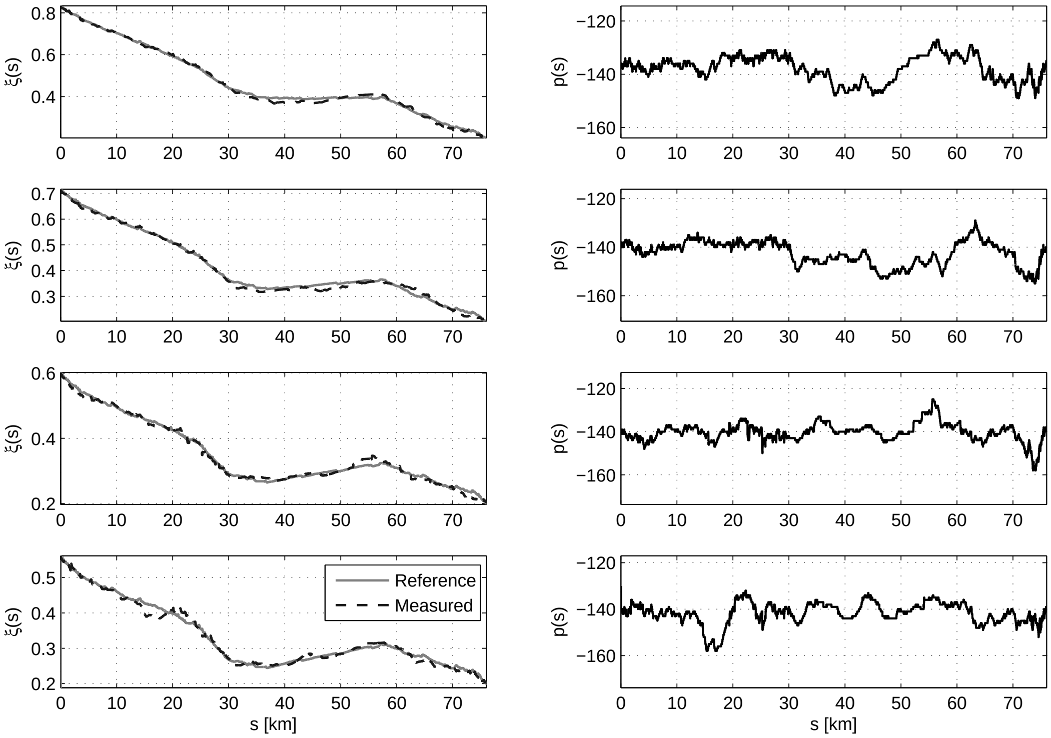

Figure 4.24: Reference and measured trajectories for $\xi(s)$ and controller output $p(s)$ for four measurements over the same cycle with different initial charging states

4.4.6 Implementation and Results

The strategy is implemented in a PHEV with a battery capacity of 7.5 kWh. As prototyping unit, a dSPACE MicroAutobox is used. The route is discretized with $N = 10000$. The calculation of the reference trajectory on the MicroAutobox takes between 16 and 26 seconds, depending on the number of iterations needed, to solve (4.86). In the final implementation, the costate p is assumed to be constant, since the variation over the entire cycle is negligible as compared to the controller offset Δp. The finite differencing step to approximate the derivative of the Hamiltonian function with respect to ξ can therefore be disregarded. To avoid frequent engine-starts, a hysteresis around T_{start} is defined and turn-on/turn-off-delays are implemented. The target value ξ_{min} is set to 0.2. As test-cycle, a route of 75 km in the area of Gifhorn, Germany is used, that contains a balanced scenario of urban, rural and highway driving situations.

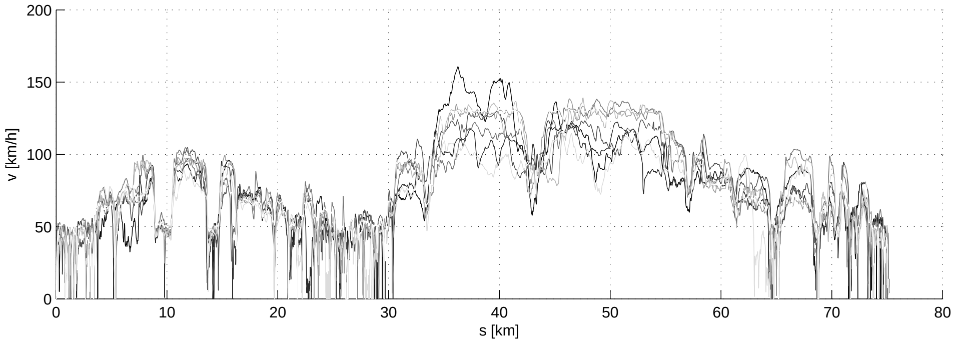

Figure 4.25: Diversity of recorded velocity profiles for the test drive cycle

The test-drives were conducted by different drivers and during different daytimes to investigate the robustness towards deviations from the predicted driving profile. Figure 4.24 depicts the reference trajectories and the measured trajectories for ξ of four measurements. The reference trajectories can be followed and the desired state of charge $\xi(s_{dist})$ is reached with a narrow tolerance of $\pm 1.2\,\%$ in all tested cases. Even though the predicted velocity profile is close to the measured profile, the controller has to correct the costate visibly.

In addition to the real-world tests, simulations were performed. To verify the robustness of the strategy against model errors, the drag parameters b_0, b_1 and b_2 were disturbed with an equally distributed factor $w \in [0.9, 1.1]$. The factor is applied to the simulated vehicle, but is not used in the prediction. Despite the model error, $\xi(s_{dist}) - \xi_{min} = 0$ is still fulfilled with a narrow tolerance, as can be seen on the left side of Fig. 4.26. From the scatter plot, it can be noted, that the distribution of the fuel consumption can be directly linked to the gearbox input energy

$$E_{gbx} = \int_{t_0}^{t_f} T_{gbx} \cdot \omega_{gbx} \cdot \frac{1\,\mathrm{h}}{3600\,\mathrm{s}} dt \tag{4.92}$$

required to propel the vehicle, which is also varied caused by the disturbance of the drag coefficients. To survey the robustness against different drivers and traffic situations, simulations were performed over 22 recorded velocity profiles. The partially unpredictable diversity in the profiles can be noticed in Fig. 4.25. The predicted velocity profile generated by the ITS and the driver model does not reflect driver types nor does it include information on traffic density. Consequently, it predicts the same velocity profile each time. The effect on $\Delta\xi$ is in this case stronger, but still within acceptable bounds, as can be seen on the right side in Fig. 4.26.

Figure 4.27 depicts the operation points of the ICE of four measurements in the efficiency map. Especially during the highway-drive, where higher engine speeds occur, the ICE operates with nearly optimal efficiency. This is also the case for lower speeds, where the spread is slightly higher due to the less constant driving conditions in urban or rural driving situations.

Figure 4.28 shows a comparison of the ICE torques selected by the predictive energy management and the fully optimal solution obtained with DP. The measurement was provided to the DP-procedure such that the same boundary conditions apply and the velocity profile and the gear sequence are exactly known. Hence, the Figure shows, how the predictive energy management should have performed, if the cycle was perfectly known. As the prediction still deviates from the actual cycle, deviations from the optimal solution cannot be avoided. However, the results are very close in terms of engine start sequence and the engine torques chosen. The difference in fuel consumption between fully optimal solution and the measurement was in all cases below 2 %.

