TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Steuerungs- und Regelungstechnik

Theory and Algorithms for Indirect Methods in Optimal Control of Hybrid Systems

Benjamin Passenberg

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Dr. rer. nat. Holger Boche

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing./Univ. Tokio Martin Buss
2. Univ.-Prof. Dr.-Ing. Olaf Stursberg,
   Universität Kassel

Die Dissertation wurde am 19.01.2012 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 27.08.2012 angenommen.

Foreword

This thesis summarizes my research conducted in the past three years at the Institute of Automatic Control Engineering, LSR, of the Technische Universität München, Germany. The time at LSR was very pleasant not only because of the wonderful research topics and the freedom that I enjoyed but mainly because of the friendly and always supportive team at the institute and the laughs that we shared.

First of all, I thank my supervisor Prof. Martin Buss for the great opportunity to work as a research assistant at the LSR, for his steady support including the extraordinary collaboration possibilities, and for the freedom of research. Next, I thank Prof. Olaf Stursberg, who was my supervisor before he changed to Kassel, for the excellent collaboration, for the steady support, and for all his help during my entire time at the LSR. Third, I thank Prof. Peter E. Caines, who warmly welcomed me as a guest at McGill, introduced me into the topic of optimal control theory, and was a source of inspiration for me. Then, my thank goes to Dr. Marion Leibold for the many discussions and superior improvements for publications. I also thank my office mate Harald Voit and Michael Scheint for proof reading the manuscript and for all the helpful discussions and the personal chats.

The thesis was also supported and influenced by the excellent work of my students Maximilian Herrmann, Magnus Kröninger, Andreas Kunz, Alexander Mergel, Christoph Passenberg, Georg Schnattinger, Dominik Sieber, Michael Steinegger, Steffen Walther, and Lu Zhong. Many thanks for the strong commitment and the contributions to this thesis.

Finally, I thank my parents Andrea and Georg, my brother Christoph, and last but not least my wife Carolina for their love, their steady support, and the numerous discussions.

Munich, November 2011 Benjamin Passenberg

Abstract

Theoretical results and algorithms for indirect methods in optimal control of hybrid systems are introduced that overcome limitations and increase the competitiveness in comparison with direct methods and dynamic programming. Indirect methods find an optimal solution by satisfying optimality conditions and provide results with high accuracy. Highly accurate solutions are important for applications like trajectory planning for space vehicles and as benchmark for other optimal control approaches. However, indirect methods suffer from several limitations, which restrict their applicability. In this thesis, several limitations are overcome and strengths are further developed:

(i) The class of hybrid optimal control problems for which optimality conditions can be provided is extended. The extensions include additionally the cases of controlled switching with simultaneous resets of the continuous state, explicitly time-varying functions, and intersecting switching manifolds in hybrid systems with partitioned continuous state space.

(ii) Two approaches are proposed for a simplified initialization of indirect methods. The first approach uses solutions from a direct method, which is initialized straightforwardly, to find initial values for an indirect method. The second approach is a novel indirect method for hybrid optimal control problems. The method is based on a so-called min-H algorithm, that is initialized with comparable ease as a direct method.

(iii) The novel min-H algorithm has an enlarged domain of convergence compared to current indirect methods, that have severe convergence limitations. In contrast to other indirect methods, the presented min-H algorithm is proven to be globally convergent to a local minimum and shows an increased numerical stability.

(iv) Two algorithms are introduced that reduce the computational complexity to find an optimal discrete state sequence for hybrid systems with autonomous switching in contrast to former algorithms, which have a combinatorial complexity. The first algorithm uses branch and bound principles. The second algorithm varies the discrete state sequence simultaneously with the continuous optimization and provably finds a locally optimal discrete state sequence for hybrid optimal control problems with partitioned state space.

(v) For an improved online solvability, two efficient algorithms are proposed that exploit structural information of the optimality conditions for hybrid optimal control problems. The first algorithm is based on model predictive control and uses neighboring extremal solutions for the reinitialization of the optimization and local, optimal feedback control laws. In the second algorithm, optimal control primitives are optimally concatenated for the solution of linear-quadratic and feedback linearizable hybrid optimal control problems.

The effectiveness of the presented methods is illustrated in several numerical examples. The competitiveness of indirect methods is increased significantly by the results derived in this thesis. It is recommended to use indirect methods if highly accurate solutions are required, if the overall computational complexity has to be low, and if online solutions are desired in cases where a precomputation as used in dynamic programming is not possible.

Zusammenfassung

Neue theoretische Ergebnisse und Algorithmen für indirekte Methoden zur Lösung von hybriden Optimalsteuerungsproblemen werden eingeführt. Das Ziel der Arbeit besteht darin Einschränkungen zu überwinden und die Konkurrenzfähigkeit im Vergleich zu anderen Lösungsmethoden wie direkten Methoden und dynamischer Programmierung zu erhöhen. Optimale Lösungen von indirekten Methoden basieren auf der Erfüllung von Optimalitätsbedingungen. Sie zeichnen sich durch eine hohe Genauigkeit aus. Diese ist für Anwendungen wie die Trajektorienplanung bei Raumschiffen und als Benchmark für andere Optimalsteuerungsansätze wichtig. Allerdings leiden indirekte Methoden unter einigen Nachteilen, die ihre Anwendbarkeit limitieren. In dieser Arbeit werden einige Einschränkungen beseitigt und Stärken weiter ausgebaut, wie im Folgenden beschrieben:

(i) Die Klasse von hybriden Optimalsteuerungsproblemen, für die Optimalitätsbedingungen bereitgestellt werden können, wird erweitert. Damit sind nunmehr auch kontrolliertes Schalten mit gleichzeitigen Sprüngen des kontinuierlichen Zustands, explizit zeitabhängige Funktionen und sich schneidende Schaltmannigfaltigkeiten bei hybriden Systemen mit partitioniertem, kontinuierlichem Zustandsraum lösbar.

(ii) Zwei Ansätze für eine vereinfachte Initialisierung von indirekten Methoden werden vorgeschlagen. Der erste Ansatz verwendet Lösungen einer einfach zu initialisierenden direkten Methode um Startwerte für eine indirekte Methode zu finden. Der zweite Ansatz stellt eine neuartige indirekte Methode für hybride Optimalsteuerungsprobleme dar. Die Methode basiert auf min-H Algorithmen, die ähnlich einfach wie direkte Methoden initialisiert werden können.

(iii) Der neuartige min-H Algorithmus hat einen größeren Konvergenzbereich als aktuelle indirekte Methoden, die bisher schwerwiegende Einschränkungen der Konvergenzfähigkeit aufweisen. Im Gegensatz zu anderen indirekten Methoden ist der vorgestellte min-H Algorithmus bewiesenermaßen zu einem lokalen Minimum global konvergent und zeichnet sich durch eine erhöhte numerische Stabilität aus.

(iv) Zwei Algorithmen werden eingeführt, um die kombinatorische Komplexität früherer Algorithmen zur Berechnung einer optimalen diskreten Zustandssequenz für hybride Systeme mit autonomem Schalten zu reduzieren. Der erste Algorithmus basiert auf Branch and Bound Prinzipien. Der zweite Algorithmus variiert die diskrete Zustandssequenz gleichzeitig mit der kontinuierlichen Optimierung und findet bewiesenermaßen eine lokal optimale, diskrete Zustandssequenz für hybride Systeme mit partitioniertem Zustandsraum.

(v) Zwei effiziente Algorithmen zur verbesserten Online-Lösbarkeit werden vorgestellt, die in Optimierungsbedingungen für hybride Optimalsteuerungsprobleme gegebene, strukturelle Informationen ausnutzen. Der erste Algorithmus basiert auf modellprädiktiver Regelung und verwendet benachbarte, extremale Lösungen zur Reinitialisierung der Optimierung und lokale, optimale Feedback-Regelgesetze. Im zweiten Algorithmus werden optimale Steuerungsprimitive optimal miteinander verknüpft, um linear-quadratische und Feedback-linearisierbare hybride Optimalsteuerungsprobleme zu lösen.

Die Effektivität der vorgestellten Methoden wird anhand mehrerer numerischer Beispiele veranschaulicht. Die Konkurrenzfähigkeit indirekter Methoden wird durch die in dieser Arbeit erreichten Ergebnisse deutlich erhöht. Es wird empfohlen indirekte Methoden zu verwenden, wenn Lösungen mit hoher Genauigkeit gefordert sind, wenn eine niedrige Berechnungskomplexität erforderlich ist oder wenn Online-Lösungen in Fällen gewünscht werden, in denen eine Vorabberechnung, wie sie in dynamischer Programmierung genutzt wird, nicht möglich ist.

Contents

1 Introduction

The biological and technical systems and coherences that men observe, analyze, and design become increasingly complex. Partly, this trend is pushed by the rapid development of computer technology. To be able to deal with complex technical systems possibly including physical systems and computers, a powerful modeling language is required. Hybrid systems are such a powerful language that can model a large class of complex systems. Hybrid systems combine continuous and discrete effects and are thus able to model systems with a close and inseparable interconnection of continuous and discrete dynamics. Hybrid systems appear in many fields, e.g. transportation, energy, healthcare, manufacturing, chemical processes, finance, and consumer compliances. On a more technical level, this includes embedded systems, that combine digital and analog devices, and mechatronic systems, where mechanical and electrical systems and informatics are closely related. Furthermore, cyber-physical systems belong to the class of hybrid systems, since a close interconnection of physical systems and computers is given. The physical systems are modeled by continuous dynamics, while computers operate time-discrete on a digital hardware and algorithms produce discrete decisions based on some logic.

Examples for hybrid systems are switched electric circuits, where the continuous dynamics is given by Kirchhoff's laws and the physical characteristics of inductors and capacitors. The discrete dynamics consists of switches, which can be opened or closed [62]. Robotic systems are hybrid if together with the continuous dynamics also impacts or different contact situations with the environment are considered. Impacts cause the continuous state to impulsively change its values [33] and different contact situations change the continuous dynamics of the robot [9]. Another example is flight control, which uses different modes of operation [153]. In a further application in flight control, a hybrid system is used to model errors in a computer based control system due to radiation [193]. A biological example is the modeling of swarms of fireflies. Fireflies have continuous dynamics and from time to time they undergo an impulsive stimulus causing them to flash. Especially interesting here is the effect of the flashing of one firefly onto the other fireflies, which reveals a strong coupling between the states of fireflies [34]. Asset strategies in stock investment are modeled with hybrid systems. The price evolves continuously and buying or selling interventions are discrete decisions [195].

Researchers and engineers usually do not only want to model complex systems but also control them to achieve a desired behavior. With increasing economic and ecological pressure, companies are more and more interested in optimally controlling their technical systems and financial processes. Frequent goals of optimal control are to minimize the consumption of time, energy, power, or tracking error and to maximize profit, throughput, and the quality of goods or services. To achieve the goals, the tasks in optimal control of hybrid systems are to find an optimal continuous control trajectory simultaneously with optimal discrete decision times and an optimal sequence of discrete decisions.

A large variety of examples can be found in literature where hybrid systems are controlled optimally. In manufacturing, the physical production process is modeled with continuous dynamics and the different production steps form the discrete dynamics. The sequence of production steps is subject to optimization along with the continuous control. The goal is to maximize the number of goods produced, while maintaining a high quality [123]. In another example, optimal control is applied to find an optimal start up procedure for a continuously stirred, chemical tank reactor with valves and changing dynamics depending on the fill level of the tank [162]. With the aim to maximize the energy efficiency, the compass gait of a biped is optimally controlled. The hybrid nature of the problem arises from different contact situations of the feet [122, 150]. The life-time of a battery for consumer electronics is maximized by an optimal discharge strategy using a cyber-physical system model [192].

In technical processes, it is often not possible to find an optimal control over the complete time horizon, since either the computational time would become too long or since information about events happening in the far future is not provided. In this case, optimal control is determined for a shortened time horizon. After the first part of the control trajectory has been applied to the system, the optimization is repeated on the receding time horizon. This control approach is called model predictive control (MPC) and a lot of specialized solution methods are developed for MPC of hybrid systems. MPC is used in a gas supply system based on a hybrid model, where gas from a steel-works is to be delivered in a certain quality to a power plant to heat different boilers [16]. A solar air conditioning plant is also controlled with hybrid MPC. There, solar collectors heat water that is transferred to an absorption machine, which produces chilled water for cooling [42]. With MPC, the speed of a truck and the selected gear are controlled, such that the fuel consumption and travel time are optimized simultaneously for maximizing the profit [209].

Optimal control for hybrid systems is challenging due to the close interconnection of continuous and discrete dynamics. Methods for purely continuous optimal control problems are not able to handle hybrid optimal control problems (HOCPs) since discrete decisions influence the continuous optimization. Similarly, methods for purely discrete optimization problems are unsuitable since the discrete optimization strongly depends on the continuous optimal control. Combining methods from continuous optimal control and discrete optimization is not straightforward. Continuous optimal control relies in general on infinitesimal variations of control and state variables and derivatives of functions. Such concepts are difficult to translate to integer-valued discrete decision problems. In contrast, discrete optimization often relies on graph based search methods, which are not applicable for continuous optimal control problems as these are infinite dimensional. In the next section, several approaches with their strengths and weaknesses are presented that tackle the problem of finding an optimal control for hybrid systems.

1.1 Solution of Hybrid Optimal Control Problems

Hybrid systems are modeled by continuous and discrete dynamics, continuous and discrete states, and continuous and discrete controls. The continuous dynamics is usually given by differential or difference equations, which are functions of the continuous state and control and which change with the discrete state. The discrete dynamics is formed by a discrete structure like automata, which includes discrete states and rules for the transition between them. Four basic types of hybrid phenomena are identified: *autonomous switching*, *controlled switching*, *autonomous impulses*, and *controlled impulses* [29]. Hybrid systems contain from one up to all of the four basic phenomena. It is even possible that autonomous switching and autonomous impulses as well as controlled switching and controlled impulses happen at the same time, respectively. The basic types of hybrid phenomena specify how a transition from one discrete state to another occurs.

Let a system evolve in discrete state 1 according to a differential equation that depends on the continuous state and the continuous control, see Fig. 1.1. Then an autonomous switching occurs when the continuous state trajectory crosses the switching manifold. The switching causes the discrete state to change - in this example from discrete state 1 to discrete state 2. The continuous states before and after the switching are equal. In the new discrete state, the continuous state trajectory follows the differential equation valid in discrete state 2. In technical systems, autonomous switching occurs for example when different contact situations in robotics are considered and when logical decisions depend on thresholds.

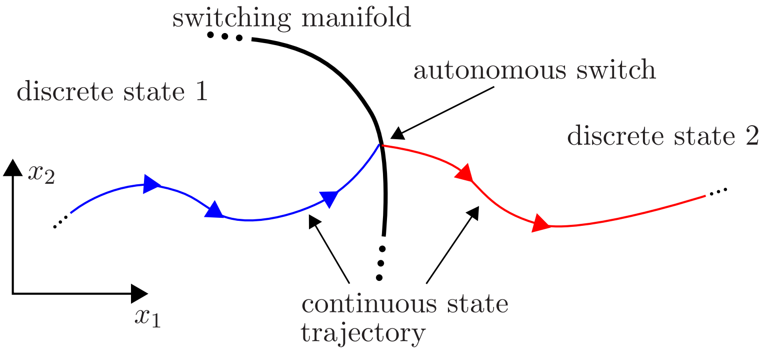


Fig. 1.1: Autonomous switching: The system switches to discrete state 2, when the continuous state trajectory in discrete state 1 crosses the switching manifold.

Controlled switching differs from autonomous switching in that the switching does not take place on a switching manifold but can be triggered externally at almost arbitrary times, see Fig. 1.2. Controlled switching models logical decisions that can be made at a desired point of time to change the system dynamics, e.g. opening or closing a valve and a switch, or changing a gear.

Consider again a system evolving in a discrete state according to some differential equations. Then an autonomous impulse resets the value of the continuous state, when the continuous state trajectory hits the manifold in Fig. 1.4. The impulse can be modeled by a function that depends on the continuous and discrete state and possibly the continuous control. After the impulse, the system is still in the same discrete state and follows the same differential equations as before. Examples for an impulsive change in the continuous state on a manifold are collisions of particles or impacts of a robot with a wall. Please note that in macro physics, impulses usually result from an abstraction of the true physical effects, as the dynamics of these effects is in general orders of magnitude faster than the

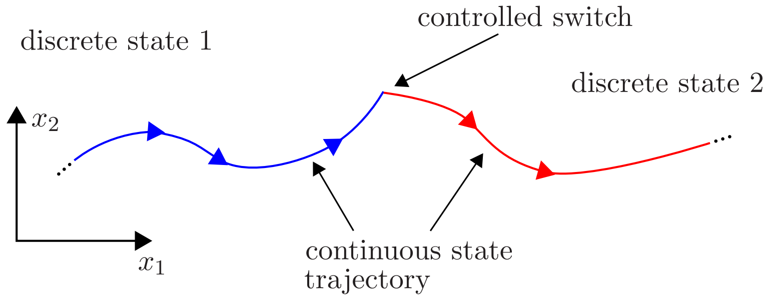


Fig. 1.2: Controlled switching: The system switches to discrete state 2, when the system receives the corresponding, externally triggered switching signal in discrete state 1.

dynamics of the remaining effects. The abstraction simplifies modeling and controlling in this case.

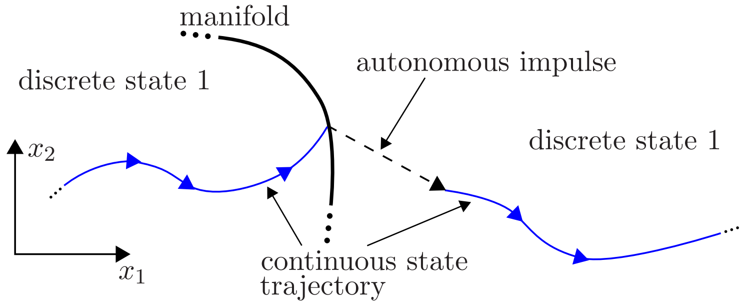


Fig. 1.3: Autonomous impulse: The continuous state is reset, when the continuous state trajectory crosses the manifold.

The difference of controlled impulses to autonomous ones is that the impulse is triggered externally when desired and is not bound to a manifold, see Fig. 1.3. Reseting a clock is an example of a controlled impulse. If controlled impulses are combined with controlled switching, then the time with open clutch during a gear shift of a car can be modeled abstractly with a jump in the time, position, and velocity of the car. As far as impulses are considered in this thesis, they are combined with autonomous and controlled switching. This still leaves the whole range of possible combinations of autonomous and controlled switching and impulses open since identity impulses and switches from a discrete state to itself are possible.

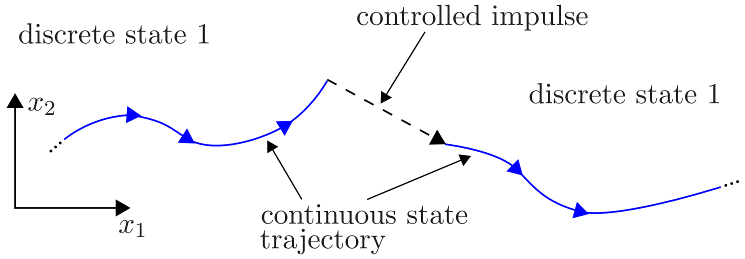


Fig. 1.4: Controlled impulse: The continuous state is reset, when the system receives the externally triggered impulse signal.

Early formulations of hybrid systems are given in [115, 182]. The existence and uniqueness theorems of executions of systems with continuous dynamics [44] have been extended to hybrid systems [157]. Attention has been paid to prove stability for hybrid systems and to design stabilizing controllers. Comprehensive and extensive overviews are provided in [62, 100]. Especially the extension of Lyapunov theory for proving the stability of continuous systems to hybrid systems in the form of multiple Lyapunov functions [28] and the development of dwell time approaches [109] are important concepts. It has also been suggested to use supervisory control to control the discrete degrees of freedom [111].

A further important topic is the optimal control of hybrid systems. One of the early versions of optimal control of hybrid systems can be found in [102]. Since then, remarkable effort has been devoted to this topic. Major theoretical contributions that provide optimality conditions for an optimal control are given among others in [29, 45, 142, 157, 165]. A vast number of algorithmic solutions to general or specialized problem classes based on various approaches have been published, e.g. [16, 134, 136, 144].

In optimal control of hybrid systems, the task is in general to find and characterize an optimal solution. This requires to determine the optimal sequence of discrete states, the optimal switching times, and the optimal continuous control. For controlled switching, the sequence of discrete states can be chosen directly with the discrete control. Furthermore, the switching times can usually be set as desired. In contrast for autonomous switching, the sequence of discrete states can only be influenced by the continuous control, which can steer the continuous state trajectory onto different switching manifolds. Similarly, the switching times can only be controlled indirectly by the continuous control.

In the following, the three main methods to solve HOCPs are presented and discussed shortly. The methods are *dynamic programming* (DP), *direct methods*, and *indirect methods*. A detailed discussion of the methods and their derivation from purely continuous optimal control problems is provided in Chap. 2. For simplicity, HOCPs are described here as minimization problems.

In DP, the goal is to find a solution of the Hamilton-Jacobi-Bellman (HJB) equation, which gives sufficient optimality conditions for an optimal solution. To find a function, called value function, satisfying the HJB partial differential equation, the continuous state and control spaces are sampled for each discrete state. The solution is iteratively approached by minimizing the sum of the costs for the next time step and the cost-to-go from the sampling point reached in the next time step to the goal. The minimization includes the selection of the optimal discrete state sequence. This procedure goes back to Bellman's principle of optimality, which proves that the remainder of an optimal trajectory is also optimal. The optimal continuous and discrete control can be calculated directly from the optimal value function. Since the value function depends on the continuous and discrete state, the optimal control is a feedback control, which is valid in the bounds of the sampled state space.

Direct methods discretize first an initially guessed continuous state and control trajectory with respect to time. The discretized state and control values at the time nodes are varied by a nonlinear programming algorithm like sequential quadratic programming (SQP), such that the cost criterion is minimized. Simultaneously with the minimization, the system dynamics and boundary conditions, which are formulated as constraints, have to be satisfied at certain points of time. Usually, discrete switches can occur at every time node possibly under consideration of switching constraints. The optimization of the discrete state sequence is often performed by organizing all switching possibilities, e.g. in a tree, and applying branch and bound or other methods to find the optimal sequence of discrete states without the need to investigate all possibilities.

In indirect methods, at first, necessary optimality conditions for an optimal solution of an HOCP are formulated and lead to a multi-point boundary value problem (MPBVP). The MPBVP consists of the differential equations of the state and its dual state, called adjoint, an algebraic equation specifying the optimal control, and boundary conditions. Second, the MPBVP is solved numerically, usually by a Newton method. The solution of the MPBVP is in general also a solution of the HOCP. In the case of autonomous switching, the optimal sequence of discrete states is found by enumerating all possibilities and solving a MPBVP for each possibility. In the case of controlled switching, an optimal discrete state sequence is determined in parallel with the solution of the MPBVP. This avoids the enumeration of all possible sequences.

The advantages and disadvantages of a solution method in comparison to the others is discussed next on a generalizing level. The statements are set up in accordance with the literature and are true for many approaches in the different categories of solution methods. Please note that individual approaches may differ in their advantages and disadvantages from the generalized statements given here. The purpose of the statements is to provide some orientation which method to choose for the solution of a certain HOCP and to motivate the focus on indirect methods in this thesis. A summary of the comparison is illustrated in Tab. 1.1.

DP offers several attractive properties. A solution found by DP is globally optimal since the HJB equation provides sufficient optimality conditions. Furthermore, DP algorithms are globally convergent in the sampled state space starting from an initialization with zeros and provide a closed-loop feedback control. The feedback control is advantageous for the stability of the control if disturbances are present and the process knowledge is imperfect. The computational complexity increases only linearly with the time horizon and in the number of discrete states with controlled and autonomous switching. The online capabilities of DP are high since basically function evaluations need to be performed online once the solution of the HJB equation is calculated. Less attractive, however, is the curse of dimensionality, which describes the exponential increase of complexity in the number of continuous states and controls. This is caused by the required sampling of the state and control spaces and currently limits the applicability of DP to systems with a continuous state and control dimension of at most 4. Due to sampling, the accuracy of solutions is generally rather low and the complexity increases relatively fast if more accurate solutions are desired. Applying DP algorithms requires a lot of knowledge. The class of HOCPs that can be solved with DP is medium size. For example, the cases, where autonomous switching manifolds intersect or certain boundary conditions are specified, are not solvable for DP. The online applicability of DP is limited if the parameters of a hybrid system are identified online, since every change in the parameters requires a new solution of the HJB equation.

Direct methods in general require less knowledge compared to DP and indirect methods about optimal control. Direct methods have a large domain of convergence compared to indirect methods, are not too difficult to initialize and often provide solutions with an acceptable accuracy. Furthermore, direct methods can be adapted flexibly to different classes of HOCPs without much effort. The growth in complexity is moderately polynomial in the number of continuous states. Disadvantages of direct methods are that solutions are only locally optimal and only open-loop controls can be determined. When increasing the accuracy, the computational complexity rises with low polynomial order. For controlled switching, the complexity increases exponentially with the time horizon since at each sample time all possible switches have to be analyzed to find the optimal switching sequence. An exception are convexification approaches that relax binary control variables, such that these variables can be optimized together with the continuous controls. In this case, the complexity is polynomial. For autonomous switching, the complexity is combinatorial in the number of switchings. The complexity increases polynomially with the number of discrete states for autonomous and controlled switching. For some special classes of HOCPs, approaches have been developed for online optimization.

Indirect methods provide solutions with high accuracy, where the accuracy is in the region of the selected accuracy of the applied integration. Indirect methods can deal with more types of HOCPs than DP but with less than direct methods. If an increasingly accurate solution is desired, the complexity grows only linearly. A longer time horizon increases the complexity moderately polynomially, which also holds for a rise of the continuous state dimension. Algorithms exist for controlled switching, such that the computational complexity only increases linearly with the number of discrete states. In contrast, the complexity grows polynomially with the number of discrete states for autonomous switching and combinatorially with the number of autonomous switches. Optimal controls found by indirect methods are locally optimal and open-loop. Compared to direct methods and DP, the domain of convergence is small. Together with the non-intuitiveness of the adjoint variables, this causes the initialization to be difficult in comparison with direct methods and DP. The small domain of convergence and the difficult initialization are two of the main drawbacks of indirect methods. Additionally, the application of indirect methods requires specialist knowledge about optimal control theory and indirect methods. The current online capabilities are limited.

1.2 Contribution

Though direct methods seem to be the most popular methods at the moment, this thesis deals with theory and algorithms for indirect methods to solve HOCPs. There are several reasons why we believe in the need for highly developed indirect methods. First, the accuracy of solutions found with indirect methods is not achievable with reasonable effort for direct methods and DP. However, some applications need highly accurate solutions. For example in space flight missions, optimal flight trajectories have to be calculated with correct payloads, otherwise the mission might fail [175]. Furthermore, the highly accurate results can be used as benchmark tests for other optimal control algorithms. Often optimality conditions used in indirect methods are also used in direct methods to check if the found solution is an optimal one.

Tab. 1.1: Qualitative comparison of dynamic programming (DP), direct methods (DM), and indirect methods (IM) for solving hybrid optimal control problems. The comparisons show average results for common algorithms based on DP, DM, or IM that are applied to the same hybrid optimal control problems.[a]

Category	**Dynamic Programming (DP)**	**Direct Methods (DM)**	**Indirect Methods (IM)**
Optimality of solution	global	local	local
Domain of convergence	global	larger than IM	smaller than DM
Ease of Initialization	higher than DM	medium	lower than DM
Accuracy of solution	lower than DM	medium	higher than DM
Class of solvable systems	smaller than IM	larger than IM	medium
Control	closed-loop	open-loop	open-loop
Increase in complexity of HOCP with increased			
- Accuracy	polynomial	polynomial	linear
- Time-horizon	linear	exponential[b]	polynomial
- #[c] of continuous states	exponential	polynomial	polynomial
- # of discrete states with			
autonomous switching	linear	polynomial	polynomial
controlled switching	linear	polynomial	linear
- # of autonomous switchings	n.c.[d]	combinatorial	combinatorial
- # of controlled switchings	n.c.	n.c.	n.c.
Online capabilities	higher than DM	medium	lower than DM
Required knowledge for application	high	smaller than DP	higher than DP

[a] Please be aware that this is a generalized comparison and that some existing approaches may not fulfill all of the statements.

[b] The exponential complexity is valid for hybrid systems with controlled switching. An exception is the convexification approach in [144], which has a polynomial increase in the complexity with the time horizon.

[c] #: number

[d] n.c.: not directly correlated

Second, indirect methods can calculate solutions with a low computational complexity. For large classes of HOCPs, the complexity of indirect methods is lower than the complexity of direct methods or DP, compare Tab. 1.1. Additionally, we realized the potential of indirect methods to reduce the complexity for HOCPs with autonomous switching.

Third, there is a large potential of indirect methods for online solutions. The reason is that a lot of information about the structure of a solution is provided by optimality conditions. The information can be exploited for online solution methods, e.g. by linearization around a given solution or by calculating certain steps analytically.

Further, we believe that several weaknesses of indirect methods can be solved to a large extend. For example, indirect methods can be implemented such that a user does not need much knowledge about indirect methods and optimal control. Partly, theoretic and algorithmic contributions presented in this dissertation will be needed. Required derivatives can be calculated by automatic differentiation. It is sufficient when a user supplies the HOCP description and initial guesses for the state or control trajectory similar to the requirements in direct methods.

The goal of this thesis is to enlarge the class of HOCPs that can be solved by indirect methods and to reduce limitations and weaknesses that prevent the use of indirect methods. This is achieved on the one hand by an extension of optimality conditions to a larger class of HOCPs. On the other hand, methods and algorithms are developed that simplify the initialization, increase the domain of convergence, reduce the complexity of HOCPs with autonomous switching, and provide online solution capabilities. This is to overcome the dominant weaknesses shown in Tab. 1.1.

In the following, the contribution of the thesis is discussed. A detailed discussion of the contribution is provided at the beginning of each main chapter.

1. **Optimality Conditions:** Optimality conditions for HOCPs are provided by the so-called *hybrid minimum principle* (HMP). Previous versions of the HMP are only valid for a certain set of HOCPs. Here, versions of the HMP are introduced for an extended set of HOCPs:

   (i) One extension provides optimality conditions for HOCPs with autonomous and controlled switching with simultaneous impulses of the continuous state. Previous versions of the HMP are not able to deal with controlled switching and simultaneous resets of the continuous state. Controlled switching with resets can be used to model some physical processes abstractly. An example is gear shifting, where the time with open clutch is modeled by a reset of the continuous state.

   (ii) A further extension considers HOCPs, where functions depend explicitly on time. Thus, explicitly time-varying dynamics like the one of a sanding vehicle can be considered in HOCPs with this extension.

   (iii) Another version of the HMP gives optimality conditions for HOCPs, where the continuous state space is partitioned and each partition is assigned to a discrete state. An autonomous switch occurs when the continuous state trajectory passes the boundary between two neighboring partitions. In this case, an optimal state trajectory might pass through an intersection of two or more switching manifolds, which previous versions of the HMP are not able to handle in contrast to the novel version. The consideration of an optimal trajectory passing an intersection of switching manifolds is important for two reasons: First, there exist cases, where an optimal trajectory naturally passes an intersection of switching manifolds. An example is a four-legged pole climbing robot, where the discrete state changes depending on the contact situation of each leg. Second, the extended HMP allows us to develop an optimal control algorithm that optimizes the discrete state sequence based on gradient information. This reduces the complexity to find an optimal control by a factor of combinatorial complexity in the number of autonomous switches, see also Item 4.

2. **Initialization:** Two approaches for a simplified initialization of indirect methods are derived in the thesis.

   (i) The first approach is to initialize an indirect method with the solution of a direct method, which is in general more straightforward to initialize. Here, two concepts are introduced. In the first concept, the direct method solves the complete HOCP at once. Then the relevant information is extracted from the solution and used to initialize the indirect method, which also solves the HOCP entirely at once. This concept is basically an extension of an existing initialization concept for purely continuous systems. In the second concept, the HOCP is decomposed into subproblems with fixed discrete state. The subproblems are connected by a choice of switching points. Each subproblem is solved separately by the direct method and the result is used to initialize an indirect method for each subproblem. On a higher level, the switching points are optimized, such that the overall optimal solution is obtained finally. The second concept is a more robust initialization scheme, while the first concept converges faster. It is even possible to apply an indirect method that solves the entire HOCP at once after the second concept has been executed. So a robust initialization with a high convergence rate is achieved.

   (ii) The second approach consists in replacing the indirect methods that are difficult to initialize by an indirect gradient method that is initialized straightforwardly and has a global domain of convergence. More details of the algorithm are given in Item 3.

3. **Domain of Convergence:** The domain of convergence of indirect methods based on multiple shooting or collocation is small due to local convergence properties and numerical instabilities, such that the practical application of these approaches is limited. Here, an indirect method based on a gradient method is proposed. The novel method is globally convergent to a locally optimal solution and the numerical stability is increased. More specifically the introduced method belongs to the class of min-H algorithms that have only been formulated for purely continuous systems up to now. In contrast to previous min-H approaches, the novel min-H approach is extended to HOCPs, leads to a convergence rate that is at least quadratic in a sufficiently close neighborhood around the optimal solution, is globally convergent, and the convergence is proven. Additionally, a similar algorithm based on a combined

first- and second-order gradient method is introduced and discussed as an alternative to the min-H algorithm.

4. **Complexity for Autonomous Switching:** If the optimal discrete state sequence is unknown in HOCPs with autonomous switching, then the computational complexity of previous solution approaches is combinatoric. The thesis introduces two concepts for reducing the computational complexity for HOCPs with autonomous switching.

   (i) In the first concept, a branch and bound method is developed for indirect methods. Therefore, hybrid systems with controlled switching are presented as relaxations of hybrid systems with autonomous switching. Tightening the relaxations step-by-step, branch and bound principles are applied to exclude non-optimal discrete state sequences from the further optimization. Depending on how tight the relaxations are, the computational complexity can be reduced considerably.

   (ii) The second concept is specialized to HOCPs with partitioned continuous state space. An algorithm is derived that optimizes the sequence of discrete states simultaneously with the continuous optimization. The discrete state sequence is varied based on a gradient algorithm with sufficient descent properties. The descent direction is obtained from the novel HMP presented in Item 1 for hybrid systems with partitioned state space. The global convergence of the algorithm to a locally optimal solution is proven. Previous approaches iterate over all possible discrete state sequences, which leads to a combinatorial complexity in the number of autonomous switches. In contrast, the order of the computational complexity of the novel algorithm is lower by a factor of combinatorial complexity.

5. **Online Methods:** The HMP and indirect methods include information about the structure of the optimal control problem. This information is exploited in two approaches with high potential for the online solution of HOCPs.

   (i) One approach is an MPC algorithm based on neighboring extremals. Neighboring extremals are optimal solutions in the neighborhood of another optimal solution and the two optimal solutions vary in their initial conditions. In the proposed MPC algorithm, neighboring extremals are calculated with the knowledge about the variation of initial conditions with receding time and a linearization around the optimal solution of the previous time step. The calculation is performed by a gradient method of second order, which is extended to the solution of HOCPs. Here, the use of a gradient method in an MPC framework is novel. Additionally, the method also converges to an optimal solution if it is linearized around a non-optimal solution, which is in the neighborhood of an optimal solution. Between two sampling points of the MPC algorithm, a local feedback-feedforward control is applied to the system, which further converges to the optimal solution. The convergence of the MPC algorithm to an optimal solution is proven. No proof of convergence exists if the discrete state sequence changes during the execution of the MPC algorithm. However, the algorithm is constructed such that it usually can handle changes in the discrete state sequence at the beginning and at the end of the prediction horizon. Those changes occur frequently in MPC with receding horizon. If the algorithm does not converge for such a case, a new initial solution for the MPC algorithm with updated discrete state sequence has to be determined by other optimal control algorithms. The MPC algorithm has a high potential for online calculations due to its structure exploiting properties. Depending on the complexity of the HOCP and the computational power, real-time computations are achievable.

(ii) In a second approach, an online method is introduced for input-to-state and input-to-output feedback linearizable systems. With such an exact linearization, a linear-quadratic HOCP is obtained. Non-hybrid, linear-quadratic optimal control subproblems are solved offline with stabilizing and anti-stabilizing solutions from the literature. Here, the solutions are interpreted as optimal control primitives that depend on some parameters. The main contribution is that the parameters are determined online by solving a system of algebraic equations, such that the overall hybrid optimal control is obtained. The main drawback of the method is that costs for the control and possibly for the state of the original, nonlinear HOCP cannot be considered.

1.3 Outline of the Thesis

After the introduction, hybrid systems are defined and a generalized class of HOCPs is formulated in Chap. 2. Afterwards, the three main solution methods DP, direct methods, and indirect methods are presented and a detailed discussion about their strengths and weaknesses is provided.

Next, in Chap. 3, the extensions of the HMP are introduced. At first, an extended version is shown that considers autonomous and controlled switching with simultaneous resets of the continuous states and switching costs. Further, the version of the HMP is derived for functions that are explicitly depending on time. The proof of the novel optimality conditions is given and a numerical example with gear shifting is presented. This part follows the publication [211]. Second, an HMP is described for HOCPs with partitioned continuous state space. Again the proof is shown and this part follows [207, 208].

The successive Chap. 4 deals with algorithms for HOCPs with a fixed discrete state sequence. First, two initialization concepts [210] are presented, where a direct method initializes an indirect method. Both concepts are compared qualitatively and with a numerical example. For a more straightforward initialization and a global domain of convergence, two versions of a gradient method are introduced afterwards. The first method is a min-H approach and the second a combined first- and second-order gradient method. A numerical example using the min-H algorithm is shown. The min-H algorithm is described in the submission [212].

Chap. 5 presents methods for reducing the computational complexity to find an optimal solution, when an optimal sequence of discrete states is not known a priori. At first, a branch and bound scheme [215] for a large class of HOCPs with autonomous switching is discussed and a numerical example illustrates the effectiveness. Second, an algorithm is introduced that varies the discrete state sequence based on a gradient method [207, 214]. The algorithm is developed for HOCPs with partitioned continuous state space. Several small numerical examples are shown. Further, the algorithm is applied with some adaptations to a cooperative transportation scenario with multiple robots, which is only partitioned in a subspace of the whole continuous state space [205].

In Chap. 6, two methods for the improved online solvability of HOCPs are presented. First, an MPC algorithm based on neighboring extremals is described and tested in simulations on an autonomous driving scenario [213]. Afterwards, an online solution based on optimal control primitives is discussed. The method is applied to a robotic manipulation task with obstacle avoidance in simulations and is published in [219].

At last, in Chap. 7, a conclusion about the achievements of the thesis and an outlook for future work is given.

2 The Hybrid Optimal Control Problem

All improvements for indirect methods that are introduced in this thesis are based on hybrid systems and solve optimal control problems for these hybrid systems. In this chapter, a general hybrid system is presented first, which is used in variations throughout the thesis. Afterwards, a general hybrid optimal control problem (HOCP) is formulated and variations are introduced, that are used in the different chapters. Finally, an overview over the state of the art of the three main solution methods is provided.

2.1 Hybrid System Formulation

Hereafter, hybrid systems are presented in a formulation that is related to the formulation used in [157]. The formulation is especially suitable for the proofs of the extended versions of the hybrid minimum principle (HMP) introduced in Chap. 3. Hybrid systems are characterized by a number of elements and a general formulation is given in the definition below as a subclass of the hybrid systems described in [29, 62].

Definition 2.1: A hybrid system is a 9-tuple

$$\mathbb{H} := \{\mathcal{Q}, \mathcal{X}, U, \mathcal{F}, \mathcal{M}, \Lambda, \Omega, \Gamma, \mathcal{H}\}\,, \tag{2.1}$$

where $\mathcal{Q}$ denotes a set of discrete states, $\mathcal{X}$ a collection of continuous state spaces, U a collection of continuous control spaces, $\mathcal{F}$ a collection of vector fields for the continuous dynamics, $\mathcal{M}$ a collection of switching manifolds, Λ a collection of reset maps, Ω a collection of discrete control spaces, Γ a discrete transition map, and $\mathcal{H}$ a collection of constraint functions. □

Hybrid systems combine discrete and continuous states, which evolve according to discrete and continuous dynamics. The continuous dynamics is usually formed by differential or difference equations. The discrete dynamics is given by a discrete structure like a discrete automaton. A change of the discrete state q can be autonomous or controlled. An autonomous switch occurs whenever the continuous state trajectory χ crosses a switching manifold. In contrast, a controlled switch is triggered externally by a switching time t_j and a discrete control ω. Both controls can be chosen freely in certain bounds and specify when the switch shall happen and what the next discrete state will be, respectively. Discrete states q are denoted with integer values taken from the set $\mathcal{Q}$. In the following, assumptions are associated with the individual elements.

Assumption 2.2: $\mathcal{Q} = \{1, 2, ..., N_q\}$ is the set of N_q discrete states $q \in \mathcal{Q}$. □

Continuous states x are defined for each discrete state q in a continuous state space $\mathcal{X}_q$.

Assumption 2.3: $\mathcal{X} = \{\mathcal{X}_q\}_{q\in\mathcal{Q}}$: collection of state spaces $\mathcal{X}_q \subseteq \mathbb{R}^{n_x}$ assigned to every discrete state q, where the dimension of the continuous state x is n_x. □

The continuous dynamics of the hybrid system can be influenced by a possibly bounded, continuous control u and the associated, measurable control function υ.

Assumption 2.4: $U = \{U_q\}_{q\in\mathcal{Q}}$: collection of compact sets $U_q \subset \mathbb{R}^{n_u}$ of admissible continuous control values u with $\dim(u) = n_u$. $\mathcal{U} = \{\mathcal{U}_q\}_{q\in\mathcal{Q}}$ is the set of admissible, Lebesgue-measurable control functions $\upsilon : [t_0, t_e] \to U$ with initial and final time t_0 and t_e. □

The continuous dynamics defines how the continuous state evolves with time t. It satisfies a Lipschitz condition and is continuously differentiable, which includes boundedness with respect to its arguments. The conditions guarantee that solutions of the according differential equations exist and the proofs in this thesis are valid.

Assumption 2.5: $\mathcal{F} = \{f_q\}_{q\in\mathcal{Q}}$: collection of vector fields $f_q : \mathcal{X}_q \times U_q \times \mathbb{R} \to \mathbb{R}^{n_x}$ defined for each $q \in \mathcal{Q}$. The vector fields are at least once continuously differentiable with respect to the continuous state x and the continuous control u and continuous with respect to time t. They fulfill a uniform Lipschitz condition, i.e. $\exists L < \infty$ such that $\|f_q(x_1, u_q, t) - f_q(x_2, u_q, t)\|_2 \leq L\|x_1 - x_2\|_2$, for any combination of $x_1, x_2 \in \mathcal{X}_q$, $u_q \in U_q$, $t \in \mathbb{R}$. □

Switching manifolds are essential for defining autonomous switchings. In hybrid systems with partitioned state space, switching manifolds also serve as the boundary between two neighboring, continuous state spaces.

Assumption 2.6: $\mathcal{M} = \{M_{i,k}\}_{i,k\in\mathcal{Q},i\neq k}$: collection of time-dependent switching manifolds $M_{i,k}(t)$. An autonomous transition from the discrete state i to k occurs at time t_j for $x(t_j)$ on the manifold $M_{i,k}(t_j)$. A switching manifold $M_{i,k}(t)$ of $\mathbb{R}^{n_x}$ has codimension 1, is at least once continuously differentiable with respect to the continuous state x, i.e. C^1, and continuous with respect to time t. Switching manifolds are locally expressed here by $M_{i,k}(t) := \{x | m_{i,k}(x,t) = 0,\ x \in \mathcal{X}_i,\ i \in \mathcal{Q}\}$ with the functions $m_{i,k}(x,t) : \mathcal{X}_i \times \mathbb{R} \to \mathbb{R}$.[1] (Note that with fixed t the Implicit Function Theorem states for $m_{i,k} \in C^\iota(\mathbb{R}^{n_x})$ with $\iota \in \mathbb{N} \cup \{\infty\}$, that $(y, \bar{m}(y))$ with $y \in \mathbb{R}^{n_x-1}$ locally gives the zero level set of $m_{i,k}(x)$, i.e. $x \in m_{i,k}^{-1}(0)$ with $x \in \mathbb{R}^{n_x}$, and $m_{i,k}(x)$ and $\bar{m}$ have the same degree of regularity [96].) For fixed time t, it is assumed that:

(i) Switching manifolds do not intersect, i.e. $M_{i,k}(t) \cap M_{i,l}(t) = \emptyset$ with $k \neq l$ for $i, k, l \in \mathcal{Q}$. (Note that this assumption can be relaxed, e.g. it can be assumed that switching manifolds intersect and that considered state trajectories do not pass through an intersection of switching manifolds.)

(ii) Let two points $\bar{x} \in \mathrm{Int}(M_{i,k}(t))$ and $x \in \mathrm{Int}(\mathcal{X}_i)$ be such that $\|x - \bar{x}\| < \|x - \hat{x}\|$ holds for all $\hat{x} \in \mathrm{Int}(M_{i,l}(t))$ with $k \neq l$ for $i, k, l \in \mathcal{Q}$. Then manifolds are oriented such that $m_{i,k}(x,t) < 0$. □

[1] A local description of the manifold $M_{i,k}(t)$ is sufficient for the proof of the optimality conditions in Chap. 3. Describing the switching manifolds by a set of functions that satisfies the C^1 differentiability property of the switching manifold is less complex than finding one function for the entire switching manifold, especially if constrained switching manifolds are considered.

In Sec. 3.2, the continuous state x may jump each time an autonomous or controlled switch occurs as specified with the collection of reset functions.

Assumption 2.7: $\Lambda = \{\varphi_{i,k}\}_{i,k\in\mathcal{Q},i\neq k}$: collection of time-dependent reset functions $\varphi_{i,k} \in C^1(\mathbb{R}^{n_x} \times \mathbb{R}, \mathbb{R}^{n_x})$ for the continuous state x, which are associated with a controlled or autonomous switching from discrete state i to k for $i, k \in \mathcal{Q}$. In the case of controlled (but not autonomous) switching, it is assumed that $x = \varphi_{k,i}(\varphi_{i,k}(x,t),t)$. □

At an autonomous or controlled switching, the successive discrete state is chosen by the discrete control ω.

Assumption 2.8: $\Omega : \{\Omega_q\}_{q\in\mathcal{Q}}$: collection of discrete sets $\Omega_q \subseteq \mathbb{N}$ of admissible discrete controls ω_q in discrete state $q \in \mathcal{Q}$. When $\omega_q \in \Omega_q$, such that $\Gamma(q, x, \omega_q) = i$ holds, a controlled or autonomous switching from discrete state $q \in \mathcal{Q}$ to discrete state $i \in \mathcal{Q}$ is triggered. In the case of autonomous switching, when the continuous state trajectory x hits $M_{q,i}(t_j)$ at time t_j, the discrete control ω_q is forced to be such that $\Gamma(q, x, \omega_q) = i$ is fulfilled. A controlled switch with $\omega_q \in \Omega_q$ can be executed whenever and wherever desired independently of any switching manifold. It is assumed that if $\Gamma(q, x, \omega_q) = i$ with $\omega_q \in \Omega_q$ holds, $\Gamma(i, x, \omega_i) = q$ with $\omega_i \in \Omega_i$ is conversely true. □

Since the discrete control is completely specified at autonomous switchings, it is omitted in this thesis if hybrid systems with only autonomous switching are considered. The discrete structure that determines which discrete states can be reached from each discrete state is denoted with Γ.

Assumption 2.9: $\Gamma : \mathcal{Q} \times \mathcal{X} \times \Omega \to \mathcal{Q}$: discrete transition map. □

In Sec. 6.2, hybrid systems are extended to include state and mixed state-control constraints. There, it is assumed in each discrete state q that boundaries of the continuous state space $\mathcal{X}_q$ and control space U_q can be expressed by at least piecewise twice continuously differentiable functions.

Assumption 2.10: $\mathcal{H} = \{h_q\}_{q\in\mathcal{Q}}$: collection of mixed state-control $h_q^{\mathrm{sc}}(x, u, t)$, pure state $h_q^{\mathrm{s}}(x, t)$, and pure control $h_q^{\mathrm{c}}(u, t)$ equality and inequality constraint vector functions, where $h_q \in C^1(\mathcal{X}_q \times U_q \times \mathbb{R}, \mathbb{R}^{n_{h_q}})$. The relative degree $r \in \mathbb{N}$ of a pure state constraint $h_q^{\mathrm{s}}(x, t)$ denotes the necessary number of time derivatives until the control u appears for the first time, i.e. $h_q^{(r),\mathrm{s}}(x, u, t)$. It is assumed that $h_q^{\mathrm{s}}(x, t)$ and $f_q(x, u, t)$ are sufficiently smooth, such that $h_q^{(r),\mathrm{s}}(x, u, t)$ exists. The equality and the active inequality constraints are summarized in the vector $h_q^{\mathrm{a}}(t) = 0$ and the inactive inequality constraints fulfill $h_q^{\mathrm{ina}}(t) < 0$, where 'a' denotes active and 'ina' inactive. □

Assumption 2.11: The switching times t_j fulfill $t_0 < t_1 < ... < t_e < \infty$, and $\forall x(t_j) \in M_{i,k}(t_j)$, $i, k \in \mathcal{Q}$, the vector fields $f_i(x, u_i, t_j)$ and $f_k(x, u_k, t_j)$ are non-vanishing at and transversal to $M_{i,k}(t_j)$ for the applied $u_i \in U_i$ and $u_k \in U_k$. The number N of autonomous and controlled switchings of an execution of a hybrid system satisfies $N + 2 \leq \bar{N} < \infty$, where $\bar{N}$ is the maximally allowed number of switchings. □

This also implies that no accumulation points of switching (Zeno points [194]) or sliding motions occur.

Assumption 2.12: At time t_0 with given initial conditions $(q(t_0), x(t_0)) = (q_0, x_0) \in \mathcal{Q} \times \mathcal{X}$, it is assumed that $x(t_0) \notin M_{q_0,i}(t_0)$ for all $i \in \mathcal{Q}$. □

Note that these assumptions are important for the existence of a unique execution of the hybrid system and are a prerequisite for the existence of an optimal control.[2]

Definition 2.13: A hybrid control is a triple (υ, τ, ω), where $\upsilon = (\upsilon_{q_0}, ..., \upsilon_{q_N})$ is a sequence of control trajectories $\upsilon_{q_j} : [t_j, t_{j+1}) \rightarrow U_{q_j}$. The sequence $\tau = (t_0, ..., t_{N+1} = t_e)$ of initial, switching, and final times is strictly increasing and N is the number of switchings. The sequence of discrete controls is specified by $\omega = (\omega_{q_0}, ..., \omega_{q_{N-1}})$. □

Note that in the thesis, the initial and final times t_0 and t_e are in general fixed. For HOCPs with autonomous switching and without controlled switching, only the control υ is considered. The reason is that the switching times t_j and the discrete controls $\omega_{q_{j-1}}$ for $j \in \{1, ..., N\}$ are determined by the control trajectory υ and the associated evolution of the continuous state trajectory.

Definition 2.14: An execution of a hybrid system satisfying the Assumptions 2.2 - 2.12 is given by $\sigma = (\tau, \varrho, \chi, \upsilon, \omega)$, where $\tau = (t_0, ..., t_e)$ is a strictly increasing sequence of times and $\varrho = (q_0, ..., q_N)$ denotes a sequence of discrete states. The term $\chi = (\chi_{q_0}, ..., \chi_{q_N})$ is a sequence of absolutely (left-) continuous state trajectories $\chi_{q_j} = \chi_{q_j}(\upsilon, t_0) : [t_j, t_{j+1}) \rightarrow \mathcal{X}_{q_j}$ evolving according to:

$$\dot{x}_{q_j}(t) = f_{q_j}(x_{q_j}(t), u_{q_j}(t), t) \tag{2.2}$$

for a.e. $t \in [t_j, t_{j+1})$, $j \in \{0, ..., N\}$, where $x_{q_0}(t_0) = x_0$ and $x_{q_j}(t_j) = \varphi_{q_{j-1},q_j}(x_{q_{j-1}}(t_j^-), t_j)$ with $t_j^- := \lim_{t \rightarrow t_j, t < t_j} t$. An autonomous transition from discrete state $q_{j-1} \in \mathcal{Q}$ to discrete state $q_j \in \mathcal{Q}$ occurs at the autonomous switching time t_j with $\Gamma(q_{j-1}, x_{q_{j-1}}(t_j^-), \omega_{q_{j-1}}) = q_j$ for $\omega_{q_{j-1}} \in \Omega_{q_{j-1}}$, if

$$x_{q_{j-1}}(t_j) \in M_{q_{j-1},q_j}(t_j)\,. \tag{2.3}$$

A controlled transition from discrete state q_{j-1} to q_j occurs at the controlled switching time t_j, if $\omega_{q_{j-1}} \in \Omega_{q_{j-1}}$ is chosen, such that

$$\Gamma(q_{j-1}, x_{q_{j-1}}(t_j^-), \omega_{q_{j-1}}) = q_j\,. \tag{2.4}$$

The left continuity means that after a switch the dynamics $f_{q_j}(x_{q_j}(t), u_{q_j}(t), t)$ of the next discrete state q_j is applied. The continuous control $\upsilon = (\upsilon_{q_0}, ..., \upsilon_{q_N})$ is a sequence of control trajectories $\upsilon_{q_j} : [t_j, t_{j+1}) \rightarrow U_{q_j}$ and ω denotes a sequence of discrete controls $(\omega_{q_0}, ..., \omega_{q_{N-1}})$. □

Considering the standing assumptions, a unique execution of the hybrid system is guaranteed until

[2] It is possible to consider hybrid system executions that contain Zeno points [3, 78], but as Zeno points are not in the scope of this thesis, they are excluded by assumption.

(i) the final time t_e,

(ii) a tangential movement along a switching manifold,

(iii) a Zeno point (infinite number of switchings in finite time), and

(iv) a point of time, where a state constraint $h_q^{\mathrm{s}}(x(t), t) \leq 0$ is violated,

is reached, compare Theorem 1 in [157].

Remark 2.15: The Assumptions 2.2 - 2.12 are close to the minimum requirements to ensure a unique execution of the hybrid system. Aspects of the standing assumptions that are a bit stronger than the minimum requirements are the C^1 properties of functions, the differentiability of state constraints up to the relative degree r, and the exclusion of Zeno phenomena. In general, physical systems satisfy the increased differentiability assumptions, such that these assumptions are not limiting the practical application. As mentioned earlier, Zeno phenomena are excluded for convenience and can be considered with specialized approaches [3, 78]. The assumptions for controlled switching that it is possible to switch back and forth between discrete states and that the corresponding reset maps are the inverse of each other is more limiting in practice than the other assumptions. If these two assumptions are not fulfilled, the optimality condition specifying the optimal discrete state sequence cannot be shown, however, all other optimality conditions derived in Sec. 3.2 remain unchanged. □

2.2 Hybrid Optimal Control Problem Formulation

2.2.1 General Hybrid Optimal Control Problem

Based on the hybrid system definition in the section before, an optimal control problem is formulated here. The goal is to control the hybrid system with the continuous and discrete controls, and the switching times, such that desired characteristics are met. In this thesis, all optimal control problems are considered to be minimization problems. Maximization problems can be reformulated as minimization problems by a change of the sign of cost functions. The considered functions to be minimized are the terminal cost function, the integral over the running cost function, and the sum of the switching costs. The terminal cost function penalizes deviations of the continuous state at the final time and possibly the final time itself from desired values.

Assumption 2.16: The terminal cost function $g : \mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}$ is at least once continuously differentiable, i.e. $g \in C^1$. □

The possibly time-dependent running cost functions penalize deviations of the continuous state and control from desired values.

Assumption 2.17: The running cost functions $\phi_q : \mathcal{X}_q \times U_q \times \mathbb{R} \to \mathbb{R}^{n_x}$ for $q \in \mathcal{Q}$ are at least once continuously differentiable with respect to the continuous state x and control u_q, i.e. $\phi_q \in C^1(\mathcal{X}_q \times U_q, \mathbb{R}_{n_x})$, and continuous with respect to time t. □

The switching costs penalize switching autonomously and controlled from one discrete state to the next.

Assumption 2.18: The switching cost functions $c_{q,i} : \mathcal{X}_q \times \mathbb{R} \to \mathbb{R}$ for $q, i \in \mathcal{Q}$ and $q \neq i$ are at least once continuously differentiable, i.e. $c_{q,i} \in C^1$. □

In Chap. 4 and 6, the optimal, continuous trajectory must pass some specified interior point constraints and must finish at a specified terminal point constraint.

Assumption 2.19: $\Psi = \{\psi_j\}_{j\in\{0,\ldots,N\}}$: collection of interior and terminal point constraints $\psi_j(x_{q_j}(t_{j+1}^-), t_{j+1}) = 0$ with dimension $0 \leq n_{\psi_j} \leq n_x$ and $\psi_j \in C^1(\mathcal{X}_{q_j} \times \mathbb{R}, \mathbb{R}^{n_{\psi_j}})$.□

The optimal control for the introduced class of hybrid systems under consideration of the different costs consists of an optimal continuous control, an optimal discrete control, and optimal switching times.

Definition 2.20: An optimal solution $\sigma^* = (\tau^*, \varrho^*, \chi^*, \upsilon^*, \omega^*)$ minimizes the cost functional

$$\begin{aligned} J = {} & g(x_{q_e}(t_e), t_e) + \sum_{j=1}^{N} c_{q_{j-1},q_j}(x_{q_{j-1}}(t_j^-), t_j) \\ & + \sum_{j=0}^{N} \int_{t_j}^{t_{j+1}} \phi_{q_j}(x_{q_j}(t), u_{q_j}(t), t)\, dt \end{aligned} \tag{2.5}$$

while satisfying initial conditions x_0 and q_0, the hybrid system dynamics $\mathbb{H}$, the Assumptions 2.2 - 2.12, and 2.16 - 2.18, and possibly defined interior and terminal constraints $\psi_{j-1}(x_{q_{j-1}}(t_j^-), t_j) = 0$ for $j = \{1, ..., N+1\}$ as in Assumption 2.19. □

Remark 2.21: The Assumptions 2.16 - 2.19 are close to the minimum requirements for the existence of an optimal solution.[3] Stronger than the minimum requirements is again the assumption about the C^1 differentiability of functions. In general, physically, socially, and economically motivated functions fulfill the differentiability assumptions and there is usually some freedom in the choice of the cost functions, such that it is possible to meet the differentiability assumptions. □

Remark 2.22: By the HOCP including the hybrid system, the cost functions, and possibly constraints, it is in general determined if a (unique) optimal solution is stable or not. In this thesis, it is always assumed that optimal solutions lead to stable behavior. If this assumption was not fulfilled, it would usually be difficult to compute an algorithmic solution in practice. Extensive overviews of tools for proving stability of hybrid systems and designing stabilizing controllers are given in [28, 62, 100, 109]. □

The HOCP above is stated in Bolza form. For the proof of novel versions of the hybrid minimum principle in Chap. 3 and the convergence proofs of the algorithms in Chap. 4 and

[3] Note that the existence of solutions is not yet guaranteed by the assumptions since the constraints, that are specified by the HOCP defined in Def. 2.20, might not be reachable, for example.

5, it is advantageous to reformulate the HOCP into a Mayer formulation with only a terminal cost [19]. For this purpose, the continuous state space is extended by the accumulated running costs $x^0(t)$ with $x^0(t_0) = 0$ leading to the extended differential equations:

$$\dot{\tilde{x}}_{q_j}(t) := \begin{pmatrix} \dot{x}^0_{q_j}(t) \\ \dot{x}_{q_j}(t) \end{pmatrix} = \begin{pmatrix} \phi_{q_j}(x_{q_j}(t), u_{q_j}(t), t) \\ f_{q_j}(x_{q_j}(t), u_{q_j}(t), t) \end{pmatrix} =: \tilde{f}_{q_j}(\tilde{x}_{q_j}(t), u_{q_j}(t), t) \tag{2.6}$$

for a.e. $t \in [t_j, t_{j+1}]$ and $j \in \{0, ..., N\}$. The terminal cost function is modified to:

$$\tilde{g}(\tilde{x}_{q_e}(t_e), t_e) := x^0_{q_e}(t_e) + g(x_{q_e}(t_e), t_e) \tag{2.7}$$

and the reset function is:

$$\tilde{\varphi}_{q_{j-1},q_j}(\tilde{x}_{q_{j-1}}(t_j^-), t_j) := \begin{pmatrix} c_{q_{j-1},q_j}(x_{q_{j-1}}(t_j^-), t_j) \\ \varphi_{q_{j-1},q_j}(x_{q_{j-1}}(t_j^-), t_j) \end{pmatrix}. \tag{2.8}$$

For convenience, $\tilde{x}$, $\tilde{f}$, $\tilde{g}$, and $\tilde{\varphi}$ are written again as x, f, g, and φ in the mentioned chapters. In the following, it is made use of so-called Hamiltonians, which are introduced for a concise notation.

Definition 2.23: For HOCPs without state constraints, the Hamiltonian

$$H_{q_j}(x(t), \lambda(t), u_{q_j}(t), t) = \lambda(t)^T f_{q_j}(x(t), u_{q_j}(t), t) \tag{2.9}$$

is defined for a.e. $t \in [t_j, t_{j+1})$ with $j \in \{0, ..., N\}$, $q \in \mathcal{Q}$, and with the adjoint variable $\lambda(t) : \mathbb{R} \to \mathbb{R}^{n_x}$. □

The adjoint variable $\lambda(t)$ is a term that results from the optimization and decodes the sensitivity of the optimal solution with respect to variations in the state differential equations. In the case of HOCPs with state constraints, the Hamiltonian is extended as follows:

Definition 2.24: For HOCPs with state constraints, the Hamiltonian

$$H_{q_j}(x(t), \lambda(t), u_{q_j}(t), t) = \lambda^T(t) f_{q_j}(x(t), u_{q_j}(t), t) + \mu^T(t) h^{(r),\mathrm{a}}_{q_j}(x(t), u_{q_j}(t), t) \tag{2.10}$$

is defined for a.e. $t \in [t_j, t_{j+1})$ with $j \in \{0, ..., N\}$ and $q \in \mathcal{Q}$. The term $\lambda(t) : \mathbb{R} \to \mathbb{R}^{n_x}$ is the adjoint variable and $\mu(t) : \mathbb{R} \to \mathbb{R}^{n_{h^{\mathrm{a}}_{q_j}}}$ is a Lagrange multiplier. □

The Lagrange multiplier $\mu(t)$ gives the sensitivity of the optimal solution with respect to the constraints.

The HOCP as given in Definition 2.20 includes different sequences of discrete states and the task is to find an optimal one. Searching for a discrete state sequence may have a high computational complexity since numerous possibilities of different sequences exist. The discrete complexity of an HOCP is specified as follows:

Definition 2.25: Two cases of discrete complexity are considered here:

- **Exponential Complexity** is given if the complexity increases exponentially with the number of switchings N and if the number of possible subsequent discrete states from every discrete state is always $N_q - 1$. The number of possible discrete state sequences is determined by $(N_q - 1)^N$ when starting from discrete state q_0.

- **Combinatorial Complexity** is defined as $(N_{cc}-1)^N$, where the number N_{cc} satisfies $1 \leq N_{cc} < N_q$ and corresponds nearly to the average number of subsequent discrete states. □

An example for exponential complexity is controlled switching if it is allowed to switch from each discrete state to every other discrete state at all times. The class of combinatorial complexity is introduced here to distinguish the case from the exponential complexity. Combinatorial complexity describes the case that it is only possible to switch from each discrete state to a subset of the other defined discrete states. Nevertheless, combinatorial complexity also shows an exponential growth in the number of switchings. Hybrid systems with autonomous switching in general show a combinatorial complexity, especially when a partitioned continuous state space is considered where each partition only has a few neighboring discrete states.

In the following chapters of the thesis, various subclasses of the general hybrid system and the HOCP defined above are analyzed. The different subclasses are defined below:

2.2.2 Hybrid Optimal Control Problem with Resets

This subclass of HOCPs includes continuous and discrete controls, autonomous and controlled switching, resets of the continuous state when switching, explicitly time-dependent functions, and switching, running, and terminal costs.

Definition 2.26 (Hybrid System with Resets): A hybrid system is an 8-tuple

$$\mathbb{H} := \{\mathcal{Q}, \mathcal{X}, \Gamma, U, \Omega, \mathcal{F}, \mathcal{M}, \Lambda\}, \tag{2.11}$$

where $\mathcal{Q}$ is a set of discrete states, Γ a discrete transition map, and $\mathcal{X}$, U, Ω, $\mathcal{F}$, $\mathcal{M}$, and Λ are collections of continuous state, continuous control, and discrete control spaces, vector fields, switching manifolds, and reset maps. □

Assumption 2.27:

1. $\mathcal{Q}$: set of discrete states as in Assumption 2.2.
2. $\mathcal{X}$: collection of continuous state spaces as in Assumption 2.3.
3. Γ: discrete transition map as in Assumption 2.9.
4. U: collection of continuous control spaces as in Assumption 2.4.
5. Ω: collection of discrete control sets as in Assumption 2.8.
6. $\mathcal{F}$: collection of vector fields as in Assumption 2.5.
7. $\mathcal{M}$: collection of switching manifolds as in Assumption 2.6.
8. Λ: collection of reset maps as in Assumption 2.7. □

Definition 2.28 (HOCP with Resets): An optimal solution $\sigma^* = (\tau^*, \varrho^*, \chi^*, \upsilon^*, \omega^*)$ minimizes the cost functional

$$\begin{aligned} J = g(x_{q_e}(t_e)) &+ \sum_{j=1}^{N} c_{q_{j-1},q_j}(x_{q_{j-1}}(t_j^-), t_j) \\ &+ \sum_{j=0}^{N} \int_{t_j}^{t_{j+1}} \phi_{q_j}(x_{q_j}(t), u_{q_j}(t), t)\, dt \end{aligned} \tag{2.12}$$

while satisfying initial conditions x_0 and q_0, the dynamics of the hybrid system $\mathbb{H}$ defined in Definition 2.26, the Assumptions 2.11, 2.12, 2.27, and 2.16 - 2.18. Here, it is assumed that the terminal costs are time-independent, i.e. $g : \mathbb{R}^{n_x} \to \mathbb{R}$. □

Remark 2.29: The class of HOCPs defined above is quite general and includes a large number of hybrid effects when comparing the HOCP from Def. 2.28 to the HOCPs used in e.g. [18, 50, 137, 157, 165]. □

2.2.3 Hybrid Optimal Control Problem with Partitioned State Space

Here, hybrid systems are considered with partitioned continuous state space, where autonomous switching occurs on the boundaries between two neighboring partitions and switching manifolds are allowed to intersect. HOCPs include continuous controls, autonomous switching, and running and terminal costs.

Definition 2.30 (Hybrid System with Partitioned State Space): A hybrid system is a 6-tuple

$$\mathbb{H} := \{\mathcal{Q}, \mathcal{X}, \Gamma, U, \mathcal{F}, \mathcal{M}\}, \tag{2.13}$$

where $\mathcal{Q}$ is a set of discrete states, Γ a discrete transition map, and $\mathcal{X}$, U, $\mathcal{F}$, and $\mathcal{M}$ are collections of continuous state and control spaces, vector fields, and switching manifolds.□

Assumption 2.31:

1. $\mathcal{Q}$: set of discrete states as in Assumption 2.2.

2. $\mathcal{X} = \bigcup_{1 \le q \le N_q} \mathcal{X}_q$ is the union of all state spaces $\mathcal{X}_q = \partial \mathcal{X}_q \cup \mathrm{Int}(\mathcal{X}_q) \subset \mathbb{R}^{n_x}$ assigned to every discrete state q, where $\mathcal{X} = \mathbb{R}^{n_x}$, the continuous state is x and $\dim(x) = n_x$. The interiors of each pair of $\mathcal{X}_q$ are disjoint: $\mathrm{Int}(\mathcal{X}_i) \cap \mathrm{Int}(\mathcal{X}_k) = \emptyset$, for $i, k \in \mathcal{Q}$, $i \neq k$, while the boundaries of neighboring state spaces have a common set $\partial \mathcal{X}_i \cap \partial \mathcal{X}_k \neq \emptyset$.

3. $\Gamma : \mathcal{Q} \times \mathcal{X} \to \mathcal{Q}$: discrete transition map.

4. U: collection of continuous control spaces as in Assumption 2.4.

5. $\mathcal{F}$: collection of vector fields as in Assumption 2.5 except for the explicit time-dependence, i.e. $f_q : \mathcal{X}_q \times U_q \to \mathbb{R}^{n_x}$.

6. $\mathcal{M} = \{M_{i,k}\}_{i,k \in \mathcal{Q}, i \neq k}$: collection of time-independent switching manifolds $M_{i,k}$ defining the boundary $\partial\mathcal{X}_i \cap \partial\mathcal{X}_k$ between neighboring partitions. An autonomous transition from the discrete state i to k occurs at time t_j for $x(t_j)$ on the manifold $M_{i,k}$. A switching manifold $M_{i,k} = \bigcup_\alpha M_{i,k}^\alpha$, $\alpha \in \mathbb{N}$, contains codimension 1 submanifolds of $\mathbb{R}^{n_x}$, which are smooth, i.e. C^∞, possibly have a boundary $\partial M_{i,k}^\alpha$, and are locally expressed here by $M_{i,k}^\alpha := \{x | m_{i,k}^\alpha(x) = 0\}$. (Note that the Implicit Function Theorem states for $m_{i,k}^\alpha \in C^\iota(\mathbb{R}^{n_x})$, $\iota \in \mathbb{N} \cup \{\infty\}$, that $(y, \hat{m}(y))$, $y \in \mathbb{R}^{n_x - 1}$, locally gives the zero level set of $m_{i,k}^\alpha(x)$, i.e. $x \in m_{i,k}^{\alpha^{-1}}(0)$ with $x \in \mathbb{R}^{n_x}$, and $m_{i,k}^\alpha(x)$ and $\hat{m}$ have the same degree of regularity [96].) It is assumed that:
   - (i) $x \in M_{i,k}^\alpha$ is such that $x \in M_{i,k}^\alpha \cap M_{i,l}^\nu$, $\alpha \neq \nu \in \mathbb{N}$ and/or $k \neq l$, $i, k, l \in \mathcal{Q}$, if and only if $x \in \partial M_{i,k}^\alpha \cap \partial M_{i,l}^\nu$.
   - (ii) If $\partial M_{i,k}^\alpha \cap \partial M_{i,l}^\nu \neq \emptyset$ then $\partial M_{i,k}^\alpha \cap \partial M_{i,l}^\nu$ is a piecewise smooth codimension 2 submanifold of $\mathbb{R}^{n_x}$ (possibly with boundary).
   - (iii) The family of switching manifold intersections as well as the family of switching submanifold intersections are locally finite.
   - (iv) On $\partial M_{i,k} \cap \partial M_{i,l}$, the set of vector fields $f_k(x, u_k)$ and $f_l(x, u_l)$ are transversally with respect to the tangent space of the switching manifold $M_{k,l}$ at x in the same direction for all $u_k \in U_k$ and $u_l \in U_l$. (Note that this assumption can be relaxed if a deterministic choice for one of the two vector fields is guaranteed otherwise.)
   - (v) Let two points $\bar{x} \in \mathrm{Int}(M_{i,k}^\alpha)$ and $x \in \mathrm{Int}(\mathcal{X}_i)$ exist such that $\|x - \bar{x}\| < \|x - \hat{x}\|$ holds for all $\hat{x} \in \mathrm{Int}(M_{i,l}^\nu)$ with $\alpha \neq \nu \in \mathbb{N}$ and/or $k \neq l$ for $i, k, l \in \mathcal{Q}$. Then manifolds are oriented such that $m_{i,k}^\alpha(x) < 0$ and $m_{i,k} = -m_{k,i}$. □

Definition 2.32 (HOCP with Partitioned State Space): An optimal solution $\sigma^* = (\tau^*, \varrho^*, \chi^*, \upsilon^*)$ minimizes the cost functional

$$J = g(x_{q_e}(t_e)) + \sum_{j=0}^{N} \int_{t_j}^{t_{j+1}} \phi_{q_j}(x_{q_j}(t), u_{q_j}(t))\, dt \tag{2.14}$$

while satisfying initial conditions x_0 and q_0, continuity of the continuous state at switchings $x(t_j^-) = x(t_j)$, the dynamics of the hybrid system $\mathbb{H}$ defined in Definition 2.30, and the Assumptions 2.11 and 2.12. The terminal and running costs fulfill the Assumptions 2.16 and 2.17 except for being time-independent, i.e. $g : \mathbb{R}^{n_x} \to \mathbb{R}$ and $\phi_q : \mathcal{X}_q \times U_q \to \mathbb{R}$. □

Remark 2.33: HOCPs with partitioned state space that satisfy Assumption 2.31 appear in several physical systems, see Chap. 3, and are used for the piecewise affine approximation of nonlinear optimal control problems. □

2.2.4 Hybrid Optimal Control Problem with State Constraints

In the following, HOCPs are defined with continuous controls, autonomous switching, state and terminal constraints, and terminal and running costs.

Definition 2.34 (Hybrid System with State Constraints): A hybrid system is a 7-tuple

$$\mathbb{H} := \{\mathcal{Q}, \mathcal{X}, U, \mathcal{F}, \mathcal{M}, \Gamma, \mathcal{H}\}, \tag{2.15}$$

where $\mathcal{Q}$ is a set of discrete states, Γ a discrete transition map, and $\mathcal{X}$, U, $\mathcal{F}$, $\mathcal{M}$, and $\mathcal{H}$ are collections of continuous state and control spaces, vector fields, switching manifolds, and constraints. □

Assumption 2.35:

1. $\mathcal{Q}$: set of discrete states as in Assumption 2.2.
2. $\mathcal{X}$: collection of continuous state spaces as in Assumption 2.3.
3. $\Gamma : \mathcal{Q} \times \mathcal{X} \to \mathcal{Q}$: discrete transition map.
4. U: collection of continuous control spaces as in Assumption 2.4.
5. $\mathcal{F}$: collection of vector fields as in Assumption 2.5 except for being time-independent, i.e. $f_q : \mathcal{X}_q \times U_q \to \mathbb{R}^{n_x}$.
6. $\mathcal{M}$: collection of switching manifolds as in Assumption 2.6 except for being time-independent, i.e. $m_{i,k} : \mathcal{X}_q \to \mathbb{R}$.
7. $\mathcal{H}$: collection of constraints as in Assumption 2.10 except for being time-independent, i.e. $h_q : \mathcal{X}_q \times U_q \to \mathbb{R}^{n_{h_q}}$. □

Definition 2.36 (HOCP with State Constraints): An optimal execution $\sigma^* = (\tau^*, \varrho^*, \chi^*, \upsilon^*)$ minimizes the cost functional

$$J = g(x_{q_e}(t_e)) + \sum_{j=0}^{N} \int_{t_j}^{t_{j+1}} \phi_{q_j}(x_{q_j}(t), u_{q_j}(t))\, dt \tag{2.16}$$

while satisfying initial conditions x_0 and q_0, continuity of the continuous state at switchings $x(t_j^-) = x(t_j)$, the dynamics of the hybrid system $\mathbb{H}$ defined in Definition 2.34, the Assumptions 2.11 and 2.12, and possibly terminal constraints ψ_N. The terminal and running costs and the terminal constraints fulfill the Assumptions 2.16, 2.17, and 2.19 except for being time-independent, i.e. $g : \mathbb{R}^{n_x} \to \mathbb{R}$, $\phi_q : \mathcal{X}_q \times U_q \to \mathbb{R}$, and $\psi_N : \mathcal{X}_{q_N} \to \mathbb{R}^{n_{\psi_N}}$. □

Remark 2.37: State constraints occur if physical variables are limited by construction or if safety boundaries have to be obeyed. □

2.2.5 Hybrid Optimal Control Problem with Raised Differentiability

This subclass describes HOCPs with at least twice continuously differentiable functions, continuous controls, autonomous switching, interior and terminal constraints, and terminal and running costs.

Definition 2.38 (Hybrid System with Raised Differentiability): A hybrid system is a 6-tuple

$$\mathbb{H} := \{\mathcal{Q}, \mathcal{X}, \Gamma, U, \mathcal{F}, \mathcal{M}\}\,, \tag{2.17}$$

where $\mathcal{Q}$ is a set of discrete states, Γ a discrete transition map, and $\mathcal{X}$, U, $\mathcal{F}$, and $\mathcal{M}$ are collections of continuous state and control spaces, vector fields, and switching manifolds.□

Assumption 2.39:

1. $\mathcal{Q}$: set of discrete states as in Assumption 2.2.
2. $\mathcal{X}$: collection of continuous state spaces as in Assumption 2.3.
3. $\Gamma : \mathcal{Q} \times \mathcal{X} \to \mathcal{Q}$: discrete transition map.
4. U: collection of continuous control spaces as in Assumption 2.4.
5. $\mathcal{F}$: collection of vector fields as in Assumption 2.5 except for being time-independent and at least twice continuously differentiable, i.e. $f_q \in C^2(\mathcal{X}_q \times U_q, \mathbb{R}^{n_x})$.
6. $\mathcal{M}$: collection of switching manifolds as in Assumption 2.6 except for being time-independent and at least twice continuously differentiable, i.e. $m_{i,k} \in C^2(\mathcal{X}_q, \mathbb{R})$. □

Definition 2.40 (HOCP with Raised Differentiability): An optimal solution $\sigma^* = (\tau^*, \varrho^*, \chi^*, \upsilon^*)$ minimizes the cost functional

$$J = g(x_{q_e}(t_e)) + \sum_{j=0}^{N} \int_{t_j}^{t_{j+1}} \phi_{q_j}(x_{q_j}(t), u_{q_j}(t))\, dt \tag{2.18}$$

while satisfying initial conditions x_0 and q_0, continuity of the continuous state at switchings $x(t_j^-) = x(t_j)$, the dynamics of the hybrid system $\mathbb{H}$ defined in Definition 2.38, and the Assumptions 2.11 and 2.12. Furthermore, interior and terminal constraints ψ_j are fulfilled as described in Assumption 2.19 and additionally ψ_j is at least twice continuously differentiable, i.e $\psi_j \in C^2(\mathcal{X}_q, \mathbb{R}^{n_{\psi_j}})$. The terminal and running costs fulfill the Assumptions 2.16 and 2.17 except for being time-independent and at least twice continuously differentiable, i.e. $g \in C^2(\mathbb{R}^{n_x}, \mathbb{R})$ and $\phi_q \in C^2(\mathcal{X}_q \times U_q, \mathbb{R})$. □

Remark 2.41: HOCPs with raised differentiability demand all functions to be at least twice continuously differentiable. In general, physical systems satisfy this assumption, so that no limitations for algorithms based on these HOCPs are expected in the application to practical problems. □

2.2.6 Hybrid Optimal Control Problem with Raised Differentiability and Constraints

Here, HOCPs are defined with at least twice continuously differentiable functions, continuous controls, autonomous switching, interior and terminal constraints, and terminal and running costs.

Definition 2.42 (Hybrid System with Raised Differentiability and Constraints): A hybrid system is a 7-tuple

$$\mathbb{H} := \{\mathcal{Q}, \mathcal{X}, \Gamma, U, \mathcal{P}, \mathcal{F}, \mathcal{M}, \mathcal{H}\}\,, \tag{2.19}$$

where $\mathcal{Q}$ is a set of discrete states, Γ a discrete transition map, $\mathcal{P}$ a parameter space, and $\mathcal{X}$, U, $\mathcal{F}$, $\mathcal{M}$, and $\mathcal{H}$ are collections of continuous state and control spaces, vector fields, switching manifolds, and constraints. □

Assumption 2.43:

1. $\mathcal{Q}$: set of discrete states as in Assumption 2.2.

2. $\mathcal{X}$: collection of continuous state spaces as in Assumption 2.3.

3. $\Gamma : \mathcal{Q} \times \mathcal{X} \to \mathcal{Q}$: discrete transition map.

4. U: collection of continuous control spaces as in Assumption 2.4. Here, $\mathcal{U}$ is the set of piecewise continuous control trajectories $\upsilon : [t_0, t_e] \to U$.

5. $\mathcal{P} \subseteq \mathbb{R}^{n_{\bar{p}}}$: set of parameters $\bar{p}$.

6. $\mathcal{F}$: collection of vector fields as in Assumption 2.5 except for additionally depending on the parameter $\bar{p}$, i.e. $f_q : \mathcal{X}_q \times U_q \times \mathcal{P} \times \mathbb{R} \to \mathbb{R}^{n_x}$, and being at least twice continuously differentiable with respect to the state x, the control u, the parameter $\bar{p}$, and time t.

7. $\mathcal{M}$: collection of switching manifolds as in Assumption 2.6 except for additionally depending on the parameter $\bar{p}$, i.e. $m_{i,k} : \mathcal{X}_q \times \mathcal{P} \times \mathbb{R} \to \mathbb{R}$, and being at least twice continuously differentiable with respect to the arguments.

8. $\mathcal{H}$: collection of constraints as in Assumption 2.10 except for additionally depending on the parameter $\bar{p}$, i.e. $h_q : \mathcal{X}_q \times U_q \times \mathcal{P} \times \mathbb{R} \to \mathbb{R}^{n_{h_q}}$, and being at least twice continuously differentiable with respect to the arguments. □

Definition 2.44 (HOCP with Raised Differentiability and Constraints): An optimal solution $\sigma^* = (\tau^*, \varrho^*, \chi^*, \upsilon^*)$ minimizes the cost functional

$$J = g(x_{q_e}(t_e), \bar{p}, t_e) + \sum_{j=0}^{N} \int_{t_j}^{t_{j+1}} \phi_{q_j}(x_{q_j}(t), u_{q_j}(t), \bar{p}, t)\, dt \tag{2.20}$$

while satisfying initial conditions x_0 and q_0, continuity of the continuous state at switchings $x(t_j^-) = x(t_j)$, the dynamics of the hybrid system $\mathbb{H}$ defined in Definition 2.42, and the Assumptions 2.11 and 2.12. Furthermore, interior and terminal constraints ψ_j are fulfilled. The terminal and running costs and the interior conditions fulfill the Assumptions 2.16, 2.17, and 2.19 except for additionally depending on the parameter $\bar{p}$, i.e. $g : \mathcal{X}_q \times \mathcal{P} \times \mathbb{R} \to \mathbb{R}$, $\phi_q : \mathcal{X}_q \times U_q \times \mathcal{P} \times \mathbb{R} \to \mathbb{R}$, and $\psi_j : \mathcal{X}_q \times \mathcal{P} \times \mathbb{R} \to \mathbb{R}^{n_{\psi_j}}$, and being at least twice continuously differentiable with respect to the arguments. □

Remark 2.45: Again physical systems, in general, fulfill the specified assumptions for HOCPs with Raised Differentiability and Constraints, such that the applicability of corresponding results is not expected to be restricted. The specified HOCP with Raised Differentiability and Constraints is solved in this thesis by a model predictive control algorithm that explicitly uses knowledge from the previous time step and accounts for changes of the online estimated parameters $\bar{p} \in \mathcal{P}$. Since the other algorithms in this thesis are, in general, developed for offline computations and thus, do not consider information from previous time steps, the parameter space $\mathcal{P}$ is only introduced here. □

Which class of hybrid systems and HOCPs is considered is stated explicitly at the beginning of each section.

2.3 State of the Art Solution Methods

In the following, different approaches to solve the presented HOCPs are shown. The solution approaches are categorized in three main types, namely dynamic programming, direct methods, and indirect methods. The presented assignment of methods to the three categories is not necessarily unique since some methods combine characteristics from several categories. The various methods in those categories focus on different subclasses of the general HOCP defined in Sec. 2.2. Main differences are if HOCPs with a fixed or free sequence of discrete states and with autonomous or controlled switching are considered. Only few approaches, e.g. [157], handle several of these properties simultaneously.

Considering the historical context of the three methods, indirect methods are by far the oldest. Indirect methods appeared first in form of calculus of variations, which go back to the 17-th century, when Johan Bernoulli invited to the solution of a novel problem, see [20] for a summary. The problem consisted in finding the time-optimal path of a particle in a gravitational field between any two points. Since then, the methodology was steadily developed theoretically and algorithmically. DP was introduced by Caratheodory in an early version in the 1930s [44] and Bellman in the 1950s [12]. DP and indirect methods are both based on complex mathematical frameworks . In contrast, direct methods are intuitively applicable and just came up recently with the increase of computational power. One of the first approaches are presented in Bock and Plitt [26].

2.3.1 Dynamic Programming

Dynamic Programming (DP) for nonlinear systems was formulated by Bellman [12] based on his principle of optimality. Assume for illustration that an optimal solution is found from an initial state to a final state. Then Bellman's principle of optimality states that the solution from any intermediate state of the optimal solution to the final state is also optimal. This fact is exploited in DP by dividing the optimal control problem into many small problems. Preliminary results of Bellmann's principle can already be found in [44]. Solution methods using DP are grouped depending on the system class underlying the optimization problem. For discrete-time or discrete-decision problems, where the dynamics can be described by difference equations, developed methods are based on a discrete DP algorithm [21]. The discrete DP algorithm is successfully applied to control a truck based on a nonlinear model in [73] with the goal to reduce the fuel consumption. In [220], the approach is extended to a hybrid model by additionally considering gear shifting.

Purely Continuous Optimal Control Problems

The continuous-time counterpart for purely continuous systems to the DP algorithm is the Hamilton-Jacobi-Bellman (HJB) equation, which is a partial differential equation [21]. If a solution to this equation, also called value function, is found, then it is guaranteed by sufficient optimality conditions that the result is an optimal solution. DP has two major advantages compared to direct and indirect solution methods. First, a solution for an optimal control problem is found by searching the whole state space and thus, a solution is globally optimal. Second, the optimal control is found in dependence on the current state of the system, which means that optimal controls are feedback controls. However, there is also a major drawback of DP. To find an optimal control for continuous-time systems, it is necessary to solve the HJB equation. Analytical solutions only exist for special system classes, e.g. linear systems with quadratic costs [32]. In most cases, the solution has to be found numerically. There the problem of the curse of dimensionality arises, since for solutions the time, state, control, and possibly output spaces have to be sampled, which means that the computational complexity increases exponentially with the dimensions of the state, control, and output spaces [133]. Efforts to resolve the curse of dimensionality have resulted in approximate dynamic programming [133]. There, for example, function approximations are used to approximate the value function or control policies, such that a coarser and possibly adaptive grid [67] can be chosen in the state and control spaces. Furthermore, iterative dynamic programming has been introduced, which only requires a grid in the neighborhood of a solution to find an optimal solution [103]. However, the global optimality of the solution cannot be guaranteed anymore in the latter approach.

Hybrid Optimal Control Problems

DP has been extended in [29, 142] to a general class of hybrid systems with autonomous and controlled switching. Like for purely continuous systems, solution approaches rely on discretizing the hybrid state and control space and approximating the value function [30, 72]. Though several algorithms have been developed, the convergence of the approximated value function to the true value function is in general still to be shown. Furthermore, it is not ensured if the control law that is applied to a system and that is based on the discretized state and control spaces is stabilizing the original hybrid system. A first result that proves the stability and convergence for discrete-time, hybrid systems is provided in

[68]. For continuous-time, hybrid systems, a first result that shows that the approximated value function converges to the true value function is given in [140, 141]. In [140], a method is also presented, where the discretization grid is refined according to the gradient of the value function to reduce the computational burden. An alternative approach is pursued in [31], where a class of hybrid systems with constant controls is abstracted into purely discrete systems using bisimulation and then DP is applied for the solution.

A different approach is pursued in [186], where the optimal switching instants are found by a direct differentiation of the value function for hybrid systems with controlled switching and a fixed sequence of discrete states. Similarly, gradients of the value function are used to determine the optimal switching state and time for autonomous switching [170]. Though using derivative information of value functions, these methods rather belong to indirect methods and mainly share their advantages and disadvantages.

Comparison with Direct and Indirect Methods

In general, a strength of DP for hybrid systems is that it only has a linear complexity increase in the number of discrete states and the increase in the number of autonomous and controlled switchings is not directly correlated to the complexity of finding a solution. In contrast, direct and indirect methods have a combinatorial complexity increase in the number of autonomous and mostly also controlled switchings. Unlike direct and indirect methods, DP delivers globally optimal solutions and is real-time capable if the HOCP can be solved offline a priori to the execution. Nevertheless, due to the curse of dimensionality, DP can in general only be applied to systems with a state space dimension of up to 4. Further, the accuracy of solutions does not reach the level of indirect methods and it is not ensured that a solution is stabilizing due to the discretization.

2.3.2 Direct Methods

Direct methods solve optimal control problems for purely continuous, nonlinear systems based on a discretization with respect to time. That means the infinite dimensional problem of finding optimal state and control trajectories is transformed into a finite dimensional optimization problem by evaluating state and control values only at a finite number of time samples. By varying the state and control values at those samples, the cost criterion is directly optimized. The optimization can be performed by linear or nonlinear programming methods like sequential quadratic programming (SQP) [117]. The optimal control problem is usually formulated as a direct collocation algorithm or as a direct multiple shooting algorithm [23, 146]. In the following, direct solution approaches based on collocation and shooting methods are presented and compared. The presentation is first for purely continuous optimal control problems and second for HOCPs.

Purely Continuous Optimal Control Problems

Direct collocation was first introduced by [172] and extended by [24, 174]. Optimization variables are the state and control values at the selected discrete time instants. In between the discretized time values, the state and control is approximated by time-dependent functions, e.g. cubic and linear polynomials of time in [174], respectively. At the time samples and at various intermediate points, the system dynamics have to be met. Furthermore, state and control constraints and initial and final conditions have to be satisfied. The optimization criterion is minimized and the equality and inequality constraints are satisfied by a solver varying the optimization variables.

In direct single shooting, the optimization variables are the initial value for the state variable and parameters for defining the control functions. Starting with an initial set of optimization variables, the system differential equations are integrated from the initial to the final time and the cost criterion is evaluated simultaneously. An optimization solver again varies the optimization variables until an optimal solution is found. This approach only possesses limited stability since the sensitivities of the optimization variables at the beginning of the trajectory is very high and changes in these variables cause considerably nonlinear changes at the end of the trajectory [22]. The problem is overcome by introducing multiple shooting in [26]. The idea is to add time instants, where the state is reinitialized by additional optimization variables, such that the stability is increased. The solution can still be computed efficiently by exploiting the structure of the optimization problem [98].

A comparison of collocation and shooting methods is given in [22]. Collocation methods usually require a finer discretization grid, so that the resulting optimization problem is larger. Furthermore, direct multiple shooting is more suitable for stiff problems since error-controlled solvers can be used for the integration of the system dynamics, whereas an adaptive grid refinement for error-control is not straightforward in collocation [144]. The finer grid in collocation methods offers slight advantages in considering state constraints. However, both methods can only enforce state constraints at nodes of the discretized time or if the sequence of constrained and unconstrained arcs of the system trajectory is known, multiple shooting can also maintain state constraints with higher precision.

Hybrid Optimal Control Problems

HOCPs that are solved by direct methods are usually formulated as mixed-integer optimization problems (MIOPs). In MIOPs, the continuous dynamics are again time discretized and the evaluation of the discrete dynamics, which are modeled by a set of integer variables, is also bound to the nodes of the time discretization. With integer variables and constraints, controlled and autonomous switching can be considered. Purely integer optimization problems are a subclass of MIOPs and it was shown that integer optimization problems are $\mathcal{NP}$-hard to solve [57]. This implies that the solution of MIOPs is also $\mathcal{NP}$-hard. For controlled switching without constraints on the switching structure, the problem complexity grows exponentially with the time horizon. For autonomous switching, the complexity still grows combinatorially. Therefore, various methods have been developed to keep the computation time in acceptable bounds. Good overviews are provided in [69, 119]. The most prominent methods are branch and bound, branch and cut, outer approximation, generalized Benders decomposition, linear programming/nonlinear programming based branch and bound, and branch, cut, and price. The methods aim at reducing the number of possible combinations of values for the integer variables. So for example, branch and bound methods are especially useful if the relaxations of the integer variables are tight and the underlying nonlinear programs can be solved efficiently [69].

In contrast, outer approximation is preferable if the solutions of the underlying nonlinear programs are time-consuming and if the occurring nonlinearities are weak [144]. Integrating the branch and bound process into nonlinear programming with an SQP solver further reduces the computational time [144]. If in non-convex optimization problems, a global solution is desired, then the mixed-integer optimization has to be combined with global optimization tools like spatial branch and bound and relaxation methods, see [116] for an overview. An alternative approach to find a locally optimal control and discrete state sequence for controlled switching is presented in [64], where a discrete state is inserted into the current discrete state sequence for local optimization. To reduce the computational load, MIOPs are only solved on a shortened optimization horizon and solutions are recomputed on a receding horizon with proceeding time [163].

The concepts for solving MIOPs are combined with direct collocation and direct multiple shooting. In [38], collocation and a branch and bound approach are used. Collocation is also applied in [180], where a locally optimal sequence of discrete states is found by including results based on optimality conditions and an algorithm from [18]. It is also stated that the algorithm finds locally optimal discrete state sequences for autonomous switching, though the accuracy of the solution is especially questionable here. Direct multiple shooting is used in conjunction with a convexification of integer variables in [89, 144]. This approach is especially interesting since the convexification of the integer variables reduces the exponential complexity in the length of the time horizon to a polynomial complexity in the number of discrete states.

MIOPs can also be formulated as logic based mixed-integer optimization problems (LB-MIOPs), which is intended to offer advantages in modeling hybrid systems [69]. MIOPs and LB-MIOPs are equivalent descriptions and can be transformed into each other. In LB-MIOPs, the discrete dynamics are described by a set of algebraic constraints including continuous and integer variables. Basically two approaches have been developed. The first one is based on the 'big-M' notation introduced in [61, 181] and used in [16] to define mixed logical dynamical systems. The second approach is disjunctive programming developed by [10, 97]. Comparing the two approaches [69, 164], disjunctive programming seems to be advantageous for the optimization task since in general less auxiliary integer variables are introduced and the resulting relaxations of the integer variables are tighter. The disadvantage of introducing additional integer variables is considered to be so severe in [161], that the MIOP is again formulated in a tree structure and solved with branch and bound techniques.

Comparison with DP and Indirect Methods

Direct methods for the solution of HOCPs have several advantages and disadvantages compared to DP and indirect methods [22, 128, 175]. Direct methods can be implemented and several software packages exist, such that only few optimal control knowledge is required to use it. The methods are stable and have a quite large domain of convergence in comparison to indirect methods. The initialization of state and control constraints is quite straightforward since these variables mainly have a physical interpretation and knowledge about the process can be incorporated. For most methods, the sequence of state-constrained and unconstrained arcs of a solution trajectory does not need to be guessed in advance but evolves during the optimization. Direct approaches are very flexible in adapting the structure of the HOCP. A major disadvantage of direct methods is that solutions do not reach the accuracy of indirect methods by several magnitudes, which is mainly due to discretization errors. Direct methods sometimes converge to local minima, which are introduced by the discretization and which do not match a true locally optimal solution [175]. In general, direct methods have an exponential complexity in the number of time nodes for controlled switching and a combinatoric complexity in the number of autonomous switchings. Approaches like branch and bound usually reduce the complexity, though it remains high. Only in the approach used in [89, 144], the exponential complexity for controlled switching is reduced to a polynomial complexity. The effort to calculate derivatives for the optimization is usually high. Direct methods provide, in general, open-loop controls.

2.3.3 Indirect Methods

In indirect methods, an optimal solution is found by satisfying optimality conditions instead of minimizing a cost criterion directly as in direct methods. Depending on the given optimal control problem, the optimality conditions lead to a two-point or multi-point boundary value problem (MPBVP). It consists of differential equations for the continuous state and adjoint variables, an algebraic equation for the continuous control, and boundary conditions for the continuous states, the adjoint variables, and time. In some very special cases with purely continuous, linear systems and quadratic costs, an analytic solution of the BVP can be found. In most cases, however, it is necessary to iteratively approach a solution with numerical methods. At first, optimality conditions are reviewed for non-hybrid and hybrid optimal control problems. Afterwards, different numerical solution methods are presented from literature, starting with methods for purely continuous optimal control problems and finishing with methods for HOCPs.

Optimality Conditions for Purely Continuous Optimal Control Problems

There is a long history in considering optimal control problems for purely continuous systems. However, it was Pontryagin and his coworkers, who achieved a breakthrough in deriving optimality conditions, which an optimal solution has to satisfy [132]. The conditions are usually called Pontryagin's Maximum Principle because it requires the optimal control to be one which maximizes the Hamiltonian at every point of time. The proof of the optimality conditions is based on a sequence of needle variations in the optimal control. The variations in the control cause variations in the state trajectory, which influences the value of the cost criterion. Analyzing the changes in the cost criterion, necessary optimality conditions of first order for a fairly general class of nonlinear systems are obtained. Almost at the same time, a similar result was derived by Hestens [74]. In the sequel, the maximum principle was proven by several authors, each relying on slightly different proof concepts based on needle variations, e.g. [46, 51]. In [27], the maximum principle is extended to provide sufficient optimality conditions for time optimality. The maximum principle has also been formulated for nonlinear systems with state constraints [76, 106]. On an arc of the state trajectory that moves along a state constraint, the adjoint variables are not defined uniquely anymore [32]. This leads to special conditions, mostly jump conditions at the entry point of a constrained arc, for the adjoint variables on active state constraints. A good overview about different approaches to deal with state constraints is contained in [70]. Based on Riccati equations, second-order necessary and sufficient conditions for an optimal solution are provided for systems with pure state and mixed state-control constraints in [190]. Optimality conditions for nonlinear systems with differential and algebraic equations are derived in [59]. DP and the maximum principle lead to equivalent optimality conditions if some conditions hold. The relation of DP and the maximum principle is reviewed in [129]. Parallel to the maximum principle, optimality conditions have been developed with variational techniques, which are suitable for sufficiently smooth system dynamics and controls, e.g. by Bryson and Ho [32]. Calculus of variations delivers optimality conditions for nonlinear systems with state, control, and mixed state-control constraints, interior points, and even discontinuities of the system dynamics at interior points.

Optimality Conditions for Hybrid Optimal Control Problems

An early extension of the maximum principle to hybrid systems with autonomous switching is provided by Witsenhausen [182]. Clarke and Vinter derive optimality conditions for optimal controls of multi-processes [45], where multi-processes consist of autonomous and controlled switching in a fixed sequence of processes. A general version of the maximum principle for hybrid systems with state resets, autonomous switching, and switching costs is presented by Sussmann [165] based on a set of needle variations and a Boltyanskii approximating cone as used similarly by Pontryagin [132]. Two versions of the maximum principle are introduced by Riedinger et al. [137] and Shaikh and Caines [157] for hybrid systems with autonomous and controlled switching. Note that in [157], the maximum principle is called hybrid minimum principle (HMP). Maximizing or minimizing a cost criterion is equivalent since both approaches can be converted into each other by inverting the sign of the cost criterion. In the following of the thesis, the term minimum principle is used for convenience independently of the original formulation in the literature. The mentioned versions of the HMP provide necessary optimality conditions that an optimal solution has to satisfy. In the case of autonomous switching, this includes state and adjoint differential equations, a minimization of the Hamiltonian with respect to the continuous control, initial and terminal conditions for the state and/or adjoint variables, jump conditions for the adjoint variables on switching manifolds, and Hamiltonian value conditions specifying the optimal switching times. However, no condition with respect to the sequence of discrete states can be given. For controlled switching, the HMP from [18, 137, 157] provides additionally necessary optimality conditions, that an optimal sequence of discrete states has to fulfill, in the form of a minimization condition of the Hamiltonian with respect to the discrete control. In [56], the problem is investigated that an optimal solution within a prespecified sequence of discrete states has to be determined. This requires modified optimality conditions if the desired sequence does not contain an optimal solution and the best solution lies on the boundary of the sequence. A version of the HMP for hybrid systems with state constraints is discussed in [39]. Optimality conditions for hybrid systems with an infinite number of discrete states are derived in [55].

Numerical Methods for Purely Continuous Optimal Control Problems

The presented optimality conditions always result in a boundary value problem. Several approaches have been suggested for finding solutions. For purely continuous systems, Polak [131], Bryson and Ho [32], and Betts [22] provide good overviews and comparisons of different approaches. The various methods are usually classified in three groups, which are neighboring extremal methods (also known as shooting methods), gradient methods, and collocation methods. The methods can be further divided into perturbation and quasilinearization methods [168]. In perturbation methods, required derivatives for updates are found by evaluating the nonlinear state and adjoint differential equations directly. Contrarily in quasilinearization methods, linearizations of the state and adjoint differential equations are used for the iterative optimization. A slightly different classification appears in [1], where collocation methods are further divided depending on the choice of approximating function and analytical methods are additionally considered. In the following, the three main solution methods are described.

Multiple shooting as solution of a boundary value problem has been introduced by Bulirsch [35] and Bock [25]. For indirect methods, multiple shooting works as follows: The time horizon is divided into several segments and at the beginning of each segment initial values for the state and the adjoint variables are guessed. Starting from those initial values, the state and adjoint differential equations are integrated to the end of the segment. Error equations are set up from initial conditions, from the mismatch between final values of the state and adjoint variables at the end of each segment and the initial values of the subsequent segment, and from final conditions. These error equations are steered to zero by an iterative Newton method, which varies the initial values. The presented concepts in [25, 35] are already capable of dealing with MPBVPs. In [36, 120], it is focused on multiple shooting as solution method for MPBVPs containing arcs with singular controls and state constraints. Local feedback control laws in the linearized tangent space around an optimal solution are precalculated and used online in [126, 127] to increase the robustness against perturbations. Multiple shooting solves optimal control problems with high accuracy and near the optimum the method converges quadratically. Additionally, multiple shooting can be implemented straightforwardly. Disadvantages are that the domain of convergence is relatively small compared to other methods due to integrating either the state or adjoint differential equations into the instable direction. This and the fact that the adjoint variables have no physically intuitive equivalence cause the initialization to be difficult, especially in comparison to other approaches. Further, the sequence of constrained and unconstrained arcs has to be known a priori.

The optimization variable in gradient methods is a history of control values from the initial to the final time [32, 65, 82, 101]. Using the initial conditions and the given control trajectory, the state differential equations are integrated forward in time until the final time. Afterwards, the adjoint differential equations are integrated backwards until the initial time. At each point of time, the optimality conditions for the control are usually violated. In the following update step, the control history is modified such that the satisfaction of the corresponding optimality conditions is iteratively approached. Depending on the update procedure for the control, gradient methods of first order [65] and second order [101] are distinguished. First-order methods have a large domain of convergence but a slow convergence rate near the optimum. Second-order methods show a small domain of convergence but a quadratic convergence rate in a sufficiently close neighborhood of the optimal solution. A gradient method, which uses conjugate search directions, is presented in [93]. The approach aims at combining the advantages of first- and second-order gradient methods and at avoiding their disadvantages. Gradient methods are capable of finding solutions with comparably high accuracy as multiple shooting at the price of a higher storage demand for the control history and a higher number of integrations. Compared to multiple shooting, the initialization of gradient methods is more straightforward since control values are more intuitive than adjoint values. Additionally, gradient methods can be set up, such that the sequence of constrained and unconstrained arcs does not have to be provided a priori.