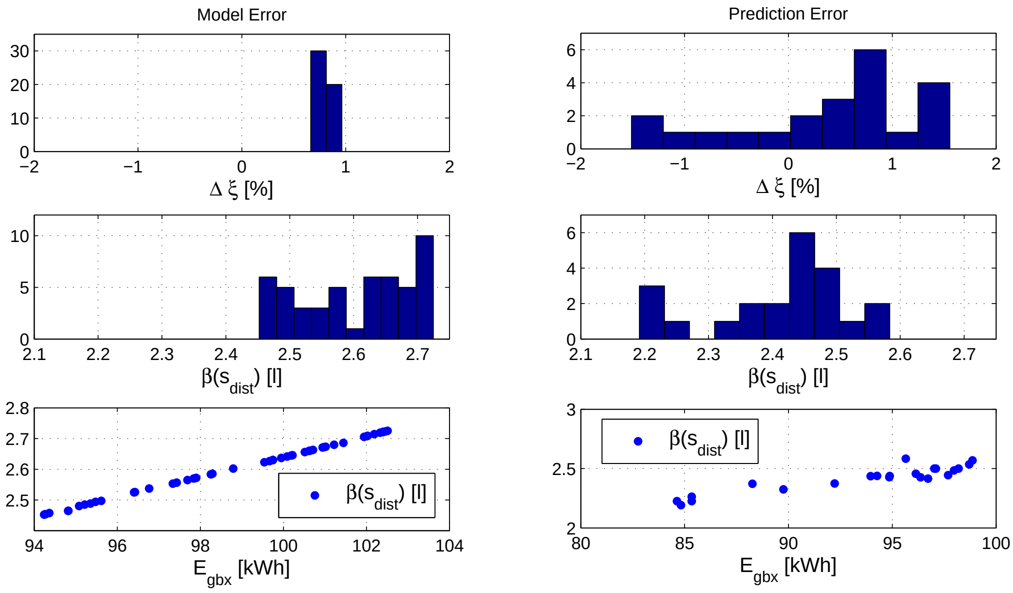

Figure 4.26: Robustness of final state attainment and fuel consumption against model error and driver behavior

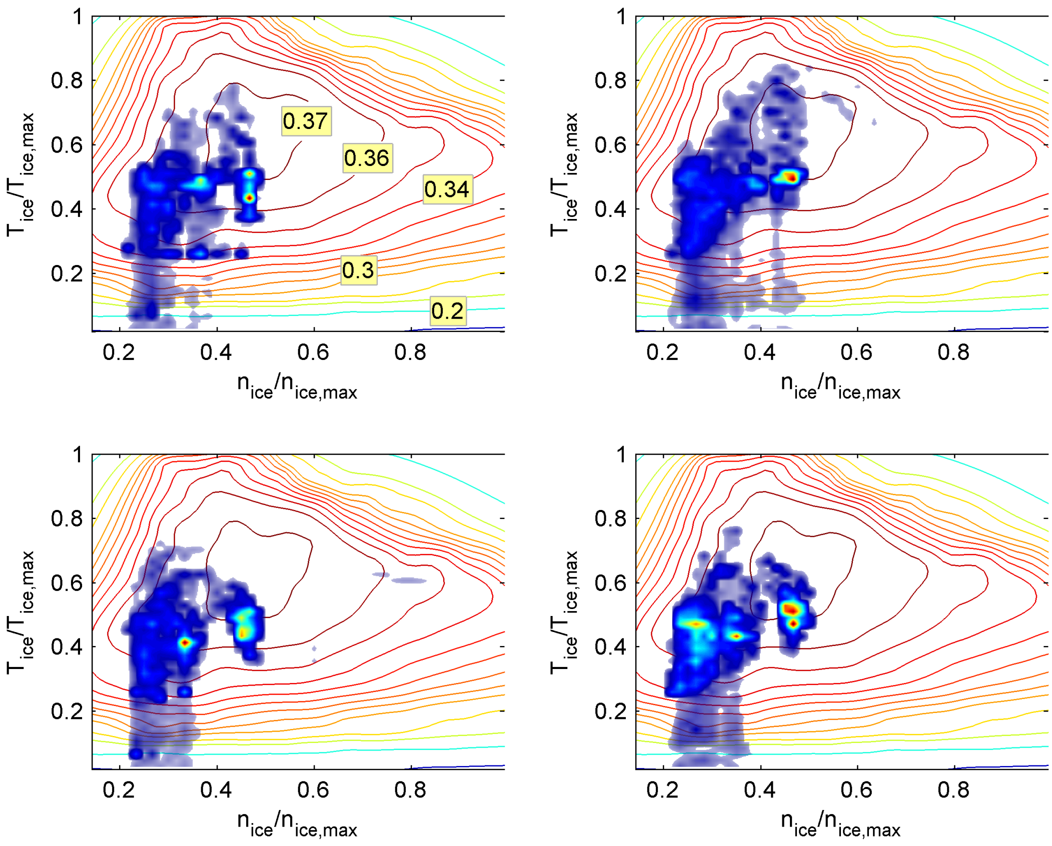

Figure 4.27: ICE efficiencies obtained with the predictive energy management

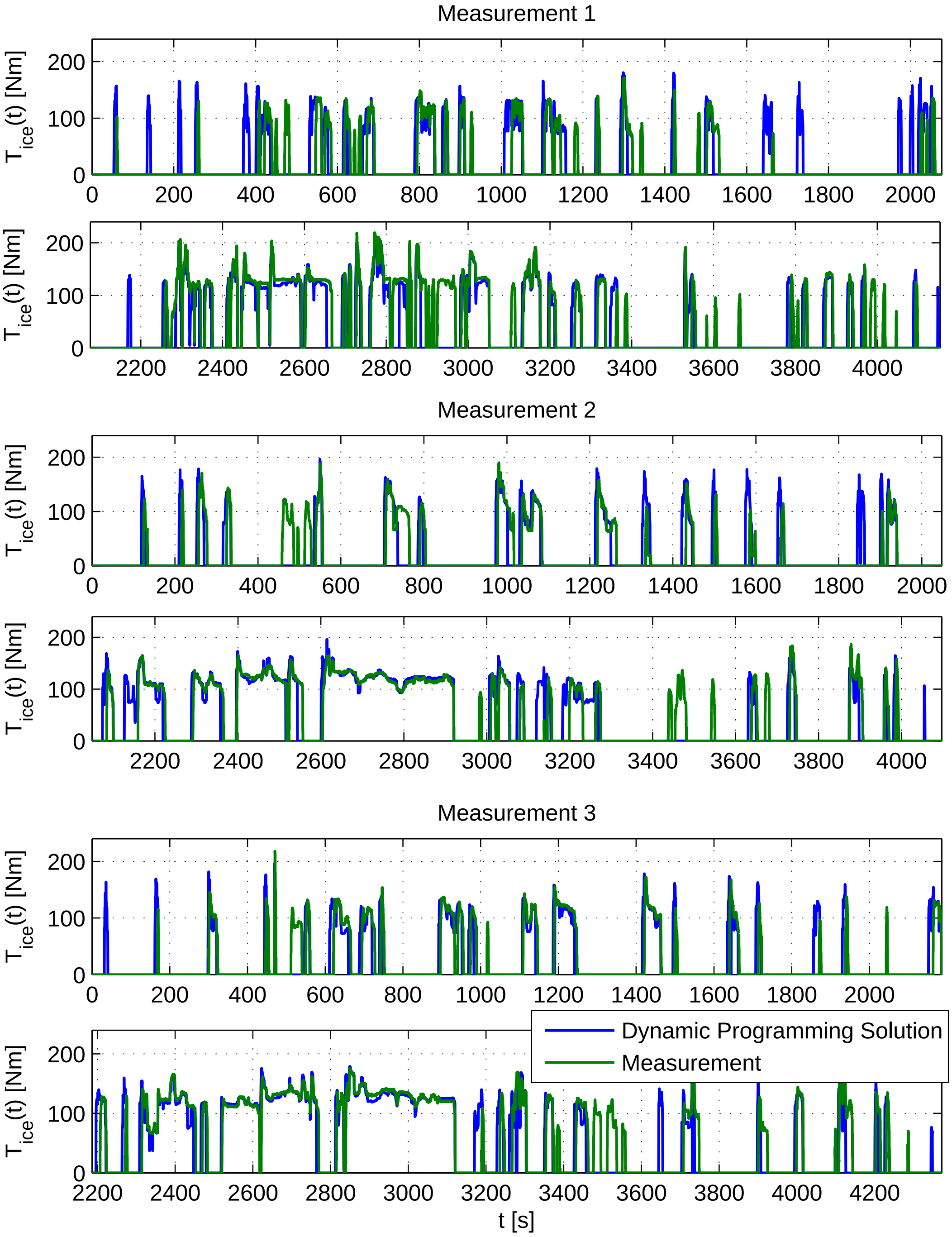

Figure 4.28: Comparison of three measurements with the dynamic programming solution

Chapter 5

Conclusion and Outlook

5.1 Conclusion

Many automotive systems can be modeled as switched systems as they involve discrete dynamics along with continuous dynamics. It is often of interest to find the control inputs and the switching sequence that minimize a given functional. The corresponding task can be defined as a Switched Optimal Control Problem. In this thesis, practical algorithms were applied to solve SOCPs for switched systems. One of the simplest algorithms, the indirect shooting algorithm is in most cases not applicable as the convergence region is very limited. Even for the simple examples provided in section 3.7, convergence was only achieved, when the first guess of the initial costate is very close to the optimal initial costate. Yet, applications exist, where this algorithm outperforms other algorithms by far, as has been demonstrated for the hybrid vehicle model. The embedding approach is very simple to implement but has in the numerical experiments attained inferior results than other algorithms. The computation times were somewhere between the indirect shooting method, if it converges, and the combined method. Yet, it still has the potential for significant reduction of the computation time by applying one or more of the following measures:

- Calculation of the gradients using automatic differentiation
- Using a collocation approach and sparse algebra for solving the quadratic sub-problem.

This applies for all methods that involve the solution of a nonlinear program and hence also for the combined method. This method has proven to obtain very good results on different problems. The computation time will still be quite high, due to the fact that the switching sequence is altered at only one time instant per main iteration. Further numerical investigations and experiments may improve the algorithm by alternating the switching sequence at more than one instant.

If the state exhibits jumps on a switching, the solvability of the SOCP is considerably impaired. Dynamic programming can in this case find the globally optimal solution with respect to a chosen discretization. While the dimension of the discrete state vector enters only linearly in the computation time for DP, the dimension of the continuous state vector increases the computational demand exponentially and hence reduces the general applicability of DP. In that case, embedding with relaxation of the jump function can yield a suboptimal solution that will in many cases be only slightly inferior than the global optimum. Dynamic programming has the potential to reduce the computation time by vectorization of the inner loops of the algorithm or by parallelization. When used for the hybrid vehicle example, the computation time could be reduced by more than 70 % using vectorization.

Optimal control of hybrid vehicles is a timely topic in the automotive industry and a big challenge. A lot of progress has already been made in terms of optimal control of the continuous inputs. In this thesis, the algorithms for the solution of SOCPs were applied to different problems arising from the calibration of HEVs. The problems were successfully solved and it is demonstrated, that for different problem formulations, different algorithms are most suitable. The necessary conditions for hybrid optimal control and the results of the HOCP can be used to calculate a set of calibration parameters for rule-based energy management systems. Many of these parameters can directly be transferred to the electronic control unit, others need further tuning. This is caused by the limitation of the mathematical models that do not include aspects like driving comfort.

It was demonstrated that the indirect shooting method is applicable on a prototyping unit and that the computational demand is no burden for the implementation on an electronic control unit. This was achieved by storing the minimum of the Hamiltonian function on a grid of its input values. With a prediction of the drive cycle, as it can be obtained with the help of an intelligent traffic system and a driver-model, the indirect shooting method can be used to implement a predictive energy management that allows for the controlled depleting of the battery. The predictive energy management has been implemented in a plug-in hybrid vehicle close to series production and has yielded excellent results in terms of fuel consumption and robustness.

5.2 Outlook

Further research in the area of switched systems will include

- Further reduction of the computation times, especially for the solution of nonlinear programs. Automatic differentiation and exploiting sparse matrix structures have the potential of significant reduction of the computation times

- The inclusion of state constraints. Experiments have shown that methods based on a direct approach, such as the combined method, are able to incorporate state constraints easily. The theoretical foundation however is questionable, since necessary conditions for optimal control of dynamical systems with constrained arcs cannot be transferred to hybrid dynamical systems

- Sensitivity analysis. Not all operating conditions of a vehicle can be captured with a single model and constant parameters. Changing environmental conditions, aging or different load conditions will occur. The question of how the continuous controls as well as the switching sequence need to be adapted if certain vehicle parameters change could be answered using the theory of sensitivity analysis.

In the area of hybrid vehicle control, future work will involve driver classification and cycle analysis. It was demonstrated, that for near-optimal control of hybrid vehicles, the determination of the costate is the major challenge. For a known drive cycle, a predicted velocity profile can be generated. If the driver can be classified and respective parameters for the driver model can be estimated, the quality of this prediction can be even more improved. If no prediction of the future drive cycle is available, the past driving behavior could be consulted to determine an appropriate costate value that leads to charge-sustaining behavior.

Appendix A

Appendix

A.1 Dynamic Programming for Purely Continuous Systems

Consider an optimal control problem

$$\min_{u(t)} \Phi(x(t), u(t), t) \tag{A.1}$$

$$\Phi(x(t), u(t), t) = \int_{t_0}^{t_f} L(x(t), u(t), t)dt + \Psi(x(t_f)) \tag{A.2}$$

$$\dot{x} = f(x(t), u(t), t) \tag{A.3}$$

$$x(t_0) = x_0 \tag{A.4}$$

The task is to find the optimal control law $u(t)$ with its corresponding optimal state trajectory $x(t)$ over the interval $t \in [t_0, t_f]$, which minimizes the objective function (A.1).

Theorem 4 (Principle of Optimality)**:** If $(x(t), u(t))$ is an optimal solution to the optimal control problem (A.1-A.4) starting at $x(t_0)$, passing through $x(t_1)$ for an arbitrary $t_1 \in [t_0, t_f]$ and ending at $x(t_f)$, then the partial trajectory from $x(t_1)$ to $x(t_f)$ is also optimal with respect to the same problem formulation and $x(t_1)$ as initial condition.

The principle of optimality is defined for general multi-stage decision processes in Bellman [1957]. For continuous optimal control problems, a definition of this principle, as well as an extensive description of the algorithmic application can be found in the works of Kirk [1970] and Föllinger and Roppenecker [1988].

To formulate the Principle of Optimality in a mathematical way, it is helpful to introduce the value function $V(x(t),t)$ that provides the optimal cost for a trajectory starting at time t and the state $x(t)$:

$$V(x(t),t) = \min_{u(t)} \left\{ \int_{t}^{t_f} L(x(t),u(t),t)dt + \Psi(x(t_f)) \right\} \tag{A.5}$$

An analytical solution for this function will be hard to find but it will be amenable for numerical computations. For an arbitrary $t_1 \in [t_0, t_f]$, it can then be stated

$$\begin{aligned} & V(x(t_0),t_0) && \text{(A.6)} \\ &= \min_{u(t)} \left\{ \int_{t_0}^{t_1} L(x(t),u(t),t)dt + \int_{t_1}^{t_f} L(x(t),u(t),t)dt + \Psi(x(t_f)) \right\} && \text{(A.7)} \\ &= \min_{u(t)} \left\{ \int_{t_0}^{t_1} L(x(t),u(t),t)dt + V(x(t_1),t_1) \right\}, && \text{(A.8)} \end{aligned}$$

where $x(t_1)$ is obtained by integration as follows:

$$x(t_1) = x(t_0) + \int_{t_0}^{t_1} f(x(t),u(t),t)dt. \tag{A.9}$$

Hence, the OCP is split into parts. Instead of seeking the controls over the entire cycle, an optimal control problem is solved over a smaller interval and the value function gives the cost for the remaining cycle. This procedure can be applied on arbitrary small intervals. Given a time grid $\bar{t}$, the same procedure can theoretically be employed to determine the Value function for one time instant t^k and a given state $x(t^k)$:

$$V(x(t^k),t^k) = \min_{u(t)} \left\{ \int_{t^k}^{t^{k+1}} L(x(t),u(t),t)dt + V(x(t^{k+1}),t^{k+1}) \right\}. \tag{A.10}$$