In collocation methods based on quasilinearization, a history of state and adjoint variables is guessed initially [32, 84] and consequently deviations from the optimal state and adjoint differential equations are obtained. Around the state and adjoint trajectories, the state and adjoint differential equations are linearized and the control is chosen, such that it satisfies the corresponding optimality condition. Additionally linearizing all boundary conditions, a linear boundary value problem is obtained, which includes the errors in the state and adjoint dynamics and the boundary conditions as inhomogeneous right-hand side. After solving the boundary value problem, the state and adjoint histories are updated. The method converges fast near the optimum as second-order gradient methods and multiple shooting do, but does not completely reach the high accuracy of gradient and multiple shooting methods. The quasilinearization method and multiple shooting are equivalent if the number of shooting nodes approaches infinity [160]. The initialization of the state history is even more intuitive than the one of the control history in gradient methods. However, the initialization of the adjoint variable is not intuitive. In some cases, the convergence may be difficult to achieve since the method starts from an initial guess that satisfies only very few optimality conditions. Collocation methods based on approximating functions for solving boundary value problems proceed similarly as direct collocation methods. Instead of minimizing a cost criterion, only boundary conditions have to be fulfilled now. An example, where collocation is applied to the solution of a boundary value problem, can be found in [5]. The initialization process is similar to the one of quasilinearization methods. If collocation methods are to deliver solutions with comparable accuracy as the previously mentioned indirect methods, the computational complexity is much higher than in those methods [160]. Additionally, we encountered singularities in the Jacobian in simulation examples, where multiple shooting was still capable of solving the problem.

Numerical Methods for Hybrid Optimal Control Problems

Partly, the presented numerical methods for purely continuous systems are transferred to HOCPs. For hybrid systems with autonomous switching, Bock [25] and Riedinger et al. [136] use multiple shooting and Shaikh and Caines [157] apply collocation methods. In [136], the introduced method is called multiple-phase multiple shooting. Each phase is assigned to a period, where the system is in a certain discrete state. In each phase, multiple shooting is applied and the different phases are connected by optimality conditions for the adjoint variables and the Hamiltonian, and boundary conditions for the state. Collocation for solving HOCPs is applied in [157]. There, an optimal control algorithm with two layers is introduced. On the upper layer, switching states and times are guessed and iteratively improved by a gradient method, such that up to this point unsatisfied optimality conditions are approached. On the lower layer, purely continuous optimal control problems are solved from one switching point to the next by a collocation method. On the upper layer of the algorithm in [152], the convergence to necessary optimality conditions is replaced by a value function approach. The intention is to find the globally optimal discrete state sequence for hybrid systems with partitioned state space, also called regional hybrid systems. Gradient methods for use with hybrid systems are shortly mentioned in [136], however, to the knowledge of the authors, a serious analysis and implementation of gradient methods for hybrid systems is missing. Similarly, nothing is known about a transfer of quasilinearization methods to hybrid systems.

All indirect methods for hybrid systems with autonomous switching share the disadvantage that if the optimal sequence of discrete states is unknown the search for the optimal sequence has a combinatorial complexity in the number of switchings. For controlled switching with continuity of the states at switchings, various approaches have been developed that only have a moderate increase in the computational complexity with respect to the number of discrete states or number of switchings. In [155], optimality zones are introduced, which permit to find an optimal sequence of discrete states with a complexity that is proportional to the number of zones. Optimality zones are created such that the optimal discrete state is determined for each pair of entry state and time and exit state and time. On the underlying layer, a collocation method is used. In [136], two concepts are presented based on multiple shooting. First, the Hamiltonian minimization condition with respect to the discrete control is directly used during the integration in the multiple shooting algorithm to decide which discrete state is applied next. When convergence is achieved, a locally optimal discrete state sequence is obtained. Second, a sequence of discrete states is initialized and the time for being in one discrete state is included in the set of optimization variables. By varying the length of the intervals up to zero, the sequence of discrete states can be changed. A locally optimal sequence of discrete states is found in [8] by inserting a discrete state for a short time span into the current sequence. The insertion delivers a gradient for optimizing the sequence. Optimal continuous controls are not included in this approach.

The algorithms in [157, 187] consider both autonomous and controlled switching for a fixed sequence of discrete states. This means finding an optimal sequence requires an enumeration of all possible sequences, which has a combinatorial complexity in the number of switchings. A special case of hybrid systems with autonomous switching is given, if the switching manifolds, which trigger the autonomous transitions from one discrete state to another, can be designed. How to design and where to locate such switching manifolds optimally is analyzed in [151, 169, 178] for fixed sequences of discrete states.

For a special class of hybrid systems, e.g. with linear dynamics, quadratic costs, and autonomous and/or controlled switching, specific solution approaches have been published [14, 92, 134, 138, 185]. Mostly, these approaches rely on Riccati equations, which are provided by special optimality conditions valid for the considered class of HOCPs. Here, the boundaries between indirect methods and dynamic programming often vanish due to equivalent aspects in the theory.

Comparison with DP and Direct Methods

A major advantage of indirect methods for HOCPs compared to direct methods and DP is the high accuracy of solutions. Further strengths are the reliability and the fast convergence in a sufficiently close neighborhood of the optimal solution. For controlled switching, algorithms exist that only have a low increase in the computational complexity with respect to the number of switchings. Simplified methods exist for special classes of hybrid systems with linear dynamics and quadratic cost. Disadvantages with respect to DP are that mainly open-loop optimal controls and only locally optimal solutions are found. Finding an optimal discrete state sequence for hybrid systems with autonomous switching has a combinatorial complexity in the number of switchings. Especially for multiple shooting and the second-order methods, the domain of convergence is small compared to direct methods and DP. In contrast to direct methods and DP, the initialization of the adjoint variables in indirect multiple shooting is difficult. The reasons are that the adjoints lack a physically intuitive equivalent and, in general, the adjoint differential equations are integrated in their instable direction, which makes the domain of convergence small. The latter often turns the optimization problem ill-conditioned. Multiple shooting implementations additionally need information about the optimal sequence of state-constrained and unconstrained arcs. In contrast to direct methods, indirect ones still require more optimal control knowledge, e.g. for deriving adjoint equations.

3 Necessary Optimality Conditions based on the Hybrid Minimum Principle

The hybrid minimum principle (HMP) provides necessary optimality conditions for a solution of a hybrid optimal control problem (HOCP) and can be used as basis for developing numerical optimal control algorithms. In this chapter, two novel versions of the HMP are introduced for extended classes of HOCPs. The first extension is to hybrid systems with autonomous switching on switching manifolds, controlled switching, jumps of the continuous state at switches, and time-varying functions for specifying the continuous and discrete dynamics. The formulation of the HOCP includes running, switching, and terminal costs. In former versions of the HMP, functions that explicitly depend on time as well as controlled switching with resets of the continuous state and switching costs cannot be considered.

The second extension is valid for a class of hybrid systems with partitioned continuous state space. Autonomous switching occurs whenever the continuous state trajectory meets the boundary between neighboring partitions. In this case, the continuous state trajectory can hit an intersection of two or more switching manifolds separating the associated partitions. In optimal control, the question arises, what happens when a trajectory passes through such intersections, since the existing versions of the HMP are not able to provide necessary optimality conditions for that case. Here, the existing HMP is extended to give the additional conditions. Further, the novel version of the HMP also enables us to propose an algorithm for hybrid optimal control that varies the discrete state sequence based on gradient information in Chap. 5. Consequently, the combinatorial complexity of former algorithms can be avoided, since not every possible discrete state sequence has to be analyzed separately anymore.

3.1 Introduction and State of the Art

In this chapter, optimality conditions are derived for effects in HOCPs, which are not covered by formerly derived optimality conditions. As a result, the class of HOCPs that can be solved by indirect and other methods is extended. Optimality conditions are important for various reasons: Numerous numerical approaches are based on evaluating optimality conditions for the solution of HOCPs, e.g. to check whether a candidate solution is an optimal one. Especially indirect methods use the optimality conditions to find results with high accuracy.

Optimality conditions were originally developed for purely continuous optimal control problems and were based on the concept of calculus of variations [32] or needle variations [132], where the latter results in the minimum (or maximum) principle. In particular optimality conditions for HOCPs based on needle variations were derived in the form of the HMP [45, 137, 182]. A HMP for a general class of hybrid systems is given in [165]. There, the hybrid system formulation includes autonomous switching on a fixed sequence of discrete states, resets of the continuous state at switches, and switching costs. Resets of the continuous state when switching autonomously are present e.g. when particles collide [62] and when contact or impact situations in robotics occur, as in bipedal locomotion [149] or in juggling tasks like ball dribbling in basketball robotics [33]. The versions of the HMP in [137, 157] can deal with hybrid systems with autonomous and controlled switching, when no resets of the continuous state at switches and no switching costs are present. No optimality conditions for the optimal sequence of discrete states with respect to autonomous switching can be provided. In contrast, conditions characterizing a discrete state sequence to be optimal exist for HOCPs with controlled switching in [18, 137, 157].

The contributions of the thesis with respect to extensions of the considerable effects in HOCPs are presented in the following. For the listed effects, no optimality conditions can be provided by the existing versions of the HMP.

(i) Controlled switchings with jumps of the continuous state variable and switching costs.

(ii) Explicitly time-dependent dynamics, running costs, jump maps, jump costs, and switching manifolds.

(iii) Intersections of switching manifolds for hybrid systems with partitioned continuous state space.

(iv) Non-differentiable points or subspaces (also called corners) of switching manifolds.

These effects are important in practice and can now be considered by the novel versions of the HMP. Below, some examples are given, where these effects occur:

(i) Controlled switching with resets of the continuous state can be found in manufacturing processes [123]. Another example is gear shifting in vehicles [13], where the dynamics of the gear shift with open clutch can be modeled abstractly by a jump in the time, position, and velocity of the vehicle [73].

(ii) Examples for time-dependent functions are the dynamics of sanding vehicles, where the mass changes with time, and supply chains, which also contain controlled switchings [124].

(iii) Optimality conditions valid at intersections of switching manifolds in hybrid systems with partitioned state space are important for two reasons: First, there exist hybrid systems, where an optimal continuous state trajectory naturally passes through an intersection of switching manifolds. An example is a four-legged pole climbing robot, which grabs the pole simultaneously with its two rear legs for maintaining stability [71]. Depending on the contact situation of each leg, the dynamics and the associated discrete state changes. Second, the optimality conditions enable us to develop an algorithm that finds the optimal sequence of discrete states based on a gradient-descent method in parallel with the continuous optimization. The order of the computational complexity of this algorithm is lower by a factor of combinatorial complexity than the complexity of existing algorithms, which have to analyze each possible sequence of discrete states.

(iv) Non-differentiable regions on a switching manifold can occur in robotics, when a robot may get into contact with an object that has a non-smooth surface.

The thesis introduces two novel versions of the HMP, which provide optimality conditions for the above described effects in HOCPs that cannot be considered up to now. The first version presented in Sec. 3.2 is for a general class of HOCPs. It includes explicitly time-dependent functions and controlled switching with resets and switching costs. Additionally, autonomous switching with resets and switching costs and purely controlled switching is also considered as in previous versions of the HMP. The second version derived in Sec. 3.3 is for special HOCPs with a small number of hybrid effects and the focus is on intersecting switching manifolds. The version is valid for hybrid systems with partitioned continuous state space, also known as *regional hybrid systems* [152]. Here, intersections of switching manifolds are formed by the boundaries of neighboring partitions. This case cannot be handled by existing versions of the HMP. Non-differentiable subspaces or corners of a switching manifold are a degenerated case of intersecting switching manifolds. Thus, optimality conditions for this case are also provided in Sec. 3.3, though the conditions are not explicitly derived and stated.

The proof of the two novel versions of the HMP is based on single needle variations. The technique was originally introduced by de la Barriere [46] and similarly used in [51] for optimal control of nonlinear systems. The technique was extended by Zabczyk [188] to nonlinear impulsive systems. Shaikh and Caines [157] applied the technique to a less general class of hybrid systems, which neither considers time-varying functions and jumps of the continuous state at autonomous and controlled switches nor intersections of or corners in switching manifolds. In the following, the HMP derived in [157] is shown exemplarily as the state of the art for hybrid systems with autonomous and controlled switching:

Theorem 3.1 ([157]): Consider a hybrid system $\mathbb{H}$ as in Definition 2.1 without resets, explicitly time-varying dynamics, and state constraints. Further, consider HOCPs as in Definition 2.20 without switching costs, explicitly time-dependent functions, and terminal and interior constraints. Let related executions σ fulfill the standing assumptions given in Sec. 2.1 and 2.2. Then all controls υ^* and ω^* (locally) minimizing the cost functional

$$\inf_{\upsilon \in \mathcal{U}, \tau, \omega \in \Omega} J(\upsilon, \tau, \omega) \tag{3.1}$$

lead to an optimal execution $\sigma^* = (\tau^*, \varrho^*, \chi^*, \upsilon^*, \omega^*)$, such that the following conditions are satisfied:

1. The differential equations (2.2) are fulfilled.

2. There exists an optimal, absolutely continuous adjoint process λ^* such that:

$$\dot{\lambda}^*(t) = -\nabla_x H_{q_j^*}^T(x^*(t), \lambda^*(t), u_{q_j^*}^*(t)) \tag{3.2}$$

for a.e. $t \in [t_j, t_{j+1})$, $j \in \{0, ..., N\}$.

The following boundary conditions hold for λ^*:

a) Terminal condition:

$$\lambda^*(t_e) = \nabla_x g^T(x^*(t_e)) \tag{3.3}$$

b) If at time t_j, $j \in \{1, ..., N\}$, an autonomous transition on $m_{q^*_{j-1},q^*_j}(x^*(t^-_j)) = 0$ with $q^*_{j-1}, q^*_j \in \mathcal{Q}$ is triggered, then:

$$\lambda^*(t^-_j) = \lambda^*(t_j) + \nabla_x m^T_{q^*_{j-1},q^*_j}(x^*(t^-_j))\, \pi^*_j \tag{3.4}$$

with constant and optimal multipliers $\pi^*_j \in \mathbb{R}$.

c) If at time t_j, $j \in \{1, ..., N\}$, a controlled transition is triggered by a discrete control $\omega_{q^*_{j-1}} = q^*_j \in \Omega_{q^*_{j-1}}$, then:

$$\lambda^*(t^-_j) = \lambda^*(t_j)\,. \tag{3.5}$$

3. The Hamiltonian has to fulfill the following conditions:

a) If at time t_j, $j \in \{1, ..., N\}$ the system switches controlled from q^*_{j-1} to q^*_j, then:

$$H_{q^*_{j-1}}(x^*(t^-_j), \lambda^*(t^-_j), u^*_{q^*_{j-1}}(t^-_j)) = H_{q^*_j}(x^*(t_j), \lambda^*(t_j), u^*_{q^*_j}(t_j))\,. \tag{3.6}$$

b) The minimization condition with respect to $u^*_{q^*_j}$, $q^*_j \in Q$ is:

$$H_{q^*_j}(x^*(t), \lambda^*(t), u^*_{q^*_j}(t)) \leq H_{q^*_j}(x^*(t), \lambda^*(t), u_v) \tag{3.7}$$

for a.e. $t \in [t_j, t_{j+1})$, $j \in \{0, ..., N\}$, and for every $u_v \in U_{q^*_j}$.

c) The minimization condition with respect to $\omega_{q^*_j} \in \Omega_{q^*_j}$, $q^*_j \in Q$, is $\forall k \in \mathcal{Q}$ with $\Gamma(q^*_j, x(t), \omega_{q^*_j}) = k$ and $\Gamma(k, x(t), \omega_k) = q^*_j$ for $\omega_k \in \Omega_k$:

$$H_{q^*_j}(x^*(t), \lambda^*(t), u^*_{q^*_j}(t)) \leq H_k(x^*(t), \lambda^*(t), u_v) \tag{3.8}$$

for a.e. $t \in [t_j, t_{j+1})$ with $j \in \{0, ..., N\}$ and for every $u_v \in U_k$. □

This version of the HMP in [157] is proven by single needle variations. To apply the technique the HOCP is reformulated into Mayer form with only a terminal cost term. The idea of the technique is to insert a needle variation in the optimal control trajectory at some point of time and to calculate the variation that this needle variation causes in the optimal state trajectory at the final time. The variation of the state at the final time leads to a variation of the optimal costs. Analyzing the variation in the costs at the time of the needle variation in the control, the adjoint differential equations, the terminal and transversality conditions for the adjoint variable valid for autonomous and controlled switching, and the Hamiltonian minimization condition with respect to the continuous control are shown. The Hamiltonian continuity condition for controlled switching is derived by a needle variation in the switching time. Similarly, the Hamiltonian minimization condition with respect to the discrete control is obtained from a needle variation in the sequence of discrete control signals.

Hereafter, the contributions of the thesis concerning the further development of single needle variations are shown. The two novel versions of the HMP modify the adjoint transversality conditions, the Hamiltonian value condition for controlled switching (formerly the Hamiltonian continuity condition), and the Hamiltonian minimization condition with respect to the discrete control. Additionally, a Hamiltonian value condition for autonomous switching is introduced and proven, which was not achieved before with single needle variations.

In the first proposed version of the HMP, the class of hybrid systems used in [157] is extended to autonomous and controlled switching with simultaneous resets of the continuous state and switching costs as well as dynamics, running costs, reset maps, switching costs, and switching manifolds that explicitly depend on time. The discussed single needle variational techniques are applied for the proof of the novel HMP with minor modifications to the extended class of hybrid systems. For example, the geometric argument used at switching manifolds for the propagation of variations is modified. This part follows our publication [211].

For the proof of the second novel version of the HMP, two major extensions of the needle variational technique are required. When optimal trajectories pass through an intersection of switching manifolds, arbitrarily small needle variations in the control lead to state variations that may pass different sequences of discrete states. The first extension is to use a convex combination of all possible variations to derive one set of optimality conditions. Further, the proof of the Hamiltonian value condition for autonomous switching requires a needle variation in the optimal switching time. The variation in the switching time is achieved by a complex construction of needle variations in the control, which is the second extension. A minor modification of the needle variational technique is again another form of the geometric expression for the propagation of variations at switching manifolds. In contrast to before, the novel argument does not need the assumption that the state vector of the difference between the varied and the optimal state trajectory is perpendicular to the optimal state trajectory. This part relies on our papers [207, 208].

3.2 The Minimum Principle for Time-Varying Hybrid Systems with State Switching and Jumps

In the following, a version of the HMP for a general class of HOCPs is proposed that considers explicitly time-dependent functions and controlled switching with resets of the continuous state and switching costs for the first time. Sec. 3.2.1 introduces the novel HMP and its proof. Sec. 3.2.2 gives a short example and Sec. 3.2.3 provides a discussion on the extended HMP.

3.2.1 Hybrid Minimum Principle

Here, hybrid systems as defined in Definition 2.26 and with the Assumption 2.27 are considered. The hybrid systems include continuous and discrete states and controls, continuous and discrete dynamics, switching manifolds, autonomous and controlled switching, resets of the continuous state at switchings, and explicitly time-depending functions. The corresponding HOCP with resets is specified in Definition 2.28 with terminal, switching, and running costs, where the switching and running costs explicitly depend on time.

The novel HMP provides necessary optimality conditions for a (locally) optimal solution minimizing (2.12). The HMP includes the conditions of the former versions [137, 157, 165], but modifies the adjoint transversality and Hamiltonian value conditions for controlled and autonomous transitions, and the Hamiltonian minimization condition with respect to the discrete control. Hereafter, the dependence of functions on variables is partly and the asterisk '*' of the optimal switching times t_j^* is sometimes omitted for a better readability.

Theorem 3.2: Consider a hybrid system $\mathbb{H}$ as in Definition 2.26 and related executions σ fulfilling Assumptions 2.27, 2.11, and 2.12. Then all controls υ^* and ω^* and switching times τ^* (locally) minimizing the cost functional

$$\inf_{\upsilon\in\mathcal{U},\tau,\omega\in\Omega} J(\upsilon,\tau,\omega) \tag{3.9}$$

lead to an optimal execution $\sigma^* = (\tau^*, \varrho^*, \chi^*, \upsilon^*, \omega^*)$, such that the following conditions are satisfied:

1) The differential equations (2.2) are fulfilled.

2) There exists an optimal, absolutely continuous adjoint process λ^* such that:

$$\dot{\lambda}^* = -\nabla_x H_{q_j^*}^T(t) \;\text{ a.e. } t\in[t_j,t_{j+1}),\; j\in\{0,...,N\} \tag{3.10}$$

The following boundary conditions hold for λ^*:

a) Terminal condition:

$$\lambda^*(t_e) = \nabla_x g^T(x^*(t_e)) \tag{3.11}$$

b) If at time t_j, $j\in\{1,...,N\}$, an autonomous transition on $m_{q_{j-1}^*,q_j^*}(x^*(t_j^-),t_j) = 0$ with $q_{j-1}^*, q_j^* \in \mathcal{Q}$ is triggered, then:

$$\lambda^*(t_j^-) = \nabla_x\varphi_{q_{j-1}^*,q_j^*}^T(x^*(t_j^-),t_j)\,\lambda^*(t_j) + \nabla_x m_{q_{j-1}^*,q_j^*}^T(x^*(t_j^-),t_j)\,\pi_j^* \tag{3.12}$$

with constant and optimal multipliers $\pi_j^* \in \mathbb{R}$.

c) If at time t_j, $j\in\{1,...,N\}$, a controlled transition to discrete state $q_j^*\in\mathcal{Q}$ is triggered, then:

$$\lambda^*(t_j^-) = \nabla_x\varphi_{q_{j-1}^*,q_j^*}^T(x^*(t_j^-),t_j)\,\lambda^*(t_j)\,. \tag{3.13}$$

3) The Hamiltonian has to fulfill the following conditions:

a) If at time t_j, $j\in\{1,...,N\}$ the system switches autonomously from q_{j-1}^* to q_j^*, then:

$$H_{q_{j-1}^*}(t_j^-) = H_{q_j^*}(t_j) - \pi_j^*\nabla_t m_{q_{j-1}^*,q_j^*} - \lambda^{*T}(t_j)\nabla_t\varphi_{q_{j-1}^*,q_j^*}\,. \tag{3.14}$$

b) If at time t_j, $j \in \{1, ..., N\}$ the system switches controlled from q^*_{j-1} to q^*_j, then:

$$H_{q^*_{j-1}}(t_j^-) = H_{q^*_j}(t_j) - \lambda^{*T}(t_j)\nabla_t \varphi_{q^*_{j-1},q^*_j} \,. \tag{3.15}$$

c) The minimization condition with respect to $u^*_{q^*_j}$, $q^*_j \in Q$ is:

$$H_{q^*_j}(x^*(t),\lambda^*(t), u^*_{q^*_j}(t), t) \leq H_{q^*_j}(x^*(t), \lambda^*(t), u_v, t) \tag{3.16}$$

for a.e. $t \in [t_j, t_{j+1})$, $j \in \{0, ..., N\}$, and for every $u_v \in U_{q^*_j}$.

d) The minimization condition with respect to $\omega_{q^*_j} \in \Omega_{q^*_j}$, $q^*_j \in Q$, is $\forall k \in \mathcal{Q}$ with $\Gamma(q^*_j, x(t), \omega_{q^*_j}) = k$ and $\Gamma(k, x(t), \omega_k) = q^*_j$ for $\omega_k \in \Omega_k$:

$$H_{q^*_j}(x^*(t), \lambda^*(t), u^*_{q^*_j}(t), t) \leq H_k(x_k(t), \lambda_k(t), u_v, t) - \lambda^{*T}(t)\nabla_t \varphi_{k,q^*_j}(x_k(t), t) \tag{3.17}$$

for a.e. $t \in [t_j, t_{j+1})$, $j \in \{0, ..., N\}$, for every $u_v \in U_k$, and with $x_k(t) = \varphi_{q^*_j,k}(x^*(t), t)$ and $\lambda_k(t) = \nabla_x \varphi^T_{k,q^*_j}(x_k(t), t)\, \lambda^*(t)$. □

In the following, a sketch of the proof is presented. Since the proof is strongly based on the scheme used in [157], only those parts are shown, where a significant deviation from the mentioned work exists. The proof consists of propagating needle variations in the controls at arbitrary time in the tangent space around an optimal solution to the end time. Suppose an optimal solution passes the discrete state sequence $0, 1, 2, ..., N$, where switching occurs in $t_1, t_2, ..., t_N$. The overall structure of the proof is as follows: (a) To prove the Hamilton minimization condition (3.16), the adjoint differential equation (3.10), and the adjoint boundary condition (3.11) in the N-th discrete state, a needle variation in that state is performed. The derivation can be found in [157]. (b) Now, step-by-step, one has to go back in the discrete state sequence to show the Hamiltonian minimization condition (3.16) and the adjoint differential equation (3.10) in the corresponding discrete states, and the Hamiltonian value condition for autonomous switchings (3.14) and the adjoint transversality conditions between discrete states (3.12) and (3.13). This is proved by propagating small variations in the state trajectory caused by needle variations in the controls in the current discrete state j to the terminal time and state. (c) The Hamiltonian value condition (3.15) for controlled transitions is derived by a needle variation in the switching time, which is also propagated to the final time. (d) The Hamiltonian minimization condition (3.17) with respect to the discrete control ω_j results from applying a controlled needle variation to the discrete control ω_j and propagating the variations in the continuous state trajectory to the final time.

Proof: **1. Propagation of Variations Through Switchings:** Let the sequence $\{\epsilon^i\}_{i=1}^{\infty}$ be monotonically decreasing with $\epsilon^i > 0$ and $\lim_{i\to\infty} \epsilon^i = 0$. Now, it is analyzed how a needle variation in the optimal input trajectory υ^* at the Lebesgue point $t_v \in [t_{j-1}, t_j)$ in discrete state $j-1 < N$ changes the optimal terminal cost $g(x^*(t_e))$. The case that a variation leads to a change in the discrete state sequence is excluded by assumption since the time ϵ^i of the variation can be chosen arbitrarily small. The variation is set up as follows, compare Fig. 3.1:

$$u^i(t) = \begin{cases} u^*(t) & t_0 \leq t < t_v - \epsilon^i \\ u_v & t_v - \epsilon^i \leq t < t_v \\ u^*_{j-1}(t) & t_v \leq t < t_j - \delta^i_j \\ u^*_j(t_j) & t_j - \delta^i_j \leq t < t_j \\ u^*(t) & t_j \leq t \leq t_e \end{cases}, \tag{3.18}$$

where $t_v \in [t_{j-1}, t_j)$, $u_v \in U_{j-1}$, and $u^*_j(t_j)$ is assumed to be a regular Lebesgue point (otherwise use the left-continuous replacement of the measurable function v^*). It is assumed that the perturbed trajectory hits the switching manifold $M_{j-1,j}$ at time $t_j - \delta^i_j$.[1]

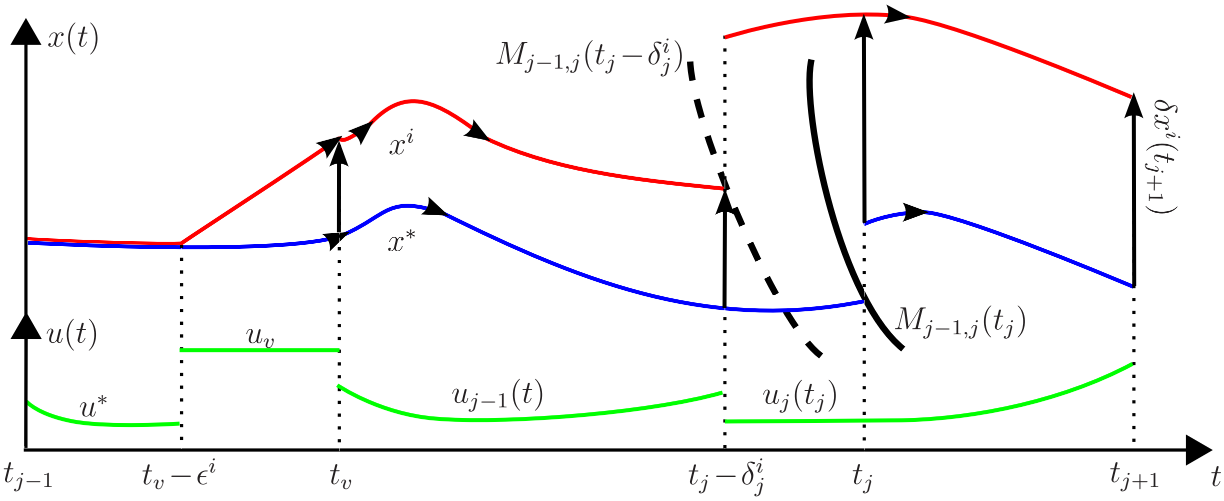


Fig. 3.1: Optimal and perturbed state trajectory with corresponding perturbed continuous control trajectory.

Having defined the perturbed continuous control trajectory u^i, the resulting difference between the perturbed and the optimal trajectory $\delta x^i(t) := x^i(t) - x^*(t)$, $\forall t \in [t_0, t_e]$ and the quantity $\xi(t) := \lim_{i\to\infty} \xi^i(t) := \lim_{i\to\infty} \frac{1}{\epsilon^i}\delta x^i(t)$ are analyzed further. Before applying the variation u_v, the disturbed and the optimal trajectory coincide: $\delta x^i(t) = 0$ $\forall t \in [t_0, t_v - \epsilon^i)$. While the variation u_v of the optimal input $u^*_{j-1}(t)$ is active for the time interval $[t_v - \epsilon^i, t_v)$, the difference between the optimal and the perturbed trajectory at $t = t_v$ is:

$$\delta x^i(t_v) = \int_{t_v - \epsilon^i}^{t_v} f_{j-1}(x^i(t), u_v, t) - f_{j-1}(x^*(t), u^*_{j-1}(t), t)\, dt\,. \tag{3.19}$$

[1] If the perturbed trajectory arrives $M_{j-1,j}$ at time $t_j + \delta^i_j$, the same results follow. This is not shown as it would be a repetition.

Rewriting (3.19), dividing by ϵ^i and forming the limit, the equation below follows:

$$\begin{aligned}\xi(t_v) &= \lim_{i\to\infty} \frac{1}{\epsilon^i} \int_{t_v-\epsilon^i}^{t_v} f_{j-1}(x^*(t), u_v, t) - f_{j-1}(x^*(t), u_{j-1}^*(t), t)\, dt \\ &\quad + \lim_{i\to\infty} \frac{1}{\epsilon^i} \int_{t_v-\epsilon^i}^{t_v} f_{j-1}(x^i(t), u_v, t) - f_{j-1}(x^*(t), u_v, t)\, dt \\ &= f_{j-1}(x^*(t_v), u_v, t_v) - f_{j-1}(x^*(t_v), u_{j-1}^*(t_v), t_v). \end{aligned} \tag{3.20}$$

The second integral in (3.20) is zero as stated in Lemma 2.1 in [157]. The proof of the lemma makes use of an approximation of $\delta x^i(t)$, $t \in [t_v - \epsilon^i, t_v)$, valid for small ϵ^i:

$$\begin{aligned}\frac{d}{dt}\delta x^i(t) &= \nabla_x f_{j-1}(x^*(t), u_{j-1}^*(t), t)\, \delta x^i(t) + \nabla_u f_{j-1}(x^*(t), u_{j-1}^*(t), t)\, \delta u_{j-1}(t) \\ &\quad + o(\epsilon^i) + o(\|\delta u_{j-1}\|) \end{aligned} \tag{3.21}$$

$$\delta x^i(t_v - \epsilon^i) = 0\,, \tag{3.22}$$

where $\delta u_{j-1}(t) := u_v - u_{j-1}^*(t)$. Furthermore, the proof of the lemma as well as the proof here need the concept of a state transition matrix $\Phi_j(\tau, \tau_0)$, that transports small deviations from the optimal trajectory χ^* along its tangent space from time τ_0 to some time $\tau > \tau_0$:

$$\frac{d}{d\tau}\Phi_j(\tau, \tau_0) = \nabla_x f_j(x^*(\tau), u_j^*(\tau), \tau)\, \Phi_j(\tau, \tau_0) \tag{3.23}$$

with the initialization $\Phi_j(\tau_0, \tau_0) = I$.

Now, the variation $\delta x^i(t_v)$ is propagated by the transition matrix until the switching manifold $m_{j-1,j}(x(t), t) = 0$ is reached, what happens at time $t_j - \delta_j^i$:

$$\delta x^i((t_j - \delta_j^i)^-) = \Phi_{j-1}(t_j - \delta_j^i, t_v)\, \delta x^i(t_v) + o(\epsilon^i) \tag{3.24}$$

and the direction of the variation is:

$$\xi(t_j^-) = \lim_{i\to\infty} \frac{1}{\epsilon^i}\delta x^i((t_j - \delta_j^i)^-) = \Phi_{j-1}(t_j, t_v)\, \xi(t_v)\,. \tag{3.25}$$

Here, the varied state $x^i((t_j - \delta_j^i)^-)$ jumps and evolves further with the dynamics in discrete state j and control $u_j^*(t_j)$ until the optimal switching time t_j is reached:

$$x^i(t_j) = \varphi_{j-1,j}(x^i((t_j - \delta_j^i)^-), t_j - \delta_j^i) + \int_{t_j-\delta_j^i}^{t_j} f_j(x^i(t), u_j^*(t_j), t)\, dt\,. \tag{3.26}$$

The optimal continuous state just after the switching time is

$$x^*(t_j) = \varphi_{j-1,j}(x^*(t_j^-), t_j)\,, \tag{3.27}$$

which gives the variation $\delta x^i(t_j) = x^i(t_j) - x^*(t_j)$. By using (3.27) and by observing that

$$x^i((t_j - \delta_j^i)^-) = x^*(t_j^-) + \delta x^i((t_j - \delta_j^i)^-) - \int_{t_j-\delta_j^i}^{t_j} f_{j-1}(x^*(t), u_{j-1}^*(t), t)\, dt + o(\epsilon^i) \tag{3.28}$$

holds, the equivalence

$$\begin{aligned}
&\lim_{i\to\infty} \frac{1}{\epsilon^i} \int_{t_j-\delta_j^i}^{t_j} f_j(x^i(t), u_j^*(t_j), t)\, dt = \\
&\qquad \lim_{i\to\infty} \frac{1}{\epsilon^i} \Big(\int_{t_j-\delta_j^i}^{t_j} f_j(x^i(t), u_j^*(t_j), t) - f_j(\varphi_{j-1,j}(x^i((t_j - \delta_j^i)^-), t_j - \delta_j^i), u_j^*(t_j), t)\, dt \\
&\qquad + \int_{t_j-\delta_j^i}^{t_j} f_j(\varphi_{j-1,j}(x^i((t_j - \delta_j^i)^-), t_j - \delta_j^i), u_j^*(t_j), t)\, dt \Big) \\
&\quad = \lim_{i\to\infty} \frac{1}{\epsilon^i} \int_{t_j-\delta_j^i}^{t_j} f_j(\varphi_{j-1,j}(x^i((t_j - \delta_j^i)^-), t_j - \delta_j^i), u_j^*(t_j), t)\, dt \\
&\quad \overset{(3.28)}{=} \lim_{i\to\infty} \frac{\delta_j^i}{\epsilon^i} f_j(\varphi_{j-1,j}(x^*(t_j^-), t_j), u_j^*(t_j), t_j)
\end{aligned} \tag{3.29}$$

is shown with Lemma 2.1 in [157]. This leads to:

$$\begin{aligned}
\xi(t_j) &= \lim_{i\to\infty} \frac{1}{\epsilon^i} \delta x^i(t_j) \\
&\overset{(3.26),(3.27),(3.29)}{=} \lim_{i\to\infty} \frac{1}{\epsilon^i} \Big(\varphi_{j-1,j}(x^i((t_j - \delta_j^i)^-), t_j - \delta_j^i) - \varphi_{j-1,j}(x^*(t_j^-), t_j) \\
&\qquad + \int_{t_j-\delta_j^i}^{t_j} f_j(\varphi_{j-1,j}(x^i((t_j - \delta_j^i)^-), t_j - \delta_j^i), u_j^*(t_j), t)\, dt \Big) \\
&\overset{(3.27),(3.28),(3.29)}{=} \lim_{i\to\infty} \frac{1}{\epsilon^i} \Big(\nabla_x \varphi_{j-1,j}(x^*(t_j^-), t_j) \big[\delta x^i((t_j - \delta_j^i)^-) - \int_{t_j-\delta_j^i}^{t_j} f_{j-1}(x^*(t), u_{j-1}^*(t), t)\, dt \big] \\
&\qquad + \nabla_t \varphi_{j-1,j}(x^*(t_j^-), t_j)\, (-\delta_j^i) + \delta_j^i\, f_j(x^*(t_j), u_j^*(t_j), t_j) \Big) \\
&\overset{(3.25)}{=} \nabla_x \varphi_{j-1,j}(x^*(t_j^-), t_j)\, \xi(t_j^-) + \lim_{i\to\infty} \frac{\delta_j^i}{\epsilon^i} \Big(f_j(x^*(t_j), u_j^*(t_j), t_j) \\
&\qquad - \nabla_x \varphi_{j-1,j}(x^*(t_j^-), t_j) f_{j-1}(x^*(t_j^-), u_{j-1}^*(t_j), t_j) - \nabla_t \varphi_{j-1,j}(x^*(t_j^-), t_j) \Big)\,.
\end{aligned} \tag{3.30}$$

The time shift δ_j^i in the switching time t_j is found by the following Taylor approximation:

$$\begin{aligned}
m_{j-1,j}(x^i((t_j - \delta_j^i)^-), t_j - \delta_j^i) = {} & m_{j-1,j}(x^*(t_j^-), t_j) + \nabla_t m_{j-1,j}(x^*(t_j^-), t_j)\, (-\delta_j^i) \\
& + \nabla_x m_{j-1,j}(x^*(t_j^-), t_j) \Big(\delta x^i((t_j - \delta_j^i)^-) \\
& - \delta_j^i\, f_{j-1}(x^*(t_j^-), u_{j-1}^*(t_j), t_j) \Big) + o(\epsilon^i)\,,
\end{aligned} \tag{3.31}$$

which leads to

$$\delta_j^i = \eta_j \nabla_x m_{j-1,j}(x^*(t_j^-), t_j)\, \delta x^i((t_j - \delta_j^i)^-) \tag{3.32}$$

with $m_{j-1,j}(x^*(t_j^-), t_j) = m_{j-1,j}(x^i((t_j - \delta_j^i)^-), t_j - \delta_j^i) = 0$, $\eta_j = \frac{1}{\nabla_x m_{j-1,j} f_{j-1}(t_j) + \nabla_t m_{j-1,j}}$, and the short notations $f_{j-1}(t_j) := f_{j-1}(x^*(t_j^-), u_{j-1}^*(t_j), t_j)$ and $m_{j-1,j} := m_{j-1,j}(x^*(t_j^-), t_j)$. Defining $\varphi_{j-1,j} := \varphi_{j-1,j}(x^*(t_j^-), t_j)$ and $f_j(t_j) := f_j(x^*(t_j), u_j^*(t_j), t_j)$ and inserting (3.32) in (3.30), the direction of the variation is finally derived at t_j:

$$\xi(t_j) = \Big(\nabla_x \varphi_{j-1,j} + \eta_j \Big[f_j(t_j) - \nabla_x \varphi_{j-1,j} f_{j-1}(t_j) - \nabla_t \varphi_{j-1,j}\Big] \nabla_x m_{j-1,j}\Big) \xi(t_j -)\,. \tag{3.33}$$

The term $\xi(t_j)$ is transported to the next switching time t_{j+1}:

$$\xi(t_{j+1}^-) = \Phi_j(t_{j+1}, t_j)\, \xi(t_j)\,, \tag{3.34}$$

and further until the final time t_e is reached with the direction of the variation $\xi(t_e)$.

2. Hamiltonian Minimization Condition: Here, for simplicity, the details of the propagation of variations are restricted to the case of one switching time, but the conditions can be extended to multiple switchings straightforwardly. Thus, $t_2 = t_e$ and the single switching time is denoted by t_1. Since χ^* is an optimal trajectory, any perturbed trajectory χ^i in a certain neighborhood has to lead to greater or equal terminal costs $g(x^i(t_2)) \geq g(x^*(t_2))$. Since the terminal costs $g(x(t_2))$ are assumed to be at least continuously differentiable, the operation below is well defined:

$$\begin{aligned} \lim_{i\to\infty} \frac{1}{\epsilon^i} \left[g(x^*(t_2) + \epsilon^i \xi(t_2)) - g(x^*(t_2))\right] &= \nabla_x g(x^*(t_2))\, \xi(t_2) \\ &= \nabla_x g(x^*(t_2))\, \Phi_1(t_2, t_1)\, \xi(t_1) \geq 0\,. \end{aligned} \tag{3.35}$$

Considering (3.33) and (3.35) and setting

$$\lambda^T(t_1) := \nabla_x g(x^*(t_2))\, \Phi_1(t_2, t_1) \tag{3.36}$$

$$\pi_1 := \eta_1 \nabla_x g(x^*(t_2))\, \Phi_1(t_2, t_1) \Big(f_1(t_1) - \nabla_x \varphi_{0,1} f_0(t_1) - \nabla_t \varphi_{0,1}\Big)\,, \tag{3.37}$$

the adjoint transversality condition (3.12) for autonomous switching is found:

$$\lambda(t_1^-) = \nabla_x \varphi_{0,1}^T\, \lambda(t_1) + \nabla_x m_{0,1}^T\, \pi_1\,. \tag{3.38}$$

This condition specializes to (3.13) in case of controlled switching, since the varied trajectory χ^i does not lead to a change δ_1^i in the controlled switching time t_1. Inserting (3.20), (3.25), and (3.33) in (3.35) and defining $\lambda(t_v) = \Phi_0^T(t_1, t_v)\, \lambda(t_1^-)$, the adjoint differential equation (3.10) is derived

$$\begin{aligned} \dot{\lambda}(t_v) &= \frac{d}{dt_v} \Phi_0^T(t_1, t_v)\, \lambda(t_1^-) \\ &= -\nabla_x f_0^T(t_v)\, \Phi_0^T(t_1, t_v)\, \lambda(t_1^-) \end{aligned}$$

$$= -\nabla_x f_0^T(t_v)\,\lambda(t_v) \tag{3.39}$$

and the Hamiltonian minimization condition for $t_v \in [t_0, t_1)$ results:

$$\lambda^T(t_v)\, f_0(x^*(t_v), u_v, t_v) \geq \lambda^T(t_v)\, f_0(x^*(t_v), u_0^*(t_v), t_v)\,. \tag{3.40}$$

3. Hamiltonian Value Condition: In the case of autonomous switching, the Hamiltonian value condition (3.14) can immediately be shown by applying the definitions of η_1, $\lambda(t_1^-)$, and $\lambda(t_1)$ and regrouping terms in (3.37). In the case of controlled switching, an approach is required, which differs from the previous one with a needle variation in the continuous control. Restricting w.l.o.g. the analysis to one controlled switching time in $[t_0, t_e]$, a needle variation in the optimal switching time t_1 is applied, such that the perturbed switching time is $t_1 - \epsilon^i$ and the corresponding continuous control is:

$$u^i(t) = \begin{cases} u_0^*(t) & t_0 \leq t < t_1 - \epsilon^i \\ u_1^*(t_1) & t_1 - \epsilon^i \leq t < t_1 \\ u_1^*(t) & t_1 \leq t \leq t_2 \end{cases}. \tag{3.41}$$

With $x^i((t_1 - \epsilon^i)^-) = x^*((t_1 - \epsilon^i)^-)$, this variation leads to:

$$x^i(t_1) = \varphi_{0,1}(x^i((t_1 - \epsilon^i)^-), t_1 - \epsilon^i) + \int_{t_1-\epsilon^i}^{t_1} f_1(x^i(t), u_1^*(t_1), t)\, dt \tag{3.42}$$

$$x^*(t_1) = \varphi_{0,1}(x^*(t_1^-), t_1)\,. \tag{3.43}$$

Expressing the varied switching point $x^i((t_1 - \epsilon^i)^-)$ by $x^*(t_1^-) - \int_{t_1-\epsilon^i}^{t_1} f_0(x^*(t), u_0^*(t), t)\, dt$ and using again Lemma 2.1 in [157], the direction of the variation $\xi(t_1)$ is obtained:

$$\begin{aligned}\xi(t_1) = & -\nabla_x \varphi_{0,1}(x^*(t_1^-), t_1)\, f_0(x^*(t_1^-), u_0^*(t_1), t_1) \\ & - \nabla_t \varphi_{0,1}(x^*(t_1^-), t_1) + f_1(x^*(t_1), u_1^*(t_1), t_1)\,. \end{aligned} \tag{3.44}$$

This results in the final value $\xi(t_2) = \Phi_1(t_2, t_1)\, \xi(t_1)$. Due to the optimality of χ^*, we again have $g(x^i(t_2)) - g(x^*(t_2)) \geq 0$. Dividing the relation by ϵ^i and passing to the limit, the following holds:

$$\nabla_x g\, \Phi_1(t_2, t_1)\, \nabla_x \varphi_{0,1}\, f_0(t_1) \leq \nabla_x g\, \Phi_1(t_2, t_1) \big(f_1(t_1) - \nabla_t \varphi_{0,1}\big)\,, \tag{3.45}$$

with $\lambda^T(t_1^-) = \nabla_x g\, \Phi_1(t_2, t_1)\, \nabla_x \varphi_{0,1}$ and $\lambda^T(t_1) = \nabla_x g\, \Phi_1(t_2, t_1)$. When the same steps are repeated with a perturbed switching time $t_1 + \epsilon^i$, a similar relation with opposite signs is obtained:

$$\lambda^T(t_1^-)\, f_0(t_1) \geq \lambda^T(t_1)\, f_1(t_1) - \lambda^T(t_1)\, \nabla_t \varphi_{0,1}\,. \tag{3.46}$$

From the two relations, we can conclude the Hamiltonian value condition (3.15) for controlled switching:

$$\lambda^T(t_1^-)\, f_0(t_1) = \lambda^T(t_1)\, f_1(t_1) - \lambda^T(t_1)\, \nabla_t \varphi_{0,1}\,. \tag{3.47}$$

4. Hamiltonian Minimization with Respect to the Discrete Control: To show condition (3.17) in discrete state j, a needle variation in the optimal discrete control ω^* is performed. The needle variation consists of switching from discrete state j to $k \in \mathcal{Q}$ with control $\omega_j \in \Omega_j$ at time $t_k - \epsilon^i$ and back to j at time t_k. This is possible since by assumption there exists $\omega_k \in \Omega_k$, such that $\Gamma(k, x(t), \omega_k) = j$, and $N + 2 \leq \bar{N} < \infty$. Let the varied discrete control sequence be $\omega^i = (..., \omega_k, \omega_j, ...)$ and the continuous control:

$$u^i(t) = \begin{cases} u^*(t) & t_0 \leq t < t_k - \epsilon^i \\ u_v & t_k - \epsilon^i \leq t < t_k \\ u^*(t) & t_k \leq t \leq t_e \end{cases} . \tag{3.48}$$

The varied state at time t_k is:

$$x^i(t_k) = \varphi_{k,j}\Big(\varphi_{j,k}(x^*(t_k - \epsilon^i), t_k - \epsilon^i) + \int_{t_k-\epsilon^i}^{t_k} f_k(x^i(t), u_v, t)\, dt,\; t_k\Big) . \tag{3.49}$$

The optimal state at time t_k can be written as:

$$x^*(t_k) = \varphi_{k,j}\big(\varphi_{j,k}(x^*(t_k - \epsilon^i), t_k - \epsilon^i), t_k - \epsilon^i\big) + \int_{t_k-\epsilon^i}^{t_k} f_j(x^*(t), u_j^*(t), t)\, dt \, . \tag{3.50}$$

Dividing the variation $\delta x^i(t_k) = x^i(t_k) - x^*(t_k)$ by ϵ^i and forming the limit, the direction of the variation is derived:

$$\begin{aligned} \xi(t_k) = & \lim_{i\to\infty} \Big(\nabla_x \varphi_{k,j}(\varphi_{j,k}(x^*(t_k - \epsilon^i), t_k - \epsilon^i), t_k - \epsilon^i) \\ & \frac{1}{\epsilon^i} \int_{t_k-\epsilon^i}^{t_k} f_k(\varphi_{j,k}(x^*(t_k - \epsilon^i), t_k - \epsilon^i), u_v, t)\, dt \\ & + \nabla_t \varphi_{k,j}(\varphi_{j,k}(x^*(t_k - \epsilon^i), t_k - \epsilon^i), t_k - \epsilon^i) - \int_{t_k-\epsilon^i}^{t_k} f_j(x^*(t), u_j^*(t), t)\, dt \Big) \\ = & \nabla_x \varphi_{k,j}(\varphi_{j,k}(x^*(t_k), t_k), t_k)\, f_k(\varphi_{j,k}(x^*(t_k), t_k), u_v, t_k) \\ & + \nabla_t \varphi_{k,j}(\varphi_{j,k}(x^*(t_k), t_k), t_k) - f_j(x^*(t_k), u_j^*(t_k), t_k). \end{aligned} \tag{3.51}$$

Following the derivation in part 2 and 3 before and assuming w.l.o.g. no further switchings, the relation below is achieved:

$$\nabla_x g\, \Phi_0(t_e, t_k) \Big(\nabla_x \varphi_{k,0}\, f_k(\varphi_{0,k}, u_v, t_k) + \nabla_t \varphi_{k,0} \Big) \geq \nabla_x g\, \Phi_0(t_e, t_k)\, f_0(t_k) \, . \tag{3.52}$$

With $\lambda^T(t_k^-) = \nabla_x g\, \Phi_0(t_e, t_k)\, \nabla_x \varphi_{k,0}$ and $\lambda^T(t_k) = \nabla_x g\, \Phi_0(t_e, t_k)$, it leads to the Hamiltonian minimization condition (3.17) with respect to the discrete control. ■

3.2.2 Example: Optimal Acceleration of a Car with Gear Shifting

1. Hybrid Optimal Control Problem: An example with controlled switching and resets is given for illustration. A car with longitudinal dynamics and two gears is modeled, where $q \in \{1, 2\}$ denotes gear, y position, v velocity, and u control and engine torque.

The HOCP in Mayer form consists in finding a solution with at most one gear shift, which minimizes the terminal costs:

$$g(x(t_e)) = \vartheta(t_e) - \gamma y(t_e) \tag{3.53}$$

with the accumulated running costs ϑ and $\gamma \in \mathbb{R}^+$.

The continuous dynamics of the hybrid system is

$$\begin{aligned}
\dot{\vartheta} &= au + (b-q)v & (3.54)\\
\dot{y} &= v & (3.55)\\
\dot{v} &= (b-q)u & (3.56)\\
\dot{\tau} &= 1\,,
\end{aligned}$$

where $u \in U_q$, $a \in \mathbb{R}^+$, $b \in \mathbb{N}$, $b > N_q = 2$, and τ is the physical time added as state variable. A gear shift, which is a controlled switch, is modeled abstractly by a jump in the continuous state $x = (\vartheta, y, v, \tau)^T$ at the switching time t_1:

$$x(t_1) = \varphi_{1,2}(x(t_1^-)) = \begin{pmatrix} \vartheta(t_1^-) + \Delta\vartheta \\ y(t_1^-) + \frac{\epsilon}{d} v(t_1^-) \\ (1-\epsilon)v(t_1^-) \\ \tau(t_1^-) + \Delta\tau \end{pmatrix} \tag{3.57}$$

with $\epsilon := 1 - e^{-d\Delta\tau}$ and $\Delta\vartheta, \Delta\tau, d \in \mathbb{R}^+$. The initial conditions are: $q(t_0) = 1$, $\vartheta(t_0) = y(t_0) = v(t_0) = \tau(t_0) = 0$, and $t_0 = 0$. The final time t_e is set to $1 - \Delta\tau$ in the case of one switch and 1 if no switch occurs, such that $\tau(t_e) = 1$ in both cases.

2. Model Derivation: The terminal costs (3.53) require a combination of minimizing fuel consumption and maximizing the traveled distance. The running costs (3.54) penalize engine torque and driving with high engine speeds, such that driving in a higher gear is more efficient. The possible acceleration in (3.56) decreases with the gear number. A gear shift works as follows: It is assumed that a gear shift takes the time $\Delta\tau$, such that τ jumps at time t_1. In the interval $\Delta\tau$, the dynamics of the car cannot be controlled since the clutch is open. Consequently, only friction with friction coefficient d is acting on the car according to

$$\dot{v} = -dv\,. \tag{3.58}$$

Solving (3.58) analytically, the velocity after the gear shift is obtained: $v(t_1) = e^{-d\Delta\tau}v(t_1^-)$. The parameter $\epsilon := 1 - e^{-d\Delta\tau}$ is the relative decrease in the velocity with a gear shift. Inserting the solution of (3.58) into (3.55), the distance $\frac{\epsilon}{d}v(t_1^-)$, that the car drives in time span $\Delta\tau$, is determined. While shifting gears, the engine is in idle speed and torque, such that a constant increase $\Delta\vartheta$ in the running costs occurs.