Applying an explicit Euler discretization to this equation yields

$$V(x^k,t^k) = \min_{u^k} \left\{ L(x^k,u^k,t^k) \cdot h^k + V(x^{k+1},t^{k+1}) \right\}, \tag{A.11}$$

where the consecutive state x^{k+1} is given by

$$x^{k+1} = x^k + f(x^k,u^k,t^k) \cdot h^k \tag{A.12}$$

and $h^k = t^{k+1} - t^k$. Equation (A.11) is called the discrete Bellman equation. In this problem formulation, we are expecting the value function $V(x^{k+1},t^{k+1})$ to be known. As has been mentioned before, an analytical formulation of the value function will seldomly be available. A workaround is the application of a backwards recursion scheme and a gridding of the state x on a rectangular grid $\mathbb{G}_x$. The aim of the dynamic programming algorithm is, to calculate the value function on every point on the grid $\bar{t} \times \mathbb{G}_x$. The value function for x^N can easily be evaluated on this grid, since

$$V(x^N, t^N) = \Psi(x^N) \tag{A.13}$$

applies. For the time instant t^{N-1}, the discrete Bellman equation (A.11) can then be written

$$V(x^{N-1}, t^{N-1}) = \min_{u^{N-1}} \left\{ L(x^{N-1}, u^{N-1}, t^{N-1}) \cdot h^{N-1} + V(x^N, t^N) \right\} \tag{A.14}$$

Herein, $V(x^N, t^N)$ is defined on the grid $\mathbb{G}_x$. For any point that falls between the grid points, an appropriate interpolation scheme can be used, such that the function can be assumed to be entirely defined over the boundaries of the grid. Consequently, the control u^{N-1} that minimizes the discrete Bellman equation can be determined. The common procedure is to grid the controls on $\mathbb{G}_u$, evaluate the Bellman equation for each point on this grid and then select the control that minimizes $V(x^{N-1}, t^{N-1})$. This procedure is repeated at time instant $N-1$ for any point on $\mathbb{G}_x$. For each state, the value $V(x^{N-1}, t^{N-1})$ is saved and the corresponding optimal control is saved in a similar structure $U(x^{N-1}, t^{N-1})$. With $V(x^{N-1}, t^{N-1})$ being defined on the grid, we can move one step backwards to $N-2$ and repeat the same procedure. To allow for more compact formulations, we define

$$g(x^k, u^k, t^k) = x^k + f(x^k, u^k, t^k) \cdot h^k. \tag{A.15}$$

such that the algorithm can be summarized as follows:

```
function DYNPROG
    Define $\mathbb{G}_x$, $\mathbb{G}_u$, $\bar{t}$
    $V^N \leftarrow \Psi(\mathbb{G}_x)$
    for $k \leftarrow N-1, -1, 1$ do
        for all $x_i \in \mathbb{G}_x$ do
            for all $u_j \in \mathbb{G}_u$ do
                $x^+ \leftarrow x_i + g(x_i, u_j, t^k)$
                $C_j \leftarrow L(x_i, u_j, t^k) \cdot h^k +$ INTP$(\mathbb{G}_x,\ V^{k+1},\ x^+)$
            end for
            $j^* \leftarrow \arg\min_j C_j$
            $V_i^k \leftarrow C_{j^*}$
            $U_i^k \leftarrow u_{j^*}$
        end for
    end for
    return $V, U$
end function
```

Algorithm 5: Dynamic programming algorithm

The procedures INTP(X, Y, x) perform an interpolation over a grid X with the corresponding grid values Y at the points x. This is neccessary as the consecutive states x^+ are unlikely to fall on grid points.

Once the functions V and U are defined on the on the grid $\bar{t} \times \mathbb{G}_x$ with the procedure DYNPROG, optimal trajectories x and u can be recovered with the procedure DPTRAJECTORY:

```
function DPTRAJECTORY($V$, $U$, $\mathbb{G}_x$, $x^1$, $\bar{t}$)
    for $k \leftarrow 1, N-1$ do
        $u^k \leftarrow$ INTP$(\mathbb{G}_x,\ U^k,\ x^k)$
        $x^{k+1} \leftarrow x^k + g(x^k, u^k, t^k)$
    end for
    return $\overline{u}, \overline{x}$
end function
```

Algorithm 6: Calculating a specific optimal trajectory

For a given starting point (t^1, x^1), the optimal control u^1 is determined from $U(x^1, t^1)$. This will again require an interpolation, as x^1 is not necessarily on the state grid. Using this control, the consecutive state x^{k+1} is computed using (A.15). This is repeated until $k = N - 1$.

A.2 Smooth Function Representation

To achieve reliable results, it is required to use a precise vehicle model. The efficiencies of the individual powertrain elements are usually determined from measurements and need to be stored in some form for later evaluation in the optimization procedure. This function can be expressed as $g(p_x, p_y)$ with the grid defining column vectors p_x and p_y and is subject to the following requirements:

- The optimization is gradient based, hence the function g should be twice continuously differentiable in the direction for which gradients are needed
- The optimization procedure needs a high amount of function evaluations. Consequently, the evaluation needs to be fast
- Measured data are usually subject to noise, which might prevent convergence of the optimization. The function should smooth the measured data up to a desired extent.

In the literature, Willans-line-methods or interpolation schemes are often used (Wei [2004]). However, the former is usually prone to considerable precision loss, while the latter only fulfills the conditions of differentiability, when a very costly spline interpolation scheme is used. Thus, a two-step approach is adopted, that satisfies the requirements mentioned above:

1. As a first step, a smooth surface on an equidistant grid is generated from the measured data
2. In the second step, a surface spline is generated that exactly represents the surface on the grid points. The coefficients of the splines are saved for later evaluation.

A.2.1 Smooth Surface Generation

The main interest in the first step is to span a smooth surface over a set of points given by the column vectors s_x and s_y and the column vector with corresponding measured values b, each of them having the dimension $\mathbb{R}^{l_m}$. The surface is to be defined on an equidistant grid of the size $l_{p_x} \times l_{p_y}$. Using a radial basis function for interpolation, any point on the surface can be calculated by

$$z_i(p_x, p_y) = \sum_{i=1}^{l_m} \pi(d_i(p_x, p_y)) \cdot w_i, \tag{A.16}$$

where d_i is the distance of the grid point (p_x, p_y) to $(s_{x,i}, s_{y,i})$ and the center of the radial basis function $\pi(d)$. The parameters w_i can be found by solving the linear system

$$\Pi(D)w = b, \tag{A.17}$$

where the matrix $\Pi(D) \in \mathbb{R}^{l_m \times l_m}$ contains the function values $\Pi_{i,j} = \pi(D_{i,j})$ of the radial basis function. The matrix entry $D_{i,j}$ indicates the distance of the two points $(s_{x,i}, s_{y,i})$ and $(s_{x,j}, s_{y,j})$. We follow Sandwell [1987] in using the radial green function

$$\pi(d) = |d|^2 \cdot (\ln |d| - 1) \tag{A.18}$$

for its minimum curvature and its smooth and stable extrapolation properties. The interpolation is additionally enhanced with a configurable regularizer as implemented in D'Errico [2010]. The regularization is given by the linear equation

$$PBz = 0, \tag{A.19}$$

where B is a sparse quadratic matrix of dimension $l_{p_x} \cdot l_{p_y} \times l_{p_x} \cdot l_{p_y}$ that gives the finite differences of second order for the grid point values when multiplied with the column vector z. The diagonal matrix P with the same dimension allows for an individual weighting of the regularization for every value in z. At the same time, we require the values in z to be built of the sum of the l_m weighted radial basis functions, which can be expressed by the linear system

$$\Pi(D_z)w = b, \tag{A.20}$$

where $D_z \in \mathbb{R}^{l_{p_x} l_{p_y} \times l_m}$ is a matrix that contains the distances of the grid points to the centers of the basis functions. Concatenating the variables w and z in the column vector

$$p = \begin{bmatrix} w & z \end{bmatrix}^T \tag{A.21}$$

and defining

$$\overline{A} = \begin{bmatrix} \Pi(D) & 0 \\ 0 & PB \end{bmatrix}, \quad \overline{b} = \begin{bmatrix} b \\ 0 \end{bmatrix}, \quad \overline{C} = \begin{bmatrix} \Pi(D_z) & -I \end{bmatrix}, \tag{A.22}$$

where I is the unity matrix, the regularized interpolation problem can then be formulated as a linear quadratic constrained optimization problem

$$\min_p ||\overline{A}p - \overline{b}||_2 \tag{A.23}$$

$$\overline{C}p = 0 \tag{A.24}$$

that can be solved with standard methods.

A.2.2 Analytical Representation

From the previous step, a smooth surface is obtained and represented as a finite set of points on an equidistant grid. This surface is to be converted to an analytic representation $g(p_x, p_y)$. If during the optimization gradients for both parameters p_x and p_y are needed, the coefficients for a tensor product spline as in de Boor [2001] are stored in a data structure for later evaluation. If only one parameter p_x or p_y is needed for gradient calculation, we use 1-dimensional blended splines, which allow for a much faster evaluation and fulfill the $\mathcal{C}^2$-condition in the required direction. The information for the blended spline is stored as a set of 1-dimensional splines over the grid points. For evaluation of the points between these splines, a linear interpolation of the neighboring spline coefficients is performed and the spline then evaluated. This dramatically reduces the time needed for an evaluation of the function $g(p_x, p_y)$ to one polynomial evaluation.

List of Figures

2.1 Hybrid phenomena, from left to right: Controlled switching, state jump and autonomous switching (Passenberg [2012]) . . . 11

2.2 Example of a switched system 14

2.3 Example trajectory with one switching 18

3.1 Insertion of the mode q^* in the interval $[t^* - \frac{1}{2}\Delta t, t^* + \frac{1}{2}\Delta t]$ . . . 45

3.2 Exemplary jump function $\hat{\delta}$ for a system with three subsystems on the surface $\Delta_1 + \Delta_2 + \Delta_3 = 0$ 54

3.3 Control and state trajectories obtained for Example 1 with the indirect shooting method, embedding and the combined method 61

3.4 Discrete state $q(t)$ and state space $x_1(t), x_2(t)$ at different iterations of the combined method 63

3.5 Control and state trajectories obtained for Example 2 with the indirect shooting method, embedding and the combined method 64

3.6 Control and state trajectories obtained for Example 3 with dynamic programming and embedding 66

3.7 $\sigma(t)$ for different iterations of the rounding scheme 66

4.1 Sketch of a parallel hybrid powertrain configuration with clutches K0 and K1 and an automatic gearbox 68

4.2 Open circuit analogy applied as battery model 72

4.3 Validation of the models for the mechanical and electrical subsystem for three different drive cycles. 75

4.4 Sketch of the elements in the exhaustsystem regarded in the model 79

4.5 Comparison of the results of the thermodynamic model compared to two measurements 82

4.6 New European Drive Cycle velocity profile and measured gearbox input torque and input shaft speed for a given gear sequence 85

4.7 First hill of the Federal Test Procedure: Velocity profile and measured gearbox input torque and input shaft speed for a given gear sequence 86

4.8 Function $B_{rem}(\xi(t_f)-\xi(t_{cyc}))$ that provides the optimal fuel consumption for the remaining drive cycle $t \in (t_f, t_{cyc}]$, depending on the difference $\xi(t_f) - \xi(t_{cyc})$ 88

4.9 Solution of problem $\mathcal{P}_1$ obtained with the indirect shooting method and the embedding method 91

4.10 Solution of problem $\mathcal{P}_2$ obtained with the indirect shooting method 92

4.11 Comparison of the results for $\mathcal{P}_3$ obtained with dynamic programming and the embedding with relaxed jump function approach 94

4.12 Evolution of the relaxed switching variable σ_1 over the iterations of the greedy rounding scheme 94

4.13 Results and approximated costates obtained for $\mathcal{P}_4$ with different bounds for the artificial emission state Z. The corresponding states of each depicted costate are written as subscript 96

4.14 LUTs defining the MG-torque in hybrid drive mode for different costate values 99

4.15 Comparison of $p(t)$ and $\xi(t)$ with and without constant costate assumption 100

4.16 LUTs defining the MG-torque in hybrid drive mode for different costate values 102

4.17 LUT defining the torque-threshold for an ICE-start 103

4.18 LUTs defining the optimal gear selection for a given driver request 103

4.19 Difference $|\Delta\mathcal{H}|$ for electric and hybrid drive mode and $p = -150$. Deviating from the recommended start torque is more acceptable at low engine speeds 104

4.20 Control strategy for the predictive energy management 107

4.21 Estimated velocity profile over a given drive cycle 109

4.22 Predicted velocity and optimal engine start sequence with respect to the prediction. Deviations to the measured trajectory of v and the measured start sequence can be noticed 110
4.23 Procedure for obtaining a suboptimal trajectory fulfilling the state constraint 111
4.24 Reference and measured trajectories for $\xi(s)$ and controller output $p(s)$ for four measurements over the same cycle with different initial charging states 112
4.25 Diversity of recorded velocity profiles for the test drive cycle . . 113
4.26 Robustness of final state attainment and fuel consumption against model error and driver behavior 115
4.27 ICE efficiencies obtained with the predictive energy management 115
4.28 Comparison of three measurements with the dynamic programming solution 116

List of Tables

2.1 Height of the jump Δx as given by the function values $\delta_{(q^-,q^+)}$ for a given state x at time t arranged in a table 11

3.1 Comparison of cost function value and computation time 61
3.2 Comparison of cost function value, computation time and fulfillment of the final state constraint 63
3.3 Comparison of cost function value and computation time 65

4.1 Discrete decisions made at any time t for a given value $q(t)$ . . . 76
4.2 Comparison of cost function value and computation time for $\mathcal{P}_1$ 91
4.3 Comparison of cost function value and computation time for $\mathcal{P}_3$ 93
4.4 Effect of the constant costate assumption on the fuel consumption 99

Nomenclature

α	Ignition angle
β	Accumulated fuel consumption
$\delta_{(q^-,q^+)}$	Jump function for a switching from q^- to q^+
Γ_f	Runge-Kutta recursion for vector field f
ι_{gbx}	Gear ratio
$\mathcal{H}$	Hamiltonian function
$\mathcal{L}$	Lagrangian function
$\mathcal{Q}$	Set of discrete states
$\mathcal{U}_q$	Set of feasible controls u for discrete state q
μ	Lagrangian multipliers
ν	Lagrangian multipliers
ω	Angular velocity
ϕ	Cost function of Mayer type
Π	Switching function for general hybrid systems
Ψ	Final state penalty term
ψ	Final state constraint
σ	Discrete state boolean vector
Θ	Switching sequence

θ	Parameterized controls
Υ	Nonlinear equation for the solution of TPBVPs
υ	Concatenated control vector
ϑ	Temperature
Ξ	Parameterized control function
ξ	Battery state of charge
ζ	Drive mode
$bsfc$	Brake-specific fuel consumption
c_u	Control restraints
e	Road inclination
f_q	Vector field of a hybrid system
G	Extended vector field for $\dot{x}$ and $\dot{p}$
g_i	Smooth multivariate function
h	Sampling time interval
I	Battery current
k	Discrete time index
L	Lagrange term
l_j	Total number of switching times
m_{cyl}	Relative cylinder charge
N	Number of sampling times
P	Power
p	Costate
Q	Battery capacity

Q	Number of discrete states
q	Discrete state
R_i	Inner resistance of the high-voltage battery
s	Driving distance
T	Torque
t	Time
t_0	Initial time
t_f	Final time
t_j	Switching time
t_{comp}	Computation time
u	Continuous system control input
V	Dynamic programming value function
v	Vehicle velocity
V_{oc}	Open circuit voltage of the high-voltage battery
W	Dynamic programming control function
w	Extended control vector, combining u and Δq
x	System state
Z	Artificial emission state
z	Extended state vector, combining x and q
ECMS	Equivalent fuel consumption minimization strategy
ECU	Electronic control unit
HEV	Hybrid electric vehicle
HOCP	Hybrid optimal control problem

ICE	Internal combustion engine
ITS	Intelligent traffic system
IVP	Initial value problem
K0	Clutch between ICE and MG
K1	Clutch between MG and gearbox
LUT	Look-up table
MG	Electrical motor/generator
MINLP	Mixed-integer nonlinear program
PHEV	Plug-in hybrid electric vehicle
PMP	Pontryagin’s minimum principle
SOCP	Switched optimal control problem
TPBVP	Two-point boundary value problem

Bibliography

M. Alamir and S. Attia. On solving optimal control problems for switched hybrid nonlinear systems by strong variations algorithm. In *Proceedings of the IFAC symposium on nonlinear control systems*, pages 558–563, 2004.