3. Solution: In the following, the optimality conditions (2.2), (3.10), (3.11), (3.13), (3.15), and (3.16) are evaluated and solved for the given HOCP. The optimality conditions (3.12) and (3.14) are not considered due to the lack of autonomous switching. The Hamiltonian minimization condition with respect to the discrete control (3.17) is not valid in this example. The proof of (3.17) needs the assumption $x(t) = \varphi_{1,2}(\varphi_{2,1}(x(t)))$, see Assumption 2.27.8, which is not satisfied here. The assumption demands that the continuous states just before and just after switching from one discrete state to another and instantly back again are equal. Considering a positive speed in this example, the state variables cost, distance, and time increase for all possible up and down gear shifts according to (3.57) and the speed always decreases. This means that the continuous state before and after switching to a gear and instantly back are not equal. Thus, the assumption is not met here and the condition (3.17) cannot be applied.

The adjoint conditions (3.10), (3.11), and (3.13) for one switch are:

$$\dot{\lambda}(t) = \begin{pmatrix} 0 & 0 & -(b-q)\lambda_\vartheta(t) - \lambda_y(t) & 0 \end{pmatrix}^T$$
$$\lambda(t_e) = \begin{pmatrix} 1 \\ -\gamma \\ 0 \\ 0 \end{pmatrix}, \quad \lambda(t_1^-) = \begin{pmatrix} \lambda_\vartheta(t_1) \\ \lambda_y(t_1) \\ (1-\epsilon)\lambda_v(t_1) + \frac{\epsilon}{d}\lambda_y(t_1) \\ \lambda_\tau(t_1) \end{pmatrix}$$

with $\lambda = (\lambda_\vartheta \; \lambda_y \; \lambda_v \; \lambda_\tau)^T$. Evaluating the conditions, one obtains for all $t \in [t_0, t_e]$ that $\lambda_\vartheta(t) = 1$, $\lambda_y(t) = -\gamma$, $\lambda_\tau(t) = 0$, and $\lambda_v(t) = \lambda_v(t_s) + (b-q-\gamma)(t_s - t)$ with $t_s = t_e$ if $q = 2$ and $t_s = t_1^-$ if $q = 1$. Since the control $u(t)$ enters the Hamiltonian

$$H_q = \lambda_\vartheta(au + (b-q)v) + \lambda_y v + \lambda_v(b-q)u + \lambda_\tau$$

only linearly, the minimum of $H_q(t)$ is obtained if the control $u(t)$ takes values on the boundaries of the admissible sets $U_q = [0,1]$, $q \in \{1,2\}$, compare (3.16):

$$\begin{aligned} u(t) &= 0 \qquad \text{for} \quad \lambda_v(t) \geq -\frac{a}{b-q} \\ u(t) &= 1 \qquad \text{for} \quad \lambda_v(t) < -\frac{a}{b-q}\,. \end{aligned}$$

Assuming constant controls u_q, $q \in \{1,2\}$, the state trajectories y and v are given by:

$$\begin{aligned} y(t) &= y(t_r) + \big(v(t_r) - t_r(b-q)u_q\big)(t - t_r) + \frac{1}{2}(b-q)(t^2 - t_r^2)u_q \\ v(t) &= v(t_r) + (b-q)(t - t_r)u_q \end{aligned}$$

with $t_r = t_1$ if $q = 2$ and $t_r = t_0$ if $q = 1$. Finally, the Hamiltonian value condition $H_1(t_1^-) = H_2(t_1)$ from (3.15) for switching from $q = 1$ to $q = 2$ is solved numerically for the switching time t_1. With $a = \Delta\tau = \Delta\vartheta = 0.1$, $d = 1$, $b = 5$, and $\gamma = 3$, the optimal switching time $t_1 = 0.26$ and the optimal costs $g(x(t_e)) = -0.0857$ are obtained with $u_1(t) = 1$ for $t \in [0, 0.26]$ and $u_2(t) = 0$ for $t \in [0.26, 0.9]$, see also Fig. 3.2. It can be verified that the discrete state sequence $\varrho = (1,2)$ is optimal, when comparing with the optimal costs $g(x(t_e)) = 0 > -0.0857$ in the case that the discrete state q is 1 in the entire time interval $[0,1]$. In this case, the optimal control u_1 is zero for $t \in [0,1]$.

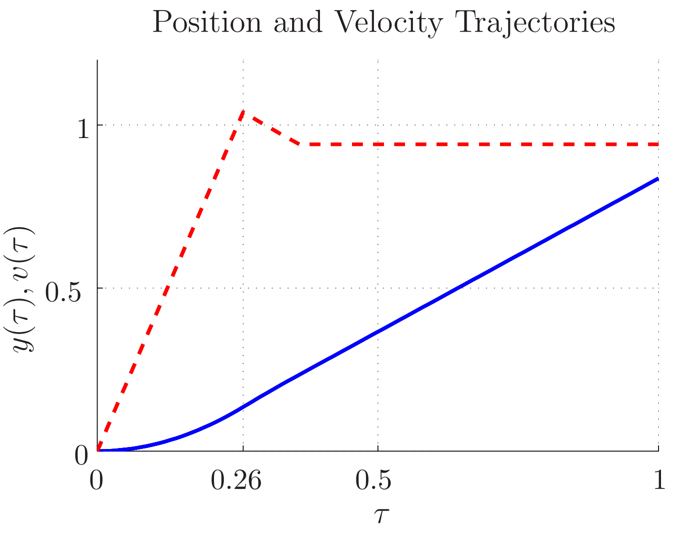


Fig. 3.2: Optimal position $y(\tau)$ (blue, solid) and velocity $v(\tau)$ (red, dashed) over time τ.

3.2.3 Discussion

The section introduces an HMP for time-varying hybrid systems with jumps of the continuous state at autonomous and controlled switchings. The novel version of the HMP enables us to find optimal controls with indirect methods for robotic scenarios in dynamic environments with impacts, where the velocity jumps and the impulse of the system remains continuous. Considering the number of hybrid effects, this version is the most general one in the literature. The versions presented in e.g. [137, 157, 165] are derived for subclasses of the class of hybrid systems used here. The novel version of the HMP includes the conditions of the former versions and additional conditions for controlled switching with resets of the continuous state and functions that are explicitly time-varying. Before this version of the HMP, an explicit dependence of functions on time could only be handled when time was considered to be a state variable. This is only possible if the corresponding functions are at least once continuously differentiable with respect to time. For the novel HMP, it is sufficient if the functions are continuous with respect to time. Further, it is usually not desired for an algorithmic solution to enlarge the state space by adding time as a further state variable.

When using the presented HMP, it is important to consider the assumptions under which the necessary optimality conditions are derived. A critical assumption is that the reset function for the switch from one discrete state to a second one is the inverse of the reset function for the switch from the second discrete state to the first one. The assumption is required to prove the Hamiltonian minimization condition with respect to the discrete control. If the assumption is not fulfilled as in the example in Sec. 3.2.2, the Hamiltonian minimization condition with respect to the discrete control is not valid. The other optimality conditions, however, are not influenced by this assumption. Some assumptions can be further relaxed. For example, the proof of the HMP also holds if all functions are only at least once continuously differentiable with respect to the state and control in a neighborhood around the optimal solution. In contrast, the standing assumptions here demand that functions are differentiable in the entire defined state and control spaces.

In the future, indirect optimal control algorithms based on the novel HMP can be developed. In principle, it is possible to use the basic indirect multiple shooting [136] or indirect collocation methods [157] with some adaptations in conjunction with the novel HMP. These optimal control algorithms find a locally optimal control for a fixed sequence of discrete states. Here, the problem may arise that no local optimum satisfying the HMP exists for the desired sequence of discrete states. In this case, the search for an optimal solution has to be changed to another sequence of discrete states. Alternatively, the optimality conditions have to be modified in the sense of a hybrid necessary principle [56], such that the best solution in a desired sequence of discrete states is found, if such a solution exists.

In general, if the optimal discrete state sequence is unknown, the optimization has to be repeated for each possible sequence of discrete states. This leads to a combinatoric complexity like in the optimal control algorithms for HOCPs with controlled and autonomous switching in [157]. For controlled switching without resets, there exist optimal control algorithms that avoid the combinatorial complexity [18, 136, 155]. A future task is to extend these concepts to controlled switching with resets and the corresponding optimality conditions in the novel HMP.

In the next section, a version of the HMP is proposed that serves as a basis for developing algorithms that avoid the combinatorial complexity to find an optimal discrete state sequence. The novel version is derived for HOCPs with partitioned continuous state space and autonomous switching.

3.3 The Minimum Principle for Hybrid Systems with Partitioned State Space and Unspecified Discrete State Sequence

Though the novel HMP presented in Sec. 3.2 is valid for a general class of HOCPs, it does not consider intersections of switching manifolds. In this section, a version of the HMP is introduced that provides optimality conditions for the case that an optimal trajectory passes an intersection of switching manifolds. The novel HMP is formulated for HOCPs with partitioned continuous state space and autonomous switching without switching costs and resets. Sec. 3.3.1 introduces some preliminary definitions and results, before the proposed HMP and a sketch of its proof are given in Sec. 3.3.2. Sec. 3.3.3 discusses the results.

3.3.1 Prerequisites

Before stating the HMP for the partitioned state space, some technical details are discussed.

In the following, hybrid systems with partitioned state space as specified in Definition 2.30 and satisfying Assumption 2.31 are analyzed. The hybrid system is set up with continuous and discrete states, continuous controls, continuous and discrete dynamics, and switching manifolds for autonomous switching. The novel HMP is developed for HOCPs with partitioned state space defined in Definition 2.32. The costs to be minimized for these HOCPs consist of terminal and running costs.

Definition 3.1: Suppose $\chi^* = \chi^*(\upsilon^*, x_0)$ to be an optimal trajectory and $x^*(t_j) \in M_{q_{j-1},q_j}$ to be an optimal switching state belonging to the closed region $\mathcal{R}_{q_{j-1}} = \mathcal{R}^0_{q_{j-1}} \cap \mathcal{R}^e_{q_{j-1}}$. Here, $\mathcal{R}^0_{q_{j-1}}$ locally denotes the maximally reachable region on the switching manifold M_{q_{j-1},q_j} for the j-th switch starting the system executions from $x(t_0)$ with all possible control trajectories $\upsilon \in \mathcal{U}$. Analogously, $\mathcal{R}^e_{q_{j-1}}$ is the region on M_{q_{j-1},q_j} that locally contains all states from which at least one control trajectory $\upsilon \in \mathcal{U}$ exists, such that the optimal terminal state $x^*(t_e)$ is reached. Then, the time interval $I_j = [a_j, b_j]$ defines the time span in which the region $\mathcal{R}_{q_{j-1}}$ can be reached. All time intervals I_j are collected in $\mathcal{I} = \{I_j\}_{j\in\{1,\ldots,N\}}$. □

Definition 3.2: Direct neighbors of the discrete state $i \in \mathcal{Q}$ are states $d_1, \ldots, d_{n_d} \in \mathcal{Q}$ whose state spaces $\mathcal{X}_{d_1}, \ldots, \mathcal{X}_{d_{n_d}}$ have a common boundary with $\mathcal{X}_i$. Switching manifolds $M_{i,d_1}, \ldots, M_{i,d_{n_d}}$ denoting the common boundaries are called *first-order neighboring switching manifolds*. *Second-order neighboring switching manifolds* M_{d_h,d_l} with $d_h, d_l \in \{d_1, \ldots, d_{n_d}\}$ separate two neighboring discrete states d_h and d_l, which are direct neighbors of state i. □

Definition 3.3: Suppose a trajectory $\chi(\upsilon, x_0)$ subject to the control υ passes through the region $\mathcal{X}_i$ corresponding to the discrete state i. Then those manifolds $M_{i,d_1}, \ldots, M_{i,d_{n_d}}$ with $i, d_1, \ldots, d_{n_d} \in \mathcal{Q}$ which the closure of the intersection of χ with R_i meets which are first-order neighboring switching manifolds are denoted $\mathcal{N}^1_i := \{M_{i,k} | m_{i,k}(x(t)) = 0, i, k \in \mathcal{Q}\}$. The second-order switching manifolds which this closure meets (necessarily in $\partial M_{d_h,d_l} \subset \partial\mathcal{X}_i$ for $d_h, d_l \in \{d_1, \ldots, d_{n_d}\}$) are denoted $\mathcal{N}^2_i$. □

In Fig. 3.3, the set $\mathcal{N}^1_{j-1}$ consists of the manifolds $M_{j-1,j}$ and $M_{j-1,p}$ for the trajectory $\chi^*(\upsilon^*, x_0)$ and the set $\mathcal{N}^2_{j-1}$ contains the manifold $M_{p,j}$.

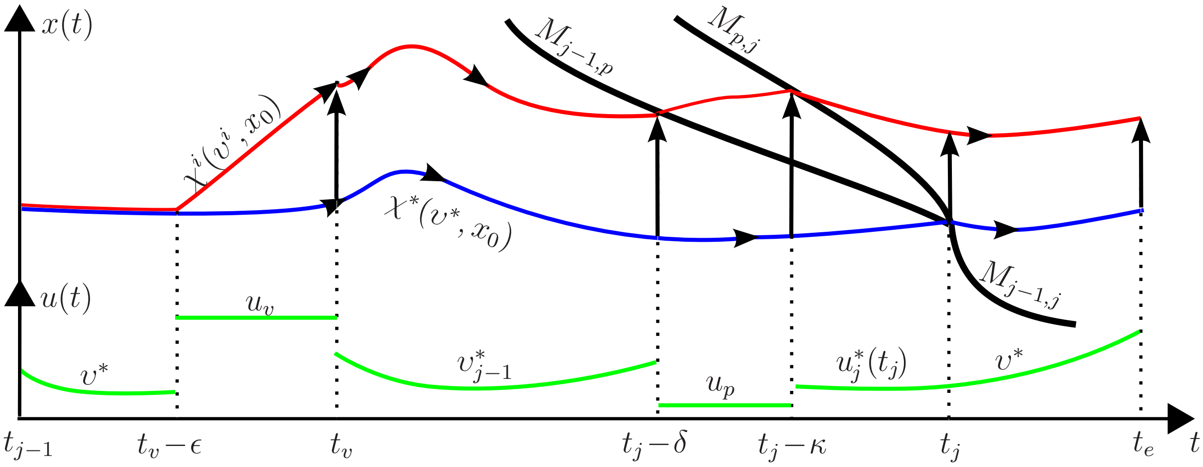


Fig. 3.3: Optimal and perturbed trajectory passing different discrete states.

The HMP classifies a state trajectory to be optimal when all possible variations in the corresponding optimal control lead to neighboring trajectories with higher costs. To prove the extended HMP, the Assumptions 2.31.4, 2.31.5, and 2.31.6 are required to ensure that variations in the optimal control lead to neighboring trajectories independently of the according discrete state sequence. Assumption 2.31.6 guarantees that for neighboring trajectories with different discrete state sequence, the time of going through additional discrete states can be steered to zero, see Fig. 3.3. Thus, the resulting cost functional is continuous across this change in the discrete state sequence:

Lemma 3.3: Assume a hybrid system and HOCP as specified in Definitions 2.30 and 2.32 and the Assumptions 2.31, 2.11, and 2.12 to hold. Then the cost functional J is continuous across the whole state space $\mathcal{X}$ for neighboring trajectories originating from $x(t_0)$ independently of the associated discrete state sequence. □

Proof: The proof has two parts: First, the continuity of J for neighboring trajectories within the same discrete state sequence follows directly from the absolute continuity of $\chi(\upsilon, x_0)$ in a discrete state, its continuity across switches $x(t_j^-) = x(t_j)$, and the continuity and differentiability of g, ϕ_q, f_q, and $m_{i,j}$ with $q, i, j \in \mathcal{Q}$. The proof of this part is not covered here.

Secondly, the continuity of J with respect to neighboring trajectories $\chi(\upsilon, x_0)$ with different discrete state sequence is shown. Consider a control trajectory $\upsilon_{j-1,-,j} \in \mathcal{U}$, which leads to a state trajectory $\chi(\upsilon_{j-1,-,j}, x_0)$ starting from x_0 with discrete state sequence $\varrho = (0, ..., j-1, j, ..., N)$. Let the control $\upsilon_{j-1,-,j}$ be such that the trajectory $\chi(\upsilon_{j-1,-,j}, x_0)$ passes at time t_j a point x_s, which lies on the intersection of switching manifolds $\partial M_{j-1,j} \cap \partial M_{j-1,p} \cap \partial M_{p,j}$.

The control trajectories $\upsilon^i_{j-1,p,j} \in \mathcal{U}$ are variations of the control $\upsilon_{j-1,-,j}$ and fulfill the condition $\lim_{i\to\infty} \|u_{j-1,-,j}(t) - u^i_{j-1,p,j}(t)\| = 0$ for a.e. $t \in [t_0, t^i_\gamma] \cup [t^i_\eta, t_e]$, where $t^i_\gamma = \min(t_j, t^i_{p1})$ and $t^i_\eta = \max(t_j, t^i_{p2})$. The controls $\upsilon^i_{j-1,p,j}$ lead to state trajectories $\chi^i(\upsilon^i_{j-1,p,j}, x_0)$ with the discrete state sequence $\varrho^i = (0, ..., j-1, p, j, ..., N)$. At the switching times t^i_{p1} and t^i_{p2}, the trajectories $\chi^i(\upsilon^i_{j-1,p,j}, x_0)$ meet the switching manifolds $M_{j-1,p}$ and $M_{p,j}$, respectively. The switching points are denoted $x^i_{j-1,p}$ and $x^i_{p,j}$, respectively. In between t^i_{p1} and t^i_{p2}, where the trajectory $\chi^i(\upsilon^i_{j-1,p,j}, x_0)$ is in discrete state p, the control $\upsilon^i_{j-1,p,j}$ is set to the constant value $u_p \in U_p$, such that $f_p(x^i_{j-1,p}, u_p)$ is transversal to $M_{j-1,p}$ and $M_{p,j}$ in direction to $\mathcal{X}_j$. The control u_p exists for a sufficiently small neighborhood around x_s due to Assumptions 2.31.6 and 2.11.

Next, the convergence of the trajectory $\chi^i(\upsilon^i_{j-1,p,j}, x_0)$ to $\chi(\upsilon_{j-1,-,j}, x_0)$ is shown. Since $\lim_{i\to\infty} \|u_{j-1,-,j}(t) - u^i_{j-1,p,j}(t)\| = 0$ is valid for a.e. $t \in [t_0, t^i_\gamma]$, the convergence $\lim_{i\to\infty} \|x(t) - x^i(t)\| = 0$ can be concluded in the same time interval. This implies that $x^i_{j-1,p}$ and t^i_{p1} converge to x_s and t_j. The switching point $x^i_{p,j}$ is given by $x^i_{p,j} := x^i_{j-1,p} + \int_{t^i_{p1}}^{t^i_{p2}} f_p(x^i(t), u_p)dt$. Due to $\lim_{i\to\infty} \|x_s - x^i_{j-1,p}\| = 0$, $\lim_{i\to\infty} |t_j - t^i_{p1}| = 0$, and the direction of $f_p(x_s, u_p)$, it follows that $\lim_{i\to\infty} \|x_s - x^i_{p,j}\| = 0$ and $\lim_{i\to\infty} |t_j - t^i_{p2}| = 0$. The remaining part of $\chi^i(\upsilon^i_{j-1,p,j}, x_0)$ also approaches $\chi(\upsilon_{j-1,-,j}, x_0)$. Thus, the convergence $\lim_{i\to\infty} \|x(t) - x^i(t)\| = 0$ follows $\forall t \in [t_0, t_e]$.

As $\lim_{i\to\infty} |t^i_{p1} - t^i_{p2}| = 0$ and $\phi_p(x(t), u_p)$ is bounded, the costs J^i_p in discrete state p fulfill $\lim_{i\to\infty} J^i_p = \lim_{i\to\infty} \int_{t^i_{p1}}^{t^i_{p2}} \phi_p(x(t), u_p)dt = 0$. This leads to the convergence of the costs $\lim_{i\to\infty} |J_{j-1,-,j} - J^i_{j-1,p,j}| = 0$, where $J^i_{j-1,p,j}$ is associated with $\upsilon^i_{j-1,p,j}$ and $J_{j-1,-,j}$ with $\upsilon_{j-1,-,j}$. The result also holds for more switching manifolds intersecting in one point. ■

3.3.2 Hybrid Minimum Principle

This section provides a novel version of the HMP with necessary optimality conditions for a (locally) optimal solution minimizing (2.14). The proposed HMP includes the conditions of the former versions, but modifies the adjoint transversality conditions at transitions in the case of passing through an intersection of switching manifolds. Consequently, the novel HMP determines gradients of the cost in intersections of switching manifolds, and therefore the search for an optimal solution is not anymore bounded to a fixed sequence of discrete states. In the following, the dependence of variables on time is often omitted for a shorter representation.

Remark 3.4: Hereafter, the HOCP, Definition 2.30, in Bolza representation is reformulated without loss of generality into Mayer form [19], as this is advantageous for the proof of the main result. □

Theorem 3.4: Given a hybrid system $\mathbb{H}$ according to Definition 2.30 and a hybrid system execution σ fulfilling the Assumptions 2.31, 2.11, and 2.12, then all controls υ^* (locally) minimizing the cost functional:

$$\inf_{\upsilon \in \mathcal{U}} J(\upsilon) \tag{3.59}$$

lead to an optimal execution $\sigma^* = (\tau^*, \varrho^*, \chi^*, \upsilon^*)$, such that the following conditions are satisfied:

1. The differential equations (2.2) are satisfied.

2. There exists an optimal, absolutely continuous adjoint process λ^* such that:

$$\dot{\lambda}^* = -\nabla_x H_{q_j^*}^T \quad \text{a.e. } t \in [t_j, t_{j+1}),\ j \in \{0, ..., N\}. \tag{3.60}$$

   The following boundary conditions hold for λ^*:

   a) Terminal condition:

$$\lambda^*(t_e) = \nabla_x g^T(x(t_e)) \tag{3.61}$$

   b) If an autonomous transition with $M_{i,k} \in \mathcal{N}_i^1$ and $M_{k,l} \in \mathcal{N}_i^2$ for $i, k, l \in \mathcal{Q}$ is triggered at time t_j, $j \in \{1, ..., N\}$, then:

$$\lambda^*(t_j^-) = \lambda^*(t_j) + \sum_{M_{i,k} \in \mathcal{N}_i^1} \nabla_x m_{i,k}^T(x(t_j))\, \pi_{j,(i,k)}^* + \sum_{M_{k,l} \in \mathcal{N}_i^2} \nabla_x m_{k,l}^T(x(t_j))\, \pi_{j,(k,l)}^* \tag{3.62}$$

   with constant and optimal Lagrange multipliers $\pi_{j,(i,k)}^*, \pi_{j,(k,l)}^* \in \mathbb{R}$ for $i, k, l \in \mathcal{Q}$.

3. The Hamiltonian has to fulfill the following conditions:

a) If at time t_j, $j \in \{1, ...N\}$ the system switches autonomously from q^*_{j-1} to q^*_j on $M_{q^*_{j-1}, q^*_j} \in \mathcal{N}^1_{q^*_{j-1}}$ for $q^*_{j-1}, q^*_j \in \mathcal{Q}$, then:

$$\begin{aligned} H_{q^*_{j-1}}(t_j^-) &= H_{q^*_j}(t_j) \quad \text{if} \quad t_j \in (a_j, b_j) \\ H_{q^*_{j-1}}(t_j^-) &\geq H_{q^*_j}(t_j) \quad \text{if} \quad t_j = a_j \\ H_{q^*_{j-1}}(t_j^-) &\leq H_{q^*_j}(t_j) \quad \text{if} \quad t_j = b_j \ . \end{aligned} \tag{3.63}$$

b) The minimization condition with respect to $u^*_{q^*}$ for $q^* \in Q$ is:

$$H_{q^*_j}(x^*, \lambda^*, u^*_{q^*_j}) \leq H_{q^*_j}(x^*, \lambda^*, u_v) \tag{3.64}$$

for a.e. $t \in [t_j, t_{j+1})$, for $j \in \{0, ...N\}$, and for every $u_v \in U_{q^*_j}$. □

In the following a sketch of the proof is presented. Since the proof is strongly based on results given in [157], only those parts are shown, where a significant deviation from the mentioned work exists. Here, the same technique is used consisting of propagating needle variations in the controls at arbitrary time in the tangent space around an optimal solution to the end time, see Fig. 3.3. Suppose an optimal solution passes the discrete state sequence $0, 1, 2, ..., N$, where switches happen at $t_1, t_2, ..., t_N$. The overall structure of the proof is as follows: (a) To prove the Hamilton minimization condition (3.64), the adjoint differential equation (3.60), and the adjoint boundary condition (3.61) in the N-th discrete state, a needle variation is introduced in this state. The derivation of the conditions is not given here, as it can be found in [157]. (b) Now, step-by-step, one has to go back in the discrete state sequence to show the Hamiltonian minimization condition (3.64) and the adjoint differential equation (3.60) in the corresponding discrete states, and the adjoint transversality condition between discrete states (3.62). This is proved by propagating small variations in the state trajectory caused by needle variations in the inputs in the current discrete state j to the terminal time and state. (c) The Hamiltonian value condition (3.63) for autonomous transitions is derived by a needle variation in the switching time, which is also propagated to the final time. The variation in the switching time is achieved by a set of needle variations in the control input around the optimal switching time. The Hamiltonian value condition can be found in a similar form in [165]. Note that the signs $\leq$ and $\geq$ there are opposite to the signs here, since the Hamiltonian is maximized there and minimized here.

Proof: The proof is structured into the three main parts: propagation of variations, Hamiltonian minimization condition, and Hamiltonian value condition.

1. Propagation of Variations

Needle Variation in the Control Let the sequence $\{\epsilon^i\}_{i=1}^{\infty}$ be monotonically decreasing with $\epsilon^i > 0$ and $\lim_{i \to \infty} \epsilon^i = 0$. Now, it is analyzed how a needle variation in the optimal input trajectory v^* at the Lebesgue point $t_v \in [t_{j-1}, t_j)$ in discrete state $j - 1 < N$ changes the optimal terminal cost $g(x(t_e))$. The case that a variation leads to a change in the discrete state sequence by inserting the discrete state p in-between $j - 1$ and j is analyzed[2]. The variation is set up as follows, compare Fig. 3.3:

$$u^i(t) = \begin{cases} u^*(t) & t_0 \leq t < t_v - \epsilon^i \\ u_v & t_v - \epsilon^i \leq t < t_v \\ u^*_{j-1}(t) & t_v \leq t < t_j - \delta^i_j \\ u_p & t_j - \delta^i_j \leq t < t_j - \kappa^i_j \\ u^*_j(t_j) & t_j - \kappa^i_j \leq t < t_j \\ u^*(t) & t_j \leq t \leq t_e \end{cases} , \tag{3.65}$$

where $t_v \in [t_{j-1}, t_j)$, $u_v \in U_{j-1}$, $u_p \in U_p$ such that $m_{p,j}(x^i(t_j - \kappa^i_j)) = 0$ with $x^i(t_j - \kappa^i_j) = x^i(t_j - \delta^i_j) + \int_{t_j-\delta^i_j}^{t_j-\kappa^i_j} f_p(x^i(t), u_p)dt$, and $u^*_j(t_j)$ is assumed to be a regular Lebesgue point (otherwise use the left-continuous replacement of the measurable function υ^*). The perturbed trajectory hits the switching manifold $M_{j-1,p}$, which may be different compared to the switching manifold crossed by the optimal trajectory, at the switching time $t_j - \delta^i_j$. Here, it is assumed that one additional discrete state is visited and thus there is another switching time $t_j - \kappa^i_j$, where $\delta^i_j > \kappa^i_j > 0$ $\forall i \in \mathbb{Z}^+$.[3]

Variation of the State Having defined the perturbed input trajectory υ^i, the resulting difference between the perturbed and the optimal trajectory $\delta x^i(t) := x^i(t) - x^*(t)$, $\forall t \in [t_0, t_e]$ and the quantity $\xi(t) := \lim_{i\to\infty} \xi^i(t) := \lim_{i\to\infty} \frac{1}{\epsilon^i}\delta x^i(t)$ are further analyzed. Before applying the variation u_v, the disturbed and the optimal trajectory coincide: $\delta x^i(t) = 0$ $\forall t \in [t_0, t_v - \epsilon^i)$. While the variation u_v of the optimal input υ^*_{j-1} is active for the time interval $[t_v - \epsilon^i, t_v)$, the difference between the optimal and the perturbed trajectory at $t = t_v$ is:

$$\begin{aligned} \delta x^i(t_v) &= x^i(t_v - \epsilon^i) + \int_{t_v-\epsilon^i}^{t_v} f_{j-1}(x^i(t), u_v)\, dt - x^*(t_v - \epsilon^i) - \int_{t_v-\epsilon^i}^{t_v} f_{j-1}(x^*(t), u^*_{j-1}(t))\, dt \\ &= \int_{t_v-\epsilon^i}^{t_v} f_{j-1}(x^i(t), u_v) - f_{j-1}(x^*(t), u^*_{j-1}(t))\, dt \end{aligned} \tag{3.66}$$

Rewriting (3.66), dividing by ϵ^i and forming the limit, the equation below follows:

$$\begin{aligned} \xi(t_v) = {} & \lim_{i\to\infty} \frac{1}{\epsilon^i} \int_{t_v-\epsilon^i}^{t_v} f_{j-1}(x^*(t), u_v) - f_{j-1}(x^*(t), u^*_{j-1}(t))\, dt \\ & + \lim_{i\to\infty} \frac{1}{\epsilon^i} \int_{t_v-\epsilon^i}^{t_v} f_{j-1}(x^i(t), u_v) - f_{j-1}(x^*(t), u_v)\, dt \\ = {} & f_{j-1}(x^*(t_v), u_v) - f_{j-1}(x^*(t_v), u^*_{j-1}(t_v)). \end{aligned} \tag{3.67}$$

The second integral in (3.67) is zero as stated in Lemma 2.1 in [157].

[2] The procedure for deriving optimality conditions for the case of skipping one or more discrete states due to the variation corresponds to the procedure shown in the discussed case and is skipped therefore.

[3] The other (meaningful) cases of relations between δ^i_j, κ^i_j, and 0 can be treated similarly. Note that the case of inserting another discrete state also includes the nominal case shown in [157], where only variations of υ^i are considered which cause the trajectory $\chi^i(\upsilon^i, x_0)$ to intersect with the switching manifold $M_{j-1,j}$.

Propagation of State Variations After the time interval with the perturbed control, the optimal control υ^*_{j-1} is applied again. Consequently, the variation $\delta x^i(t)$, $t \in [t_v, t_j - \delta^i_j)$, can be approximated by:

$$\frac{d}{dt}\delta x^i(t) = \nabla_x f_{j-1}(x^*(t), u^*_{j-1}(t))\, \delta x^i(t) + o(\epsilon^i) \tag{3.68}$$

Based on the approximation, a state transition matrix $\Phi_{j-1}(\tau, \tau_0)$ is introduced, that transports small deviations from the optimal trajectory χ^* along its tangent space from time τ_0 to some time $\tau > \tau_0$:

$$\frac{d}{d\tau}\Phi_{j-1}(\tau, \tau_0) = \nabla_x f_{j-1}(x^*(\tau), u^*_{j-1}(\tau))\, \Phi_{j-1}(\tau, \tau_0) \tag{3.69}$$

with the initialization $\Phi_{j-1}(\tau_0, \tau_0) = I$. Now, the variation $\delta x^i(t_v)$ is propagated to the first jump time $t_j - \delta^i_j$ by the transition matrix:

$$\delta x^i(t_j - \delta^i_j) = \Phi_{j-1}(t_j - \delta^i_j, t_v)\, \delta x^i(t_v) + o(\epsilon^i). \tag{3.70}$$

From there, the deviation between the perturbed and the optimal trajectory evolves according to the equations below up to the switching time t_{j+1}:

$$\delta x^i(t_j - \kappa^i_j) = \delta x^i(t_j - \delta^i_j) + \int_{t_j - \delta^i_j}^{t_j - \kappa^i_j} f_p(x^i(t), u_p) - f_{j-1}(x^*(t), u^*_{j-1}(t))\, dt \tag{3.71}$$

$$\delta x^i(t_j) = \delta x^i(t_j - \kappa^i_j) + \int_{t_j - \kappa^i_j}^{t_j} f_j(x^i(t), u^*_j(t_j)) - f_{j-1}(x^*(t), u^*_{j-1}(t))\, dt \tag{3.72}$$

$$\delta x^i(t^-_{j+1}) = \Phi_j(t_{j+1}, t_j)\delta x^i(t_j) + o(\epsilon^i)\,. \tag{3.73}$$

The given deviations allow us to define $\xi^\delta(t^-_j)$, $\xi^\kappa(t^-_j)$ and $\xi(t_j)$ from (3.70), (3.71), and (3.72):

$$\xi^\delta(t^-_j) := \lim_{i\to\infty} \frac{1}{\epsilon^i}\delta x^i(t_j - \delta^i_j) = \Phi_{j-1}(t_j, t_v)\ \left[f_{j-1}(x^*(t_v), u_v) - f_{j-1}(x^*(t_v), u^*_{j-1}(t_v))\right] \tag{3.74}$$

$$\xi^\kappa(t^-_j) := \lim_{i\to\infty} \frac{1}{\epsilon^i}\delta x^i(t_j - \kappa^i_j) = \xi^\delta(t^-_j) + \lim_{i\to\infty} \frac{1}{\epsilon^i} \int_{t_j - \delta^i_j}^{t_j - \kappa^i_j} f_p(x^i(t), u_p) - f_{j-1}(x^*(t), u^*_{j-1}(t))\, dt \tag{3.75}$$

$$\xi(t_j) := \lim_{i\to\infty} \frac{1}{\epsilon^i}\delta x^i(t_j) = \xi^\kappa(t^-_j) + \lim_{i\to\infty} \frac{1}{\epsilon^i} \int_{t_j - \kappa^i_j}^{t_j} f_j(x^i(t), u^*_j(t_j)) - f_{j-1}(x^*(t), u^*_{j-1}(t))dt. \tag{3.76}$$

The set of points in the intersection of $\partial M_{j-1,j} \cap \partial M_{j-1,p} \cap \partial M_{p,j}$ is called $\mathcal{X}_s$ and one of its elements x_s.

Propagation of Variations through Switchings Two different cases have to be distinguished now:

(i) The optimal switching state $x^*(t_j)$ is on $M_{j-1,j}$ but not in $\mathcal{X}_s$ in $M_{j-1,j}$, which means there exists an arbitrarily small neighborhood $B(x^*(t_j), \rho)$, $\rho < \|x^*(t_j) - x_s\| \; \forall x_s \in \mathcal{X}_s$, around $x^*(t_j)$ on $M_{j-1,j}$ that is not part of $\mathcal{X}_s$. For a sufficiently small ϵ^i, $x^i(t_j - \delta_j^i) \in B(x^*(t_j), \rho)$, such that no change of the discrete state sequence occurs. As this case corresponds to the nominal case without discrete state changes discussed in [157], it need not to be described in further detail here.

(ii) The optimal switching state $x^*(t_j)$ is an element of $\mathcal{X}_s$, such that there exist directions of variation that still switch on $M_{j-1,j}$ corresponding to the nominal case. Additionally, there exist variations that lead to a change in the discrete state sequence for arbitrarily small ϵ^i since $x^i(t_j - \delta_j^i)$ is on $M_{j-1,p}$. This second case is further analyzed:

Clearly, $\delta_j^i = \delta_j^i(\epsilon^i)$ and $\kappa_j^i = \kappa_j^i(\epsilon^i)$ are functions of ϵ^i and due to the differentiability of m and f, the limits of the integrals in (3.75) and (3.76) exist. Forming the sum of the limits of the two integrals in (3.76) with $f_{j-1}^p = f_p(x^*(t_j), u_p) - f_{j-1}(x^*(t_j), u_{j-1}^*(t_j))$ and $f_{j-1}^j = f_j(x^*(t_j), u_j^*(t_j)) - f_{j-1}(x^*(t_j), u_{j-1}^*(t_j))$, they can be rewritten as:

$$\begin{aligned}
\gamma_{j-1,p} + \gamma_{j-1,j} = \; & \lim_{i\to\infty} \frac{1}{\epsilon^i} \left[\int_{t_j-\delta_j^i}^{t_j-\kappa_j^i} f_p(x^i(t), u_p) - f_{j-1}(x^*(t), u_{j-1}^*(t))\, dt \right. \\
& \left. + \int_{t_j-\kappa_j^i}^{t_j} f_j(x^i(t), u_j^*(t_j)) - f_{j-1}(x^*(t), u_{j-1}^*(t))\, dt \right] \\
= \; & \lim_{i\to\infty} \frac{1}{\epsilon^i} \left[(\delta_j^i - \kappa_j^i) f_{j-1}^p + \kappa_j^i f_{j-1}^j + o(\epsilon^i) \right]. \qquad (3.77)
\end{aligned}$$

Next, an expression for the limits $\frac{\delta_j^i}{\epsilon^i}$ and $\frac{\kappa_j^i}{\epsilon^i}$ is derived and illustrated in Fig. 3.4:

$$\begin{aligned}
\lim_{i\to\infty} \frac{\delta_j^i(\epsilon^i)}{\epsilon^i} &= \lim_{i\to\infty} \frac{1}{\epsilon^i} \frac{\|\Delta_{j-1,p}\|}{\|f_{j-1}\|} \\
&= \lim_{i\to\infty} \frac{\|\Delta_{j-1,p}\|}{\|\delta x^i(t_j - \delta_j^i)\|} \frac{\|\delta x^i(t_j - \delta_j^i)\|}{\epsilon^i} \frac{\|\nabla_x m_{j-1,p}\| \, |\cos\vartheta|}{\|f_{j-1}\| \, \|\nabla_x m_{j-1,p}\| \, |\cos\vartheta|} \\
&= \frac{\operatorname{sgn}(\nabla_x m_{j-1,p} \, \xi^\delta(t_j^-))}{|\sin\theta| \, \|f_{j-1}\| \, \|\nabla_x m_{j-1,p}\|} \nabla_x m_{j-1,p} \, \xi^\delta(t_j^-) \\
&= \frac{1}{|\nabla_x m_{j-1,p} \, f_{j-1}|} \nabla_x m_{j-1,p} \, \xi^\delta(t_j^-), \qquad (3.78)
\end{aligned}$$

where $f_{j-1} = f_{j-1}(x^*(t_j), u_{j-1}(t_j^*))$, $m = m(x^*(t_j))$, $\Delta_{j-1,p} = x^*(t_j) - x^*(t_j - \delta_j^i)$, $\cos(\frac{\pi}{2} - \theta) \, \|f_{j-1}\| \, \|\nabla_x m_{j-1,p}\| = \nabla_x m_{j-1,p} \, f_{j-1}$, and $\cos\vartheta \, \|\xi^\delta(t_j^-)\| \, \|\nabla_x m_{j-1,p}\| = \nabla_x m_{j-1,p} \, \xi^\delta(t_j^-)$. The equivalences $\frac{1}{|\sin\theta|} = \frac{\|\Delta_{j-1,p}\|}{\|\delta x^i(t_j - \delta_j^i)\| \, |\cos\vartheta|}$ is derived by a geometric construction using the perpendicular to $M_{j-1,p}$ through $x^*(t_j - \delta_j^i)$. Then, $|\cos\vartheta| > 0$, $|\sin\theta| > 0$, $\nabla_x m_{p,j} \xi(t_j^-) >$

0, $\|f_{j-1}\| > 0$, and $\eta_{j-1,p} > 0$ follows from Assumption 2.11 and similarly results in:

$$\begin{aligned}
\lim_{i\to\infty} \frac{\kappa_j^i(\epsilon^i)}{\epsilon^i} &= \frac{1}{|\nabla_x m_{p,j}\, f_{j-1}|}\, \nabla_x m_{p,j}\, \xi^\kappa(t_j^-) \\
&= \eta_{p,j}\, \nabla_x m_{p,j} \left[\xi^\delta(t_j^-) + \lim_{i\to\infty} \frac{\delta_j^i - \kappa_j^i}{\epsilon^i}\, f_{j-1}^p\right] \\
\Rightarrow \lim_{i\to\infty} \frac{\kappa_j^i(\epsilon^i)}{\epsilon^i} &= \left[\frac{\eta_{p,j}}{1+\eta_{p,j}\nabla_x m_{p,j}\, f_{j-1}^p}\nabla_x m_{p,j} + \frac{\eta_{j,p}\eta_{p,j}\, \nabla_x m_{p,j}\, f_{j-1}^p}{1+\eta_{p,j}\nabla_x m_{p,j}\, f_{j-1}^p}\nabla_x m_{j-1,p}\right]\, \xi^\delta(t_j^-)
\end{aligned} \tag{3.79}$$

with $\eta_{j-1,p} = \frac{1}{\nabla_x m_{j-1,p}\, f_{j-1}}$ and $\eta_{p,j} = \frac{1}{\nabla_x m_{p,j}\, f_{j-1}}$. The equivalence $\frac{1}{|\sin\theta|} = \frac{\|\Delta_{j-1,p}\|}{\|\delta x^i(t_j-\delta_j^i)\|\,|\cos\vartheta|}$ is applied again and similarly $\frac{1}{|\sin\psi|} = \frac{\|\Delta_{p,j}\|}{\|\delta x^i(t_j-\kappa_j^i)\|\,|\cos\varphi|}$ is derived using the perpendicular to $M_{p,j}$ through $x^*(t_j - \kappa_j^i)$. This step is skipped here. Combining (3.79), (3.78), and the right-hand side of (3.77), the equations

$$\gamma_{j-1,p} = \left[\left(\frac{\eta_{j-1,p}}{1+\eta_{p,j}\nabla_x m_{p,j}\, f_{j-1}^p}\nabla_x m_{j-1,p} - \frac{\eta_{p,j}}{1+\eta_{p,j}\nabla_x m_{p,j}\, f_{j-1}^p}\nabla_x m_{p,j}\right)\, \xi^\delta(t_j^-)\right]\, f_{j-1}^p \tag{3.80}$$

$$\gamma_{j-1,j} = \left[\left(\frac{\eta_{j-1,p}\eta_{p,j}\, \nabla_x m_{p,j}\, f_{j-1}^p}{1+\eta_{p,j}\nabla_x m_{p,j}\, f_{j-1}^p}\nabla_x m_{j-1,p} + \frac{\eta_{p,j}}{1+\eta_{p,j}\nabla_x m_{p,j}\, f_{j-1}^p}\nabla_x m_{p,j}\right)\, \xi^\delta(t_j^-)\right]\, f_{j-1}^j \tag{3.81}$$

are obtained.

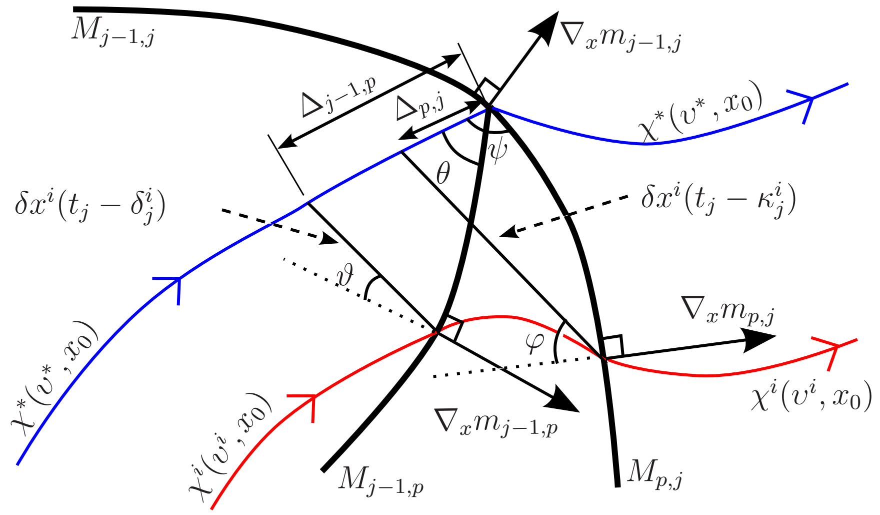

$M_{j-1,j}$
$\nabla_x m_{j-1,j}$
$\Delta_{j-1,p}$
$\Delta_{p,j}$
$\chi^*(\upsilon^*, x_0)$
ψ
θ
$\delta x^i(t_j - \delta_j^i)$
$\delta x^i(t_j - \kappa_j^i)$
ϑ
$\nabla_x m_{p,j}$
φ
$\chi^*(\upsilon^*, x_0)$
$\chi^i(\upsilon^i, x_0)$
$\chi^i(\upsilon^i, x_0)$
$\nabla_x m_{j-1,p}$
$M_{j-1,p}$
$M_{p,j}$

Fig. 3.4: Geometrical relations between the optimal trajectory χ^* passing an intersection $\mathcal{X}_s$ of several switching manifolds and a perturbed trajectory χ^i visiting an additional discrete state p.

Types of Variations As mentioned before, $x^*(t_j) \in \mathcal{X}_s$, which means that there exist variations $\xi_p(t_j) = \xi(t_j^-)+\gamma_{j-1,p}+\gamma_{j-1,j}$ that lead to a change in the discrete state sequence and nominal variations $\xi_n(t_j) = \xi(t_j^-) + \gamma_n$ that maintain the old sequence, where $\gamma_n = \eta_{j-1,j} \nabla_x m_{j-1,j}(x^*(t_j))\, \xi(t_j^-)\, f_{j-1}^j$ and $\eta_{j-1,j} = \frac{1}{\nabla_x m_{j-1,j}\, f_{j-1}}$ (not derived here). A third meaningful case exists, where variations move along the intersection $\partial M_{j-1,j} \cap \partial M_{j-1,p}$. Here, it is possible to show that the limit $\lim_{i\to\infty} \frac{1}{\epsilon^i} \int_{t_j-\delta_j^i}^{t_j} f_{j-1}^j\, dt = \gamma_{con}$ can be expressed by some convex combination of the normal vectors of the switching surfaces $M_{j-1,j}$ and $M_{j-1,p}$:

$$\begin{aligned} \gamma_{con} &= \eta_{con} \left[\nu_1 \nabla_x m_{j-1,j} + \nu_2 \nabla_x m_{j-1,p}\right] \xi(t_j^-)\, f_{j-1}^j \\ &= \left[\eta_{con,1} \nabla_x m_{j-1,j} + \eta_{con,2} \nabla_x m_{j-1,p}\right] \xi(t_j^-)\, f_{j-1}^j \end{aligned} \tag{3.82}$$

with $\sum_{l=1}^2 \nu_l = 1$, $\nu_l \in \mathbb{R}$ and $\nu_l \geq 0$. This gives a further type of variation in the limit $\xi_{con}(t_j) = \xi(t_j^-) + \gamma_{con}$.

2. Hamiltonian Minimization Condition

Variation in Terminal Costs Here, for simplicity, the details of the propagation of variations are restricted to the case of one switching time, but the conditions can be extended to multiple switchings straightforwardly. Thus, $t_2 = t_e$ and the single switching time is denoted by t_1. Since $\chi^*(\upsilon^*, x_0)$ is an optimal trajectory, any perturbed trajectory $\chi^i(\upsilon^i, x_0)$ in a certain neighborhood has to lead to greater or equal terminal costs $g(x^i(t_2)) \geq g(x^*(t_2))$. As shown in Lemma 3.3, the costs are continuous for arbitrary variations and so the terminal condition can be rewritten as:

$$\lim_{i\to\infty} \frac{1}{\epsilon^i} \left[g(x^*(t_2) + \epsilon^i \xi(t_2)) - g(x^*(t_2))\right] = \nabla_x g(x^*(t_2))\, \xi(t_2) \geq 0\,. \tag{3.83}$$

Since (3.83) holds for all possible variations $\xi_p(t_2)$, $\xi_n(t_2)$, and $\xi_{con}(t_2)$, (3.83) is replaced by:

$$\min\left(\nabla_x g(x^*(t_2))\, \xi_p(t_2), \nabla_x g(x^*(t_2))\, \xi_n(t_2), \nabla_x g(x^*(t_2))\, \xi_{con}(t_2)\right) \geq 0\,, \tag{3.84}$$

where $\nabla_x g(x^*(t_2))\, \xi_{con}(t_2) = \min_{\nu_1,\nu_2} \nabla_x g(x^*(t_2))\;\, \xi_{con}(\nu_1,\nu_2,t_2)$. Furthermore, with $\nabla_x g = \nabla_x g(x^*(t_2))$, $\pi_{1,p} := \frac{\eta_{0,p}\, f_0^p + \eta_{0,p}\eta_{p,1}\, \nabla_x m_{p,1}\, f_0^p\, f_0^1}{1+\eta_{p,1}\nabla_x m_{p,1}\, f_0^p}$, and $\pi_{p,2} := \frac{\eta_{p,1}(f_0^1 - f_0^p)}{1+\eta_{p,1}\nabla_x m_{p,1}\, f_0^p}$:

$$\begin{aligned} \nabla_x g\, \xi_p(t_2) &\overset{(3.73)}{=} \nabla_x g\, \Phi_1(t_2,t_1)\, \xi_p(t_1) \\ &\overset{(3.76),(3.80),(3.81)}{=} \Big(\Phi_0^T(t_1,t_v)\Big[\Phi_1^T(t_2,t_1)\nabla_x g^T + \pi_{0,p}\nabla_x m_{0,p}^T + \pi_{p,1}\nabla_x m_{p,1}^T\Big]\Big)^T \xi(t_v) \geq 0 \end{aligned} \tag{3.85}$$

$$\nabla_x g\, \xi_n(t_2) = \Big(\Phi_0^T(t_1,t_v)\Big[\Phi_1^T(t_2,t_1)\nabla_x g^T + \pi_{0,1}\nabla_x m_{0,1}^T\Big]\Big)^T \xi(t_v) \geq 0 \tag{3.86}$$

$$\nabla_x g\, \xi_{con}(t_2) = \Big(\Phi_0^T(t_1,t_v)\Big[\Phi_1^T(t_2,t_1)\nabla_x g^T + \pi_{con,1}\nabla_x m_{0,1}^T + \pi_{con,2}\nabla_x m_{0,p}^T\Big]\Big)^T \xi(t_v) \geq 0\,. \tag{3.87}$$

Backpropagation of Variations in Costs Now, the goal is to derive the optimality conditions (3.60), (3.62), and (3.64) from (3.84). Therefore, the three variations $\xi_p(t_2)$, $\xi_n(t_2)$, and $\xi_{con}(t_2)$ in (3.84) have to be replaced by a single variation $\xi(t_2)$. This is achieved by introducing the convex combination $\xi(t_2) = \nu_1'\xi_p(t_2) + \nu_2'\xi_n(t_2) + \nu_3'\xi_{con}(t_2)$. If the min-condition in (3.84) is satisfied for the convex combination $\xi(t_2)$, then it is also satisfied for (3.84) since $\xi(t_2)$ includes the single variations $\xi_p(t_2)$, $\xi_n(t_2)$, and $\xi_{con}(t_2)$, i.e.

$$\min\left(\nabla_x g\,\xi_p(t_2), \nabla_x g\,\xi_n(t_2), \nabla_x g\,\xi_{con}(t_2)\right) \geq \min_{\nu_1',\nu_2',\nu_3'}\left(\nabla_x g\,\xi(t_2)\right). \tag{3.88}$$

By setting $\lambda_1(t_1) = \Phi_1^T(t_2,t_1)\,\nabla_x g^T(x^*(t_2))$ and using (3.85) - (3.88), (3.84) can be written as:

$$\begin{aligned}
&\min_{\nu_1',\nu_2',\nu_3'}\left(\nu_1'\nabla_x g\,\xi_p(t_2) + \nu_2'\nabla_x g\,\xi_n(t_2) + \nu_3'\nabla_x g\,\xi_{con}(t_2)\right)\\
&\quad= \min_{\nu_1',\nu_2',\nu_3'}\Big(\Big[\Phi_0^T(t_1,t_v)[\lambda_1(t_1) + (\nu_1'\pi_{0,p} + \nu_3'\pi_{con,2})\nabla_x^T m_{0,p}\\
&\qquad+ (\nu_2'\pi_{0,1} + \nu_3'\pi_{con,1})\nabla_x m_{0,1}^T + \nu_1'\pi_{p,1}\nabla_x m_{p,1}^T]\Big]^T \xi(t_v)\Big)\\
&\quad= \Big[\Phi_0^T(t_1,t_v)[\lambda_1(t_1) + \pi_1\nabla_x m_{0,1}^T + \pi_2\nabla_x m_{0,p}^T + \pi_3\nabla_x m_{p,1}^T]\Big]^T \xi(t_v) \geq 0\,.
\end{aligned} \tag{3.89}$$

where $\sum_{l=1}^{3}\nu_l' = 1$, $\nu_l' \in \mathbb{R}$ and $\nu_l' \geq 0$. By defining

$$\lambda_0(t_v) := \Phi_0^T(t_1,t_v)[\lambda_1(t_1) + \pi_1\nabla_x m_{0,1}^T + \pi_2\nabla_x m_{0,p}^T + \pi_3\nabla_x m_{p,1}^T] \tag{3.90}$$

and setting $t_v := t_1^-$ with $\Phi_0(t_1,t_1) = I$, the transversality condition for the adjoint variables at autonomous switchings is obtained:

$$\lambda_0(t_1^-) = \lambda_1(t_1) + \pi_1\nabla_x m_{0,1}^T(x^*(t_1)) + \pi_2\nabla_x m_{0,p}^T(x^*(t_1)) + \pi_3\nabla_x m_{p,1}^T(x^*(t_1))\,. \tag{3.91}$$

The Hamiltonian minimization condition (3.64) can be derived by inserting (3.67) and (3.90) in (3.89):

$$\lambda_0^T(t_v)f_0(x^*(t_v),u_v) \geq \lambda_0^T(t_v)f_0(x^*(t_v),u_0^*(t_v)) \tag{3.92}$$

for a.e. $t_v \in [t_0,t_1)$, for all $u_v \in U_0$, and with $H_0(x,\lambda,u) = \lambda_0^T f_0(x,u)$. Differentiating (3.90) with respect to time and using the definition of the transition matrix (3.69), the adjoint differential equation in discrete state 0 is obtained:

$$\dot{\lambda}_0(t_v) = -\nabla_x f_0^T(x^*(t_v),u_0^*(t_v))\,\lambda_0(t_v) \tag{3.93}$$

for a.e. $t_v \in [t_0,t_1)$. To complete the proof, the Hamiltonian value condition (3.63) is proven in the next part.

3. Hamiltonian Value Condition

Needle Variations In the following, the Hamiltonian value condition (3.63) is derived with needle variational techniques, which was not presented in [157]. Here, the Hamiltonian value condition is shown for the case of intersecting switching manifolds. This includes the nominal case without intersecting switching manifolds as a special case. Deriving the Hamiltonian value property for autonomous transitions requires a slightly different approach than used in part 1 and 2 of this proof. The roles of ϵ^i and κ_j^i are exchanged, such that a desired difference in arrival times κ_j^i on the corresponding switching manifolds between the optimal and the perturbed trajectory is obtained by an adequate variation interval ϵ^i of the continuous input. This means that now: $\epsilon^i = \epsilon^i(\kappa_j^i)$. The second key point in the proof is that the auxiliary geometrical terms $\gamma_{j-1,p}$ and $\gamma_{p,j}$ are written in dependence of $f_{j-1}(x^*(t_j), u_{j-1}^*(t_j))$ instead of $\xi^\delta(t_j^-)$ as before. Let $\{\kappa_j^i\}_{i=1}^\infty$ and $\{\delta_j^i\}_{i=1}^\infty$ with $\delta_j^i > \kappa_j^i > 0\ \forall i$ be two monotonically decreasing sequences and let $\epsilon^i > 0\ \forall i$ and $t_v = t_j - \delta_j^i$:

$$u^i(t) = \begin{cases} u^*(t) & t_0 \le t < t_v - \epsilon^i \\ v_1 & t_v - \epsilon^i \le t < t_v \\ v_2 & t_j - \delta_j^i \le t < t_j - \kappa_j^i \\ u_j^*(t_j) & t_j - \kappa_j^i \le t < t_j \\ v_3 & t_j \le t < t_j + \kappa_j^i \\ v_4 & t_j + \kappa_j^i \le t < t_j + 2\kappa_j^i \\ v_5 & t_j + 2\kappa_j^i \le t < t_j + 3\kappa_j^i \\ u^*(t) & t_j + 3\kappa_j^i \le t \le t_e \end{cases} . \qquad (3.94)$$

Here, $v_1 \in U_{j-1}$, $v_3, v_4, v_5 \in U_j$, and $v_2 \in U_p$ is such that $\lim_{i\to\infty} \frac{\delta_j^i}{\kappa_j^i} = \zeta$ for some feasible $\zeta \in \mathbb{R}$, $\zeta > 1$.

Propagation of Variations Then, the following variations can be specified:

$$\delta x^i(t_v) = \delta x^i(t_j - \delta_j^i) = \int_{t_v-\epsilon^i}^{t_v} f_{j-1}(x^i(t), v_1) - f_{j-1}(x^*(t), u_{j-1}^*(t))\, dt \qquad (3.95)$$

$$\begin{aligned} \delta x^i(t_j + 3\kappa_j^i) &= \delta x^i(t_v) + \int_{t_j-\delta_j^i}^{t_j-\kappa_j^i} f_p(x^i(t), v_2) - f_{j-1}(x^*(t), u_{j-1}^*(t))\, dt \\ &+ \int_{t_j-\kappa_j^i}^{t_j} f_j(x^i(t), u_j^*(t_j)) - f_{j-1}(x^*(t), u_{j-1}^*(t))\, dt + \int_{t_j}^{t_j+\kappa_j^i} f_j(x^i(t), v_3) - f_j(x^*(t), u_j^*(t))\, dt \\ &+ \int_{t_j+\kappa_j^i}^{t_j+2\kappa_j^i} f_j(x^i(t), v_4) - f_j(x^*(t), u_j^*(t))\, dt + \int_{t_j+2\kappa_j^i}^{t_j+3\kappa_j^i} f_j(x^i(t), v_5) - f_j(x^*(t), u_j^*(t))\, dt . \end{aligned} \qquad (3.96)$$

The integrands of the variations above are abbreviated by $f_{j-1}^{v_1}$, $f_{j-1}^{v_2}$, f_{j-1}^{j}, $f_j^{v_3}$, $f_j^{v_4}$, and $f_j^{v_5}$, respectively, where $x^i(t)$ is replaced by $x^*(t)$ according to Lemma 2.1 in [157] as already used in (3.67). Dividing the variation $\delta x^i(t_j + 3\kappa_j^i)$ by κ_j^i and taking the limit, the following is obtained:

$$\begin{aligned}\xi(t_j^+) &:= \lim_{i\to\infty}\frac{1}{\kappa_j^i}\,\delta x^i(t_j+3\kappa_j^i)\\ &= \lim_{i\to\infty}\frac{1}{\kappa_j^i}\left[\epsilon^i f_{j-1}^{v_1}+(\delta_j^i-\kappa_j^i)f_{j-1}^{v_2}+\kappa_j^i f_{j-1}^j+\kappa_j^i f_j^{v_3}+\kappa_j^i f_j^{v_4}+\kappa_j^i f_j^{v_5}+o(\kappa_j^i)\right].\end{aligned} \tag{3.97}$$

Using the relations $\lim_{i\to\infty}\frac{\|\Delta_{\epsilon^i}\|}{\|\delta x^i(t_v)\|}=\frac{\|f_{j-1}\|}{\|f_{j-1}^{v_1}\|}$, $t_v=t_j-\delta_j^i$ and $\lim_{i\to\infty}\frac{\|\delta x^i(t_j-\delta_j^i)\|}{\|\Delta_{\delta_j^i}\|}=\frac{|\sin\theta|}{|\cos\vartheta|}$, the expression

$$\begin{aligned}\lim_{i\to\infty}\frac{\epsilon^i}{\kappa_j^i}=\lim_{i\to\infty}\frac{\zeta\epsilon^i}{\delta_j^i}&=\lim_{i\to\infty}\frac{\zeta\|\Delta_{\epsilon^i}\|\,\|f_{j-1}\|}{\|\Delta_{\delta_j^i}\|\,\|f_{j-1}\|}\\ &=\frac{\zeta\|\nabla_x m_{j-1,p}\|\,\|f_{j-1}\|\,|\sin\theta|}{\|\nabla_x m_{j-1,p}\|\,\|f_{j-1}^{v_1}\|\,|\cos\vartheta|}\\ &=\frac{\zeta}{\nabla_x m_{j-1,p}\,f_{j-1}^{v_1}}\,\nabla_x m_{j-1,p}\,f_{j-1}\end{aligned} \tag{3.98}$$

is derived with $\nabla_x m_{j-1,p}=\nabla_x m_{j-1,p}(x^*(t_j))$ and $f_{j-1}=f_{j-1}(x^*(t_j),u_{j-1}^*(t_j))$. Inserting (3.98) in (3.97), the variation in the limit can be rewritten:

$$\xi(t_j^+)=\frac{\zeta\,f_{j-1}^{v_1}}{\nabla_x m_{j-1,p}\,f_{j-1}^{v_1}}\,\nabla_x m_{j-1,p}\,f_{j-1}+(\zeta-1)f_{j-1}^{v_2}+f_{j-1}^j+f_j^{v_3}+f_j^{v_4}+f_j^{v_5}\,.$$

Variation in Costs By transferring the change of the variation in the limit to the end time (assuming only one switch as before)

$$\xi(t_2)=\Phi_1(t_2,t_1)\xi(t_1^+)\,, \tag{3.99}$$

the difference in the terminal costs between the disturbed and the optimal trajectory can be calculated:

$$g(x^*(t_2)+\kappa_1^i\xi(t_2))-g(x^*(t_2))\geq 0\,, \tag{3.100}$$

which is greater or equal zero since the trajectory χ^* is optimal.

Backpropagation of Variations Dividing by κ_1^i and forming the limit, (3.100) can be transformed into:

$$\begin{aligned}\nabla_x g\,\xi(t_2)&=\nabla_x g\,\Phi_1(t_2,t_1)\left[\frac{\zeta\,f_0^{v_1}}{\nabla_x m_{0,p}\,f_0^{v_1}}\,\nabla_x m_{0,p}\,f_0+(\zeta-1)f_0^{v_2}+f_0^1+f_1^{v_3}+f_1^{v_4}+f_1^{v_5}\right]\\ &=\nabla_x g\,\Phi_1(t_2,t_1)\left[\left(\frac{\zeta\,f_0^{v_1}}{\nabla_x m_{0,p}\,f_0^{v_1}}+\frac{(\zeta-1)f_0^{v_2}+f_1^{v_3}}{\nabla_x m_{0,p}\,f_0}\right)\nabla_x m_{0,p}\,f_0\right.\\ &\qquad\left.+\frac{f_1^{v_4}}{\nabla_x m_{p,1}\,f_0}\nabla_x m_{p,1}\,f_0+\frac{f_1^{v_5}}{\nabla_x m_{0,1}\,f_0}\nabla_x m_{0,1}\,f_0+f_1-f_0\right]\geq 0\,.\end{aligned} \tag{3.101}$$

Setting $\lambda_1^T(t_1) = \nabla_x g(x^*(t))\,\Phi_1(t_2,t_1)$ and separating the terms ending with f_0, or f_1 respectively, we receive

$$\begin{aligned}\lambda_1^T(t_1) f_1 \geq \Bigg[&\lambda_1^T(t_1) - \lambda_1^T(t_1)\left(\frac{\zeta\, f_0^{v_1}}{\nabla_x m_{0,p}\, f_0^{v_1}} + \frac{(\zeta-1) f_0^{v_2} + f_1^{v_3}}{\nabla_x m_{0,p}\, f_0}\right)\nabla_x m_{0,p} \\ &- \lambda_1^T(t_1)\frac{f_1^{v_4}}{\nabla_x m_{p,1}\, f_0}\nabla_x m_{p,1} - \lambda_1^T(t_1)\frac{f_1^{v_5}}{\nabla_x m_{0,1}\, f_0}\nabla_x m_{0,1}\Bigg]\, f_0 \,. \end{aligned} \tag{3.102}$$

The design variables ϵ^i, v_1, and v_2 have already been chosen, such that the desired κ_1^i and ζ are achieved. So, the remaining design variables v_3, v_4, and v_5 are set, such that the three last terms in (3.102) become equal with the multipliers π_1, π_2 and π_3 in (3.89):

$$\pi_1 = -\lambda_1^T(t_1)\,\frac{f_1^{v_5}}{\nabla_x m_{0,1}\, f_0} \tag{3.103}$$

$$\pi_2 = -\lambda_1^T(t_1)\left(\frac{\zeta\, f_0^{v_1}}{\nabla_x m_{0,p}\, f_0^{v_1}} + \frac{(\zeta-1) f_0^{v_2} + f_1^{v_3}}{\nabla_x m_{0,p}\, f_0}\right) \tag{3.104}$$

$$\pi_3 = -\lambda_1^T(t_1)\,\frac{f_1^{v_4}}{\nabla_x m_{p,1}\, f_0}\,, \tag{3.105}$$

which yields with $\lambda_0(t_1^-) = \lambda_1(t_1) + \pi_1 \nabla_x m_{0,1}^T + \pi_2 \nabla_x m_{0,p}^T + \pi_3 \nabla_x m_{p,1}^T$:

$$\lambda_1^T(t_1)\, f_1 \geq \lambda_0^T(t_1^-)\, f_0 \tag{3.106}$$

$$H_1(t_1) \geq H_2(t_1^-)\,. \tag{3.107}$$

The whole derivation can be repeated with delayed switchings $t_j+\delta_j^i$ and $t_j+\kappa_j^i$, $\kappa_j^i > \delta_j^i > 0$, compared to the optimal switching time t_j, which gives:

$$H_1(t_1) \leq H_0(t_1^-)\,. \tag{3.108}$$

Combining (3.107) and (3.108), the equality condition

$$H_1(t_1) = H_0(t_1^-)\,. \tag{3.109}$$

can be shown. In the cases, where only variations κ_j^i exist that lead to either an early or delayed arrival compared to the optimal switching instant t_j since $t_j = b_j$ or $t_j = a_j$, then either only the condition (3.107) or (3.108) can be proven, respectively. Altogether, the Eq. (3.107), (3.108), and (3.109) provide the Hamiltonian value condition (3.63). In cases, where no controls v_3, v_4, and v_5 can be found that satisfy the Eq. (3.103)-(3.105) due to the constrained control space U_1, the lengths of the intervals in which v_3 to v_5 are applied have to be varied in multiples of κ_j^i and possibly ϵ^i and the time points of the variations have also to be varied.

The proof includes the case, where variations κ_j^i lead to perturbed state trajectories that do not cross an additional discrete state p inbetween switching from $j-1$ to j. Furthermore, the proof extends straight forwardly to more intersecting switching manifolds. This concludes the sketch of the proof of Theorem 3.4. ■

Remark 3.5: Consider an optimal autonomous switch from discrete state $j-1$ to j, where the switch does not occur at an intersection of switching manifolds. Then the Hamiltonian value condition (3.63) reduces to the continuity condition of the Hamiltonian value condition. The continuity condition can be directly calculated from the equation for the multiplier

$$\pi = \frac{1}{\nabla_x m_{j-1,j}(x^*(t_j))\, f_{j-1}(x^*(t_j), u^*_{j-1}(t_j))}\, \lambda_j^T(t_j) \Big(f_j(x^*(t_j), u^*_j(t_j)) - f_{j-1}(x^*(t_j), u^*_{j-1}(t_j)) \Big) \tag{3.110}$$

by regrouping terms and applying the definition $\lambda_{j-1}(t_j^-) = \lambda_j(t_j) + \nabla_x m^T_{j-1,j}(x^*(t_j))\, \pi$:

$$\lambda_j^T(t_j)\, f_j(x^*(t_j), u^*_j(t_j)) = \lambda^T_{j-1}(t_j^-)\, f_{j-1}(x^*(t_j), u^*_{j-1}(t_j))\,. \tag{3.111}$$

Since the expression (3.110) for π is also valid on the boundaries of the interval of possible switching times $I_j = [a_j, b_j]$, the Hamiltonian equality condition also holds on the boundaries of I_j. □

Remark 3.6: The proof concepts can be applied in straightforward manner to give optimality conditions for optimal trajectories passing through corner points of piecewise-smooth switching manifolds. The corner case can be extended to a more general class of hybrid systems, e.g. with smooth functions resetting the continuous state at switching times. □

3.3.3 Discussion

The version of the HMP presented here provides necessary optimality conditions for HOCPs with partitioned continuous state space. The adjoint transversality conditions and the Hamiltonian continuity conditions of former versions are modified to a different adjoint transversality condition and a Hamiltonian value condition in the case that the continuous state trajectory passes an intersection of switching manifolds.

For the proof of the novel HMP, it is essential that the costs are continuous for neighboring trajectories independent of the discrete state sequence. This is the case for HOCPs with partitioned state space. The continuity of the costs enables us to calculate directional derivatives of the costs for neighboring continuous state trajectories with the same sequence of discrete states. Suppose that the optimal trajectory passes an intersection of switching manifolds, such that neighboring trajectories with different discrete state sequence exist. To derive a Hamiltonian minimization condition and only one set of optimality conditions and not one for each discrete state sequence, the directional derivatives of the different discrete state sequences are combined in a convex derivative. The weights for the convex combination are determined, such that the convex derivative is minimal. The convex combination allows us to introduce a novel HMP. However, the convex combination also inserts some conservativeness. The reason is that there may exist optimal solutions, where the convex combination with the minimal convex derivative does not satisfy the ”necessary” optimality conditions.

The proof of the HMP is based on several assumptions. Especially interesting is Assumption 2.31.6(iv), which serves two purposes:

(i) Consider an optimal trajectory to pass through an intersection of switching manifolds. Then the assumption guarantees that the time that a variation of the optimal trajectory is in an additional discrete state can be steered to zero. This property is important to show that the cost function is continuous, such that the costs of neighboring trajectories in different discrete state sequences can be compared by directional derivatives of the costs.

(ii) The assumption determines which discrete state is the subsequent one after an autonomous switch on an intersection of different switching manifolds. This choice is made to ensure a deterministic behavior of the hybrid system. The assumption can be changed if desired to less strict versions. For example, it can be a priori decided which discrete state is selected subsequently. The risk of making a decision about the subsequent discrete state a priori is to exclude optimal solutions from the analysis. The most general approach appears to consist in considering the discrete control ω as optimization variable whenever the state trajectory meets an intersection of switching manifolds. The discrete control determines the subsequent discrete state. In this case, additional optimality conditions for the optimal choice of the discrete control have to be derived. In our preliminary analysis, these conditions consist of a comparison of integrals of the Hamiltonians over the remaining optimization horizon for the different choices of the discrete control. This condition basically means that the costs of the state trajectories have to be compared for the different possibilities of subsequent discrete state sequences. In this case, the discrete state sequences start with different possible discrete states subsequent to the considered autonomous switching on the intersection of switching manifolds. Future research should be directed to rigorously derive the relevant optimality conditions and to find algorithms for computing the optimal discrete controls. Relaxing the assumption, it is further required that there exist continuous controls such that the time in an additional discrete state can approach zero for arbitrarily small variations of the optimal trajectory.

For future research, a further extension of the derived results is discussed in the following. Consider a hybrid system with non-partitioned continuous state space, where switching manifolds are allowed to intersect. Assume that when a state trajectory passes an intersection of switching manifolds, it is already chosen a priori what the subsequent discrete state is. For this class of HOCPs, it is in general not possible to obtain neighboring continuous state trajectories for different subsequent discrete states in contrast to hybrid systems with partitioned state space. This means that it is not possible as in the just presented approach to find gradients of the costs for the different discrete state sequences and to combine them in a convex combination. The reason is that the costs may jump from one discrete state sequence to the next. Consequently, a Hamiltonian minimization condition with respect to the continuous control can only be derived as our preliminary analysis suggests if an additional condition is considered. The additional condition compares the costs for trajectories that pass an arbitrarily small neighborhood around the intersection of switching manifolds and follow from there different discrete state sequences. A future task is to extend our preliminary studies and rigorously derive optimality conditions for this class of HOCPs.

3.4 Summary

This chapter presents two novel versions of the HMP. The HOCPs considered for the first version consist of autonomous and controlled switching with simultaneous resets of the continuous state, terminal, running, and switching costs, and explicitly time-dependent functions. The proposed version of the HMP provides necessary optimality conditions for the optimal continuous control, the optimal switching times, and the optimal discrete control, which determines the optimal discrete state sequence. The version of the HMP is proven with single needle variations in the continuous and discrete controls and in the controlled and autonomous switching times. In contrast to former versions, the novel HMP can handle controlled switching with resets and explicitly time-dependent functions. Controlled switching with resets occurs, for example, in manufacturing processes and it is used to model uncontrollable, physical behavior during a controlled switch abstractly by a jump in the continuous state. This simplifies the analysis of HOCPs, especially in modeling, simulating, and searching for the optimal control. An example for the abstract consideration of gear shifting is provided. An example for an explicitly time-depending dynamics is a sanding vehicle, where the mass changes with time.

The HOCPs in the second version consider autonomous switching on switching manifolds and switching manifolds are allowed to intersect. In former versions of the HMP, intersections of switching manifolds cannot be handled. Besides the known optimality condition of former versions, the novel HMP gives additional optimality conditions for the case that an optimal trajectory passes an intersection of switching manifolds. The additional conditions are a modified adjoint transversality condition and a Hamiltonian value condition, which are both valid at an autonomous switching on an intersection of switching manifolds. The proof is based on single needle variations in the continuous control and the autonomous switching times. Further, the Hamiltonian continuity condition is shown, which has not been achieved with single needle variations before. Optimal trajectories naturally pass through intersections of switching manifolds e.g. when considering a four-legged, pole climbing robot. In addition, the optimality conditions on intersections of switching manifolds form gradients of the costs with respect to the switching point and time. These gradients can be used for optimal control algorithms that find the optimal discrete state sequence in parallel with the optimal continuous control, see Sec. 5.3.

With the two novel versions of the HMP, the class of HOCPs is extended for which optimality conditions can be provided. As a result, the applicability of indirect methods is increased, as indirect methods use optimality conditions to find optimal controls for HOCPs. In the next chapter, the applicability of indirect methods is further improved by facilitating the initialization and enlarging the domain of convergence.

4 Optimal Control for Hybrid Systems with Fixed Discrete State Sequence

Existing algorithms for optimal control of hybrid systems based on indirect multiple shooting provide solutions with high accuracy. However, in contrast to dynamic programming (DP) and direct methods, the algorithms converge only in a small domain and correspondingly the initialization is difficult. For the solution of hybrid optimal control problems (HOCPs) with a fixed sequence of discrete states, this chapter introduces two approaches that simplify the initialization and show a larger domain of convergence than the existing algorithms. At first, two initialization concepts are presented based on direct methods, as direct methods generally have a large domain of convergence and are initialized more straightforwardly than indirect methods. In the first concept, a direct method solves the HOCP with an accuracy that is sufficient for a successful initialization of the indirect method. The second concept decomposes the hybrid optimal control problem into non-hybrid subproblems, where each subproblem is initialized separately by a direct method. This results in a significantly higher robustness of the initialization compared to the first concept. However, the accuracy of the solution achieved in the first concept is higher. The two concepts are compared based on results from a numerical example.

Secondly, a min-H algorithm for the optimal control of hybrid systems with autonomous switching, control constraints, and interior point constraints is introduced. The algorithm is intuitively initialized and finds optimal solutions with high accuracy. The proposed min-H approach combines the advantages of first-order and second-order indirect gradient methods, namely a large domain of convergence and fast convergence near the optimal solution, respectively. Contrary to previous min-H algorithms, the introduced algorithm is globally convergent to a locally optimal solution and the convergence is proven. Furthermore, the extension of a min-H approach to hybrid optimal control problems is novel. The algorithm finds the optimal continuous control and optimal switching instants for hybrid systems with a fixed sequence of discrete states. A numerical example shows the efficiency of the proposed min-H algorithm.

4.1 Introduction and State of the Art

As discussed in Chap. 1, indirect methods have the advantage of providing results with high accuracy for the solution of HOCPs in contrast to direct methods and DP. However, the applicability of indirect methods is limited in practice due to initialization difficulties and a small domain of convergence. The small domain of convergence compared to direct methods and DP results from the local convergence properties of indirect methods and numerical instabilities. The goal of this chapter is to propose two approaches based on indirect methods that are initialized straightforwardly and have an enlarged domain of convergence. The approaches are designed for HOCPs with autonomous switching and a fixed sequence of discrete states, where the sequence is not subject to optimal control. Algorithms, which optimize the discrete state sequence, are presented in Chap. 5.

In general, indirect methods solve a multi-point boundary value problem (MPBVP) that originates from necessary optimality conditions which an optimal solution has to satisfy. The optimality conditions are derived by calculus of variations in Bryson and Ho [32] or Pontryagin's minimum principle in Pontryagin et al. [132]. For HOCPs, the minimum principle was extended to a hybrid minimum principle (HMP), e.g. by Sussmann [165] and Shaikh and Caines [157]. Common methods to solve the MPBVP for purely continuous optimal control problems are given by gradient methods [32, 65], indirect multiple shooting [22, 25, 35], and indirect collocation [160]. Multiple shooting has been extended for the solution of HOCPs with autonomous switching and a fixed sequence of discrete states in the form of multiple-phase multiple-shooting [25, 136]. Indirect collocation has also been applied to the solution of HOCPs on the lower level of a two level algorithm [157]. Algorithms that use indirect methods for HOCPs with controlled switching are introduced in [50, 155], for hybrid systems with autonomous switching in [214], and for hybrid systems with controlled and autonomous switching in [156, 187].

Especially for indirect multiple shooting, it holds that solutions are obtained with almost arbitrary accuracy and the convergence in a sufficiently small neighborhood around the optimal solution is at least quadratic due to Newton update steps [154, 175]. A disadvantage is the relatively small domain of convergence. Two reasons lead to the small domain of convergence: First, the nonlinear optimization problem that results from the MPBVP is solved by methods based on gradient information, e.g. Newton methods. This limits the convergence to a local region around an optimal solution. Outside the local region, indirect methods converge to local minima, if present, of the nonlinear optimization problem, but the local minima are not necessarily solutions of the MPBVP and thus neither of the HOCP. Second, indirect methods have difficulties with the numerical stability. The reason for indirect multiple shooting is that either the state or adjoint differential equations have to be integrated in their instable direction, which may turn the optimization problem ill-conditioned. For indirect collocation methods, the numerical instabilities are caused by the high sensitivities of the differential equations, such that used Jacobians may become ill-conditioned. Further, the adjoint variables have to be guessed for initialization but are physically not intuitive. Both the small domain of convergence and the non-intuitiveness cause the initialization process to be difficult. Additionally, the sequence of state constrained and unconstrained arcs has, in general, to be known beforehand.

In this chapter, two approaches are introduced for a simplified initialization and a large domain of convergence. At first, two concepts are presented based on two-stage approaches known from the literature. There, a direct method solves the optimal control problem first and the solution is used to initialize an indirect method, which improves the solution. Afterwards, a so called "min-H" algorithm as a special class of gradient methods is developed. The min-H algorithm integrates the state and adjoint differential equations in their stable direction and the Hamiltonian function is minimized directly for updates of the control, which is the optimization variable. The contributions are the following:

(i) In the first presented initialization concept, a direct method solves an HOCP completely and the solution initializes an indirect method. The concept is an extension of known concepts to HOCPs.

(ii) A novel, second initialization concept consists in decomposing the HOCP into purely continuous optimal control subproblems. The subproblems are solved separately: first by a direct method and second by an indirect method, where the indirect method is initialized by the solution of the direct method.

(iii) A min-H algorithm is proposed that is globally convergent to a locally optimal solution and the global convergence is proven. This is in contrast to former min-H algorithms and other indirect methods, which are only locally convergent to a locally optimal solution.

(iv) Limitations due to numerical instabilities are reduced for the proposed min-H algorithm compared to other indirect methods as verified in numerical simulations.

(v) The novel min-H algorithm solves HOCPs with autonomous switching and interior point constraints, whereas former min-H algorithms are limited to purely continuous optimal control problems.

(vi) By a careful design, the linear multi-point boundary value problem is solved non-iteratively with a single forward-backward integration at each iteration of the min-H algorithm in contrast to indirect gradient methods like the one in [108].

(vii) An alternative algorithm to the min-H algorithm is introduced that combines features of first- and second-order gradient methods to a locally convergent method with fast convergence. The algorithm is also formulated for HOCPs.

Below, the two presented approaches are described briefly.

Initialization by a Direct Method

At first, direct methods and their strengths and weaknesses are revisited shortly. Direct methods transform the infinite dimensional problem of finding optimal state and control trajectories into a finite dimensional, nonlinear programming problem by evaluating state and control values only at a finite number of time samples. By varying the state and control values at those samples, the cost criterion is directly optimized which can be performed by nonlinear programming methods like sequential quadratic programming. The nonlinear program is usually formulated as a direct multiple shooting algorithm [23, 26] or as a direct collocation algorithm [174]. Direct methods show several advantages compared to indirect ones. Direct methods are easier to initialize due to a larger domain of convergence and the physically intuitive meaning of the optimization variables. Furthermore, the sequence of constrained and unconstrained arcs does not need to be guessed in advance but evolves during the optimization. Constrained and unconstrained arcs are segments of the state trajectory without active or with the same active state constraints, respectively. However, direct methods deliver less accurate results, primarily due to discretization errors. Additionally, direct methods sometimes converge to local minima, which are introduced by the discretization and which do not match a true locally optimal solution [175].

The idea of combining the advantages of direct and indirect methods has already been realized in [66, 154, 175]. Basically, this is developed for non-hybrid systems. In principle, a direct method is applied to initialize an indirect one and information about the sequence of state constrained and unconstrained arcs is gained from the direct approach. A direct collocation and a direct multiple shooting method are used to determine initial values and the sequence of constraints for an indirect multiple shooting method in [175] and [66]. Estimates of the adjoint variables are obtained directly from the applied numerical optimization algorithms. Furthermore, it is proved in [66] that those estimates converge to the exact optimal adjoint values if the discretization is continuously refined. In [154], the fact is exploited that adjoint variables form the sensitivity of the performance with respect to the initial state. Thus, solving a series of optimization problems with perturbed initial state by direct methods provides the desired initial values for the adjoint variables.

The chapter introduces and discusses two initialization concepts for the indirect solution of HOCPs with autonomous switching. In the first concept, a direct and an indirect multiple shooting method solve the HOCP entirely. First, the direct method is applied and delivers estimates of the state and adjoint variables. As those estimates of the adjoint variables do not match the corresponding values in the indirect method on state constrained arcs, a backward integration scheme is applied to determine a suitable initialization. The first concept suffers from the complexity of the HOCP and the required accuracy of adjoint variables to successfully initialize the indirect method, what can lead to cases of non-successful initialization. Therefore, a second initialization concept is introduced, where the HOCP is decomposed into non-hybrid subproblems. Each of those subproblems is solved separately from the others, at first, by a direct method for non-hybrid systems. Secondly, each subproblem is solved by an indirect method for non-hybrid systems based on an initialization with the solution of the direct method. This simplifies the initialization significantly due to the reduced complexity of solving subproblems instead of the entire HOCP at once. The subproblems are connected by switching points and times, which are varied until the entire HOCP satisfies all necessary optimality conditions. An algorithm using the decomposition technique is presented in Chap. 5 and [207, 214]. We published the two initialization concepts in [210].

Min-H Algorithm

The min-H algorithm belongs to the class of gradient methods, which are a third type of indirect methods and which, in general, were only applied successfully to non-hybrid optimal control problems up to now. Gradient methods are more intuitive to initialize than indirect multiple shooting and indirect collocation since basically control values have to be guessed initially instead of adjoint values. Gradient methods consist of gradient methods of first order, gradient methods of second order, and mixtures of both types [32]. Gradient methods of first order [82] have a large domain of convergence compared to the other indirect methods but converge very slow near the optimum and therefore usually provide solutions with low accuracy for a reasonable number of iterations. Gradient methods of second order [101] only converge in a convex region around a minimum but converge at least quadratically close to the minimum. Second-order gradient methods in general find solutions with an accuracy that is slightly lower than the accuracy of solutions from indirect multiple shooting but considerably higher than the accuracy of solutions from direct methods and dynamic programming. Mixtures of both types of gradient methods try to combine a large domain of convergence with a high convergence speed. For example, algorithms based on gradient methods of first order were developed that accelerate the convergence by using conjugate search directions [93, 107, 108] and weighting matrices [108]. A disadvantage of the approaches of [108] is the computational complexity as a boundary value problem is solved iteratively at each iteration of the method. Built upon gradient methods of second order and using penalty function approaches, fast convergence and a relatively large domain of convergence is achieved in [83]. In [65], a min-H approach is introduced, where the Hamiltonian H is minimized directly, such that a large domain of convergence is combined with a high convergence speed. The disadvantage of the method is that a required assumption about which update direction leads to lower costs is not satisfied by many HOCPs.

The contribution is that a min-H algorithm is developed that is provably globally convergent and converges at least quadratically in a close neighborhood around the minimum. No existing gradient method provides these properties simultaneously. We are able to remove the restrictive assumption from [65] by compensating changes in the state and its adjoint variable (and the switching times in HOCPs), that are caused by updates of the control and could prevent the convergence. We solve the underlying boundary value problem, which is linearized in the tangent space of the current trajectory, by a single sweep similar to Riccati solutions for linear-quadratic optimal control problems [32, 190]. Thus, an iterative solution embedded in the iterative main algorithm is not necessary as in [108]. A further contribution is that we extended our min-H algorithm to the solution of HOCPs for hybrid systems with autonomous switching and a fixed discrete state sequence. To our knowledge, this is the first successful application of gradient methods to HOCPs with autonomous switching. The method can also handle interior point, state, and control constraints. The min-H approach provides highly accurate solutions. If nevertheless refinement of accuracy is still desired, the algorithm can be used to initialize an indirect multiple-phase multiple shooting method.

As alternative to the novel min-H algorithm, an algorithm is introduced that combines first- and second-order gradient methods. The alternative algorithm differs from the min-H algorithm in the choice of the updates for the control. In locally convex regions of the HOCP, Newton update steps are applied for a rapid convergence. In regions of the HOCP that are not locally convex or contain active control constraints, the updates of the control are formed by a steepest descent method. The alternative algorithm is not globally convergent as it is based on local gradients in contrast to the min-H algorithm, which uses a direct minimization of the Hamiltonian for a reference of the optimal control.

4.2 Initialization Concepts for Indirect Multiple Shooting Methods

At first, the two novel initialization concepts for the solution of HOCPs with indirect multiple shooting are presented. The section is structured as follows: In Sec. 4.2.1, the

HOCP is described and optimality conditions are formulated. Sec. 4.2.2 introduces and discusses two initialization concepts for the solution of the HOCP. Sec. 4.2.3 applies both concepts to a numerical example and Sec. 4.2.4 compares the two concepts.

4.2.1 Problem Formulation

The two-stage initialization concepts are designed for hybrid systems with continuous and discrete states, continuous controls, continuous and discrete dynamics, switching manifolds for autonomous switching, and state and control constraints. This class of hybrid systems is described in Definition 2.34 and it satisfies the Assumption 2.35. The corresponding HOCP is formulated in Definition 2.36 and includes terminal and running costs and terminal constraints.

In the following, it is assumed that the discrete state sequence ϱ is fixed and given by $\varrho = (0, 1, ..., N)$. It is further assumed that the constraint functions $h_j(x(t), u(t))$ only consist of state constraints $h_j^{\mathrm{s}}(x(t)) \leq 0$. The abbreviation

$$\vartheta_j(x(t)) = \begin{pmatrix} h_j^{\mathrm{s,a}}(x(t)) \\ h_j^{(1),\mathrm{s,a}}(x(t)) \\ \dots \\ h_j^{(r-1),\mathrm{s,a}}(x(t)) \end{pmatrix} . \tag{4.1}$$

contains the time derivatives of the active state constraint $h_j^{\mathrm{s,a}}$ up to one below the relative degree $r \in \mathbb{N}$. The relative degree r is the derivative $h_j^{(r),\mathrm{s,a}}(x(t), u_j(t))$, where the control vector u appears explicitly for the first time. State constraints are active if the state $x(t)$ satisfies $h_j^{\mathrm{s,a}}(x(t)) = 0$. For simplified notation, it is assumed that only one state constraint is active at a time t. The extension to multi-dimensional state constraints is straightforward. The Hamiltonian function

$$H_j(x(t), \lambda(t), u(t)) = \phi_j(x(t), u(t)) + \lambda^T(t) f_j(x(t), u(t)) + \mu^T(t) h_j^{(r),\mathrm{s,a}}(x(t), u(t)) \tag{4.2}$$

is defined as in Definition 2.24, where the time derivative $h_i^{(r),\mathrm{s,a}}$ with relative degree r is added to (4.2). The optimality conditions presented in Chap. 3 are extended in Theorem 4.1 with additional optimality conditions at the entry times t_c of state constraints. For a concise notation, the dependence on time t is often omitted in the following equations.

Theorem 4.1: Given a hybrid system $\mathbb{H}$ according to Definition 2.34 and a hybrid system execution σ fulfilling the Assumptions 2.35, 2.11, and 2.12, then all controls υ^* (locally) minimizing the cost functional:

$$\min_{\upsilon \in \mathcal{U}} J(\upsilon) = \min_{\upsilon \in \mathcal{U}} \Big(g(x(t_e)) + \sum_{j=0}^{N} \int_{t_j}^{t_{j+1}} \phi_j(x(t), u_j(t))\, dt \Big) \tag{4.3}$$

lead to an optimal execution $\sigma^* = (\tau^*, \varrho^*, \chi^*, \upsilon^*)$, such that the following conditions are satisfied:

$$\dot{x}^* = f_j(x^*, u^*) \quad \text{a.e. } t \in [t_j, t_{j+1}) \tag{4.4}$$

$$\dot{\lambda}^* = -\nabla_x H_j \quad \text{a.e. } t \in [t_j, t_{j+1}) \tag{4.5}$$

$$\lambda^*(t_e) = \nabla_x g(x^*(t_e)) + \nabla_x \psi_N^T(x^*(t_e))\,\nu^* \tag{4.6}$$

$$\lambda^*(t_j^-) = \lambda^*(t_j) + \nabla_x m_{j-1,j}^T(x^*(t_j))\,\pi_j^* \quad \text{for } j \neq 0 \tag{4.7}$$

$$H_{j-1}(t_j^-) = H_j(t_j) \quad \text{for } j \neq 0 \tag{4.8}$$

$$\lambda^*(t_c^-) = \lambda^*(t_c) + \nabla_x \vartheta_j^T(x^*(t_c))\,\kappa_c^* \tag{4.9}$$

$$H_j(t_c^-) = H_j(t_c) \tag{4.10}$$

$$\mu^{*T} h_j^{\mathrm{s}}(x^*) = 0 \tag{4.11}$$

$$\mu^* \geq 0 \tag{4.12}$$

$$u_j^* = \operatorname*{arg\,min}_{u \in U_j} H_j(x^*, u, \lambda^*) \tag{4.13}$$

$$0 = m_{j-1,j}(x^*(t_j)) \quad \text{for } j \neq 0 \tag{4.14}$$

$$0 = \psi_N(x^*(t_e)) \tag{4.15}$$

$$0 \geq h_j^{\mathrm{s}}(x^*) \tag{4.16}$$

$$x^*(t_0) = x_0\,, \tag{4.17}$$

where $j \in \{0, ..., N\}$, $c \in \{1, ..., N_c\}$, N_c is the number of inequalities met, $\nu \in \mathbb{R}^{n_{\psi_N}}$, $\kappa_c \in \mathbb{R}^r$, and $\pi_j \in \mathbb{R}$. □

Proof: For the proof, see e.g. [32]. ■

4.2.2 Initialization

Initializing an indirect multiple shooting algorithm for solving an HOCP is comparably difficult to the case of non-hybrid optimal control problems with state constraints, which allows the adjoint variables to jump. The difficulty arises from the physically non-intuitive adjoint variables and the small domain of convergence compared to direct methods and DP. The latter is mainly due to integrating the adjoint differential equations in the instable direction, which turns the optimization problem ill-conditioned. Additionally, good estimates of the switching times and states are required in the fixed sequence of discrete states.

In the following, a direct and an indirect multiple shooting algorithm for solving HOCPs are explained shortly as they form the basis for two initialization concepts introduced later in the section.

Direct Multiple Shooting

The continuous time HOCP is time discretized with time samples t_s with $s \in \{0, 1, ..., N_s\}$ and $t_{N_s} = t_e$ to transform the HOCP into a nonlinear program [26]. For the nonlinear program, the continuous state $x[s] := x(t_s)$ and control variables $u[s] := u(t_s)$ for the samples s form the optimization variables. From the set of samples $s \in \{0, 1, ..., N_s\}$, so called key samples s_j are picked and are associated with the initial, hybrid switching, and final time instants t_j for $j \in \{0, ..., N+1\}$, such that $x[s_j] := x(t_j)$. As the time instants t_j of hybrid switches will vary during the optimization, it is necessary to define further optimization variables $\Delta t_j = t_{j+1} - t_j$ with $j \in \{0, ..., N\}$ for the duration of being in discrete state j. Equally distributed inter-sampling times $\Delta t[s] := t_{s+1} - t_s = \frac{\Delta t_j}{s_{j+1} - s_j}$ are determined for all samples $s \in \{s_j, ..., s_{j+1} - 1\}$ relative to Δt_j. The solver for the nonlinear program varies $x[s]$, $u[s]$, and Δt_j such that the discretized cost functional

$$J = g[N_s] + \sum_{j=0}^{N} J_j \tag{4.18}$$

with

$$J_j = \frac{1}{2} \Delta t[s_j]\, \phi_j[s_j] + \frac{1}{2} \Delta t[s_{j+1}]\, \phi_j[s_{j+1}] + \sum_{s=s_j+1}^{s_{j+1}-1} \frac{1}{2} \big(\Delta t[s-1] + \Delta t[s]\big)\, \phi_j[s] \tag{4.19}$$

is minimized with discretized equality and inequality constraints:

$$0 = \frac{1}{2}(f_j[s+1] + f_j[s]) - \frac{x[s+1] - x[s]}{\Delta t[s]} = \gamma[s] \qquad \text{for } s \in \{s_j, s_{j+1} - 1\} \tag{4.20}$$

$$0 = m_{j-1,j}[s_j] \quad \text{for } j \neq 0 \tag{4.21}$$

$$0 = \psi_N[N_s] \tag{4.22}$$

$$0 = x[0] - x_0 \tag{4.23}$$

$$0 = t_e - t_0 - \sum_{j=0}^{N} \Delta t_j \tag{4.24}$$

$$0 \geq h_j[s] \tag{4.25}$$

$$0 \geq c_j[s] \tag{4.26}$$

$$0 \geq \; - \Delta t_j \,. \tag{4.27}$$

Here, $c_j[s]$ is a set of functions approximating the boundaries of the set U_j for the control $u_j[s]$. A function followed by $[s]$ means that the function is evaluated with all its arguments at time t_s. Instead of a trapezoidal approximation of the system dynamics (4.20) any other method like Runge-Kutta or a direct integration of the differential equations (4.4) can be used if desired. Note that in this formulation, the inequality constraints (4.25) and (4.26) are only required to be satisfied at the sampling times t_s. There is no need to prespecify the sequence or times of active inequality constraints. It is found during the optimization. Depending on the complexity of the HOCP, the necessary accuracy of the initialization for a successful solution with the direct method varies but is in general far from the accuracy required by indirect methods. For HOCPs with low complexity, an initialization with zero state $x[s]$ and control $u[s]$ values and equally distributed time intervals Δt_j is often enough.

For solving the nonlinear program, a suitable solver adjoins the equality and inequality constraints (4.20) - (4.27) with multipliers to the costs (4.18). The multipliers for adjoining the system dynamics (4.20), the state constraints (4.25), and the interior and terminal constraints are called $\lambda[s]$, $\mu[s]$, $\pi[s_j]$, and $\nu[N_s]$, respectively. It can be shown that for

$N_s \to \infty$ the costs of the time discretized optimal control problem (4.18) converge to the costs (4.3) of the originial HOCP [63]. Furthermore, the sampled multipliers converge for $N_s \to \infty$ to the optimal multipliers $\lambda^*(t_s)$, $\mu^*(t_s)$, $\pi^*(t_j)$, and $\nu^*(t_e)$ [66] except for $\lambda[s]$ in the case of active inequality constraints. Thus, the sampled multipliers provided by the solver can in general be used as estimates for the optimal multipliers.

Indirect Multiple Shooting

The goal of indirect methods is to solve the MPBVP resulting from the optimality conditions given in Sec. 4.2.1 [25, 136]. Before solving the MPBVP, the sequence of hybrid switches, constrained, and unconstrained arcs has to be pre-specified. A set of time instants t_k, $k \in \{1, 2, ..., N_k\}$ is chosen with $t_0 < t_1 < ... < t_{N_k} < t_e$ where N_k is large enough to assign one time instant t_k to each autonomous switching instant t_j and constraint entry time t_c. At each time t_k, values for the continuous state x_k and the adjoint variable λ_k have to be guessed. Those values are used as initialization for integrating the system and adjoint differential equations (4.4) and (4.5) with the control υ given by (4.13) from t_k to t_{k+1}^- to obtain the state and adjoint variables $x(t_{k+1}^-)$ and $\lambda(t_{k+1}^-)$. This enables us to define a set of error equations ς_k for $k \in \{1, ..., N_k + 1\}$, which a solver steers simultaneously to zero by varying the initial values x_k and λ_k. The error equations are set up depending on if a hybrid switch (4.28), an entry onto a state constraint arc (4.29), or neither the first nor the second case (4.30) occurs at time t_k:

$$\varsigma_k = \begin{pmatrix} x(t_k^-) - x_k \\ \lambda(t_k^-) - \lambda_k - \nabla_x m_{j-1,j}^T(x(t_k^-))\pi_j \\ H_{j-1}(t_k^-) - H_j(t_k) \\ m_{j-1,j}(x(t_k^-)) \end{pmatrix} \tag{4.28}$$

$$\varsigma_k = \begin{pmatrix} x(t_k^-) - x_k \\ \lambda(t_k^-) - \lambda_k - \nabla_x \vartheta_j^T(x(t_k^-))\kappa_c \\ H_j(t_k^-) - H_j(t_k) \\ \vartheta_j(x(t_k^-)) \end{pmatrix} \tag{4.29}$$

$$\varsigma_k = \begin{pmatrix} x(t_k^-) - x_k \\ \lambda(t_k^-) - \lambda_k \end{pmatrix} , \tag{4.30}$$

for $k \in \{1, ..., N_k\}$. Additionally, the terminal error with dimension n_x is given by:

$$\varsigma_e = \begin{pmatrix} \psi_N(x(t_e)) \\ \lambda(t_e) - \nabla_x g(x(t_e)) - \nabla_x \psi_N^T(x(t_e))\, \nu \end{pmatrix} . \tag{4.31}$$

There exist various possibilities how to determine π_j, κ_c, ν, and t_k, e.g. the parameters can also be treated as optimization variables. At the initial time t_0, only an adjoint optimization variable λ_0 is introduced since the state variable is set to the known value x_0. The error equations are usually solved by a modified Newton-method for root finding.

Concept 1

In the first concept, a complete solution of the HOCP is obtained by the direct multiple shooting algorithm in Sec. 4.2.2 and is used to initialize the indirect multiple shooting algorithm presented in Sec. 4.2.2. The first step of the initialization of the indirect method is to specify the sequence of autonomous switches, constrained and unconstrained arcs. The sequence of constraints is taken from the values of the multipliers $\mu[s]$ provided by the solution of the direct method and is fitted into the prespecified hybrid sequence.

Next, the optimization horizon of the indirect method has to be divided into multiple intervals. The most intuitive way is to set $t_k = t_s$. However, the number of intervals N_k in the indirect method may also be chosen significantly lower than the number of samples N_s in the direct method. The domain of convergence of the indirect method is relatively small, thus it is essential to have good estimates for the initial values x_k and λ_k of each interval (e.g. adjoint estimates must equal the optimal values for at least two digits for the system discussed in [37]). In the following, a concept for obtaining estimates x_k and λ_k from the solution of the direct multiple shooting is shown. If no state inequality constraints appear, the values x_k and λ_k can be calculated with $x[s]$ and $\lambda[s]$. However, for active state inequality constraints, the optimal adjoint variables are not unique in the subspace $\Theta_j(x(t))$ that is spanned by the row vectors of $\nabla_x \vartheta_j(x(t))$. This implies that the values $\lambda[s]$ may be completely different from the optimal adjoint variables $\lambda^*(t_s)$ in the subspace $\Theta_j(x(t_s))$. Only the uniquely determined part of $\lambda[s]$, which is in the complementary subspace $\bar{\Theta}_j(x(t_s))$, is used for updates. The other part is obtained by a backward integration of previously and uniquely known estimates of the adjoint variables. To distinguish between the uniquely determined part of λ and the non-unique part, projections into the non-unique and the unique subspace are applied

$$P_{B(\Theta_j)} = B(\Theta_j)\, B(\Theta_j)^T \tag{4.32}$$

$$P_{\bar{B}(\Theta_j)} = I - P_{B(\Theta_j)}\,, \tag{4.33}$$

respectively, with the identity matrix I. The projections are based on $B(\Theta_j)$, which is a normalization and an orthogonalization of the subspace $\Theta_j(x(t_s))$ that can be achieved, for example, by a Gram-Schmidt process [179].