Matthias Althoff, Olaf Stursberg, and Martin Buss. Computing reachable sets of hybrid systems using a combination of zonotopes and polytopes. *Nonlinear Analysis: Hybrid Systems*, 4(2):233 – 249, 2010.

H. Axelsson, Y. Wardi, M. Egerstedt, and E.I. verriest. Gradient descent approach to optimal mode scheduling in hybrid dynamical systems. In *Journal of Optimization Theory and Applications, vol. 136, issue 2*, pages 167–186, 2008.

M. Back. Prädiktive Antriebsregelung zum energieoptimalen Betrieb von Hybridfahrzeugen. In *PhD-dissertation, Universität Karlsruhe*, 2005.

R. Van Basshuysen. *Ottomotor mit Direkteinspritzung: Verfahren, Systeme, Entwicklung, Potenzial.* Atz/Mtz-Fachbuch. Springer Fachmedien Wiesbaden, 2013.

R. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

S.C. Bengea and R.A. DeCarlo. Optimal and suboptimal control of switching systems. In *Proceedings of the 42nd IEEE Conference on Decision and Control*, pages 5295–5300, 2003.

J.T. Betts. *Practical Methods for Optimal Control and Estimation using Nonlinear Programming, second Edition.* Society for Industrial and Applied Mathematics, 2010.

J.T. Betts and W.P. Huffman. Exploiting sparsity in the direct transcription method for optimal control. *Computational Optimization and Applications*, 14(2):179–201, 1999.

H. G. Bock and K. J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proceedings 9th IFAC world*, pages 243–247, 1984.

T.J. Boehme, M. Schori, B. Frank, M. Schultalbers, and W. Drewelow. A predictive energy management for hybrid vehicles based on optimal control theory. In *Proceedings of the American Control Conference*, pages 6004–6010, 2013.

H.A. Borhan, A. Vahidi, A.M. Phillips, M.L. Kuang, and I.V. Kolmanovsky. Predictive energy management of a power-split hybrid electric vehicle. In *Proceedings of the American Control Conference*, pages 3970–3976, 2009.

M.S. Branicky and S.K. Mitter. Algorithms for optimal hybrid control. In *Proceedings of the 34th Conference on Decision and Control*, pages 2661–2666, 1995.

M.S. Branicky, V.S. Borkar, and S.K. Mitter. A unified framework for hybrid control. In *Proceedings of the 33rd Conference on Decision and Control*, pages 4228–4234, 1994.

M.S. Branicky, V.S. Borkar, and S.K. Mitter. A unified framework for hybrid control: Model and optimal control theory. *IEEE Transactions on Automatic Control*, 43(1):31–45, January 1998.

C.G. Broyden, J.E. Dennis, and J.J. Moré. On the local and superlinear convergence of quasi-newton methods. *IMA Journal of Applied Mathematics*, 12(3):223–245, 1973.

C. Büskens. *Optimierungsmethoden und Sensitivitätsanalyse für optimale Steuerprozesse mit Steuer- und Zustandsbeschränkungen.* PhD thesis, Universität Münster, 1998.

J.S. Chen and M. Salman. Learning energy management strategy for hybrid electric vehicles. In *Vehicle Power and Propulsion, 2005 IEEE Conference*, pages 68–73, Sept 2005.

F. Clarke and R. Vinter. The relationship between the maximum principle and dynamic programming. *SIAM Journal on Control and Optimization*, 25(5): 1291–1311, 1987.

W. Dahmen and A. Reusken. *Numerik für Ingenieure und Naturwissenschaftler*. Springer-Verlag Berlin Heidelberg, 2008.

C. de Boor. *A Practical Guide to Splines*. Springer-Verlag New York, Inc., 2001.

B. de Jager, T. van Keulen, and J. Kessels. *Optimal Control of Hybrid Vehicles*. Springer London, 2013.

J. D'Errico. Surface fitting using gridfit. `http://www.mathworks.com/matlabcentral/fileexchange/8998-surface-fitting-using-gridfit`, 2010.

A.V. Dmitruk and A.M. Kaganovich. The hybrid maximum principle is a consequence of pontryagin maximum principle. *Systems and Control Letters*, 57:964–970, 2008.

M. Dowell and P. Jarratt. The pegasus method for computing the root of an equation. *BIT Numerical Mathematics*, 12:503–508, 1972.

M. Egerstedt, Y. Wardi, and H. Axelsson. Transition-time optimization for switched-mode dynamical systems. *IEEE Transactions on Automatic Control*, 51, 2006.

P.J. Enright and B.A. Conway. Discrete approximations to optimal trajectories using direct transcription and nonlinear programming. *Journal of Guidance Control and Dynamics*, 15(4):994–1002, 1992.

O. Föllinger and G. Roppenecker. *Optimierung dynamischer Systeme: eine Einfuehrung fuer Ingenieure*. Oldenbourg, 1988.

A. Forsgren, P.E. Gill, and M.H. Wright. Interior methods for nonlinear optimization. *SIAM Review*, 44:525–597, 2003.

R.W. Freund and R.H. Hoppe. *Stoer Bulirsch: Numerische Mathematik 2.* Springer-Verlag Berlin Heidelberg, 2011.

M. Gerdts. *Optimal Control of ODEs and DAEs.* Walter de Gruyter, Berlin, 2012.

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

R. Goebel, R.G. Sanfelice, and A. Teel. Hybrid dynamical systems. *Control Systems, IEEE*, 29(2):28–93, April 2009.

Q. Gong, P. Tulpule, V. Marano, S. Midlam-Mohler, and G. Rizzoni. The role of its in phev performance improvement. In *Proceedings of American Control Conference*, pages 2119–2124, 2011.

I.E. Grossmann and Z. Kravanja. Mixed-integer nonlinear programming: A survey of algorithms and applications. In *THE IMA Volumes in Mathematics and its Applications, vol. 93*, pages 73–100, 1993.

L. Guzzella and A. Sciarretta. *Vehicle Propulsion Systems. Introduction to Modeling and Optimization.* Berlin: Springer-Verlag, 2005.

L. Guzzella and A. Sciarretta. *Vehicle Propulsion Systems: Introduction to Modeling and Optimization.* Springer-Verlag Berlin Heidelberg New York, 2007.

S. Hedlund and A. Rantzer. Optimal control of hybrid systems. In *Proceedings of the 39th IEEE Conference on Decision and Control*, pages 0191–2216, 1999.

S. Hedlund and A. Rantzer. Convex dynamic programming for hybrid systems. In *IEEE Transactions on Automatic Control, vol. 47, no. 9*, pages 1536–1540, 2002.

P. Hofmann. *Hybridfahrzeuge: Ein Alternatives Antriebskonzept für die Zukunft.* Springer Wien, New York, 2010.

D.H. Jacobsen, M.M. Lele, and J.L. Speyer. New necessary conditions of optimality for control problems with state-variable inequality constraints. *Journal of Mathematical Analysis and Applications*, 35(2):255–284, 1972.

L. Johannesson, S. Petterson, and B. Egardt. Predictive energy management of a 4qt series-parallel hybrid electric bus. *Control Engineering Practice*, 17: 1440–1453, 2009.

M. Kamgarpour and C. Tomlin. On optimal control of non-autonomous switched systems with a fixed mode sequence. *Automatica*, 48(6):1177 – 1181, 2012.