As the adjoint variable $\lambda[s]$ is related to the equality constraint $\gamma[s]$, $\lambda[s]$ corresponds to its continuous-time equivalent $\lambda(\bar{t}_s)$ with $\bar{t}_s = \frac{1}{2}(t_{s+1} + t_s)$. At each time $\bar{t}_s$, the backward integration of Eq. (4.4) and (4.5) is updated according to

$$\hat{x}(\bar{t}_s) = \frac{1}{2}(x[s] + x[s+1]) \tag{4.34}$$

$$\hat{\lambda}(\bar{t}_s) = P_{\bar{B}(\Theta_j)}\Big[\alpha\lambda[s] + (1-\alpha)\tilde{\lambda}(\bar{t}_s)\Big] + P_{B(\Theta_j)}\tilde{\lambda}(\bar{t}_s)\,, \tag{4.35}$$

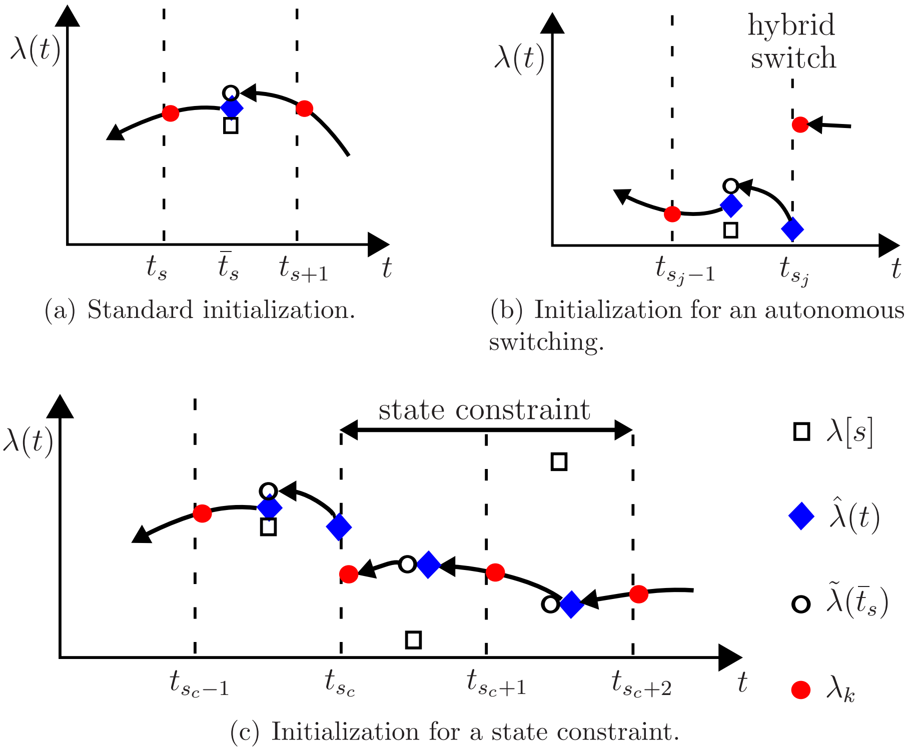
where the uniquely determined part of the adjoint variable is chosen as a convex combination of the values $\lambda[s]$ from the direct solution and $\tilde{\lambda}(\bar{t}_s)$ from the backward integration with $\alpha \in \mathbb{R}, \alpha \in [0,1]$, see Fig. 4.1(a). In contrast, the state $x(\bar{t}_s)$ is always updated only by the values $x[s]$ and $x[s+1]$ from the direct method due to their good quality. At the final time t_e, at the switching times $t_j = t_{s_j}$, see Fig. 4.1(b), and at the entry times of constrained arcs $t_c = t_{s_c}$, see Fig. 4.1(c), the backward integration is initialized by

$$\hat{\lambda}(t_e) = \nabla_x g(x[N_s]) + \nabla_x \psi_N^T(x[N_s])\, \nu[N_s] \tag{4.36}$$

$$\hat{\lambda}(t_j^-) = \tilde{\lambda}(t_j) + \nabla_x m_{i-1,i}^T(x[s_j])\, \pi[s_j] \tag{4.37}$$

$$\hat{\lambda}(t_c^-) = P_{\bar{B}(\Theta_j)} \tilde{\lambda}(t_c) + P_{B(\Theta_j)} \lambda[s_c - 1] \tag{4.38}$$

as no adjoint variables $\lambda[N_s]$, $\lambda[s_j^-]$, and $\lambda[s_c^-]$ can be provided by the direct solver. The samples s_j and s_c denote an autonomous switching and an entry onto a state constraint. Here, the values $\tilde{\lambda}(t_j)$ and $\tilde{\lambda}(t_c)$ are determined by the update (4.35) and a backward integration from $\bar{t}_{s_j}$ to t_j and from $\bar{t}_{s_c}$ to t_c, respectively. The multipliers $\nu[N_s]$ and $\pi[s_j]$ are given by the direct method. As no multiplier $\kappa[s_c]$ can be obtained from the solver, $\lambda[s_c^-]$ can only be approximated by $P_{B(\Theta_j)}\lambda[s_c - 1]$, which uses the neighboring sample $s_c - 1$. In the scalar example shown in Fig. 4.1(c), the new starting values $\hat{\lambda}(\bar{t}_{s_c})$ and $\hat{\lambda}(\bar{t}_{s_c+1})$ can only be determined by the backward integration according to (4.35) while a state constraint is active. Note that for unique updates the update rule (4.35) requires the assumption that two overlapping state constraints cause jumps in the adjoint variables in orthogonal subspaces Θ_j.



(a) Standard initialization.

(b) Initialization for an autonomous switching.

(c) Initialization for a state constraint.

Fig. 4.1: Examples of the initialization process with backward integration to find initial adjoint variables λ_k for the indirect method from the solution of the direct method $\lambda[s]$. Vertically aligned groups of points have different values, while horizontally aligned groups of points denote equal values.

Concept 2

The second concept is based on an algorithm presented in [214]. There, the HOCP (4.3) is decomposed into a two-layered problem. On the higher level, a sequence of switching points $x(t_j)$ and entry $x(t_c)$ and exit points $x(t_c^{\text{out}})$ of state constrained arcs is moved along the corresponding switching manifolds and state constraints with a gradient method until the optimality conditions (4.7) - (4.10) are satisfied. On the lower level, optimal control subproblems with constant discrete state and constant active state constraints are solved from one point to the next in the sequence of switching points and entry and exit points of state constraints. The optimal control of the subproblems is performed by indirect multiple shooting as introduced in Sec. 4.2.2. Here, only the error Eq. (4.30) and (4.31) are considered as the subproblems are non-hybrid and have constant active state constraints.

For the initialization of the algorithm, a sequence of discrete states and corresponding switching points is guessed. This leads to optimal control subproblems with constant discrete state but possibly varying active state constraints between two consecutive switching points $x(t_j)$ and $x(t_{j+1})$. Then, each subproblem is solved separately from the others by the direct method from Sec. 4.2.2, where the costs in (4.18) reduces to $J = J_j$ for subproblem j defined between the samples s_j and s_{j+1}. Furthermore, (4.21) is replaced by the boundary conditions $x(t_j) - x[s_j] = 0$ and $x(t_{j+1}) - x[s_{j+1}] = 0$. Thus, the entire HOCP does not need to be solved at once by the direct optimization method. From the solution of the direct method, estimates of the optimal state and adjoint variables and the sequence of constrained and unconstrained arcs with estimates of the optimal entry and exit points are obtained for each subproblem. The estimates of the sequence of constrained and unconstrained arcs are used to divide the subproblems further into subproblems with constant discrete states and constant active state constraints. These new subproblems are now solved individually by indirect multiple shooting with the initial estimates of the state and adjoint variables from the direct method. These estimates are determined with the backward integration scheme introduced in Concept 1. Varying the switching, entry and exit points to find the optimal state and control trajectory of the entire HOCP problem is performed by the algorithm introduced in [214]. Note that the algorithm there is set up for hybrid systems with partitioned state spaces, so that the optimal sequence of discrete states can be found during the optimization and does not need to be known a priori. However, by removing the algorithm's ability to change the discrete state sequence during the optimization, the algorithm can be applied to the more general class of hybrid systems specified in Sec. 4.2.1.

4.2.3 Numerical Example

Autonomous Vehicle

Both initialization concepts are used for optimizing a drive-around maneuver of an autonomous car. The goal is to find the optimal state and control trajectory to drive around a parked car. The autonomous car is modeled as a *unicycle* or *Dubin's vehicle* [145]:

$$\begin{aligned} \dot{x} &= v \cos\theta && (4.39)\\ \dot{y} &= v \sin\theta && (4.40)\\ \dot{\theta} &= u && (4.41)\\ |u| &\leq 0.8\,, && (4.42) \end{aligned}$$

where (x, y) is the position of the car, θ the orientation, v the constant velocity, and the steering input u is the velocity of a change in the orientation. The problem becomes hybrid by the model of the environment and the associated cost functions. A two lane road is modeled with three discrete states $q = \{1, 2, 3\}$, such that if the center point of the car is in discrete state $q = 1$ or $q = 3$, then the whole car is completely on the right or left lane. The transition region for the center point of the car between the right and the left lane is discrete state $q = 2$, which corresponds to driving in the middle of the road, see Fig. 4.2:

$$\begin{aligned} q = 1: &\quad -2.15 \leq y \leq -0.85 && (4.43)\\ q = 2: &\quad -0.85 \leq y \leq 0.85 && (4.44)\\ q = 3: &\quad 0.85 \leq y \leq 2.15\,. && (4.45) \end{aligned}$$

The hybrid formulation provides the advantage that clear decisions to drive on the right, middle, or left part of the road are made by the autonomous car. Furthermore, by the choice of the assigned cost functions

$$\begin{aligned} \phi_1 &= c_1 u^2 + c_2 (y + 1.5)^4 && (4.46)\\ \phi_2 &= c_1 u^2 + c_3 && (4.47)\\ \phi_3 &= c_1 u^2 + c_4\,, && (4.48) \end{aligned}$$

the car can be manipulated to drive on the right lane except for special situations like another car blocking the right lane. A car blocking the right lane is modeled with two state inequality constraints

$$\begin{aligned} h_1^{\text{rear}} = h_2^{\text{rear}} &= 0.375\,x - y - 10.4 \leq 0 && (4.49)\\ h_1^{\text{front}} = h_2^{\text{front}} &= -0.375\,x - y + 15.85 \leq 0\,, && (4.50) \end{aligned}$$

which demand the autonomous car to drive around the obstacle on the left lane, such that no collision can occur. Further state inequality constraints are represented by the right border of the right lane and the left border of the left lane.

The optimization task is to find the optimal state and control trajectory around the parked car with discrete state sequence $(1, 2, 3, 2, 1)$. The optimal solution minimizes the costs accumulated by (4.46) - (4.48) over a time horizon of 6 sec under consideration of the car dynamics (4.39) - (4.41), the control (4.42), and state constraints. Here, the constants are chosen to be: $v = 15$, $c_1 = 20$, $c_2 = 10$, $c_3 = 5$, and $c_4 = 10$.

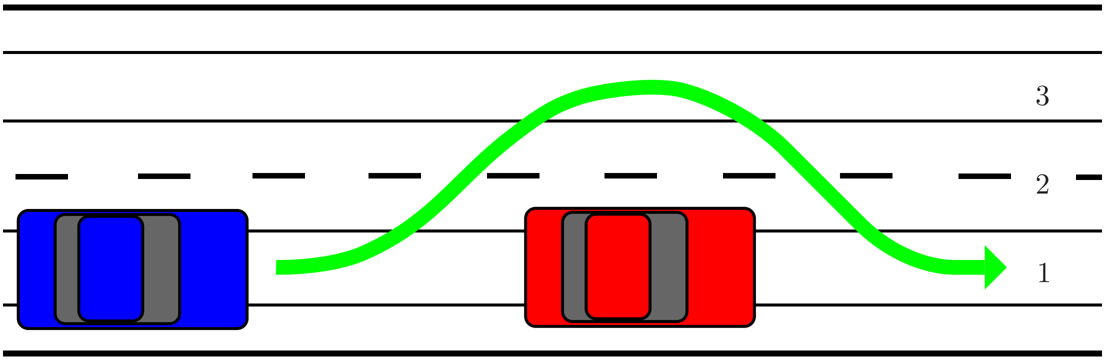


Fig. 4.2: Autonomous car driving around a parked car in a division of the road into three discrete states.

Results

The results of optimizing with concepts 1 and 2 are illustrated in Fig. 4.3 and Fig. 4.4, respectively. The accuracy of the solution of concept 1 was higher, which can be seen in the optimality conditions, where each condition is satisfied in concept 1 to an accuracy of at least 10^{-3} compared to $7 \cdot 10^{-1}$ in concept 2. Consequently, the costs of the reached optima differ with $J = 17.34$ in concept 1 and $J = 17.68$ in concept 2. The costs found by the direct method in concept 1 are $J = 26.39$. Our experience shows that the initialization with concept 2 is more robust and needs less tuning of optimization parameters to achieve convergence. Additionally, the initialization for the direct method in the individual discrete states may be less accurate. The computation time was 4.5 h in concept 1 and 15.1 h in concept 2. The computations were performed on an AMD 3 GHz processor running Windows and Matlab 2009a.

4.2.4 Discussion

Two concepts are introduced for initializing indirect multiple shooting methods with direct methods for the optimal control of hybrid systems. In concept 1 the whole optimization problem is initialized directly, whereas in concept 2 the problem is decomposed such that only subproblems have to be initialized.

In the following, a short comparison of the advantages and disadvantages of the two concepts is given. Concept 1 delivers solutions with higher accuracy than concept 2 as observed in the example in Sec. 4.2.3. This means that for the same number of iterations and comparable initial values, the solution of concept 1 is closer to the optimal solution. Solving the complete HOCP at once, the direct method in concept 1 allows to apply indirect multiple shooting for the solution of the entire HOCP, which achieves a very high accuracy with its Newton-iterations. In contrast, the upper layer of the indirect approach in concept 2, which shifts the switching points, uses updates based on first-order gradients. In general, algorithms based on first-order gradient approaches show a lower accuracy for a comparable number of iterations and a larger domain of convergence than algorithms with second-order Newton update steps. The latter argument directly leads to the major disadvantage of concept 1, which is the lower robustness. This means the reported number

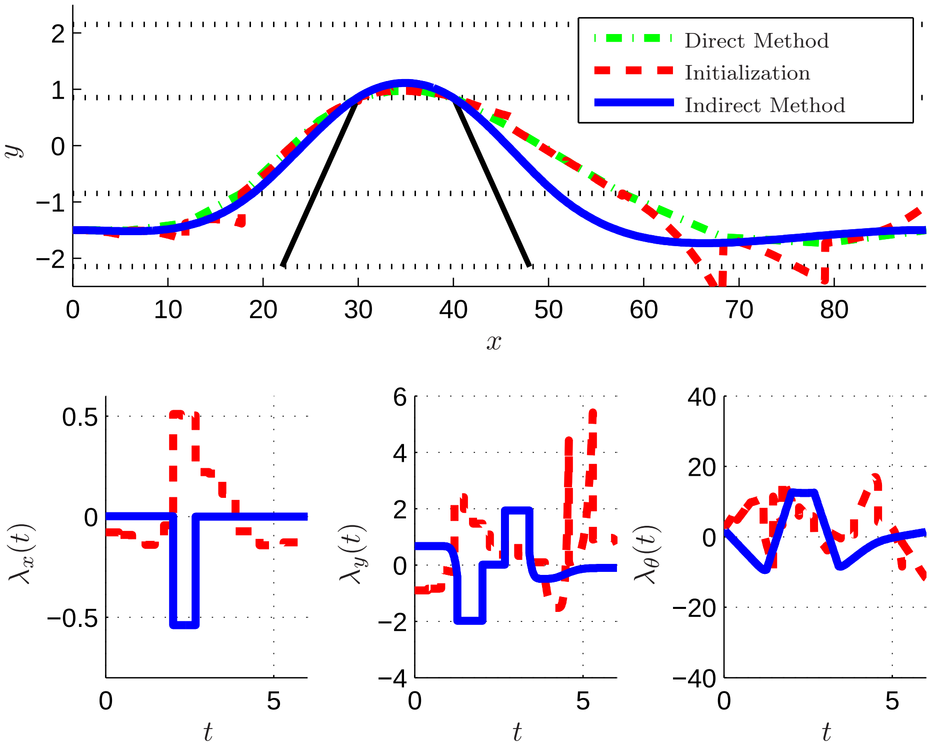


Fig. 4.3: Comparison of the trajectories found by concept 1 after optimizing with the direct method (green dotted line), after initializing the indirect method with the direct one (red dashed line), and after optimizing with the indirect one after the initialization (blue solid line).

of failing in the initialization is higher. An initialization fails if the indirect method does not converge to an optimal solution. In such a case, the accuracy of the solution from the direct method has to be increased, e.g. by augmenting the number of sampling points or refining the initialization of the direct method. In concept 1, a very complex optimization problem is to be solved, which causes difficulties in achieving the required accuracy for initializing the indirect multiple shooting with Newton-iterations successfully. The gradient method applied in concept 2 is capable of converging to the optimal solution, even if the initial switching points are far from the optimal switching points. Furthermore, the first approach needs a higher accuracy in the estimates of the optimal state and adjoint state variables. This is due to the influence of small errors of the initial values compared to the optimal ones on the complete HOCP. In contrast, the impact of initialization errors in concept 2 remains bounded to each subproblem. Additionally, the accuracy of the estimates is higher and the estimates can be found faster as the decomposition into non-hybrid and smaller subproblems simplifies the solution for the direct method. The direct method in concept 2 tolerates a less accurate initialization due to the lower complexity of the optimal control subproblems in contrast to the complexity of the entire HOCP. In total, this leads to a higher robustness of the initialization in concept 2.

Drawing a conclusion from the comparison of the two initialization concepts, the fol-

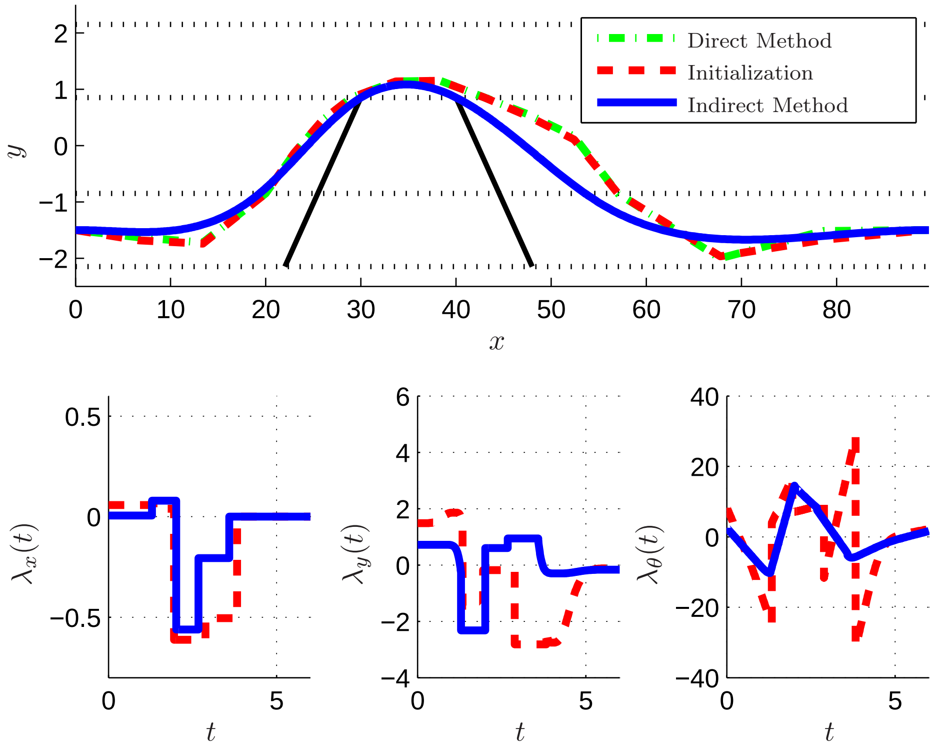


Fig. 4.4: Comparison of the trajectories found by concept 2 after optimizing with the direct method (green dotted line), after initializing the indirect method with the direct one (red dashed line), and after optimizing with the indirect one after the initialization (blue solid line).

lowing is recommended:

- For strongly nonlinear systems with many hybrid switches, concept 2 is the initialization scheme, that is more robust against inaccuracies of the solution from the direct method.
- If a high accuracy of the solution is desired, it is recommended to apply concept 1.
- It is even possible to use concept 2 for initializing an indirect method that solves the entire HOCP at once. With this approach, HOCPs can be solved robustly and with high accuracy.

Consider the case of hybrid systems with partitioned state space, where the optimal discrete state sequence is unknown. There, concept 2 can be initialized with some feasible sequence of discrete states and during the optimization the sequence is varied until a locally optimal sequence is found [214]. In comparison, the direct method in concept 1 either remains unchanged, such that it searches for an optimal trajectory in the prespecified hybrid sequence. Then, the optimal sequence has to be found by solving an HOCP for all possible sequences, which exhibits combinatoric computational complexity. Alternatively, the direct method could be modified such that it finds the locally optimal sequence automatically, but here it is especially difficult to reach the necessary accuracy for the initialization.

A task for the future is to explore how an initial discrete state sequence can be found in the neighborhood of the optimal discrete state sequence.

4.3 A Globally Convergent Min-H Algorithm

In the following, an indirect optimal control method for a simplified initialization and a larger domain of convergence is introduced. The section is structured as follows: Sec. 4.3.1 describes the hybrid optimal control problem and presents necessary optimality conditions. The algorithm is derived in Sec. 4.3.2 and its convergence is proved in Sec. 4.3.3. Sec. 4.3.4 shows a numerical example and Sec. 4.3.5 discusses observations about the method.

4.3.1 Hybrid Optimal Control Problem

In this section, hybrid systems as defined in Definition 2.38 are applied with the Assumption 2.39. The hybrid systems include continuous and discrete states, continuous controls, continuous and discrete dynamics, and switching manifolds for autonomous switching. The class of HOCPs contains terminal and running costs, and interior and terminal constraints, see Definition 2.40. All functions are at least twice continuously differentiable.

In general, the novel min-H algorithm is also able to deal with an extended class of hybrid systems, that includes resets of the continuous state at autonomous switchings, state constraints, switching costs, etc. These effects are skipped here in order to keep the derivation concise.

Hereafter, interior conditions $\psi_{j-1}(x(t_j)) = 0$ are combined with the corresponding switching manifold $m_{q_{j-1},q_j}(x(t_j)) = 0$ for $q_{j-1}, q_j \in \mathcal{Q}$ and $j \in \{1, ..., N\}$, to the n_j-dimensional vector $\tilde{m}_j$ with $n_j \leq n_x$:

$$\tilde{m}_j(x(t_j)) := \begin{pmatrix} m_{q_{j-1},q_j}(x(t_j)) \\ \psi_{j-1}(x(t_j)) \end{pmatrix} = 0\,. \tag{4.51}$$

Remark 4.1: In the following, the general HOCP, which is specified in Definition 2.40 and given in Bolza representation, is reformulated without loss of generality into Mayer form [19]. This has advantages for the proof of the convergence of the algorithm. □

An optimal solution of the HOCP satisfies necessary optimality conditions that are given by the HMP as defined in Theorem 4.2. In the following, the dependence of variables on time t is partly omitted as well as the asterisk '*' for optimal switching times t_j^*. The Hamiltonian $H_q(x(t), \lambda(t), u_q(t)) = \lambda(t)^T f_q(x(t), u_q(t))$ is as defined in Definition 2.23.

Theorem 4.2: Given a hybrid system $\mathbb{H}$ according to Definition 2.38 and a hybrid system execution σ fulfilling the Assumptions 2.39, 2.11, and 2.12, then all controls υ^* (locally) minimizing the cost functional:

$$J(\upsilon^*) = \inf_{\upsilon \in \mathcal{U}} J(\upsilon) \tag{4.52}$$

lead to an optimal system execution $\sigma^* = (\tau^*, \varrho^*, \chi^*, \upsilon^*)$, such that the following conditions are satisfied:

$$\dot{x}^* = f_{q_j^*}(x^*(t), u^*_{q_j^*}(t)) \quad \text{a.e. } t \in [t_j, t_{j+1}),\, \forall j \in \{0, ..., N\} \tag{4.53}$$

$$\dot{\lambda}^* = -\nabla_x H^T_{q_j^*} \quad \text{a.e. } t \in [t_j, t_{j+1}),\ \forall j \in \{0, ..., N\} \tag{4.54}$$

$$\lambda^*(t_e) = \nabla_x g^T(x^*(t_e)) + \nabla_x \psi_N^T(x^*(t_e))\, \nu^* \tag{4.55}$$

$$\lambda^*(t_j^-) = \lambda^*(t_j) + \nabla_x \tilde{m}^T_{q^*_{j-1}}(x^*(t_j))\, \tilde{\pi}_j^* \quad j \in \{1, ..., N\} \tag{4.56}$$

$$H_{q^*_{j-1}}(t_j^-) = H_{q_j^*}(t_j) \quad j \in \{1, ..., N\} \tag{4.57}$$

$$H_{q_j^*}(x^*, \lambda^*, u^*_{q_j^*}) \leq H_{q_j^*}(x^*, \lambda^*, u_{q_j^*}) \quad \text{a.e. } t \in [t_j, t_{j+1}],\ j \in \{0, ..., N\},\ \forall u_{q_j^*} \in U_{q_j^*} \tag{4.58}$$

$$\tilde{m}_j(x^*(t_j)) = 0 \quad j \in \{1, ..., N\} \tag{4.59}$$

$$\psi_N(x^*(t_e)) = 0 \tag{4.60}$$

with the optimal and constant Lagrange multipliers $\tilde{\pi}_j^* \in \mathbb{R}^{n_j}$ and $\nu^* \in \mathbb{R}^{n_{\psi_N}}$. □

Proof: The proof of the theorem builds on a combination of results derived with needle variations for continuous optimal control problems with state constraints [76], for HOCPs with autonomous switching [165, 208], for HOCPs with autonomous and controlled switching [137, 157], and for HOCPs with state constraints [39]. Similar results are found with variational techniques in [19, 32, 74]. Due to this previous work, the proof of Theorem 4.2 is here omitted. ■

In the following, the analysis is restricted to a fixed sequence of discrete states $\varrho = (0, 1, ..., N)$. In HOCPs with fixed discrete state sequence, the HMP cannot be applied in general since no optimal control might exist for that specific sequence of discrete states, compare [56]. This case is excluded by Assumption 4.2 below:

Assumption 4.2: It is assumed that an optimal control exists for the considered HOCP with the fixed discrete state sequence ϱ, such that the costs (4.52) are minimized. □

Remark 4.3: The assumption that an optimal control exists for the given discrete state sequence is required to prove the convergence of the algorithm to an optimal solution, as the algorithm is not able to optimize the discrete state sequence in the presented version. If the algorithm is applied to a discrete state sequence without an optimal solution, the algorithm converges to a solution that minimizes the distance in the optimality conditions to an optimal solution. If the convergence to such a non-optimal solution is observed, one has to restart the algorithm with a different discrete state sequence and/or initialization since the algorithm is locally not able to reach an optimal solution. □

4.3.2 Algorithm

In this section, the novel min-H algorithm is presented. For better illustration, the algorithm is divided into a main algorithm that controls the updates of the optimization variables and the step sizes and a second algorithm that computes the update direction based on the step sizes specified by the main algorithm. Before the complete min-H algorithm is introduced in its full complexity, the main ideas that lead to the algorithm are explained in the following Subsection 4.3.2. Please note that no derivations or proofs for the equations in Subsection 4.3.2 are given here. A detailed proof together with the precise derivation of the algorithm is provided in Sec. 4.3.3. First, the idea is described how the convergence of the min-H algorithm for non-hybrid optimal control problems can be ensured. Secondly, required extensions to apply the algorithm to HOCPs are discussed.

Concept

Continuous Optimal Control Problem Indirect multiple shooting methods have a small domain of convergence since the convergence is only local and the numerical stability is limited as either the state or adjoint differential equations have to be integrated in their unstable direction. In the following, an indirect gradient algorithm is introduced, where starting from the initial conditions the system dynamics is integrated forward in time in its stable direction until the final time is reached. Afterwards, the adjoint differential equations are integrated backwards in time in their stable direction until the initial time is reached. This backward integration is initialized with the relevant optimality conditions at final time. To perform the integrations and the initialization of the adjoint variable, a control as a function of time has to be guessed initially as well as some constant multipliers. These variables are the optimization variables, which are iteratively varied until an optimal solution is found. The control function is a physically intuitive variable, what simplifies its initialization. After each pair of forward-backward integrations, the optimization variables are updated. Hereafter, it is shown how the update of the control is calculated.

In gradient methods of second order [32], it is assumed that the Hamiltonian is convex with respect to the control, i.e. $\nabla_{uu}H(x(t),\lambda(t),u(t)) > 0$. This implies that the Hamiltonian minimization condition (4.58) is satisfied for any values x and λ if the optimal control u^* is chosen such that $\nabla_u H(x(t),\lambda(t),u^*(t)) = 0$. If this gradient is not zero for the control $u(t)$ in the current iteration of the gradient method, the optimal control $u^*(t)$ is approached by a Newton step. This is achieved by linearization of the current gradient $\nabla_u H(x(t),\lambda(t),u(t))$, which leads to the update formula

$$\begin{aligned}\delta u(t) = -\big(\nabla_{uu}H(x(t),\lambda(t),u(t))\big)^{-1}\Big(&\nabla_{ux}H(x(t),\lambda(t),u(t))\delta x(t)\\ &+\nabla_u f^T(x(t),u(t))\delta\lambda(t)+\epsilon\nabla_u H(x(t),\lambda(t),u(t))\Big)\end{aligned} \tag{4.61}$$

of the control $u(t)$. The step size ϵ with $0 < \epsilon \leq 1$ determines how fast the optimal control $u^*(t)$ satisfying $\nabla_u H(x(t),\lambda(t),u^*(t)) = 0$ is approached. Due to the convexity assumption, second-order gradient methods are only locally convergent to an optimal solution.

For both a large domain of convergence and fast convergence, a min-H approach was proposed by [65] instead of a second-order gradient method. There, the Hamiltonian $H(x(t),\lambda(t),u(t))$ was minimized directly with the current values of $x(t)$ and $\lambda(t)$ to obtain the currently optimal control $u^*(t)$. The update of the control was

$$\delta u(t) = -\epsilon\Delta u(t) \tag{4.62}$$

with $\Delta u(t) = u(t) - u^*(t)$ and $0 < \epsilon \leq 1$. Here, $u(t)$ is the currently best value of the control in the algorithm. The major drawback of this approach is that the update (4.62) is not guaranteed to be improving. The reason is the following: Let the current control $u(t)$ be updated by $\delta u(t)$ from (4.62). Then changes $\delta x(t)$ and $\delta\lambda(t)$ of the state $x(t)$ and adjoint $\lambda(t)$ are introduced by the updated control in the next forward-backward integration of the corresponding differential equations. The changes $\delta x(t)$ and $\delta\lambda(t)$ lead to changes $\delta u^*(t)$ in the optimal control. Depending on the optimal control problem at hand, the change $\delta u^*(t)$ can cause the control $u(t)$ to diverge from the optimal value $u^*(t)$ in each iteration of the algorithm.

This severe restriction is removed with our novel min-H approach. We designed the update of the control

$$\delta u(t) = -\epsilon \Delta u(t) + \delta u^*(t)\,, \tag{4.63}$$

such that the change $\delta u^*(t)$ in the optimal control is compensated up to first order. Note that the change $\delta u^*(t)$ compensates a change in the optimal control $u^*(t)$ directly and not a change around the current control $u(t)$ as in second-order gradient methods, compare (4.61). This means the convergence also works if the current control $u(t)$ is not in a convex region of the Hamiltonian $H(x(t), \lambda(t), u(t))$ in contrast to second-order gradient methods. Consequently, this specific compensation guarantees that our min-H approach is globally convergent to a locally optimal solution. Additionally, the update $\delta u(t)$ in the novel min-H algorithm ensures that the convergence is at least quadratic in a close neighborhood around the optimum. In the unconstrained and nonsingular case, $\nabla_u H(x(t), \lambda(t), u^*(t))$ is zero and $\nabla_{uu} H(x(t), \lambda(t), u^*(t)) > 0$, such that the predicted variation

$$\begin{aligned}\delta u^*(t) = -\big(\nabla_{uu} H(x(t), \lambda(t), u^*(t))\big)^{-1} \Big(&\nabla_{ux} H(x(t), \lambda(t), u^*(t))\,\delta x(t) \\ &+ \nabla_u f^T(x(t), u^*(t))\,\delta\lambda(t)\Big)\end{aligned} \tag{4.64}$$

is obtained by linearization of $\nabla_u H(x(t), \lambda(t), u^*(t))$. The variation $\delta u^*(t)$ can be easily adapted for control constraints and singular controls.

To compute the update $\delta u(t)$, it is required to find the unknown values $\delta x(t)$ and $\delta\lambda(t)$ needed for $\delta u^*(t)$. To obtain the predicted values $\delta x(t)$ and $\delta\lambda(t)$, the following efficient procedure is applied:

(i) The state and adjoint differential equations are linearized and the expression for $\delta u(t)$ from (4.63) and (4.64) is inserted. This results in a linear two-point boundary value problem (TPBVP) of the form

$$\delta\dot{x}(t) = A(t)\delta x(t) - B(t)\delta\lambda(t) - d(t) \tag{4.65}$$

$$\delta\dot{\lambda}(t) = -C(t)\delta x(t) - F(t)\delta\lambda(t) + \gamma(t)\,, \tag{4.66}$$

where the matrices and vectors A, B, C, F, d, and γ result from the linearization of the differential equations.[1]

[1] Compare also the matrices and vectors in (4.85) - (4.90) for details in the min-H algorithm for HOCPs.

(ii) The TPBVP is solved by a single backward sweep of final conditions to the initial time by using Riccati-matrices and vectors S, R, T, Q, e, and η. Replacing $\delta\lambda(t)$ by the approach

$$\delta\lambda(t) = S(t)\delta x(t) + R(t)d\nu + e(t) \tag{4.67}$$

and differentiating the expression, differential equations for $S(t)$, $R(t)$, and $e(t)$ are obtained. Using a similar approach to express the change in the still violated boundary conditions

$$d\psi = T(t)\delta x(t) + Q(t)d\nu + \eta(t)\,, \tag{4.68}$$

differential equations for $T(t)$, $Q(t)$, and $\eta(t)$ are determined. At the final time, the Riccati-matrices and vectors are initialized by comparison with the linearized boundary conditions (4.55) and (4.60).

(iii) After integrating the Riccati-matrices and vectors backwards to the initial time, the update $d\nu$ can be calculated, see (4.117). Afterwards, the differential equation (4.65) is integrated forward in time with the help of Equation (4.67), such that $\delta x(t)$ and $\delta\lambda(t)$ are found for all t and the control update $\delta u(t)$ can be determined. The trajectory $\delta\upsilon$ of the control update is such that the optimal control trajectory υ^* and the boundary condition $\psi(x(t_e)) = 0$ are approached simultaneously.

By careful design, the solution of the TPBVP is derived with a single pair of backward-forward integrations and an iterative procedure is avoided.

Hybrid Optimal Control Problem The idea to use a gradient method for the optimal control of hybrid systems has already been mentioned in [136]. Here, the proposed min-H method for nonlinear systems is extended to hybrid systems with autonomous switching, interior point constraints, and in Sec. 4.3.5 also to state constraints. The control updates $\delta\upsilon$ serve two purposes simultaneously: First, the updates are such that the Hamiltonian minimization condition (4.58) is approached. Second, the updates steer the system trajectory, such that the interior and terminal constraints are approached. The algorithm is designed such that the control update $\delta\upsilon$ in the interval between the initial and the first interior point constraint makes the system approach the first interior point constraint, the control update $\delta\upsilon$ between the first and the second interior point constraint makes the system approach the second interior point constraint and so on. This setup prevents that the Riccati-matrices are growing with each point constraint as in [183].

The necessary modifications for the extended system class are the following:

(i) If during the forward integration a switching manifold is hit, then the corresponding switch in the discrete state is performed and the integration is continued with the dynamics valid in the new discrete state.

(ii) At each switching time of the forward integration, the backward integration is reinitialized with the according optimality conditions for the adjoint variables at autonomous switchings, see (4.74).

(iii) Constant multipliers for the interior point constraints that must be met at autonomous switchings are added to the set of optimization variables.

(iv) Due to variations in the switching times with each update, additional compensation terms have to be added to the trajectory of the control update $\delta \upsilon$. The resulting modifications become obvious when comparing (4.63) and (4.118).

(v) At each switching time, the backward integration of the Riccati terms is reinitialized with (4.100) - (4.105) if interior point constraints are specified at this switching time and with (4.106) - (4.111) if no interior point constraints are specified.

(vi) At each switching time t_j, the variation $\delta x(t_j)$ is reinitialized according to (4.115) during the forward integration of the linearized differential equations (4.113).

In the following, the complete min-H algorithm for the solution of HOCPs with autonomous switching is introduced.

Algorithm

The novel min-H algorithm is shown in a concise form below and it is divided into two parts for a better overview. In Algorithm 4.1, the main algorithm controlling the step size and the updates is proposed. Algorithm 4.2 introduces the computation of the update directions including the compensation terms and the associated prediction of changes in the state and adjoint variables. The derivation is provided in the subsequent section together with the proof of convergence. Here, the dependence of the switching times t_j on the iteration counter l is omitted. The constant Lagrange multipliers $\hat{\pi}_j \in \mathbb{R}^{n_{\psi_{j-1}}}$ for $j \in \{1, ..., N\}$ are associated with the interior constraints $\psi_{j-1}(x(t_j)) = 0$.

Algorithm 4.1:

Step 0 **Initialization:** Initialize the optimization variables[2]: $\hat{\pi}_j^0 \in \mathbb{R}^{n_{\psi_{j-1}}}$ for $j \in \{1, ..., N\}$, $\nu^0 \in \mathbb{R}^{n_{\psi_N}}$, and $\upsilon_q^0 : [t_0, t_e] \to \mathcal{U}_q \ \forall q \in \mathcal{Q}$, such that the desired discrete state sequence ϱ, w.l.o.g. $\varrho = (0, ..., N)$, results. Choose $\beta \in (0, 1)$ and $\tau_0 \in (0, 1)$. Set $p = 0$ and the iteration counter $l := -1$. Execute Steps 2 and 3 once to find the trajectories χ^0 and λ^0, then set $l := 0$ and go to Step 1.

Step 1 **Update:** Get the updates $\delta u(t)$ for $t \in [t_0, t_e]$, $d\hat{\pi}_j$ for $j \in \{1, ..., N\}$ with $n_{\psi_{j-1}} > 0$, and $d\nu$ of the optimal control, the interior, and terminal multipliers from Algorithm 4.2 with update step size $\epsilon := \beta^p$. Update the control function and the multipliers

$$u_j^{l+1}(t) = u_j^l(t) + \delta u(t) \tag{4.69}$$

$$\nu^{l+1} = \nu^l + d\nu \tag{4.70}$$

$$\hat{\pi}_j^{l+1} = \hat{\pi}_j^l + d\hat{\pi}_j \tag{4.71}$$

[2] In the implementation, the control trajectory υ_q is specified by a number of control-time point pairs and the current control $u_q(t)$ is read out by interpolating with splines over υ_q.

for $j \in \{1, ..., N\}$ with $n_{\psi_{j-1}} > 0$ and all $t \in [t_0, t_e]$. If the updated control $u_j^{l+1}(t)$ is not in U_j, it is projected onto the boundary ∂U_j.

Step 2 **Forward Integration:** Solve (4.53) from t_0 to t_e with the initial state $x(t_0) = x_0$, the discrete state sequence $\varrho = (0, ..., N)$, the (updated) control trajectories υ_j^{l+1}, the autonomous switchings from $j-1$ to j if $m_{j-1,j}(x^{l+1}(t_j)) = 0$, and the continuity of the state variables across switchings $x_{j-1}^{l+1}(t_j^-) = x_j^{l+1}(t_j)$. Save χ^{l+1} and t_j $\forall j \in \{1, ..., N\}$.

Step 3 **Backward Integration:**

(a) Set $\lambda^{l+1}(t_e) = \nabla_x g^T(x^{l+1}(t_e)) + \nabla_x \psi_N^T(x^{l+1}(t_e))\, \nu^{l+1}$. Solve

$$\dot{\lambda}^{l+1}(t) = -\nabla_x f_N^T(x^{l+1}(t), u_N^{l+1}(t))\, \lambda^{l+1}(t) \tag{4.72}$$

for a.e. $t \in [t_N, t_e]$ backwards in time from t_e to t_N with $\lambda^{l+1}(t_e)$ as specified above. Set $j := N$.

(b) Set:

$$\begin{aligned}\pi_j^{l+1} = &\frac{\lambda^{l+1,T}(t_j)\Big(f_j(x^{l+1}(t_j), u_j^{l+1}(t_j)) - f_{j-1}(x^{l+1}(t_j), u_{j-1}^{l+1}(t_j))\Big)}{\nabla_x m_{j-1,j}(x^{l+1}(t_j))\, f_{j-1}(x^{l+1}(t_j), u_{j-1}^{l+1}(t_j))} \\ &- \frac{\hat{\pi}_j^{l+1,T}\, \nabla_x \psi_{j-1}(x^{l+1}(t_j))\, f_{j-1}(x^{l+1}(t_j), u_{j-1}^{l+1}(t_j))}{\nabla_x m_{j-1,j}(x^{l+1}(t_j))\, f_{j-1}(x^{l+1}(t_j), u_{j-1}^{l+1}(t_j))} \end{aligned}\tag{4.73}$$

$$\begin{aligned}\lambda^{l+1}(t_j^-) = &\lambda^{l+1}(t_j) + \nabla_x \psi_{j-1}^T(x^{l+1}(t_j))\, \hat{\pi}_j^{l+1} \\ &+ \nabla_x m_{j-1,j}^T(x^{l+1}(t_j))\, \pi_j^{l+1}, \end{aligned}\tag{4.74}$$

where $m_{j-1,j}(x^{l+1}(t_j))$ is the switching manifold that was hit during the forward integration. The corresponding multiplier π_j^{l+1} is determined by solving the Hamiltonian continuity condition (4.57) with (4.56) for π_j^{l+1}.

(c) Integrate

$$\dot{\lambda}^{l+1}(t) = -\nabla_x f_{j-1}^T(x^{l+1}(t), u_{j-1}^{l+1}(t))\, \lambda^{l+1}(t) \tag{4.75}$$

with "initial" conditions (4.74) from t_j backwards to t_{j-1}. Save the trajectory λ^{l+1} and the multiplier π_j^{l+1}. Decrease $j := j - 1$. If $j > 0$, repeat Step 3(b), else go to Step 4.

Step 4 **Descent Criterion:** Check if the Armijo-criterion [4]

$$\Omega(\chi^{l+1}, \lambda^{l+1}, \upsilon^{l+1}) - \Omega(\chi^l, \lambda^l, \upsilon^l) \leq -2\tau_0\, \epsilon\, \Omega(\chi^l, \lambda^l, \upsilon^l) \tag{4.76}$$

is satisfied, where $\|a\|^2 := a^T a$ for any vector a, $\Delta u_j^l(t) := u_j^l(t) - u_j^{*l}(t)$, and

$$\varpi(\chi^l) = \sum_{j=1}^{N} \|\psi_{j-1}(x^l(t_j))\|^2 + \|\psi_N(x^l(t_e))\|^2 \tag{4.77}$$

$$\Omega(\chi^l, \lambda^l, \upsilon^l) = \frac{1}{2}\varpi(\chi^l) + \frac{1}{2}\sum_{j=0}^{N}\int_{t_j}^{t_{j+1}} \|\Delta u_j^l(t)\|^2 \, dt\,. \tag{4.78}$$

□

The optimal control $u_j^{*l}(t)$ in iteration l at time t is found by minimizing the Hamiltonian $H_j(x^l(t), \lambda(t)^l, u_v)$ with respect to $u_v \in U_j$. If the Armijo criterion does not hold, then go back to Step 1 and repeat it with a reduced step size by setting $p := p + 2$.[3] If either $p > p_{\max}$ or (4.76) is fulfilled together with $\Omega(\chi^l, \lambda^l, \upsilon^l) - \Omega(\chi^{l-1}, \lambda^{l-1}, \upsilon^{l-1}) > -\epsilon_\Omega$ for $0 < \epsilon_\Omega \ll 1$, then stop. Else set $p := p - 1$ and $l := l + 1$ and go to Step 1.

Algorithm 4.2 calculates the updates $\delta u(t)$ for $t \in [t_0, t_e]$, $d\hat{\pi}_j$ for $j \in \{1, ..., N\}$ with $n_{\psi_{j-1}} > 0$, and $d\nu$. The updates are required in Step 1 of Algorithm 4.1 to determine the updated control and multipliers.

Algorithm 4.2:

Step 0 **Initialization:** Obtain the current trajectories χ^l, λ^l, and υ^l, the multipliers π_j^l and ν^l, the step size $\epsilon = \beta^p$, and the errors in the boundary conditions $\psi_{j-1}(x^l(t_j))$ and $\psi_N(x^l(t_e))$ from Algorithm 4.1.

Step 1(a) **Backward Sweep with Riccati-Equations:** Solve the system of Riccati differential equations

$$\dot{S}_j = -S_j A_j - F_j S_j + S_j B_j S_j - C_j \tag{4.79}$$
$$\dot{R}_j = -F_j R_j + S_j B_j R_j \tag{4.80}$$
$$\dot{T}_j = -T_j A_j + T_j B_j S_j \tag{4.81}$$
$$\dot{Q}_j = T_j B_j R_j \tag{4.82}$$
$$\dot{e}_j = -(F_j - S_j B_j) e_j + S_j d_j + \gamma_j \tag{4.83}$$
$$\dot{\eta}_j = T_j B_j e_j + T_j d_j \tag{4.84}$$

backwards from t_e to t_0, where j goes from N to 0 and changes at each jump time t_j. At each point of time, the matrices and vectors

$$A_j(t) := D_j^{-1}\Big(\nabla_x f_j(t) - \nabla_u f_j(t)\Xi_j(t)\Big) \tag{4.85}$$
$$B_j(t) := D_j^{-1}\nabla_u f_j(t) L_j(t) \tag{4.86}$$
$$C_j(t) := H_{j,xx}(t) - H_{j,xu}(t)\Big(\Xi_j(t) + \frac{1}{2}\Delta u_j^l(t)\zeta_j^T(t)A_j(t)\Big) \tag{4.87}$$

[3] The update $p := p + 2$ is chosen instead of $p := p + 1$ for a faster convergence.

$$F_j(t) := \nabla_x f_j^T(t) - H_{j,xu}(t)\Big(L_j(t) - \frac{1}{2}\Delta u_j^l(t)\zeta_j^T(t)B_j(t)\Big) \tag{4.88}$$

$$d_j(t) := \epsilon\, D_j^{-1}\, \nabla_u f_j(t)\Delta u_j^l(t) \tag{4.89}$$

$$\gamma_j(t) := H_{j,xu}(t)\Big(\epsilon\, \Delta u_j^l(t) - \frac{1}{2}\Delta u_j^l(t)\zeta_j^T(t)d_j(t)\Big) \tag{4.90}$$

are calculated with $D_j(t) := I + \frac{1}{2}\,\nabla_u f_j(t)\Delta u_j^l(t)\zeta_j(t)^T$, the identity matrix I, and the abbreviations $f_j(t) := f_j(x^l(t), u_j^l(t))$, $H_{j,xx}(t) := \nabla_{xx}H_j(x^l(t), \lambda^l(t), u_j^l(t))$, $H_{j,xu}(t) := \nabla_{xu}H_j(x^l(t), \lambda^l(t), u_j^l(t))$, and $r_{j-1}(t_j^-) := \nabla_x m_{j-1,j}^T(x^l(t_j))$. The auxiliary function $\zeta_j(t)$ is defined as the linear combination of ζ_j and $\tilde{\zeta}_{j-1}$:

$$\zeta_j(t) := \frac{t - t_j}{t_{j+1} - t_j}\,\zeta_j + \frac{t_{j+1} - t}{t_{j+1} - t_j}\,\tilde{\zeta}_{j-1} \tag{4.91}$$

with $\zeta_{j-1} := -\frac{r_{j-1}(t_j^-)}{r_{j-1}^T(t_j^-)\,f_{j-1}(t_j^-)}$ and $\tilde{\zeta}_{j-1} := \frac{\zeta_{j-1}}{1+\zeta_{j-1}^T(f_{j-1}(t_j^-)-f_j(t_j))}$. The matrices $\Xi_j(t)$, $K_j(t)$, and $L_j(t)$ are determined to

$$\Xi_j(t) = \frac{1}{2}\,\Delta u_j^l(t)\dot{\zeta}_j^T(t) + \Delta\dot{u}_j^l(t)\zeta_j^T(t) + K_j(t) \tag{4.92}$$

$$K_j(t) = P^T(t)\big(P(t)H_{j,uu}^*(t)P^T(t)\big)^{-1}P(t)H_{j,ux}^*(t) \tag{4.93}$$

$$L_j(t) = P^T(t)\big(P(t)H_{j,uu}^*(t)P^T(t)\big)^{-1}P(t)\nabla_u f_j^{*T}(t)\,. \tag{4.94}$$

Here, $H_{j,uu}^*(t) := \nabla_{uu}H_j(x^l(t), \lambda^l(t), u_j^{*l}(t))$ and $f_j^*(t) := f_j(x^l(t), u_j^{*l}(t))$.

Step 1(b) Control Constraints: Whenever $u_j^{*l}(t) \in \mathrm{Int}(U_j)$ and $H_{j,uu}^*(t) > 0$, then $P \in \mathbb{R}^{n_u}$ is the identity matrix I. If the optimal control $u_j^{*l}(t)$ is on the boundary ∂U_j, at first, the orthonormalized subspace $B_{U_j}(t)$ with dimension n_B is found, which is spanned by n_B derivatives $\nabla_u \partial U_j(u_j^*(t))$. Note that there can only exist more than one derivative $\nabla_u \partial U_j(u_j^{*l}(t))$, if $u_j^{*l}(t)$ is on an edge or corner of the boundary ∂U_j and $\nabla_u \partial U_j(u_j^{*l}(t))$ summarizes the different directional derivatives that exist at this edge or corner. Then $P(t)$ is formed by the difference $I - B_{U_j}(t)$ and by canceling n_B of the rows in $I - B_{U_j}(t)$, which are affected by $B_{U_j}(t)$. E.g. in the case of box constraints for the controls, the rows are canceled from I, where the corresponding component of the optimal control $u_j^{*l}(t)$ is on one of its two constraints. If the Hessian $H_{j,uu}^*(t)$ is not positive definite, then the rows in $P(t)$ are canceled, which correspond to the eigenvalues that are zero or negative.

Step 1(c) (Re-)Initialization of Riccati Variables: The backward integrations of the Riccati-matrices and vectors are initialized at t_e with:

$$S_N(t_e) = \nabla_{xx}g(x^l(t_e)) + \nabla_x(\nabla_x \psi_N^T(x^l(t_e))\,\nu^l) \tag{4.95}$$

$$R_N(t_e) = T_N^T(t_e) = \nabla_x \psi_N^T(x^l(t_e)) \tag{4.96}$$

$$Q_N(t_e) = 0 \tag{4.97}$$

$$e_N(t_e) = 0 \tag{4.98}$$

$$\eta_N(t_e) = 0\,. \tag{4.99}$$

In the case that interior constraints $\psi_{j-1}(x^l(t_j))$ are specified at a switching time t_j, the transversality conditions for the Riccati-matrices and vectors are

$$\begin{aligned}
S_{j-1}(t_j^-) &= S_j(t_j) - R_j(t_j)Q_j^{-1}(t_j)T_j(t_j) - r_{j-1}(t_j^-)v_{r,j}^{-1}v_{x,j} \\
&\quad + \nabla_x(\nabla_x \tilde{m}_{j-1}^T(x^l(t_j))\,\tilde{\pi}_j^l) + \Big[\nabla_x H_{j-1}^T(t_j^-) - \nabla_x H_j^T(t_j) \\
&\quad + \big(S_j(t_j) - R_j(t_j)Q_j^{-1}(t_j)T_j(t_j)\big)\big(f_{j-1}(t_j^-) - f_j(t_j)\big) \\
&\quad + \nabla_x(\nabla_x \tilde{m}_{j-1}^T(x^l(t_j))\,\tilde{\pi}_j^l)\, f_{j-1}(t_j^-)\Big]\,\zeta_{j-1}^T && (4.100)\\
R_{j-1}(t_j^-) &= \nabla_x \psi_{j-1}^T(x^l(t_j)) - r_{j-1}(t_j^-)v_{r,j}^{-1}v_{R,j} && (4.101)\\
T_{j-1}(t_j^-) &= \nabla_x \psi_{j-1}(x^l(t_j))(I + f_{j-1}(t_j^-)\,\zeta_{j-1}^T) && (4.102)\\
Q_{j-1}(t_j^-) &= 0 && (4.103)\\
e_{j-1}(t_j^-) &= R_j(t_j)Q_j^{-1}(t_j)(d\psi_k(x^l(t_{k+1})) - \eta_j(t_j)) \\
&\quad + e_j(t_j) - r_{j-1}(t_j^-)v_{r,j}^{-1}v_{e,j} && (4.104)\\
\eta_{j-1}(t_j^-) &= 0\,, && (4.105)
\end{aligned}$$

where the values $v_{x,j}$, $v_{R,j}$, $v_{r,j}$, and $v_{e,j}$ are given in (4.122) - (4.125). The index k is chosen as the minimal index in $\{j, ..., N\}$ with $n_{\psi_k} > 0$, which corresponds to specified interior constraints. In the case that no interior constraints exist at a switching time t_j, the conditions change to

$$\begin{aligned}
S_{j-1}(t_j^-) &= S_j(t_j) - r_{j-1}(t_j^-)v_{r,j}^{-1}v_{x,j} + \nabla_x(\nabla_x m_{j-1,j}^T(x^l(t_j))\,\pi_j^l) \\
&\quad + \Big[\nabla_x H_{j-1}^T(t_j^-) - \nabla_x H_j^T(t_j) + S_j(t_j)\big(f_{j-1}(t_j^-) - f_j(t_j)\big) \\
&\quad + \nabla_x(\nabla_x m_{j-1,j}^T(x^l(t_j))\,\pi_j^l)\, f_{j-1}(t_j^-)\Big]\,\zeta_{j-1}^T && (4.106)\\
R_{j-1}(t_j^-) &= R_j(t_j) - r_{j-1}(t_j^-)v_{r,j}^{-1}v_{R,j} && (4.107)\\
T_{j-1}(t_j^-) &= T_j(t_j)\Big(I + \big(f_{j-1}(t_j^-) - f_j(t_j)\big)\,\zeta_{j-1}^T\Big) && (4.108)\\
Q_{j-1}(t_j^-) &= Q_j(t_j) && (4.109)\\
e_{j-1}(t_j^-) &= e_j(t_j) - r_{j-1}(t_j^-)v_{r,j}^{-1}v_{e,j} && (4.110)\\
\eta_{j-1}(t_j^-) &= \eta_j(t_j)\,. && (4.111)
\end{aligned}$$

Here, $v_{R,j}$ and $v_{e,j}$ are redefined in (4.129) and (4.130). The values of $d\psi_j$ for all $j \in \{0, ..., N\}$ with $n_{\psi_j} > 0$ are set to:

$$d\psi_j = -\epsilon\,\psi_j(x^l(t_{j+1})) \tag{4.112}$$

Store the trajectories of S_j, R_j, T_j, Q_j, A_j, B_j, d_j, e_j, and η_j.

Step 2 Forward Integration: Integrate the linearized state differential equations

$$\delta\dot{x}(t) = A_j(t)\delta x(t) - B_j(t)\delta\lambda(t) - d_j(t) \tag{4.113}$$

forward in time from t_0 to t_e with

$$\delta x(t_0) = 0 \tag{4.114}$$

$$\delta x(t_j) = \Big(I + \big(f_{j-1}(t_j^-) - f_j(t_j)\big)\,\zeta_{j-1}\Big)\,\delta x(t_j^-) \tag{4.115}$$

$$\delta\lambda(t) = S_j(t)\delta x(t) + R_j(t)d\hat{\pi}_{k+1} + e_j(t) \tag{4.116}$$

$$d\hat{\pi}_{k+1} = Q_j^{-1}(t_j)\Big(d\psi_k - T_j(t_j)\delta x(t_j) - \eta_j(t_j)\Big) \tag{4.117}$$

for $k = \min\big(\{k \in \{j, ..., N\} | n_{\psi_k} > 0\}\big)$ and store $d\nu := d\hat{\pi}_{N+1}$, $d\hat{\pi}_{j+1}$, $\delta\chi$ and $\delta\lambda$.

Step 3 Control Update: Determine the update of the control:

$$\delta u(t) = \; -\epsilon\, W_j^{-1}(t)\Delta u_j^l(t) - \tilde{K}_j(t)\delta x(t) - \tilde{L}_j(t)\delta\lambda(t)\,. \tag{4.118}$$

The matrices $W_j(t)$, $\tilde{K}_j(t)$, and $\tilde{L}_j(t)$ are given by

$$W_j(t) = I + \frac{1}{2}\Delta u_j^l(t)\zeta_j^T(t)\nabla_u f_j(t) \tag{4.119}$$

$$\begin{aligned}\tilde{K}_j(t) = W_j^{-1}(t)\Big(&\frac{1}{2}\,\Delta u_j^l(t)\dot{\zeta}_j^T(t) + \Delta\dot{u}_j^l(t)\zeta_j^T(t) + K_j(t)\\ &+ \frac{1}{2}\,\Delta u_j^l(t)\zeta_j^T(t)\nabla_x f_j(t)\Big)\end{aligned} \tag{4.120}$$

$$\tilde{L}_j(t) = W_j^{-1}(t)L_j(t)\,. \tag{4.121}$$

□

Remark 4.4: For purely continuous optimal control problems, $D_j(t)$ becomes the identity matrix and $\Xi_j(t)$ in (4.92) equals $K_j(t)$. The reinitializations in (4.100) - (4.111) are superfluous as well as the errors in the interior conditions $d\psi_j$ in (4.112) and the reinitializations of $\delta x(t_j)$ in (4.115). The expressions for $W_j(t)$, $\tilde{K}_j(t)$, and $\tilde{L}_j(t)$ reduce to I, $K_j(t)$, and $L_j(t)$, respectively. □

Variable Definitions

Here, various variables used as abbreviations in the algorithm are introduced. At switchings with specified interior conditions, the terms $v_{x,j}$, $v_{R,j}$, $v_{r,j}$, and $v_{e,j}$ are given by:

$$\begin{aligned}v_{x,j} = {}& \nabla_x H_{j-1}(t_j^-) - \nabla_x H_j(t_j) + \big(v_{a,j} - v_{b,j}\big)\,v_{c,j}\\ &+ v_{a,j}\,\nabla_x(\nabla_x \tilde{m}_{j-1}^T(x^l(t_j))\,\tilde{\pi}_j^l) - \nabla_u H_{j-1}(t_j^-)\tilde{K}_{j-1}(t_j^-) + \nabla_u H_j(t_j)\tilde{K}_j(t_j)\\ &+ \Big[\Big(\big(v_{a,j} - v_{b,j}\big)\,v_{c,j} - \nabla_x H_j(t_j)\Big)\,\Big(f_{j-1}(t_j^-) - f_j(t_j)\Big)\\ &+ v_{a,j}\Big(\nabla_x(\nabla_x \tilde{m}_{j-1}^T(x^l(t_j))\,\tilde{\pi}_j^l)\,f_{j-1}(t_j^-) + \nabla_x H_{j-1}^T(t_j^-) - \nabla_x H_j^T(t_j)\Big)\\ &+ \nabla_u H_{j-1}(t_j^-)\,\dot{u}_{j-1}^l(t_j^-) - \nabla_u H_j(t_j)\,\dot{u}_j^l(t_j)\Big]\,\zeta_{j-1}^T\end{aligned} \tag{4.122}$$

$$v_{R,j} = v_{a,j}\,\nabla_x\psi_{j-1}^T(x^l(t_j)) \tag{4.123}$$

$$v_{r,j} = v_{a,j}\,r_{j-1}(t_j) \tag{4.124}$$

$$\begin{aligned}\upsilon_{e,j} &= \epsilon \nabla_u H_j(t_j)\Big(1 - \frac{1}{2}\zeta_j^T(t_j) D_j^{-1}(t_j) \nabla_u f_j(t_j) \Delta u_j^l(t_j)\Big)\Delta u_j^l(t_j) \\ &\quad - \epsilon \nabla_u H_{j-1}(t_j^-)\Big(1 - \frac{1}{2}\zeta_{j-1}^T(t_j^-) D_{j-1}^{-1}(t_j^-) \nabla_u f_{j-1}(t_j^-) \Delta u_{j-1}^l(t_j^-)\Big)\Delta u_{j-1}^l(t_j^-) \\ &\quad + \big(\upsilon_{a,j} - \upsilon_{b,j}\big)\Big(R_j(t_j) Q_j^{-1}(t_j)\big(d\psi_k - \eta_j(t_j)\big) + e_j(t_j)\Big) \qquad (4.125)\end{aligned}$$

with $k = \min\big(\{k \in \{j, ..., N\} | n_{\psi_k} > 0\}\big)$ and the auxiliary variables

$$\upsilon_{a,j} = f_{j-1}^T(t_j^-) - \nabla_u H_{j-1}(t_j^-)\tilde{L}_{j-1}(t_j^-) \qquad (4.126)$$

$$\upsilon_{b,j} = f_j^T(t_j) - \nabla_u H_j(t_j)\tilde{L}_j(t_j) \qquad (4.127)$$

$$\upsilon_{c,j} = S_j(t_j) - R_j(t_j) Q_j^{-1}(t_j) T_j(t_j)\,. \qquad (4.128)$$

At switchings without any specified interior conditions, the variables $\upsilon_{R,j}$, $\upsilon_{e,j}$ and $\upsilon_{c,j}$ are redefined:

$$\upsilon_{R,j} = \big(\upsilon_{a,j} - \upsilon_{b,j}\big)\,R_j(t_j) \qquad (4.129)$$

$$\begin{aligned}\upsilon_{e,j} &= \epsilon \nabla_u H_j(t_j)\Big(1 - \frac{1}{2}\zeta_j^T(t_j) D_j^{-1}(t_j) \nabla_u f_j(t_j) \Delta u_j^l(t_j)\Big)\Delta u_j^l(t_j) \\ &\quad - \epsilon \nabla_u H_{j-1}(t_j^-)\Big(1 - \frac{1}{2}\zeta_{j-1}^T(t_j^-) D_{j-1}^{-1}(t_j^-) \nabla_u f_{j-1}(t_j^-) \Delta u_{j-1}^l(t_j^-)\Big)\Delta u_{j-1}^l(t_j^-) \\ &\quad + \big(\upsilon_{a,j} - \upsilon_{b,j}\big)\,e_j(t_j) \qquad (4.130)\end{aligned}$$

$$\upsilon_{c,j} = S_j(t_j)\,. \qquad (4.131)$$

4.3.3 Convergence

In the following, it is shown that Algorithm 4.1 converges globally to a solution satisfying all necessary optimality conditions (4.53) - (4.60). By the design of the Steps 0, 2, and 3 in Algorithm 4.1, it is ensured in every iteration that the necessary optimality conditions (4.53) - (4.57) are satisfied. The remaining conditions (4.58) - (4.60) are iteratively approached by updating the optimization variables ν^l, $\hat{\pi}_j^l$, and $u_j^l(t)$ for all $j \in \{0, ..., N\}$ and all $t \in [t_0, t_e]$. The update is performed by a method, which is similar to a modified Newton method.

The convergence analysis is started at iteration l with trajectories χ^l and λ^l, which fulfill the optimality conditions (4.53) - (4.57). The variations $d\nu$, $d\hat{\pi}_j$, and $\delta\upsilon$ cause variations $\delta\chi$ and $\delta\lambda$ in the trajectories χ^l and λ^l. The variations $d\nu$, $d\hat{\pi}_j$, and $\delta\upsilon$ compensate for these variations such that the satisfaction of the optimality conditions (4.53) - (4.57) is not degraded for the trajectories $\chi^l + \delta\chi$ and $\lambda^l + \delta\lambda$. Here, this is an essential piece in proving convergence to the fulfillment of all optimality conditions (4.53) - (4.60). Additionally, convergence can only be shown if the resulting trajectories χ^{l+1} and λ^{l+1} equal $\chi^l + \delta\chi$ and $\lambda^l + \delta\lambda$, what is proven to be the case in Proposition 4.3:

Proposition 4.3: Let the Assumptions 2.39, 2.11, and 2.12 hold and Algorithm 4.1 to be executed. Let $x^l(t)$, $\lambda^l(t)$, $x^{l+1}(t)$ and $\lambda^{l+1}(t)$ be the solutions from Steps 2 and 3 of the algorithm in iteration l and $l+1$ and let the variations $\delta x(t)$ and $\delta\lambda(t)$ be specified as in Step 1 of iteration l. Then updates of $u^l(t)$, ν^l, and $\hat{\pi}_j^l$ according to (4.69), (4.70), and

(4.71) with the variations $d\nu$, $d\hat{\pi}_j$, and $\delta u(t)$ from (4.117) and (4.118) are such that

$$\lim_{\epsilon\to 0}\frac{1}{\epsilon}\big(x^{l+1}(t)-x^{l}(t)-\delta x(t)\big)=0 \tag{4.132}$$

$$\lim_{\epsilon\to 0}\frac{1}{\epsilon}\big(\lambda^{l+1}(t)-\lambda^{l}(t)-\delta \lambda(t)\big)=0 \tag{4.133}$$

for all $t\in[t_0,t_e]$ and for all $l\geq 0$. □

Proof: At first, Taylor expansions of all equations defining the trajectories χ^{l+1} and λ^{l+1} around χ^l and λ^l are formed. Secondly, it is shown that all resulting equations are taken into account by the approach presented in Step 1 of the algorithm. Finally, the statement of the proposition is proven by showing that the difference between $x^{l+1}(t)$ and $x^l(t)+\delta x(t)$ respectively $\lambda^{l+1}(t)$ and $\lambda^l(t)+\delta\lambda(t)$ is of order $o(\epsilon)$.

Due to the standing assumptions, especially about trajectories being transversal to switching manifolds and the C^2 properties of all functions, the equations that are satisfied by the updated trajectories χ^{l+1} and λ^{l+1} can be approximated with Taylor expansions. Defining the changes $\delta\bar{x}(t)=x^{l+1}(t)-x^l(t)$ and $\delta\bar{\lambda}(t)=\lambda^{l+1}(t)-\lambda^l(t)$, the Taylor expansions of the system dynamics (4.53) and (4.54) are:

$$\begin{aligned}\dot{x}^{l+1}(t) &= f_j(x^{l+1}(t),u^{l+1}(t))\\ &= f_j(x^l(t),u^l(t))+\nabla_x f_j(x^l(t),u^l(t))\,\delta\bar{x}(t)\\ &\quad+\nabla_u f_j(x^l(t),u^l(t))\,\delta u(t)+o(\|\delta\bar{x}\|)+o(\|\delta u\|)\\ &= \dot{x}^l(t)+\delta\dot{\bar{x}}(t) \qquad (4.134)\\ \dot{\lambda}^{l+1}(t) &= -\nabla_x H_j^T(x^{l+1}(t),\lambda^{l+1}(t),u^{l+1}(t))\\ &= -\nabla_x H_j^T(x^l(t),\lambda^l(t),u^l(t))-\nabla_{xx}H_j^T(x^l(t),\lambda^l(t),u^l(t))\,\delta\bar{x}(t)\\ &\quad-\nabla_x f_j^T(x^l(t),u^l(t))\,\delta\bar{\lambda}(t)-\nabla_{xu}H_j^T(x^l(t),\lambda^l(t),u^l(t))\,\delta u(t)\\ &\quad+o(\|\delta\bar{x}\|)+o(\|\delta u\|)+o(\|\delta\bar{\lambda}\|)\\ &= \dot{\lambda}^l(t)+\delta\dot{\bar{\lambda}}(t)\,, \qquad (4.135)\end{aligned}$$

which leads to the linearized differential equations

$$\delta\dot{\bar{x}}(t)=\nabla_x f_j(x^l(t),u^l(t))\,\delta\bar{x}(t)+\nabla_u f_j(x^l(t),u^l(t))\,\delta u(t)+o(\|\delta\bar{x}\|)+o(\|\delta u\|) \tag{4.136}$$

$$\begin{aligned}\delta\dot{\bar{\lambda}}(t)= &-\nabla_{xx}H_j^T(x^l(t),\lambda^l(t),u^l(t))\,\delta\bar{x}(t)-\nabla_x f_j^T(x^l(t),u^l(t))\,\delta\bar{\lambda}(t)\\ &-\nabla_{xu}H_j^T(x^l(t),\lambda^l(t),u^l(t))\,\delta u(t)+o(\|\delta\bar{x}\|)+o(\|\delta u\|)+o(\|\delta\bar{\lambda}\|)\,. \qquad (4.137)\end{aligned}$$

The related boundary conditions $x(t_0)=0$, and (4.55) - (4.57) are expressed by a Taylor series:

$$0=x^{l+1}(t_0)=x^l(t_0)+\delta\bar{x}(t_0)=\delta\bar{x}(t_0) \tag{4.138}$$

$$\begin{aligned}0&=\lambda^{l+1}(t_e)-\nabla_x g^T(x^{l+1}(t_e))-\nabla_x\psi_N^T(x^{l+1}(t_e))\,\nu^{l+1}\\ &=\lambda^l(t_e)+\delta\bar{\lambda}(t_e)-\nabla_x g^T(x^l(t_e))-\nabla_x\psi_N^T(x^l(t_e))\,\nu^l-\nabla_x\psi_N^T(x^l(t_e))\,d\nu\\ &\quad-\Big(\nabla_{xx}g(x^l(t_e))+\nabla_x(\nabla_x\psi_N^T(x^l(t_e))\,\nu^l)\Big)\,\delta\bar{x}(t_e)+o(\|\delta\bar{x}\|) \qquad (4.139)\end{aligned}$$

$$\begin{aligned}
0 &= \lambda^{l+1}(t_j^-) - \lambda^{l+1}(t_j) - \nabla_x \tilde{m}_{j-1}^T(x^{l+1}(t_j))\, \tilde{\pi}_j^{l+1} \\
&= \lambda^l(t_j^-) + d\bar{\lambda}(t_j^-) - \lambda^l(t_j) - d\bar{\lambda}(t_j) - \nabla_x \tilde{m}_{j-1}^T(x^l(t_j))\, \tilde{\pi}_j^l \\
&\quad - \nabla_x(\nabla_x \tilde{m}_{j-1}(t_j)\, \tilde{\pi}_j^l)\, d\bar{x}(t_j) - \nabla_x \tilde{m}_{j-1}^T(x^l(t_j))\, d\tilde{\pi}_j + o(\|\delta\bar{x}\|) + o(|dt_j|)
\end{aligned} \tag{4.140}$$

$$\begin{aligned}
0 &= H_{j-1}(x^{l+1}(t_j), \lambda^{l+1}(t_j^-), u_{j-1}^{l+1}(t_j^-)) - H_j(x^{l+1}(t_j), \lambda^{l+1}(t_j), u_j^{l+1}(t_j)) \\
&= H_{j-1}(x^l(t_j), \lambda^l(t_j^-), u_{j-1}^l(t_j^-)) + dH_{j-1}(x^l(t_j), \lambda^l(t_j^-), u_{j-1}^l(t_j^-)) \\
&\quad - H_j(x^l(t_j), \lambda^l(t_j), u_j^l(t_j)) - dH_j(x^l(t_j), \lambda^l(t_j), u_j^l(t_j)) \\
&\quad + o(\|\delta\bar{x}\|) + o(\|\delta u\|) + o(\|\delta\bar{\lambda}\|) + o(|dt_j|)\,.
\end{aligned} \tag{4.141}$$

Now, the variations $\delta x(t)$ and $\delta\lambda(t)$ are introduced. They satisfy Eq. (4.136) - (4.141) without considering the higher order terms. Consequently, it can be concluded that

$$\delta\bar{x}(t) = \delta x(t) + o(\|\delta\bar{x}\|) + o(\|\delta u\|) + o(|dt_j|) \tag{4.142}$$

$$\delta\bar{\lambda}(t) = \delta\lambda(t) + o(\|\delta\bar{x}\|) + o(\|\delta u\|) + o(\|\delta\bar{\lambda}\|) + o(|dt_j|) \tag{4.143}$$

and by recursive application of (4.142) and (4.143) the residuals $o(\|\delta\bar{x}\|)$ and $o(\|\delta\bar{\lambda}\|)$ can be replaced by $o(\|\delta x\|)$ and $o(\|\delta\lambda\|)$. Using $\delta u(t)$ from (4.118), the error $o(\|\delta u\|)$ is split into the parts $o(\epsilon)$, $o(\|\delta x\|)$, and $o(\|\delta\lambda\|)$, which is possible in the case of dealing with orders despite of the norm $\|\delta u\|$. The differential equations for $\delta x(t)$ and $\delta\lambda(t)$ are obtained from (4.136), (4.137), and (4.118):

$$\delta\dot{x}(t) = A_j(t)\, \delta x(t) - B_j(t)\delta\lambda(t) - d_j(t) \tag{4.144}$$

$$\delta\dot{\lambda}(t) = \; - C_j(t)\, \delta x(t) - F_j(t)\delta\lambda(t) + \gamma_j(t) \tag{4.145}$$

for all j and t and with $A_j(t)$, $B_j(t)$, $C_j(t)$, $F_j(t)$, $d_j(t)$, and $\gamma_j(t)$ specified in (4.85) - (4.90). The corresponding boundary conditions are:

$$\delta x(t_0) = 0 \tag{4.146}$$

$$\delta\lambda(t_e) = \Big(\nabla_{xx} g(x^l(t_e)) + \nabla_x(\nabla_x \psi_N^T(x^l(t_e))\, \nu^l)\Big)\, \delta x(t_e) + \nabla_x \psi_N^T(x^l(t_e))\, d\nu \tag{4.147}$$

$$\begin{aligned}
d\lambda(t_j^-) &= d\lambda(t_j) + \nabla_x(\nabla_x \tilde{m}_{j-1}(t_j)\, \tilde{\pi}_j^l)\, dx(t_j) + \nabla_x \psi_{j-1}^T(x^l(t_j))\, d\hat{\pi}_j \\
&\qquad + \nabla_x m_{j-1,j}^T(x^l(t_j))\, d\pi_j \qquad \forall j \in \{1, ..., N\}
\end{aligned} \tag{4.148}$$

$$dH_{j-1}(t_j^-) = dH_j(t_j)\,, \tag{4.149}$$

where at switching times the approximations of first order

$$d\lambda(t_j^-) = \delta\lambda(t_j^-) + \dot{\lambda}(t_j^-)\, dt_j \tag{4.150}$$

$$d\lambda(t_j) = \delta\lambda(t_j) + \dot{\lambda}(t_j)\, dt_j \tag{4.151}$$

$$\begin{aligned}
dx(t_j) &= \delta x(t_j^-) + \dot{x}(t_j^-)\, dt_j \\
&= \delta x(t_j) + \dot{x}(t_j)\, dt_j
\end{aligned} \tag{4.152}$$

hold. The equations (4.144) - (4.152) form a linear multi-point boundary-value problem (MPBVP). For the solution of the MPBVP, a two-step, non-iterative procedure is chosen in the algorithm, which is based on a Riccati-approach. First, the Riccati-matrices S_j, R_j, T_j,

Q_j, e_j, and η_j are integrated backwards. In a second step, the linearized dynamics (4.144) are integrated forwards to find $\delta\chi$ and the terms $\delta\lambda$ and $d\hat{\pi}_j$ are determined simultaneously by the matrices S_j, R_j, T_j, Q_j, e_j, η_j, and $\delta\chi$. To avoid an iterative procedure for increased computational speed, all unknown variables except for $\delta x(t)$ need to be replaced in the backward integration by expressions only depending on known variables and $\delta x(t)$.

In the Riccati-approach, variations in the adjoint variable $\delta\lambda(t)$ and in the interior and terminal constraints $d\psi_j$ are expressed as functions of the variation in the state $\delta x(t)$ and the multipliers $d\hat{\pi}_{j+1}$ and $d\nu$:

$$\delta\lambda(t) = S_j(t)\,\delta x(t) + R_j(t)\,d\hat{\pi}_{k+1} + e_j(t) \tag{4.153}$$

$$d\psi_k = T_j(t)\,\delta x(t) + Q_j(t)\,d\hat{\pi}_{k+1} + \eta_j(t) \tag{4.154}$$

for all $j \in \{0, ..., N\}$, $t \in [t_j, t_{j+1})$, and $k = \min\big(\{k \in \{j, ..., N\} | n_{\psi_k} > 0\}\big)$. Forming the absolute derivative with respect to time

$$\delta\dot{\lambda}(t) = \dot{S}_j(t)\,\delta x(t) + S_j(t)\,\delta\dot{x}(t) + \dot{R}_j(t)\,d\hat{\pi}_{k+1} + \dot{e}_j(t) \tag{4.155}$$

$$0 = \dot{T}_j(t)\,\delta x(t) + T_j(t)\,\delta\dot{x}(t) + \dot{Q}_j(t)\,d\hat{\pi}_{k+1} + \dot{\eta}_j(t)\,, \tag{4.156}$$

inserting the differential equations (4.144) and (4.145) and the expression for $\delta\lambda(t)$ (4.153), and regrouping terms

$$\begin{aligned} 0 = {} & \Big[\dot{S}_j(t) + C_j(t) + F_j(t)\,S_j(t) + S_j(t)\,A_j(t) - S_j(t)\,B_j(t)\,S_j(t)\Big]\,\delta x(t) \\ & + \Big[\dot{R}_j(t) + F_j(t)\,R_j(t) - S_j(t)\,B_j(t)\,R_j(t)\Big]\,d\hat{\pi}_{k+1} \\ & + \Big[\dot{e}_j(t) - \gamma_j(t) + (F_j(t) - S_j(t)\,B_j(t))\,e_j(t) - S_j(t)\,d_j(t)\Big] \qquad (4.157) \\ 0 = {} & \Big[\dot{T}_j(t) + T_j(t)\,A_j(t) - T_j(t)\,B_j(t)\,S_j(t)\Big]\,\delta x(t) \\ & + \Big[\dot{Q}_j(t) - T_j(t)\,B_j(t)\,R_j(t)\Big]\,d\hat{\pi}_{k+1} + \Big[\dot{\eta}_j(t) - T_j(t)\,(B_j(t)\,e_j(t) + d_j(t))\Big]\,, \qquad (4.158) \end{aligned}$$

the system of Riccati differential equations (4.79) - (4.84) is found. Here, all brackets have to be zero to ensure for any $\delta x(t)$ and $d\hat{\pi}_{k+1}$ that the equations become zero. Analogously, the same result is obtained in the last discrete state q_N. By (4.153) and (4.154) evaluated at $j = N$, (4.147), and $d\psi_N = \nabla_x\psi_N(x(t_e))\delta x(t_e)$, the terminal conditions (4.95) - (4.99) follow immediately. Starting from the terminal conditions, the differential equations (4.79) - (4.84) can be integrated backward from t_e to t_N. Next, the terminal values of the Riccati-matrices $S_{N-1}(t_N^-)$, $R_{N-1}(t_N^-)$, $T_{N-1}(t_N^-)$, $Q_{N-1}(t_N^-)$, $e_{N-1}(t_N^-)$, and $\eta_{N-1}(t_N^-)$ have to be found. There, two cases have to be distinguished: (i) If interior conditions are specified at t_N ($n_{\psi_{N-1}} > 0$), then the time span $[t_N, t_e]$ is used to approach the terminal conditions $\psi_N(x(t_e)) = 0$ and in the preceding interval $[t_{N-1}, t_N]$ the interior condition $\psi_{N-1}(x(t_N)) = 0$ is approached. Therefore, (4.154) is evaluated at t_N and solved for $d\nu := d\hat{\pi}_{N+1}$ (4.117) if $Q_N(t_N)$ is invertible, which is assumed here.[4] The linearized

[4] The assumption is reasonable since the structure of $Q_j(t_j)$ and the associated differential equations is almost quadratic and the assumption was always satisfied in simulation examples. Note that it is sufficient if $Q_j(t)$ can be inverted for some time $t \in [t_j, t_{j+1})$.

transversality condition (4.148) is expressed only in terms of $\delta x(t_N^-)$, $d\hat{\pi}_N$, and the known values from discrete state N by the approach

$$\delta\lambda(t_N^-) = S_{N-1}(t_N^-)\,\delta x(t_N^-) + R_{N-1}(t_N^-)\,d\hat{\pi}_N + e_{N-1}(t_N^-)\,. \tag{4.159}$$

Subsequently, (4.148) is grouped into terms depending on $\delta x(t_N^-)$, $d\hat{\pi}_N$, and none of both, which delivers the reinitialization conditions (4.100), (4.101), and (4.104). The values for $T_{N-1}(t_N^-)$, $Q_{N-1}(t_N^-)$, and $\eta_{N-1}(t_N^-)$ as in (4.102), (4.103), and (4.105) are determined by comparing (4.154) with

$$d\psi_{N-1} = \nabla_x \psi_{N-1}(x^l(t_N))\,(\delta x(t_N^-) + \dot{x}(t_N^-)\,dt_N)\,. \tag{4.160}$$

(ii) In the second case, no interior conditions are specified at t_j and the entire interval $[t_{N-1}, t_e]$ is used to converge to the terminal condition $\psi_N(x(t_e)) = 0$. Here, the approach

$$\delta\lambda(t_N^-) = S_{N-1}(t_N^-)\,\delta x(t_N^-) + R_{N-1}(t_N^-)\,d\hat{\pi}_{N+1} + e_{N-1}(t_N^-) \tag{4.161}$$

is chosen and the transversality condition (4.148) is again regrouped, such that the Eq. (4.106), (4.107), and (4.110) are obtained. The expression for the desired change in the terminal condition $d\psi_N$ is now continued in discrete state $N-1$

$$\begin{aligned} d\psi_N &= T_N(t_N)\delta x(t_N) + Q_N(t_N)d\hat{\pi}_{N+1} + \eta_N(t_N) \\ &= T_{N-1}(t_N^-)\Big(\delta x(t_N^-) + \big(\dot{x}(t_N^-) - \dot{x}(t_N)\big)\,dt_N\Big) + Q_{N-1}(t_N^-)d\hat{\pi}_{N+1} + \eta_{N-1}(t_N^-)\,, \end{aligned} \tag{4.162}$$

which leads to (4.108), (4.109), and (4.111). Both cases require to find expressions for dt_N and $d\pi_N$ in dependence of the known values. The forward integration of Step 1 in Algorithm 4.1 has been stopped when hitting the switching manifold $m_{N-1,N}(x^l(t_N)) = 0$ and for sufficiently small $dx(t_N)$, the integration will again stop on the switching manifold. Therefore, the change $dm_{N-1,N}$ for the absolute variation $dx(t_N)$ is zero:

$$dm_{N-1,N} = \nabla_x m_{N-1,N}(x^l(t_N))\,(\delta x(t_N^-) + \dot{x}(t_N^-)\,dt_N) = 0\,. \tag{4.163}$$

With $r_{N-1}(t_N^-) = \nabla_x m_{N-1,N}^T(x^l(t_N))$, this leads in an approximation of first order to

$$dt_N = \zeta_{N-1}^T\,\delta x(t_N^-) = -\frac{r_{N-1}^T(t_N^-)\,\delta x(t_N^-)}{r_{N-1}^T(t_N^-)\,f_{N-1}(t_N^-)}\,. \tag{4.164}$$

Putting the terms for dt_N, $d\nu$, $d\lambda(t_N^-)$, $d\lambda(t_N)$, $dx(t_N)$, and $\delta\lambda(t)$ into the linearized Hamiltonian continuity condition (4.149), the unknown

$$d\pi_N = -v_{r,N}^{-1}\Big(v_{x,N}\,\delta x(t_N^-) + v_{R,N}\,d\hat{\pi}_N + v_{e,N}\Big) \tag{4.165}$$

is calculated. The procedure can be repeated until t_0 is reached and with the obtained values of $S_j(t)$, $R_j(t)$, $T_j(t)$, $Q_j(t)$, $e_j(t)$, and $\eta_j(t)$ for $t \in [t_0, t_e]$ the solution trajectories $\delta\chi$ and $\delta\lambda$ of the MPBVP (4.144) - (4.149) are found.

The next step in the proof is to show that the variations $\delta x(t)$ and $\delta\lambda(t)$ are scaled with

ϵ. Let ϵ approach zero, then the terms $d_j(t)$, $\gamma_j(t)$, and $d\psi_j$ also immediately approach zero, compare (4.89), (4.90), and (4.112). This implies that $e_j(t)$, $\eta_j(t)$, and $v_{e,j}$ go to zero as well from (4.83), (4.84), (4.98), (4.99), (4.104), (4.105), (4.125), and (4.130). Bringing the system of boundary conditions (4.146) - (4.148) and (4.152) into matrix-vector notation

$$B^+ \begin{pmatrix} \delta x(t_0) \\ \delta\lambda(t_0) \\ \delta x(t_1) \\ \vdots \end{pmatrix} + B^- \begin{pmatrix} \delta x(t_1^-) \\ \delta\lambda(t_1^-) \\ \delta x(t_2^-) \\ \vdots \end{pmatrix} = 0 \tag{4.166}$$

and considering the differential equations (4.144) and (4.145) of the linear MPBVP, it can be observed that one solution is:

$$\delta x(t) = 0 \tag{4.167}$$

$$\delta\lambda(t) = 0 \tag{4.168}$$

for all $t \in [t_0, t_e]$ and $\epsilon \to 0$. Though Theorem 6.1 in [6] can be extended to linear MPBVPs and the matrices $A_j(t)$ to $F_j(t)$ are piecewise continuous in all intervals, the uniqueness of the solution cannot be shown in general. This is the case since the term $B^+ + B^-(\Phi_B^T(t_1^-, t_0) \;\; \Phi_B^T(t_2^-, t_1) \;\; ...)^T$ may become singular in pathological cases, where the fundamental solution is given by $\Phi_B(t_j, t_j) = I \in \mathbb{R}^{2n_x \times 2n_x}$, $j \in \{0, ..., N\}$, and

$$\frac{d\Phi_B(t, t_j)}{dt} = \begin{pmatrix} A_j(t) & -B_j(t) \\ -C_j(t) & -D_j(t) \end{pmatrix} \Phi_B(t, t_j)\,. \tag{4.169}$$

However, it can be shown that the zero solution is the one that the algorithm picks in all cases if ϵ approaches zero. With (4.146) and (4.117), it follows that $d\hat{\pi}_{k+1}$, $k = \min\big(\{k \in \{0, ..., N\} | n_{\psi_k} > 0\}\big)$, is zero, which leads to $\delta\lambda(t_0) = 0$ from (4.116). Thus, $\delta x(t)$ and $\delta\lambda(t)$ remain zero during the subsequent integration. Following the procedure, the zero solution results. Consequently, we can conclude that:

$$\lim_{\epsilon \to 0} \|\delta x(t)\| = 0 \tag{4.170}$$

$$\lim_{\epsilon \to 0} \|\delta\lambda(t)\| = 0 \tag{4.171}$$

for all $t \in [t_0, t_e]$. Augmenting the equations with $\frac{\|\delta x(t)\|}{\|\delta x(t)\|}$ and $\frac{\|\delta\lambda(t)\|}{\|\delta\lambda(t)\|}$ respectively, the relations

$$\lim_{\epsilon \to 0} \frac{o(\|\delta x(t)\|)}{\|\delta x(t)\|} = 0 \tag{4.172}$$

$$\lim_{\epsilon \to 0} \frac{o(\|\delta\lambda(t)\|)}{\|\delta\lambda(t)\|} = 0 \tag{4.173}$$

are found. This enables us to replace all residuals $o(\|\delta x\|)$ and $o(\|\delta\lambda\|)$ in (4.142) and (4.143) by $o(\epsilon)$. Note that $o(|dt_j|)$ depends on $o(\|\delta x\|)$ since dt_j is a function of $\delta x(t_j^-)$.

Finally, the statement of the proposition can be proven:

$$\lim_{\epsilon \to 0} \frac{1}{\epsilon}\big(x^{l+1}(t) - x^l(t) - \delta x(t)\big) = \lim_{\epsilon \to 0} \frac{1}{\epsilon}\big(\delta x(t) + o(\epsilon) - \delta x(t)\big) = 0 \tag{4.174}$$

$$\lim_{\epsilon \to 0} \frac{1}{\epsilon}\big(\lambda^{l+1}(t) - \lambda^l(t) - \delta \lambda(t)\big) = \lim_{\epsilon \to 0} \frac{1}{\epsilon}\big(\delta \lambda(t) + o(\epsilon) - \delta \lambda(t)\big) = 0\,. \tag{4.175}$$

■

Next, it is shown that the satisfaction of the optimality conditions (4.58) - (4.60) is iteratively approached by the choice of the updates $d\nu$, $d\hat{\pi}_j$, and $\delta\upsilon$. This is achieved by deriving the boundedness of the function Ω used in the convergence criterion (4.76) and the sufficient descent property of the algorithm to an optimal solution.

Lemma 4.4: Let the standing assumptions hold. Then, the function Ω (4.78) is bounded for any feasible trajectory χ and λ with initial condition x_0 at t_0 and finite final time t_e.□

Proof: From Assumption 2.39.5, $f_j(x(t), u_j(t))$ is Lipschitz continuous and bounded with respect to its arguments. This means that for all $j \in \{0, ..., N\}$ and $u_j(t) \in U_j$, the trajectory χ is in a compact set. By definition, the optimal control $u_j^*(t)$ is also in U_j. Thus, the integral part of the convergence criterion Ω (4.78) is bounded. Since the constraints $\psi_j(x^l(t_{j+1}))$ are also bounded with respect to their arguments due to their C^2-property, ϖ (4.77) is bounded as well. Consequently, $|\Omega|$ is bounded from above by a constant independent of iteration l. ■

Definition 4.5: Algorithm 4.1 has *sufficient descent* if for every iteration l and every $\kappa > 0$, there exists $\rho > 0$ such that, if $\Omega(\chi^l, \lambda^l, \upsilon^l) - \Omega(\chi^*, \lambda^*, \upsilon^*) > \kappa$, then

$$\Omega(\chi^{l+1}, \lambda^{l+1}, \upsilon^{l+1}) - \Omega(\chi^l, \lambda^l, \upsilon^l) \leq -\rho\,. \tag{4.176}$$

□

Proposition 4.5: Consider the standing assumptions to be fulfilled. If Algorithm 4.1 has sufficient descent, then it is convergent. □

Proof: If the algorithm has sufficient descent, then in every iteration l the convergence criterion Ω will decrease by a non-zero amount. As $\Omega(\chi^l, \lambda^l, \upsilon^l)$ is bounded, it must converge to $\Omega(\chi^*, \lambda^*, \upsilon^*)$. ■

In the following, it is shown that the Armijo-condition (4.76) always holds.

Proposition 4.6: Consider the standing assumptions to hold, let $Q_j(t_j)$ be invertible for all $j \in \{0, ..., N\}$, and assume that υ_j^{*l+1} does not vanish, i.e. $u_j^{*l+1}(t)$ is approximated in first order by $u_j^{*l}(t) - K_j(t)\delta x(t) - L_j(t)\lambda(t)$ for all $t \in [t_j, t_{j+1})$, compare (4.185). Then there exists a constant $p \in \mathbb{N}_0$ in every iteration l with $\tau_0 \in (0, 1)$ and $\beta \in (0, 1)$, such that

$$\Omega(\chi^{l+1}, \lambda^{l+1}, \upsilon^{l+1}) - \Omega(\chi^l, \lambda^l, \upsilon^l) \leq -2\tau_0\, \beta^p\, \Omega(\chi^l, \lambda^l, \upsilon^l)\,, \tag{4.177}$$

where $\Omega(\chi^l, \lambda^l, \upsilon^l)$ is as specified in (4.78). □

Proof: In the proof, it is shown for the terminal condition $\|\psi_N(x^l(t_e))\|^2$, every specified interior condition $\|\psi_j(x^l(t_{j+1}))\|^2$, and the integrated Hamiltonian minimization condition $\sum_{j=0}^{N}\int_{t_j}^{t_{j+1}}\|\Delta u_j^l(t)\|^2\,dt$ separately that the corresponding descent condition in $\Omega(\chi^l,\lambda^l,\upsilon^l)$ holds. Finally, by adding the interior, terminal, and Hamiltonian conditions up, the proposition is proven.

At first, it is derived that in every iteration l and for every $j \in \{0,...,N\}$ with $n_{\psi_j} > 0$ there exists a $p \in \mathbb{N}_0$, such that

$$\frac{1}{2}\Big(\|\psi_j(x^{l+1}(t_{j+1}^{l+1}))\|^2 - \|\psi_j(x^l(t_{j+1}^l))\|^2\Big) \le -\tau_0\beta^p\|\psi_j(x^l(t_{j+1}^l))\|^2\,. \tag{4.178}$$

The proof is given by contradiction following the concept in [58]. Let us assume that there does not exist a $p \in \mathbb{N}_0$, such that Eq. (4.178) holds. Thus, the equation below is assumed to be true:

$$\frac{1}{2}\Big(\|\psi_j(x^{l+1}(t_{j+1}^{l+1}))\|^2 - \|\psi_j(x^l(t_{j+1}^l))\|^2\Big) > -\tau_0\beta^p\|\psi_j(x^l(t_{j+1}^l))\|^2\,. \tag{4.179}$$

Consider that interior conditions are specified at t_j and at t_{k+1} with $k = \min\big(\{k \in \{j,...,N\} | n_{\psi_{k+1}} > 0\}\big)$ and none are given in-between t_j and t_{k+1}. Replacing j by k in (4.179), dividing both sides by β^p, and passing p to ∞, the derivative of the left side is obtained:

$$\begin{aligned}
&\lim_{p\to\infty}\frac{1}{\beta^p}\frac{1}{2}\Big(\|\psi_k(x^{l+1}(t_{k+1}^{l+1}))\|^2 - \|\psi_k(x^l(t_{k+1})^l)\|^2\Big)\\
&\quad= \lim_{p\to\infty}\frac{1}{\beta^p}\psi_k^T(x^l(t_{k+1}^l))\Big(\nabla_x\psi_k(x^l(t_{k+1}^l))\,dx(t_{k+1}^l) + o(\beta^p)\Big)\\
&\quad\overset{(4.152),(4.164)}{=} \lim_{p\to\infty}\frac{1}{\beta^p}\psi_k^T(x^l(t_{k+1}^l))\,\nabla_x\psi_k(x^l(t_{k+1}^l))\,\big(I + f_k(t_{k+1}^{l-})\,\zeta_{k+1}^T\big)\\
&\qquad\Big[\prod_{i=j+1}^{k}\big(I + (f_{i-1}(t_i^{l-}) - f_i(t_i^l))\,\zeta_i^T\big)\,\delta x(t_j^l)\\
&\qquad+\sum_{i=j}^{k}\Big(\prod_{r=i+1}^{k}\big(I + (f_{r-1}(t_r^{l-}) - f_r(t_r^l))\,\zeta_r^T\big)\int_{t_i^l}^{t_{i+1}^l}\delta\dot{x}(t)\,dt\Big)\Big] + \lim_{p\to\infty}\frac{1}{\beta^p}\,o(\beta^p)\\
&\quad\overset{(4.102),(4.113),(4.116)}{=} \lim_{p\to\infty}\frac{1}{\beta^p}\psi_k^T(x^l(t_{k+1}^l))\,T_k(t_{k+1}^{l-})\\
&\qquad\Big[\prod_{i=j+1}^{k}\big(I + (f_{i-1}(t_i^{l-}) - f_i(t_i^l))\,\zeta_i^T\big)\,\delta x(t_j^l)\\
&\qquad+\sum_{i=j}^{k}\Big(\prod_{r=i+1}^{k}\big(I + (f_{r-1}(t_r^{l-}) - f_r(t_r^l))\,\zeta_r^T\big)\\
&\qquad\int_{t_i^l}^{t_{i+1}^l}(A_i - B_iS_i)\,\delta x - B_iR_id\hat{\pi}_{k+1} - B_ie_i - d_i\,dt\Big)\Big]\,,
\end{aligned} \tag{4.180}$$

where the dependence on time t is omitted and $\prod_{r=k+1}^{k}(.) = I$. The state transition matrix is defined with the properties $\Phi_i(t_i, t_i) = I$, $\Phi_i(t, t_i) = \Phi_i(t, t')\,\Phi_i(t', t_i)$, and

$$\frac{d}{dt}\Phi_i(t, t_i) = \big(A_i(t) - B_i(t)S_i(t)\big)\,\Phi_i(t, t_i)\,. \tag{4.181}$$

With these properties, the following can be derived:

$$\begin{aligned}
\frac{d}{dt}\Phi_i(t_{i+1}, t) &= \lim_{\delta \to 0} \frac{\Phi_i(t_{i+1}, t+\delta) - \Phi_i(t_{i+1}, t)}{\delta} \\
&= \lim_{\delta \to 0} \frac{\Phi_i(t_{i+1}, t+\delta)\big(I - \Phi_i(t+\delta, t)\big)}{\delta} \\
&= \lim_{\delta \to 0} \frac{-\Phi_i(t_{i+1}, t+\delta)\big(\Phi_i(t+\delta, t) - \Phi_i(t, t)\big)}{\delta} \\
&= -\,\Phi_i(t_{i+1}, t)\,\big(A_i(t) - B_i(t)S_i(t)\big)\,.
\end{aligned} \tag{4.182}$$

Eq. (4.180) can be further modified by using (4.182), (4.81), and (4.117):

$$\begin{aligned}
&\lim_{p \to \infty} \frac{1}{\beta^p}\frac{1}{2}\Big(\|\psi_k(x^{l+1}(t^{l+1}_{k+1}))\|^2 - \|\psi_k(x^l(t^l_{k+1}))\|^2\Big) \\
&\quad = \lim_{p \to \infty} \frac{1}{\beta^p}\,\psi_k^T(x^l(t^l_{k+1}))\,\Big(T_j(t^l_j)\,\delta x(t^l_j) - \sum_{i=j}^{k}\int_{t^l_i}^{t^l_{i+1}} T_i(B_i e_i + d_i)\,dt \\
&\qquad - \sum_{i=j}^{k}\Big[\int_{t^l_i}^{t^l_{i+1}} T_i B_i R_i\,dt\Big]\,Q_j^{-1}(t^l_j)\big(d\psi_k - T_j(t^l_j)\,\delta x(t^l_j) - \eta_j(t^l_j)\big)\Big) \\
&\quad \overset{(4.82),(4.84),(4.103),(4.105)}{=} \lim_{p \to \infty} \frac{1}{\beta^p}\psi_k^T(x^l(t^l_{k+1}))\,\Big(T_j(t^l_j)\,\delta x(t^l_j) \\
&\qquad - \sum_{i=j}^{k}\int_{t^l_i}^{t^l_{i+1}} \dot{\eta}_i\,dt - (-Q_j(t^l_j))\,Q_j^{-1}(t^l_j)\big(d\psi_k - T_j(t^l_j)\,\delta x(t^l_j) + \eta_k(t^{l-}_{k+1}) - \eta_j(t^l_j)\big)\Big) \\
&\quad \overset{(4.112)}{=} \lim_{p \to \infty} \frac{1}{\beta^p}\psi_k^T(x^l(t^l_{k+1}))\,\big(-\,\beta^p\psi_k(x^l(t^l_{k+1}))\big) \\
&\quad = -\,\|\psi_k(x^l(t^l_{k+1}))\|^2 < -\tau_0\,\|\psi_k(x^l(t^l_{k+1}))\|^2\,,
\end{aligned} \tag{4.183}$$

which contradicts our assumption (4.179). Consequently, (4.178) is proven. The derivation can be repeated analogously for all $j \in \{0, ..., N\}$ with $n_{\psi_j} > 0$.

Next, it is shown that there exists a $p \in \mathbb{N}_0$ such that

$$\frac{1}{2}\sum_{j=0}^{N}\Big(\int_{t^{l+1}_j}^{t^{l+1}_{j+1}}\|\Delta u^{l+1}_j(t)\|^2\,dt - \int_{t^l_j}^{t^l_{j+1}}\|\Delta u^l_j(t)\|^2\,dt\Big) \le -\tau_0\beta^p\sum_{j=0}^{N}\int_{t^l_j}^{t^l_{j+1}}\|\Delta u^l_j(t)\|^2\,dt\,. \tag{4.184}$$

That (4.184) holds is again shown by contradiction. That means it is assumed that no $p \in \mathbb{N}_0$ exists such that (4.184) is true. The change from the optimal control $u^{*l}_j(t)$ to $u^{*l+1}_j(t)$ is given by

$$\delta u^*(t) = -K_j(t)\delta x(t) - L_j(t)\delta\lambda(t) + o(\beta^p)\,. \tag{4.185}$$

The applied control in iteration $l+1$ is:

$$u_j^{l+1}(t) = u_j^l(t) + \delta u(t)\,, \tag{4.186}$$

where $\delta u(t)$ from (4.118) can be rewritten with (4.119) - (4.121), (4.136) and (4.142) to

$$\begin{aligned} \delta u(t) = \; & -\beta^p \Delta u_j^l(t) - \Big(K_j(t) + \frac{1}{2}\,\Delta u_j^l(t)\dot{\zeta}_j^T(t) + \Delta\dot{u}_j^l(t)\zeta_j^T(t)\Big)\delta x(t) \\ & - \frac{1}{2}\,\Delta u_j^l(t)\zeta_j^T(t)\delta\dot{x}(t) - L_j(t)\delta\lambda(t)\,. \end{aligned} \tag{4.187}$$

The product

$$\begin{aligned} \Delta u_j^{l+1}(t)^T \Delta u_j^{l+1}(t) = {} & (1-2\beta^p)\Delta u_j^l(t)^T\Delta u_j^l(t) - \Delta u_j^l(t)^T\Delta u_j^l(t)\zeta_j^T(t)\delta\dot{x}(t) \\ & - \Delta u_j^l(t)^T\Big(\,\Delta u_j^l(t)\dot{\zeta}_j^T(t) + 2\Delta\dot{u}_j^l(t)\zeta_j^T(t)\Big)\delta x(t) + o(\beta^p) \end{aligned} \tag{4.188}$$

is determined and the antiderivative of the two middle terms is calculated to be

$$\begin{aligned} \frac{d}{dt}\Big(\Delta u_j^l(t)^T\Delta u_j^l(t)\,\zeta_j^T(t)\delta x(t)\Big) = {} & \Delta u_j^l(t)^T\Delta u_j^l(t)\zeta_j^T(t)\delta\dot{x}(t) \\ & + \Delta u_j^l(t)^T\Big(\,\Delta u_j^l(t)\dot{\zeta}_j^T(t) + 2\Delta\dot{u}_j^l(t)\zeta_j^T(t)\Big)\delta x(t)\,. \end{aligned} \tag{4.189}$$

Note that $dt_j = \zeta_j^T(t_j^l)\delta x(t_j^l)$, $dt_{j+1} = \zeta_j^T(t_{j+1}^{l-})\delta x(t_{j+1}^{l-})$, and $\lim_{p\to\infty}\|\delta x(t)\| = 0$. Dividing (4.184) by β^p and forming the limit, the following is obtained:

$$\begin{aligned} & \lim_{p\to\infty}\frac{1}{2\beta^p}\sum_{j=0}^{N}\Big(\int_{t_j^l+dt_j}^{t_{j+1}^l+dt_{j+1}}\|\Delta u_j^{l+1}(t)\|^2\,dt - \int_{t_j^l}^{t_{j+1}^l}\|\Delta u_j^l(t)\|^2\,dt\Big) \\ & \overset{(4.188),(4.189)}{=} \lim_{p\to\infty}\frac{1}{2\beta^p}\sum_{j=0}^{N}\Big(\int_{t_j^l}^{t_{j+1}^l}(1-2\beta^p)\|\Delta u_j^l(t)\|^2 - \|\Delta u_j^l(t)\|^2 + o(\beta^p)\,dt + o(\beta^p) \\ & \qquad - \Big[\|\Delta u_j^l(t)\|^2\,\zeta_j^T(t)\delta x(t)\Big]_{t_j^l}^{t_{j+1}^{l-}} + dt_{j+1}\,\|\Delta u_j^{l+1}(t_{j+1}^{l-})\|^2 + dt_j\,\|\Delta u_j^{l+1}(t_j^l)\|^2\Big) \\ & \overset{(4.164),(4.188)}{=} \lim_{p\to\infty}\frac{1}{2\beta^p}\sum_{j=0}^{N}\Big(\int_{t_j^l}^{t_{j+1}^l} -2\beta^p\|\Delta u_j^l(t)\|^2 + o(\beta^p)\,dt - dt_{j+1}\|\Delta u_j^l(t_{j+1}^{l-})\|^2 \\ & \qquad + dt_j\|\Delta u_j^l(t_j^l)\|^2 + dt_{j+1}\|\Delta u_j^l(t_{j+1}^{l-})\|^2 - dt_j\|\Delta u_j^l(t_j^l)\|^2 + o(\beta^p)\Big) \\ & = -\sum_{j=0}^{N}\int_{t_j^l}^{t_{j+1}^l}\|\Delta u_j^l(t)\|^2\,dt < -\tau_0\sum_{j=0}^{N}\int_{t_j^l}^{t_{j+1}^l}\|\Delta u_j^l(t)\|^2\,dt\,, \end{aligned} \tag{4.190}$$

which again contradicts the assumption. Since there always exists a p, such that the individual descent conditions (4.178) for all $j \in \{0, ..., N\}$ and (4.184) are fulfilled, there also exists a p, such that (4.177), which is the summation over all these conditions, is satisfied. The simple summation is possible since the individual conditions do not degrade the convergence of the other conditions in first order. ■

Below, the sufficient descent property of the algorithm is derived.