D. Karbowski, A. Rousseau, S. Pagerit, and P. Sharer. Plug-in vehicle control strategy: from global optimization to real time application. In *22nd Electric Vehicle Symposium, EVS22, Yokohama, Japan*, 2006.

S. Kermani, S. Delprat, T.M. Guerra, R. Trigui, and B. Jeanneret. Predictive energy management for hybrid vehicle. *Control Engineering Practice*, 20: 408–420, 2012.

N. Kim, D. Lee, S. W. Cha, and H. Peng. Optimal control of a plug-in hybrid electric vehicle (phev) based on driving patterns. In *International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium*, 2009.

N. Kim, S. Cha, and H. Peng. Optimal control of hybrid electric vehicles based on pontryagin's minimum principal. *IEEE Transaction on Control System Technology*, 19(5):1279–1287, 2011.

D.E. Kirk. *Optimal Control Theory: An Introduction.* Englewood Cliffs, N.J., Prentice-Hall, 1970.

D. Kum, H. Peng, and N. Bucknor. Supervisory control of parallel hybrid electric vehicles for fuel and emission reduction. *ASME Journal of Dynamic Systems, Measurement and Control*, 133(6):4498–4503, 2011.

D. Lee, S.W. Cha, A. Rousseau, N. Kim, and D. Karbowski. Optimal control strategy for phevs using prediction of future driving schedule. In *Proceedings of the 26th Electric Vehicle Symposium*, 2012.

C. Lemaréchal. Nondifferentiable optimization. In *Handbooks in Operations research and management science*. Elsevier Science Publishers B.V. (North-Holland), 1989.

A.S. Lewis and M.L. Overton. Nonsmooth optimization via quasi-newton methods. In *Math. Program., Ser. A(2013) 141*, pages 135–163, 2012.

S. Leyffer. Complementary constraints as nonlinear equations: Theory and numerical experience. *Springer Optimization and Its Applications*, 2:169–208, 2006.

C.C. Lin, J.M. Kang, J.W. Grizzle, and H. Peng. Energy management strategy for a parallel hybrid electric truck. In *American Control Conference, 2001. Proceedings of the 2001*, volume 4, pages 2878–2883, 2001.

J. Liu and H. Peng. Control optimization for a power-split hybrid vehicle. In *Proceedings of the 2006 American Control Conference, Minneapolis*, pages 466–471, June 2006.

L. Lukšan and J. Vlček. Globally convergent variable metric method for convex and nonsmooth unconstrained minimization. In *Journal of Optimization Theory and Applications, vol. 102, Issue 3*, pages 593–613, 1999.

J. Lygeros, K.H. Johansson, S.N. Simic, J. Jun Zhang, and S.S. Sastry. Dynamical properties of hybrid automata. *Automatic Control, IEEE Transactions on*, 48(1):2–17, Jan 2003.

B.S. Mordukhovich. On difference approximations of optimal control systems. *Journal of Applied Mathematics and Mechanics*, 42(3):452 – 461, 1978.

B.S. Mordukhovich. *Variational analysis and generalized differentiation II, Applications*. Grundlehren der mathematischen Wissenschaften. Springer, Berlin, 2006.

C. Musardo, G. Rizzoni, and B. Staccia. A-ecms: An adaptive algorithm for hybrid electric vehicle energy management. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pages 1816–1823, Dec 2005.

J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer Science and Business Media, LLC., 2006.

T. Nüesch, P. Elbert, M. Flankl, C. Onder, and L. Guzzella. Convex optimization for the energy management of hybrid electric vehicles considering engine start and gearshift costs. *Energies*, 7:834–856, 2014.

H.J. Oberle and W. Grimm. Bndsco: A program for the numerical solution of optimal control problems. *Hamburger Beiträge zur Angewandten Mathematik, Reihe B, Bericht 36*, 515:1–64, 2001.

G. Paganelli, Y. Guezennec, and G. Rizzoni. Optimizing control strategy for hybrid fuel cell vehicle. In *SAE World Congress*, 2002. Paper no.: SAE 2002-01-0102.

B. Passenberg. *Theory and Algorithms for Indirect Methods in Optimal Control of Hybrid Systems.* PhD thesis, Technische Universität München, 2012.

B. Passenberg, M. Leibold, O. Stursberg, and P. Caines M. Buss. The minimum principle for time-varying hybrid systems with state switching and jumps. In *Proceedings of the IEEE Conference on Decision and Control*, pages 6723 – 6729, 2011.

B. Piccoli. Necessary conditions for hybrid optimization. In *Proceedings of the 38th Conference on Decision and Control*, pages 410–415, 1989.

L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, and E.F. Mishchenko. *The Mathematical Theory of Optimal Processes.* John Wiley and Sons, 1962.

M.J.D. Powell. A fast algorithm for nonlinearly constrained optimization calculation. *Springer Lecture Notes in Mathematics*, 630:144–157, 1987.

L.B. Rall. *Automatic Differentiation: Techniques and Applications*, volume 120 of *Lecture Notes in Computer Science.* Springer Berlin New York, 1981.

C. Ress, D. Balzer, A. Bracht, S. Durekovic, and J. Löwenau. Adasis protocol for advanced in-vehicle applications. `http://www.ertico.com/assets/pdf/ADASISv2-ITS-NY-Paper-Finalv4.pdf`.

P. Riedinger, F. Kratz, C. Iung, and C. Zannes. Linear quadratic optimization for hybrid systems. In *Proceedings of the 38th IEEE conference on Decision and Control*, pages 3059–3064, 1999.

P. Riedinger, C. Iung, and F. Kratz. An optimal control approach for hybrid systems. *European Journal of Control*, 9:449–458, 2003.

P. Riedinger, J. Daafouz, and C. Iung. About solving hybrid optimal control problems. In *17th IMACS World Congress*, 2005.

G. Rousseau, D. Sinoquet, and P. Rouchon. Constrained optimization of energy management for a mild-hybrid vehicle. *Oil and Gas Science and Technology - Rev. IFP*, 62(4):623–634, 2007.

M. Rungger. *On the Numerical Solution of Nonlinear and Hybrid Optimal Control Problems.* PhD thesis, Universität Kassel, 2011.

J. Raisch S.A. Attia, V. Azhmyakov. State jump optimization for a class of hybrid autonomous systems. In *Proceedings of the 16th IEEE International Conference on Control Applications*, pages 1408 – 1413, 2007.

C.A. Sagastizábal. Nonsmooth optimization. In *Numerical Optimization - Theoretical and Practical Aspects, second edition.* Springer-Verlag Berlin Heidelberg New York, 1997.

S. Sager. *Numerical Methods for mixed-integer Optimal Control Problems.* PhD thesis, Universität Heidelberg, 2005.

S. Sager, M. Jung, and C. Kirches. Combinatorial integral approximation. *Mathematical Methods of Operations Research*, 73(3):363–380, 2011.

S. Sager, H.G. Bock, and M. Diehl. The integer approximation error in mixed-integer optimal control. *Mathematical Programming*, 133(1-2):1–23, 2012.

D.T. Sandwell. Biharmonic spline interpolation of geos-3 and seasat altimeter data. In *Geophysical research letters, Vol. 14, Issue 2*, pages 139–142, 1987.

M. Schori, T.J. Boehme, U. Becker, and M. Schultalbers. Verfahren zur Lösung von hybriden Optimalsteuerungsproblemen und deren Anwendung auf den

Betrieb von Hybridfahrzeugen. *at Automatisierungstechnik*, 61(12):831–840, 2013a.

M. Schori, T.J. Boehme, B. Frank, and M. Schultalbers. Solution of a hybrid optimal control problem for a parallel hybrid vehicle. In *7th IFAC Symposium on Advances in Automotive Control (AAC), Tokyo*, pages 109–114, 2013b.

M. Schori, T.J. Boehme, B. Frank, and M. Schultalbers. Calibration of parallel hybrid vehicles based on hybrid optimal control theory. In *Proceedings of the 9th IFAC Symposium on Nonlinear Control Systems, Toulouse, France*, pages 475–480, 2013c.

M. Schori, T.J. Boehme, B. Frank, and M. Schultalbers. Control optimization of discontinuous hybrid systems using embedding. In *Preprints of the 12th International Workshop on Discrete Event Systems*, pages 326–331, 2014a.

M. Schori, T.J. Boehme, T. Jeinsch, and M. Schultalbers. Optimal catalytic converter heating in hybrid vehicles. In *SAE Technical Paper 2014-01-1351*, 2014b.

M. Schori, T.J. Boehme, T. Jeinsch, and B. Lampe. Costate approximation from direct methods for switched systems with state jumps. In *Proceedings of the European Control Conference, accepted for publication*, 2015a.

M. Schori, T.J. Boehme, T. Jeinsch, and M. Schultalbers. A robust predictive energy management for plug-in hybrid vehicles based on hybrid optimal control theory. In *Proceedings of the American Control Conference, accepted for publication*, 2015b.

M. Schori, T.J. Boehme, B. Frank, and B. Lampe. Optimal calibration of map-based energy management for plug-in parallel hybrid configurations: a hybrid optimal control approach. *IEEE Transactions on Vehicular Technology*, 2015, accepted for publication.

N.J. Schouten, M.A. Salman, and N.A. Kheir. Fuzzy logic control for parallel hybrid vehicles. *Control Systems Technology, IEEE Transactions on*, 10(3): 460–468, May 2002.

A. Sciarretta, M. Back, and L. Guzzella. Optimal control of parallel hybrid electric vehicles. *Control Systems Technology, IEEE Transactions on*, 12(3): 352–363, May 2004.

T.I. Seidman. Optimal control for switching systems. In *Proceedings of the 21st Annual Conference on Information Science and Systems*, pages 485–489, 1987.

L. Serrao and G. Rizzoni. Optimal control of power split for a hybrid refuse vehicle. In *Proceedings of the American Control Conference*, pages 4498–4503, 2008.

M.S. Shaikh. *Optimal Control of Hybrid Systems: Theory and Algorithms.* PhD thesis, McGill University, Montreal, 2004.

A. Siburian. *Numerical Methods for Robust, Singular and Discrete Valued Optimal Control Problem.* PhD thesis, Curtin University of Technology, Perth, Australia, 2004.

C.M. Silva, T.L. Farias, H.C. Frey, and N.M. Rouphail. Evaluation of numerical models for simulation of real-world hot-stabilized fuel consumption and emissions of gasoline light-duty vehicles. *Transportation Research part D*, 1 (5):377–385, 2006.

M. Sivertsson, C. Sundstroem, and L. Eriksson. Adaptive control of a hybrid powertrain with map-based ecms. In *Proceedings of the IFAC World Congress*, 2011.

S. Stockar, V. Marano, M. Canova, G. Rizzoni, and L. Guzzella. Energy-optimal control of plug-in hybrid electric vehicles for real-world driving cycles. *IEEE Transactions on Vehicular Control*, 60(7):2949–2962, 2011.

H.J. Sussmann. A maximum principle for hybrid optimal control problems. In *Proceedings of the IEEE Conference on Decision and Control*, pages 425 – 430, 1999.

L. Tavernini. Differential automata and their discrete simulators. *Nonlinear Analysis, Theory, Methods and Applications*, 11:665–583, 1987.

L. Tavernini. Generic asymptotic error estimates for the numerical simulation of hybrid systems. *Nonlinear Analysis: Hybrid Systems*, 3(2):108 – 123, 2009.

M. Treiber and A. Kesting. *Verkehrsdynamik und -simulation: Daten, Modelle und Anwendungen der Verkehrsflussdynamik.* Springer-Verlag Berlin Heidelberg, 2010.

V. Turau. *Algorithmische Graphentheorie.* Oldenbourg Verlag, 2009.

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, June 2008.

R.B. Vinter. New results on the relationship between dynamic programming and the maximum principle. *Mathematics of Control, Signals and Systems*, 1(1):97–105, 1988. doi: 10.1007/BF02551239.

O. von Stryk. *Numerische Lösung otpimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der adjungierten Variable.* Fortschritts-Berichte VDI, Reihe 8, Nr. 441. VDI-Verlag, 1995.

O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37:357 – 373, 1992.

X. Wei. *Modeling and control of a hybrid electric drivetrain for optimum fuel economy, performance and drivability.* PhD thesis, Ohio State University, 2004.

R.B. Wilson. *A simplicial algorithm for concave programming.* PhD thesis, Graduate School of Business Administration, Harvard University, 1963.

H.S. Witsenhausen. A class of hybrid-state continuous-time dynamic systems. *IEEE Transactions on Automatic Control*, 11:161–167, 1966.

X. Xu. *Analysis and Design of Switched Systems.* PhD thesis, University of Notre Dame, 2001.

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*, volume 2, pages 1822–1827, 2000a.

X. Xu and P.J. Antsaklis. Optimal control of switched systems: New results and open problems. In *Proceedings of the American Control Conference*, pages 2683–2687, 2000b.

X. Xu and P.J. Antsaklis. Results and perspectives on computational methods for optimal control of switched systems. *Hybrid Systems: Computation and Control, Lecture Notes in Computer Science*, 2623:540–555, 2003.

X. Xu and P.J. Antsaklis. Optimal control of switched systems based on parameterization of the switching instants. *IEEE Transactions on Automatic Control*, 49(1):2–16, 2004.

N. Yamashita. Sparse quasi-newton updates with positive definite matrix completion. *Mathematical programming*, 115(1):1–30, 2008.

J. Zhang, K.H. Johansson, J. Lygeros, and S. Sastry. Zeno hybrid systems. *International Journal of Robust and Nonlinear Control*, 11(2):435–451, 2001.

Theses

- The solution of switched optimal control problems poses special requirements to a solution procedure, such that standard approaches for purely continuous optimal control problems cannot be applied without further adaptation
- The solution method needs to be selected carefully after a detailed analysis of the underlying system and the problem formulation
- The indirect shooting method can be extended for switched systems without state jumps without extensive modifications
- The use of the indirect shooting method is, as it is the case for purely continuous problems, often prevented by the strong sensitivity towards the first guess of the initial costate value
- The weaker necessary conditions for optimal control and the jump conditions of the costate prevent the use of the indirect shooting method for systems with state jumps
- An approximation of the costate, including the jump conditions, can be obtained from the solution of an optimal control problem via direct methods with fixed switching sequence
- The proposed combined method uses the advantages of both, indirect and direct methods, and is therefore a robust solution procedure
- When state jumps are present, dynamic programming will be able to find the globally optimal solution with respect to the chosen discretization. Yet, for systems with high dimension of the continuous state, the use will be prevented by the curse of dimensionality
- For systems with high dimension of the continuous state, the embedding approach with relaxation of the jump function can be a procedure for obtaining slightly suboptimal solution, while other methods may fail
- Direct solution methods still have significant potential for reduction of the computation time by using a collocation approach and exploiting the sparsity structures of the resulting nonlinear program

- The energy conversion in hybrid vehicles can be modeled with very good accuracy, such that model-based calibration can be applied
- With the help of optimal control theory, many aspects in the calibration of hybrid vehicles can be facilitated and the quality of the results can be improved
- A large set of calibration parameters for rule-based energy management systems in hybrid vehicles can be calculated analytically using the theory of switched optimal control
- The process of catalytic converter heating in hybrid vehicles can be modeled using a switched system formulation and the correspondingly formulated optimal control problem can be solved with the combined method
- With minor simplifications, the indirect shooting method can be applied for the solution of a switched optimal control problem on an electronic control unit for the development of a predictive energy management
- The velocity profile for a given route can be estimated with good accuracy based on information available in modern navigation systems
- The estimated velocity profile can then be used to formulate and solve a switched optimal control problem
- The predictive energy management proposed is able to reliably deplete the battery to a given value over the predicted driving route
- The predictive energy management is highly robust against model errors for the driver and for the vehicle model
- The energy management achieves nearly optimal fuel consumption, which can be demonstrated by comparing the results with dynamic programming solutions over the known drive cycle