Proposition 4.7: Let the standing assumptions hold. For every $l \in \mathbb{N}$ and every $\kappa > 0$, there exists $\rho > 0$, such that, if $\Omega(\chi^l, \lambda^l, \upsilon^l) > \kappa$, then

$$\Omega(\chi^{l+1}, \lambda^{l+1}, \upsilon^{l+1}) - \Omega(\chi^l, \lambda^l, \upsilon^l) \leq -\rho\,. \tag{4.191}$$

□

Proof: In Proposition 4.6, it has been found that $-2\Omega(\chi^l, \lambda^l, \upsilon^l)$ is the derivative of $\Omega(\chi^l, \lambda^l, \upsilon^l) \geq 0$. For $\Omega(\chi^l, \lambda^l, \upsilon^l) > \kappa$, it follows that $\Omega(\chi^l, \lambda^l, \upsilon^l)$ is not equal to the optimum $\Omega(\chi^*, \lambda^*, \upsilon^*) = 0$, which exists due to Assumption 4.2. From Proposition 4.6, the sufficient descent property can be concluded:

$$\Omega(\chi^{l+1}, \lambda^{l+1}, \upsilon^{l+1}) - \Omega(\chi^l, \lambda^l, \upsilon^l) \leq -2\tau_0\, \beta^p\, \Omega(\chi^l, \lambda^l, \upsilon^l) \leq -2\tau_0 \beta^p \kappa = -\rho \tag{4.192}$$

with $\rho = 2\tau_0\beta^p\kappa$. ■

Theorem 4.8: Under the standing assumptions specified in Sec. 4.3.1 and with $\varrho^0 = \varrho^*$, every sequence $\{\Omega(\chi^l, \lambda^l, \upsilon^l)\}_{l=1}^{\infty}$ generated by Algorithm 4.1 converges to a solution ϱ^*, χ^*, λ^*, and υ^*, which satisfies all optimality conditions of Theorem 4.2. □

Proof: As a direct consequence of Propositions 4.3 and 4.7, the algorithm converges to $\Omega(\chi^*, \lambda^*, \upsilon^*) = 0$, which can only occur iff $\psi_{j-1}(x^*(t_j)) = 0$ for $j \in \{1, ..., N\}$, $\psi_N(x^*(t_e)) = 0$, and $u_j^*(t) = \arg\min_{u \in U_j} H_j(x^*(t), \lambda^*(t), u)$ for $j \in \{0, ..., N\}$ and $t \in [t_0, t_e]$. Since Proposition 4.3 is also valid, the proof is concluded. ■

4.3.4 Numerical Example

The algorithm is applied to a drive-around maneuver, where an autonomous car shall drive around a car parked on the right lane of a road with two lanes. The autonomous car is modeled as a unicycle

$$\dot{x}_1 = v \cos\theta \tag{4.193}$$

$$\dot{x}_2 = v \sin\theta \tag{4.194}$$

$$\dot{v} = u_1 \tag{4.195}$$

$$\dot{\theta} = u_2\,, \tag{4.196}$$

where (x_1, x_2) is the position of the car, v the velocity, θ the orientation, u_1 the accelerating or braking force, and the steering input u_2 is the velocity of a change in the orientation. The state and control vectors are defined as $x := (x_1, x_2, v, \theta)^T$ and $u := (u_1, u_2)^T$. The control constraints are:

$$|u_1| \leq 2 \tag{4.197}$$

$$|u_2| \leq 2\,. \tag{4.198}$$

The road is divided into three regions

$$q = 1: \qquad -2.15 \leq x_2 \leq -0.85 \tag{4.199}$$

$$q = 2: \qquad -0.85 \leq x_2 \leq 0.85 \tag{4.200}$$

$$q = 3: \qquad 0.85 \leq x_2 \leq 2.15\,. \tag{4.201}$$

If the center point of the car is in region 1 or 3, then the car is entirely on the right or left lane, respectively. Region 2 is the transition region between right and left lane. The idea behind the division of the road into three regions is that an optimal control algorithm is enabled to take an optimal, discrete decision on the sequence of lanes for the car to drive. Except for short transition times, a car should always drive on either the right or left lane in usual traffic situations. For each discrete state, a different cost function is chosen

$$\phi_1 = 0.05\,\|u\|^2 \tag{4.202}$$

$$\phi_2 = 0.0002 + 0.05\,\|u\|^2 \tag{4.203}$$

$$\phi_3 = 0.0005 + 0.05\,\|u\|^2\,, \tag{4.204}$$

such that the car will in general drive on the right lane due to the lowest costs. On the right lane at position $x = (80, -1.5, 0, 0)^T$, a car is parked and for safety reasons the autonomous car is not allowed to drive in the right and middle regions $q = 1$ and $q = 2$ for $40 < x_1 < 100$. The autonomous car has initially the state value $x(t_0) = (0, -1.5, 20, 0)^T$. The optimal control task is to find an optimal trajectory for the autonomous car with the a-priori fixed discrete state sequence $\varrho = (1, 2, 3, 2, 1)$. Additionally, the interior and terminal conditions

$$\psi_1(x(t_2)) = x_1(t_2) - 40 = 0 \tag{4.205}$$

$$\psi_2(x(t_3)) = x_1(t_3) - 100 = 0 \tag{4.206}$$

$$\psi(x(t_e)) = \begin{pmatrix} x_1(t_e) - 160 \\ x_2(t_e) + 1.5 \\ \theta(t_e) \end{pmatrix} = 0 \tag{4.207}$$

have to be satisfied, such that the covered distance in direction of x_1 in discrete state 3 is minimal, see also Fig. 4.5. The final time t_e is 8 and the switching times are free.

The algorithm converges to an optimal solution with a cost value $J = 0.0027$ and a final state $x(t_e) = (160, -1.5, 20.0317, 0)^T$, see Fig. 4.6. For initialization, the unknown multipliers $\hat{\pi}_2$, $\hat{\pi}_3$, and ν and the trajectory u_1 are set to zero. Only u_2 is initialized with constant and non-zero values, such that the desired switching manifolds are hit. The integration over the squared errors in the minimization condition of the Hamiltonian fulfills

$$\sum_{j=0}^{4} \int_{t_j}^{t_{j+1}} \|\nabla_u H_{q_j}(x(t), \lambda(t), u_{q_j}(t))\|^2 \, dt < 1 \cdot 10^{-11} \tag{4.208}$$

$$\sum_{j=0}^{4} \int_{t_j}^{t_{j+1}} \|\Delta u_{q_j}(t)\|^2 \, dt < 5 \cdot 10^{-10}\,. \tag{4.209}$$

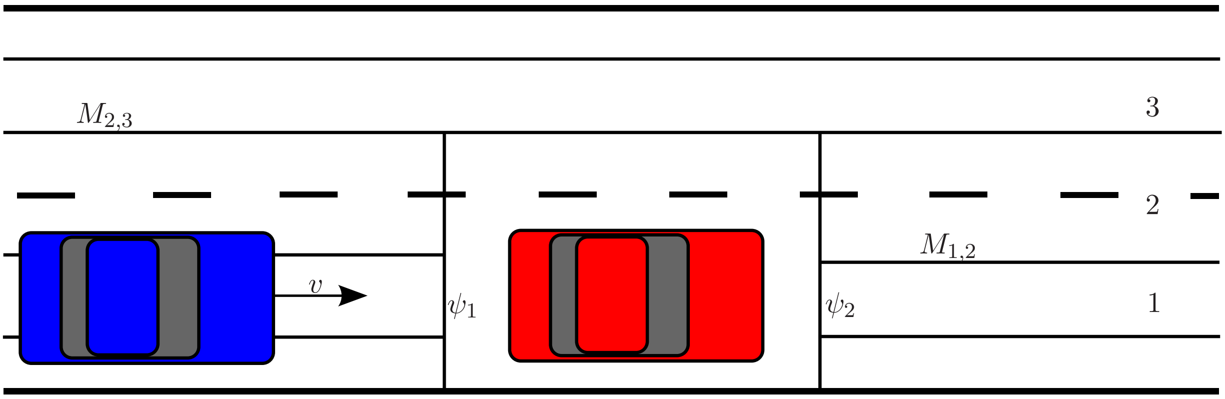


Fig. 4.5: Road with three regions, the autonomous car (on the left), the parked car, and the safety zone around the parked car.

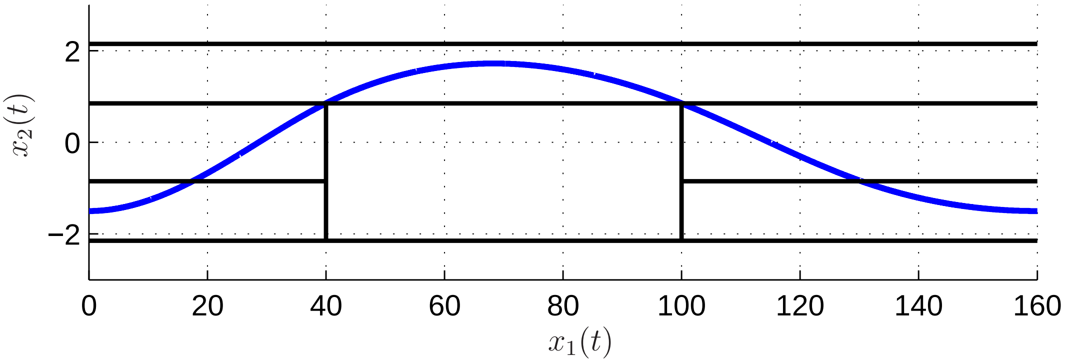


Fig. 4.6: Optimal trajectory of the autonomous car driving around a parked car under consideration of two interior point constraints, a boundary condition, and different costs in the different regions of the road.

The summarized error in the squared interior and terminal conditions is

$$\|\psi_1(x(t_2))\|^2 + \|\psi_2(x(t_3))\|^2 + \|\psi(x(t_e))\|^2 < 2 \cdot 10^{-9} \,. \tag{4.210}$$

To achieve this accuracy, the control trajectory is created with 3201 samples that are interpolated with splines for determining the control values in the integrations. When increasing the number of samples in the control trajectory, only the number of evaluations for the control update in (4.69) increases linearly. Since the integrations in the algorithm are independent of the update evaluations, the computational complexity increases linearly with an increased demand in accuracy. The algorithm needs 10 iterations and a computational time of 1042 sec. The computations are executed on a 3.0 GHz AMD Phenom II processor using Matlab 2009a.

4.3.5 Discussion

In this section, a min-H algorithm for optimal control of hybrid systems is proposed that integrates the state and adjoint differential equations in their stable directions. The idea behind this approach is to enlarge the domain of convergence by reducing the numerical difficulties that appear in indirect multiple shooting and also indirect collocation. In all tested numerical examples, the numerical stability of the novel algorithm was significantly higher than the numerical stability of previous algorithms. The min-H algorithm converged without difficulties for the presented example in Sec. 4.3.4. In contrast, it was very difficult and required considerable tuning to achieve convergence with the indirect multiple shooting algorithm, which is presented in Sec. 4.2 and which solves the entire HOCP at once. The difficulties arose though indirect multiple shooting was applied to less complex vehicle dynamics with constant velocity in Sec. 4.2.3 and was initialized by a direct method.

Nevertheless, the novel algorithm also suffers from a numerical limitation. The limitation is that the Riccati-matrices sometimes tend towards infinity during the backward integration. The reason for this growth is the quadratic differential equation (4.79) of the Riccati-matrix S_j. The problem was observed when we optimized the trajectories for a robotic manipulator consisting of kinematic chains with 14 degrees of freedom, which leads to system dynamics with 29 dimensions in the optimization. The problem occurs when the initialization is far from the optimal solution and the considered time horizon is relatively long for the dynamics. The numerical stability of the backward integration is increased by a modified version of the Davison-Maki algorithm presented in [85]. The approach separates the Riccati-matrix S_j into two matrices X_j and Y_j, such that only linear differential equations have to be integrated. If this problem is observed despite the use of the modified Davison-Maki algorithm, it is suggested to extend the time horizon of the optimal control problem step-by-step from the initial time until the desired time horizon is reached. This simplifies the optimal control task for the algorithm. Additionally, it might be necessary to scale down the cost functions and the interior and terminal constraint functions. In face of the problems, the novel algorithm still outperforms existing indirect algorithms in view of numerical stability.

A further advantage of the novel min-H algorithm in comparison to indirect multiple shooting and indirect collocation is the simplified initialization. A first reason for the simplified initialization is the enlarged domain of convergence. A second reason is that, instead of physically non-intuitive adjoint variables, a physically intuitive control history $\upsilon^0_{q_j}$ with $q \in \mathcal{Q}$ and $j \in \{0, ..., N\}$ and some multipliers $\hat{\pi}^0_j$ and ν^0 with $j \in \{1, ..., N\}$ have to be guessed for initialization. In our experience, it is sufficient for successful convergence to initialize the multipliers ν^0 and $\hat{\pi}^0_j$ with zero and the control with some constant functions $\upsilon^0_q \in \mathcal{U}_q$, such that the desired switching manifolds are reached.

The proposed min-H algorithm is especially efficient for smooth control trajectories, but also works for non-smooth controls that belong to the class $\mathcal{U}$ of bounded and measurable functions. Here, it is noticeable that the applied update $\delta u(t) = -\tilde{K}_j(t)\delta x(t) - \tilde{L}_j(t)\delta\lambda(t) - \epsilon\left(1 - \frac{1}{2}\zeta^T_j(t)D^{-1}_j(t)\nabla_u f_j(t)\Delta u^l_j(t)\right)\Delta u^l_j(t)$ contains the derivative $\Delta\dot{u}^l_j(t) = \dot{u}^l_j(t) - \dot{u}^{*l}_j(t)$. However, since by Assumption 2.39.4 υ^l_j is a measurable function, which may jump and may have corners, the derivative might not be defined for some $t \in [t_j, t_{j+1})$ with measure zero. This means $\dot{\upsilon}^l_j$ is also a measurable function. At times t, where the control $u^l_j(t)$ has singular points and the derivative $\dot{u}^l_j(t)$ is not defined, the left-continuous replacement of the measurable function υ^l_j and its derivative is used. Note that the solution of the integral in (4.190) is not affected by υ^l_j and $\dot{\upsilon}^l_j$ being measurable functions since all singular points have measure zero.

The min-H algorithm has been described for HOCPs with autonomous switching, control constraints, and interior point constraints. An extension to state constraints is straightforward. Consider a hybrid system with bounded state spaces $\mathcal{X}_q$, where the boundaries are partly formed by switching manifolds $M_{q,i}$ for $q, i \in \mathcal{Q}$ and by a set of state inequality constraints $h^{\mathrm{s}}_{q,k}(x(t)) \leq 0$ with $k \in \{0, 1, ..., N_{h_q}\}$. The number of state constraints is N_{h_q} in discrete state q. The constraints $h^{\mathrm{s}}_{q,k}(x(t))$ are oriented to the outside of $\mathcal{X}_q$. Let the control $u_q(t)$ appear in the r-th time derivative $h^{(r),\mathrm{s}}_{q,k}(x(t), u_q(t))$ for the first time with $r \in \mathbb{N}_0$. Assume the functions $h^{\mathrm{s}}_{q,k}(x(t))$ and $f_q(x(t), u_q(t))$ to be sufficiently smooth, such that $h^{(r),\mathrm{s}}_{q,k}(x(t), u_q(t))$ exists. Assume furthermore that there always exists a control $u_q(t) \in U_q$, such that $h^{(r),\mathrm{s}}_{q,k}(x(t), u_q(t)) = 0$. Then the algorithm can also handle state constraints. On a state constraint, $h^{(r),\mathrm{s}}_{q,k}(x(t), u_q(t)) = 0$ represents an additional constraint for the optimal control. State constraints are similar to the switching case, but with optimality conditions

$$\lambda(t_c^-) = \lambda(t_c) + \sum_{\varsigma=0}^{r-1} (\nabla_x h^{(\varsigma),\mathrm{s}}_{q,k}(x(t_c)))^T \mu_{c,\varsigma} \tag{4.211}$$

$$H_q(t_c^-) = H_q(t_c) \tag{4.212}$$

at the times t_c of hitting a state constraint and with the multipliers $\mu_{c,\varsigma}$ for $\varsigma \in \{0, ..., r-1\}$.

The proposed algorithm finds optimal open-loop controls. This means that when the optimal solution is applied in practice, where usually model uncertainties, sensor noise, etc. are present, it is recommended to add a form of feedback control, preferably an optimal feedback controller. An optimal feedback controller can be extracted straightforwardly from the Riccati-matrices and vectors that have been computed during the optimization with the min-H algorithm.

4.4 A Combined First- and Second-Order Gradient Algorithm

In the following, an alternative algorithm to the novel min-H algorithm is proposed. The alternative algorithm mainly differs from the min-H algorithm in the choice of the update direction of the control. In the alternative algorithm, the update direction is determined as in first- and second-order gradient methods.

4.4.1 Algorithm

The alternative algorithm resembles the min-H algorithm in most parts except for the update direction. Therefore, only the difference in the update direction is presented below. In the min-H algorithm, the update (4.118)

$$\delta u(t) = -\tilde{K}_j(t)\delta x(t) - \tilde{L}_j(t)\delta\lambda(t) - \epsilon W_j^{-1}(t)\Delta u_j^l(t)$$

of the control consists of the term $-\epsilon\, W_j^{-1}(t)\Delta u_j^l(t)$ that approaches the optimal control and the term $-\tilde{K}_j(t)\delta x(t) - \tilde{L}_j(t)\delta\lambda(t)$ that compensates for changes in the state and adjoint variables due to updates of the control. In the alternative algorithm, properties of first- and second-order gradient algorithms are combined, which leads to two different updates of the control:

(i) If the Hamiltonian $H_j(t)$ is convex, i.e. $H_{j,uu}(t) > 0$, and the current control is not on a constraint, i.e. $u_j^l(t) \in \mathrm{Int}(U_j)$, then an update $\delta u(t) = -\epsilon H_{j,uu}^{-1}(t)\nabla_u H_j^T(t) - H_{j,uu}^{-1}(t)H_{j,ux}(t)\delta x(t) - H_{j,uu}^{-1}(t)f_{j,u}^T(t)\delta\lambda(t)$ is performed, which corresponds to a Newton update step. This update is used in second-order gradient methods.

(ii) If $H_{j,uu}(t) \leq 0$ or $u_j^l(t) \in \partial U_j$, then an update $\delta u(t) = -\epsilon P_j \nabla_u H_j^T(t) - H_{j,uu}^{-1}(t)H_{j,ux}(t)\delta x(t) - H_{j,uu}^{-1}(t)f_{j,u}^T(t)\delta\lambda(t)$ is executed, also known as steepest descent, to ensure convergence. The matrix P_j is positive semi-definite to create an artificial convex region possibly with constraints, such that the update $\delta u(t)$ of the control $u_j(t)$ approaches a control $u_j^*(t)$ with $\nabla_u H_j(u_j^*(t)) = 0$ that denotes a minimum of the Hamiltonian and not a maximum. The update resembles the updates used in first-order gradient methods. However, in updates of previous first-order gradient methods, the compensation terms for changes in the state and adjoint variables are sometimes not considered, such that convergence cannot be guaranteed.

Note that the updates are illustrated here for purely continuous optimal control problems. In the case of HOCPs with autonomous switching, additional compensation terms for changes in the switching times due to the updated control are required to prove local convergence, compare Sec. 4.3.3.

4.4.2 Comparison

Hereafter, advantages and disadvantages of the alternative algorithm compared to the min-H algorithm are discussed. The alternative method has the advantage that no minimization is necessary at every time t to find the optimal control $u_j^{*l}(t)$, which minimizes the Hamiltonian $H_j(x^l(t), \lambda^l(t), u_j^l(t))$. This may save computational time, if the minimization cannot be executed analytically. A disadvantage is that the convergence is not as fast as in the min-H approach if the steepest descent method has to be used due to nonconvex optimization regions. Further, the global convergence to a local minimum cannot be guaranteed any longer. The reason is that the method always follows gradients and has no locally optimal control $u_j^*(t)$ as reference for the convergence. In contrast, the min-H algorithm uses an optimal control $u_j^*(t)$ as reference for the convergence, which leads to the discussed global convergence property.

4.5 Summary

A difficult initialization process and a small domain of convergence are two limitations of indirect methods, when indirect methods are compared to direct methods and DP based on average results for the same HOCPs. This chapter provides two initialization concepts and a min-H algorithm for a simplified initialization and an enlarged domain of convergence.

In the initialization concepts, a direct method is applied first to the HOCP and the solution is used for initializing an indirect method since direct methods are more straightforward to initialize than indirect methods. Further, direct methods have in general a larger domain of convergence than indirect methods, such that the overall size of the domain of convergence is increased by the first approach. Two concepts are presented for this approach: First, a direct multiple shooting method solves an entire HOCP at once and the solution serves an indirect multiple shooting algorithm to refine an optimal solution of the complete HOCP also at once. In a second concept, the HOCP is decomposed into non-hybrid subproblems, which are solved separately by direct and indirect multiple shooting. On an upper layer of the algorithm, the switching points, which connect the subproblems, are optimized. By the decomposition the robustness of the initialization scheme is further improved. However, the convergence rate is decreased as the optimization of switching points is based on steepest descent. In future, the initialization concept based on the decomposition can be extended, such that the solution is used to initialize an indirect multiple shooting method for further refinement.

The presented min-H algorithm is proposed for the optimal control of hybrid systems with autonomous switching and interior point constraints. The algorithm is initialized straightforwardly, converges globally to a locally optimal solution, and delivers results with high accuracy. If a higher accuracy is needed, this increases the computational complexity linearly. The algorithm is numerically stable and converges robustly in the discussed numerical example. In contrast, it has been difficult to achieve convergence with an algorithm based on multiple-phase multiple shooting, which has been applied to a simplified version of the numerical example with fixed velocity. The novel min-H algorithm operates more efficiently if it is initialized with relatively smooth control trajectories. Applied to a highly nonlinear manipulation task with chained kinematics and 29 dimensional state space, some values in the Riccati matrices tend towards infinity during the backward integration. In this case, it is suggested to extend the time horizon of the optimal control problem step-by-step from the initial time until the original time horizon is reached. Additionally, it is recommended to scale down the cost functions and interior and terminal constraints, in order to simplify the optimal control task for the algorithm. Future work includes to extend the algorithm to a larger class of hybrid systems with free final time, time-varying dynamics, jumps in the state variable at switches, switching costs, and controlled switching.

In this chapter, limitations of indirect methods are overcome and for simplicity the analysis was restricted to HOCPs with fixed discrete state sequence. In the next chapter, the computational complexity is reduced if the optimal sequence of discrete states is unknown and subject to optimal control.

5 Optimal Control for Hybrid Systems with Variable Discrete State Sequence

This chapter introduces two optimal control algorithms that reduce the computational complexity for hybrid optimal control problems (HOCPs) with autonomous switching when the optimal discrete state sequence is unknown a priori. Former optimal control algorithms have to analyze each possible discrete state sequence separately, which is of combinatorial complexity in the number of autonomous switchings. The first optimal control algorithm is based on a tree search, where the tree encodes all possible discrete state sequences. Branch and bound principles are used to prune the tree and reduce the computational complexity for finding an optimal discrete state sequence. Determining lower bounds for the branch and bound method is inexpensive by relaxing the HOCP with autonomous switching to an HOCP with controlled switching. The efficiency of the algorithm is illustrated with a numerical example.

The second optimal control algorithm is for HOCPs with partitioned continuous state space. The algorithm varies the discrete state sequence based on gradient information during the search for an optimal continuous control trajectory. The version of the hybrid minimum principle (HMP) from Sec. 3.3 is used in the algorithm. The novel HMP allows optimal state trajectories to pass through intersections of switching manifolds, which enables the algorithm to vary the sequence. Consequently, the complexity to find a locally optimal solution is reduced by a factor of combinatorial complexity in contrast to former algorithms. The convergence of the algorithm is proven and several numerical examples demonstrate the efficiency of the algorithm.

5.1 Introduction and State of the Art

Indirect methods show a combinatorial complexity for HOCPs with autonomous switching and unknown optimal discrete state sequence. This chapter introduces two indirect optimal control algorithms that reduce the computational complexity for this class of HOCPs. This extends the set of HOCPs, in which indirect methods have a lower complexity than direct methods and dynamic programming (DP).

In the following, HOCPs are considered where the search for an optimal solution includes to find the optimal sequence of discrete states besides the optimal continuous control and the optimal switching times. In general, direct and indirect methods find an optimal solution by solving an HOCP for each possible discrete state sequence, e.g. [105, 136, 187]. Sometimes the search is bounded to each possible discrete state sequence in a certain neighborhood around a nominal sequence [156]. Afterwards, the methods compare the costs of the optimal solutions in the different discrete state sequences, as far as optimal solutions exist, and select the discrete state sequence with the minimal costs. The disadvantage of the procedure is that the computational complexity is combinatorial for HOCPs with autonomous switching and exponential for HOCPs with controlled switching, compare Definition 2.25. In contrast, solution methods based on value function approaches like in [143] only show a linear increase in the complexity with the number of discrete states. The disadvantage of value function approaches is the exponential increase in complexity with the dimension of the continuous state, which limits the applicability to low dimensional systems with at most 4 continuous states.

A challenge consists in developing efficient numerical optimization algorithms that avoid the combinatorial or exponential complexity for direct and indirect methods. A major difficulty is to find approaches that are able to calculate the optimal discrete state sequence simultaneously with the continuous optimal solution, such that the high complexity is avoided. For direct methods and HOCPs with controlled switching, several approaches are proposed that reduce the complexity by using branch and bound principles, e.g. [164], or by optimizing over an HOCP with relaxed discrete controls [144]. Considering direct methods and HOCPs with autonomous switching, one of the few approaches for reducing the complexity is given in [15], where numerical techniques for solving continuous optimization problems are combined with symbolic techniques for solving constraint satisfaction problems of logic variables.

Efficient algorithms based on indirect methods that avoid the combinatorial or exponential complexity exist mainly for hybrid systems with controlled switching. Controlled switching simplifies finding an optimal control. The reason is that optimality conditions exist in this case for the locally optimal discrete state sequence and switches are not constrained to switching manifolds. Known algorithms that only consider controlled switching include the following four representative approaches, which find locally optimal, open-loop controls: (a) An algorithm finds the optimal discrete state schedule in a single run by the construction and pre-computation of optimality zones [40]. Optimality zones are created by finding the optimal discrete state for all possible pairs of entry and exit states and times based on combining a value function approach with the HMP. The overall computational complexity is linear in the number of switchings. (b) Algorithms are applied to determine the optimal discrete state sequence based on relaxations of the discrete control in [18] and [136]. This shows a polynomial complexity in number of discrete states. (c) Another approach consists in imposing a sequence of discrete states and associated time intervals [136]. The discrete state sequence can be changed and thus optimized by reducing the length of some time intervals to zero. The complexity depends linearly on the number of imposed time intervals, which is usually related to the number of discrete states. (d) Needle variations of the discrete state are inserted in a given discrete state sequence and gradient methods are applied to find the optimal switching sequence and times [50]. The computational complexity is linear in the number of discrete states.

For indirect methods and HOCPs with autonomous switching, it is still an open problem to reduce the combinatorial complexity. Hereafter, two approaches with a decreased computational complexity are introduced. The contributions are the following:

(i) A tree search algorithm is proposed that reduces the complexity by using branch and bound principles for a general class of hybrid systems.

(ii) For the tree search algorithm, a relaxation of HOCPs with autonomous switching to controlled switching is formulated.

(iii) A multiple-phase multiple shooting algorithm is presented that is capable of solving HOCPs with both autonomous and controlled switching.

(iv) For HOCPs with partitioned state space, an algorithm is introduced that finds the optimal discrete state sequence based on gradient information provided by the novel HMP from Sec. 3.3 and the convergence is proven. With the approach, the computational complexity is reduced by a factor of combinatorial complexity compared to former algorithms.[1]

The optimal control algorithm that is presented first organizes all possible discrete state sequences in a tree. In a first step, the HOCP with autonomous switching is relaxed to a problem with controlled switching. Next, the relaxation is tightened iteratively, which leads to upper and lower bounds on the optimal cost for the different sequences of discrete states. A branch and bound scheme is used to prune the tree and reduce the search space for the optimal discrete state sequence. The underlying HOCPs with autonomous and controlled switching are solved by indirect multiple shooting based on the HMP derived in [157]. The algorithm is similar to multiple-phase multiple shooting as in [136] and extended to consider controlled switching additionally. Here, the discrete state sequence for the parts with controlled switching is determined according to the Hamiltonian minimization condition with respect to the discrete control from [157]. Thus, finding a locally optimal discrete state sequence for controlled switching has a linear dependence of the complexity on the number of discrete states. The overall complexity of the algorithm is usually considerably reduced compared to the combinatorial complexity of former algorithms. The algorithm has been published in [215].

The second optimal control algorithm is developed for hybrid systems with partitioned state space and autonomous switching and is based on the novel HMP in Sec. 3.3. The algorithm varies the discrete state sequence simultaneously with the search for an optimal continuous control. The algorithm is initialized with a discrete state sequence and corresponding switching states and times. Then, an underlying optimization routine determines the optimal solution from switching point to switching point. By evaluating optimality conditions from the novel HMP at the switching points, the switching points can be shifted. The shift of a switching point is also possible across intersections of switching manifolds. Consequently, the discrete state sequence can be varied with a provably convergent gradient descent method until an optimal solution is obtained. The algorithm reduces the computational complexity to find a locally optimal state trajectory and discrete state sequence by a factor of combinatorial complexity in contrast to former algorithms. The novel algorithm has been presented in [207, 214].

[1] Former algorithms based on the HMP cannot handle the case of intersecting switching manifolds, which bounds their search for a locally optimal solution to a differentiable segment of a switching manifold. Therefore, a search of all candidates for an optimal solution has combinatorial complexity as all possible sequences have to be analyzed separately [136, 157].

5.2 A Tree Search Algorithm Based on Branch and Bound

Hereafter, the tree search algorithm is introduced in order to reduce the computational complexity in comparison with existing indirect optimal control algorithms. Sec. 5.2.1 describes the considered HOCP. Sec. 5.2.2 introduces the proposed tree search algorithm, Sec. 5.2.3 provides a numerical example, and Sec. 5.2.4 gives a discussion on the algorithm.

5.2.1 Formulation and Relaxation of the Hybrid Optimal Control Problem

Again hybrid systems following Definition 2.34 and Assumption 2.35 are considered. They include continuous and discrete states, continuous controls, continuous and discrete dynamics, switching manifolds for autonomous switching, and state and control constraints. However, for a concise presentation, state constraints are not further taken into account in the remainder of this section, though the tree search algorithm can handle state constraints. Assumption 2.35.6 is relaxed in that intersections of switching manifolds are permitted as in Sec. 3.3. The HOCP under consideration contains terminal and running costs and possibly terminal constraints as specified in Definition 2.36. Additionally, the HOCP considers the target $T(t_e)$, that may contain a specified final continuous state x_e given by the terminal constraints and discrete state q_e.

For the relaxation of the HOCP from autonomous to controlled switching, a discrete control as a function of time is introduced:

Definition 5.1: Σ is the set of control functions $\varsigma = (\varsigma_1, ..., \varsigma_{N_q})^T : [t_0, t_e] \rightarrow \{0,1\}^{N_q}$ with the constraint $\sum_{i=1}^{N_q} \varsigma_i(t) = 1$. The binary variable ς is chosen in discrete state q, such that $\varsigma_q = 1$ and $\varsigma_i = 0$ for $i \neq q$ with $i, q \in \mathcal{Q}$. The variable $\varsigma(t)$ is piecewise constant for $t \in [t_0, t_e]$. □

Here, the discrete control ς is a function of time in contrast to the discrete control ω that is time-independent. It is advantageous to reformulate the considered class of hybrid systems for the application to the tree search algorithm. The reformulation resembles the class of hybrid systems in [18, 137] and is additionally modified, such that autonomous switching is included. In the reformulation of the system dynamics

$$\dot{x}(t) = \sum_{i=1}^{N_q} \varsigma_i(t)\ f_i(x(t), u_i(t)) \quad \text{a.e. } t \in [t_j, t_{j+1}),\ j \in \{0, ..., N\}\,, \tag{5.1}$$

the binary variable ς is constant in the intervals $[t_j, t_{j+1})$ for $j \in \{0, ..., N\}$. Since only the element ς_q of the discrete control ς is nonzero in discrete state q, only the dynamics f_q is active. This reformulation causes a redefinition of the execution of a hybrid system $\sigma = (\tau, \varrho, \chi, \upsilon, \varsigma)$, such that the discrete control function ς is included now. The cost function corresponding to the reformulated HOCP is:

$$J = g(x(t_e)) + \sum_{j=0}^{N} \int_{t_j}^{t_{j+1}} \sum_{i=1}^{N_q} \varsigma_i(t)\ \phi_i(x(t), u_i(t))\, dt\,. \tag{5.2}$$

Please note that the discrete control ς changes its values if and only if the continuous state trajectory meets a switching manifold and the system switches autonomously from one discrete state to the next.

The proposed branch and bound approach for solving this HOCP requires a relaxed version of the hybrid system defined above:

Definition 5.2: A relaxation $R(0)$ of the hybrid system with autonomous switching is a system, where (5.1) and all further conditions given in Definition 2.34 are fulfilled except for the autonomous switching structure, which is relaxed to controlled switching. This means that switching is no longer bound to $x \in M_{q,l}$ for $q, l \in \mathcal{Q}$, but it may occur in the whole state space $\mathcal{X}_q$. Furthermore, switching is not limited to the entries of the transition map Γ any more, but it is permitted to switch to any discrete state in $\mathcal{Q}$ from each state q. Controlled switching may take place at every time instant $t \in [t_0, t_e]$ as long as only finitely many switching instances occur. The term $R(a)$ with $a \in \{0, ..., N+1\}$ denotes a relaxation that is only valid from the a-th autonomous switching instance of the original system on. $a = 0$ means that the system is relaxed from the initial time t_0 to the end t_e, and $R(N+1)$ is the original hybrid system with autonomous switching. □

The relaxation is introduced to obtain the optimal discrete state at each time instant t for controlled switches with the HMP in a similar way as the optimal continuous control $u_q(t)$ is found at each time instant t. Using the Hamiltonian minimization condition with respect to the discrete control[2], a locally optimal discrete state sequence can be found directly. It is not necessary to compare the costs of all the different possible sequences to find an optimal schedule as it is in the case of autonomous switching. The relaxation here differs from the relaxations often used in the context of the HMP or mixed-integer programming, where usually the binary variables are relaxed to values in $[0, 1]$. Here, not the values of the binary variables $\varsigma_q(t)$ for $q \in Q$ are relaxed but the time instants at which the variables are permitted to change their values. In the original system, $\varsigma_q(t)$ had to be constant until a switching manifold was hit – now $\varsigma_q(t)$ may always vary. An advantage of the proposed relaxation compared to versions that relax the binary variable $\varsigma_q(t)$ to a continuous one is that the dynamics imposed by the vector fields f_q are never mixed. Below, the dependence of the variables on t is omitted for brevity.

Definition 5.3: An execution $\sigma_{R(a)} = (\tau, \varrho, \chi, \upsilon, \varsigma)$ of the partly relaxed hybrid system $R(a)$ for $a \in \{0, ..., N+1\}$ is defined as follows: τ is a sequence of initial, switching, and final times t_j for $j \in \{0, ..., N+1\}$. $\varrho = (q_0, ..., q_N)$ is a sequence of discrete states. $\chi = (\chi_{q_0}, ..., \chi_{q_N})$ is a sequence of absolutely continuous state trajectories $\chi_{q_j} : [t_j, t_{j+1}) \to \mathcal{X}_{q_j}$, which evolve according to

$$\dot{x}_{q_j}(t) = \sum_{i=1}^{N_q} \varsigma_i(t)\ f_i(x_{q_j}(t), u_i(t)) \tag{5.3}$$

for a.e. $t \in [t_j, t_{j+1})$ and for $j \in \{0, ..., N\}$ and satisfy $x_{q_{j-1}}(t_j^-) = x_{q_j}(t_j)$ at the switching times t_j. When the system is in discrete state q_{j-1} for $j \in \{1, ..., a\}$, it next switches

[2] The condition can also be derived for the considered time-varying discrete control ς.

autonomously on the switching manifold $m_{q_{j-1},q_j}(x_{q_{j-1}}(t_j^-)) = 0$ to discrete state q_j. When the system is in discrete state q_{j-1} for $j \in \{a+1, ..., N\}$, it next switches controlled to the discrete state q_j. $\upsilon = (\upsilon_{q_0}, ..., \upsilon_{q_N})$ is a sequence of measurable control functions $\upsilon_{q_j} : [t_j, t_{j+1}) \to U_{q_j}$. ς is a piecewise constant, binary function $\varsigma : [t_j, t_{j+1}) \to \{0,1\}^{N_q}$ for $j \in \{0, ..., N\}$, where $\varsigma_{q_j}(t) = 1$ and $\varsigma_i(t) = 0$ hold for $t \in [t_j, t_{j+1})$, $i \neq q_j$, and $i, q_j \in \mathcal{Q}$. The binary function $\varsigma(t)$ for $t \in [t_j, t_{j+1})$ and $j \in \{1, ..., a\}$ changes its value according to $\Gamma\big(q_{j-1}, x_{q_{j-1}}(t_j^-), \varsigma_{q_{j-1}}(t_j^-)\big)$ if and only if $m_{q_{j-1},q_j}(x_{q_{j-1}}(t_j^-)) = 0$, i.e. only autonomous switching is considered in this part. For $t \in [t_j, t_{j+1})$ and $j \in \{a+1, ..., N\}$, the discrete control $\varsigma(t)$ is allowed to change whenever desired, i.e. only controlled switching is considered in this part. □

Definition 5.4: An optimal solution $\sigma^*_{R(a)}$ for $a \in \{0, ..., N+1\}$ of the relaxed system $R(a)$ minimizes

$$J_{R(a)} = g(x(t_e)) + \sum_{j=0}^{\bar{N}} \int_{t_j}^{t_{j+1}} \sum_{i=1}^{N_q} \varsigma_i(t)\ \phi_i(x(t), u_i(t))\, dt \tag{5.4}$$

with a maximum number of permitted jumps $\bar{N}$, which is chosen to be large $\infty > \bar{N} \gg N$ with $\bar{N} \in \mathbb{N}$. Up to the a-th transition, the autonomously switched system must be fulfilled and from that switch on the relaxed system properties with controlled switching are considered. □

The original system is a subclass of the relaxed system, as each feasible system execution σ of the autonomously switched system can also be performed by the relaxed system. The relaxed system includes hybrid system executions $\sigma_{R(.)}$, however, which are not possible for the original system. An obvious consequence is that the globally optimal solution of the relaxed system $R(0)$ incurs the costs

$$\min_{\upsilon \in \mathcal{U}, \varsigma \in \Sigma} J_{R(0)} \leq \min_{\upsilon \in \mathcal{U}} J\ , \tag{5.5}$$

which are less or equal to the minimal achievable costs for the original system.

To find a minimizing solution σ^* for the autonomously switched system, the optimality conditions are used, which are stated in Theorems 3.1 and 3.4 and are adapted to the considered class of HOCPs. They lead to a multi-point boundary value problem (MPBVP), which has to be solved for each possible sequence of discrete states specified in Γ and starting at q_0. The number of possible discrete state sequences increases combinatorially with the number of autonomous switchings. A major advantage in the solution of the system with controlled switching is that the optimal discrete state sequence is found in parallel with the search for an optimal continuous control. By always switching to the discrete state q with the lowest Hamiltonian H_q in the sense of (3.8), the optimal discrete state sequence is found by solving the related MPBVP with an indirect multiple shooting algorithm. For computational simplicity, the following assumption is introduced.

Assumption 5.5: It is assumed in the following that no singular cases occur, where two or more Hamiltonians are equal for a nonzero time interval[3]. Furthermore, it is assumed

[3] Solutions for singular cases can be found in the literature, e.g. in [18] and [139].

that at most one optimal solution per discrete state sequence exists. □

Remark 5.6: Singular cases are excluded since those cases are beyond the scope of this thesis. The existence of at most one optimal solution per discrete state sequence is assumed as this guarantees that if a solution is found, this solution is globally optimal. If the assumption about the existence of at most one solution per discrete state sequence does not hold, it is required to use approaches to find the globally optimal solution. For example, the optimization could be repeated with numerous different initializations, such that the optimization space is covered sufficiently dense for obtaining global results. If no global optimization techniques are applied, the overall tree search algorithm provides locally optimal solutions. □

5.2.2 Algorithm

The tree search algorithm based on branch and bound principles is developed to reduce the computational effort to find the globally optimal discrete state sequence when the HMP is applied, consider also Remark 5.6. Some parts of the concept of the algorithm follow the algorithm published in [162] combined with branch and bound principles. The main differences are the definition of the hybrid system and that indirect methods based on the HMP are used here in the underlying optimization compared to direct methods embedding nonlinear programming techniques. In [162], nodes of the tree correspond to controlled switches with discrete inputs, which are not considered in the unrelaxed HOCP solved here.

Formulation of the Tree

The key idea here is to create a tree and assign a branch of the tree to each possible sequence of discrete states such that the tree represents a reachability tree. A node n in the tree corresponds to a discrete state sequence ϱ with the last discrete state q. For a node n, successive nodes are determined by the discrete states that can be reached from discrete state q according to the discrete transition map Γ. The change from one node in the tree to its successor occurs when the continuous state trajectory meets a switching manifold and an autonomous switch is triggered. The required definitions are provided in the following:

Definition 5.7: A node n_i for $i \in \mathcal{N} = \{0, 1, ..., N_{\max}\}$ is a (sub)sequence of discrete states $\varrho_i = (q_0, ..., q_j)$ for $j \in \{0, ..., N\}$. $N_{\max}$ is the maximum number of nodes considering the discrete transition map Γ for a maximum of $\bar{N}$ autonomous switchings. Any node n_i in the tree can be in the set of terminal nodes $\mathcal{N}_e$, which is the case if the last discrete state q_j in the sequence of discrete states ϱ_i fulfills the target $T(t_e)$. □

The definition of the set of terminal nodes implies that the number of autonomous switchings contained in the optimal solution is not fixed beforehand.

Definition 5.8: An edge between two nodes n_i and n_w for $i, w \in \{0, 1, ..., N_{\max}\}$ and $i < w$ exists if and only if the subsequence of discrete states $\varrho_i = (q_0, ..., q_j)$ is contained in the subsequence $\varrho_w = (\varrho_i, q_{j+1})$ and discrete state q_{j+1} is reachable from q_j with an autonomous switching on the switching manifold $m_{q_j,q_{j+1}}(x(t_{j+1})) = 0$. □

Definition 5.9: A tree consists of nodes n_i for $i \in \{0, ..., N_{\max}\}$ and edges connecting the nodes given in Definitions 5.7 and 5.8. The base node is node $n_0 = (q_0)$. From n_0, the tree is spanned by considering all possible edges and resulting nodes up to the $\bar{N}$-th switch, i.e. the tree contains the sequences $\big\{\varrho = (q_0, ..., q_\nu, ..., q_j) | j \in \{0, ..., \bar{N}\}, q_\nu = \Gamma(q_{\nu-1}, x(t_\nu), \varsigma(t_\nu))$ for $\nu \in \{1, ..., j\}\big\}$. □

Corresponding to the definition of the tree, a relaxed hybrid system is defined based on the current node in the tree and the number of considered autonomous switchings.

Definition 5.10: A relaxed hybrid system $R(j, n_i)$ considers only autonomous switching up to the j-th switching following the discrete state sequence ϱ_i specified by the node n_i. With the j-th autonomous switching, the discrete state q_{j-1} is left and q_j is entered. Afterwards, controlled switching occurs. Note that $R(N+1, \cdot)$ means that no relaxation is considered for the hybrid system here, it does not give any statement on the number of autonomous switchings (except being less than $N+1$) in this sequence. □

Definition 5.11: The execution $\sigma_{R(j,n_i)}$ and costs $J_{R(j,n_i)}$ are the execution σ and costs J associated with the relaxed hybrid system $R(j, n_i)$. An optimal execution and optimal costs are denoted by $\sigma^*_{R(j,n_i)}$ and $J^*_{R(j,n_i)}$. If a discrete state sequence ϱ_i associated with node n_i does not possess a feasible solution as the relevant switching manifolds are not reached, then the costs are set to $J^*_{R(j,n_i)} = \infty$. □

Branch and Bound Concept

In principle, the optimal discrete state sequence for autonomous switching is found by comparing the costs of all possible discrete state sequences, which are the branches in the tree. In contrast to former algorithms based on the HMP, the algorithm introduced here avoids to analyze several branches by applying branch and bound methods. The concept of branch and bound is based on the fact that a branch can be pruned as soon as the lower bound on the minimally achievable costs in this branch is higher than the upper bound on the optimal cost over all branches known up to this point. This implies that whole bundles of branches originating from the same base branch specified by node n_i do not need to be analyzed if $J^*_{R(j,n_i)} > J^*_{R(N+1,n_s)}$ with $i, s \in \mathcal{N}$ and $j \in \{0, ..., N\}$. Here, $J^*_{R(N+1,n_s)}$ is the optimal cost for the discrete state sequence finishing in the terminal node n_s at time t_e.

A lower bound on the costs for a base branch with the discrete state subsequence ϱ_i is determined by solving the HOCP for the relaxed system $R(j, n_i)$, if more autonomous switchings are permitted after node n_i. Finding a locally optimal discrete state sequence for the part of the relaxed hybrid system $R(j, n_i)$ with controlled switching is inexpensive. In this algorithm, an optimal discrete state sequence is calculated by applying the Hamiltonian minimization condition with respect to the discrete control (3.8) directly in the indirect method.

During the optimization for a lower bound solution, it is unclear where to optimally switch on the switching manifold $m_{q_{j-1},q_j}(x(t_j)) = 0$ with $q_{j-1}, q_j \in \mathcal{Q}$ and $j \in \{1, ..., N\}$, which is the last autonomous switching considered in the node n_i. The optimal switching state $x^*(t_j)$ depends on the choice of the subsequent switching manifolds and the corresponding subsequent discrete state sequences, which are taken to the target $T(t_e)$ based on the subsequence ϱ_i. For different subsequent sequences of discrete states, the optimal switching state $x^*(t_j)$ has different values, which makes it difficult to determine a lower bound on the costs. By considering controlled switching from node n_i on, the optimal switching state $x^*(t_j)$ is picked on the manifold $m_{q_{j-1},q_j}(x(t_j)) = 0$, such that no other choice of a switching state $x(t_j)$ leads to smaller overall costs for branches emerging from node n_i. By this choice, it is ensured that the optimal overall costs of associated solutions provide a lower bound on the costs for all branches starting from node n_i.

Branch and Bound Algorithm

The tree search algorithm is shown in Algorithm 5.1. The selection of the next nodes to be analyzed is based on some sets and different search modes, which are defined and described in the following. The structure $\hat{M}$ contains all theoretically possible sequences of discrete states starting from discrete state q_0 with increasing length up to $\bar{N}$ switching instances. The sequences are based on the autonomous switching structure specified by the discrete transition map Γ with switching manifolds (function *createPossibleSequences*):

Definition 5.12: $\hat{M} = \big\{\varrho = (q_0, ..., q_\nu, ..., q_j) \mid j \in \{0, ..., \bar{N}\}$ and autonomous switchings according to $q_\nu = \Gamma(q_{\nu-1}, x(t_\nu), \varsigma(t_\nu))$ for $\nu \in \{1, ..., j\}\big\}$. □

The set E includes all possible extensions from node n_i according to the sequences in $\hat{M}$ (found by the function *getAllSeqStartingFromCurrentNodeWithOneMoreSwitch*):

Definition 5.13: $E = \big\{\varrho = (\varrho_i, q_{j+1}, ..., q_\nu, ..., q_{\bar{\nu}}) \mid \varrho_i \in \hat{M},\ j \in \{0, ..., \bar{N}-1\}$, and $\nu, \bar{\nu} \in \{1, ..., \bar{N}\}\big\}$. □

Furthermore, L is the set of live nodes from where the tree has still to be analyzed further, G are the nodes generated in the last iteration, and S is the set of nodes to be analyzed at the moment. Based on these sets, different search modes are applied. Following experience, a strategy of *depth-first* ($S = \{n_{\text{best}}\}$, where n_{best} is the node with lowest $J^*_{R(j,n_i)}$ from G), is selected initially, until a solution $\sigma^*_{R(N+1,n_i)}$ with an upper bound $J^*_{R(N+1,n_i)}$ on the optimal cost J^* is found such that pruning of branches can start early. After obtaining an upper bound, a strategy of *breadth-first* ($S = L$) or *best-first* ($S = \{n_{\text{best}}\}$ with n_{best} being the node with lowest costs from L) is applied. Which search mode is currently active is decided in the function *selectSearchMode*. The search begins at node n_0 corresponding to the initial state q_0 and moves step-by-step through parts of the tree where the costs indicate that the optimal solution may be found. The information stored with each node is $\hat{\Psi} = (\sigma^*_{R(j,n_i)}, J^*_{R(j,n_i)})$.

Algorithm 5.1:

```
$\hat{M}$ = $createPossibleSequences(N+1)$
$L = G = \{n_0\}$
$S = \hat{\Psi} = \emptyset$
upperBound = $\infty$
if ($n_0 \in \mathcal{N}_e$) then
    $(\sigma^*_{R(N+1,n_0)}, J^*_{R(N+1,n_0)}) = solveHOCP(R(N+1, n_0))$
    if $\sigma^*_{R(N+1,n_0)} \in T(t_e)$ then
```

```
        $\hat{\Psi} = \hat{\Psi} \cup \{(\sigma^*_{R(N+1,n_0)}, J^*_{R(N+1,n_0)})\}$
        upperBound = $J^*_{R(N+1,n_0)}$
    **end if**
**end if**
**while** $(L \neq \emptyset)$ **do**
    $S =$ *selectSearchMode*$(L, G)$
    $G = \emptyset$
    **for** ALL $(n \in S)$ **do**
        **if** $J^*_{R(j,n)} \leq$ upperBound **then**
            $E =$ *getAllSeqStartingFromCurrentNodeWithOneMoreSwitch*$(\hat{M}, j+1, n)$
            **for** ALL $(n^+ \in E)$ **do**
                **if** $n^+ \in \mathcal{N}_e$ **then**
                    $(\sigma^*_{R(N+1,n^+)}, J^*_{R(N+1,n^+)}) =$ *solveHOCP*$(R(N+1, n^+))$
                    **if** $\sigma^*_{R(N+1,n^+)} \in T(t_e)$ **then**
                        $\hat{\Psi} = \hat{\Psi} \cup \{(\sigma^*_{R(N+1,n^+)}, J^*_{R(N+1,n^+)})\}$
                        **if** $J^*_{R(N+1,n^+)} <$ upperBound **then**
                            upperBound = $J^*_{R(N+1,n^+)}$
                        **end if**
                    **end if**
                **end if**
                **if** $j+1 < N$ **then**
                    $(\sigma^*_{R(j+1,n^+)}, J^*_{R(j+1,n^+)}) =$ *solveRelaxedHOCP*$(R(j+1, n^+))$
                    **if** $J^*_{R(j+1,n^+)} <$ upperBound **then**
                        $G = G \cup \{n^+\}$
                    **end if**
                **end if**
            **end for**
        **end if**
    **end for**
    $L = (L \cup G) \setminus S$
**end while**
$(\sigma^*, J^*) = \min_{J \in \hat{\Psi}.J} (\sigma, J)$                                      □
```

Indirect Multiple Shooting

The functions *solveHOCP* and *solveRelaxedHOCP* use indirect multiple shooting to find the optimal solution for the given hybrid systems $R(N+1, n_i)$ and $R(j, n_i)$. The multiple shooting algorithm is set up similarly to the one in [136]. Here, the time interval $[t_0, t_e]$ is divided into several contiguous subintervals denoted by $I_k = [t_k, t_{k+1})$, $k \in \{0, 1, ..., N_k\}$ with $N_k \geq N$. All switching instants t_j with $j \in \{0, ..., N\}$ and the final time t_e form a boundary of such subintervals. Further boundaries can be specified, where continuity conditions on $x(t_k)$, $\lambda(t_k)$ and $H_q(t_k)$ for $q \in \mathcal{Q}$ have to hold. The additional boundaries can be introduced for increasing the numerical stability of the optimization. For each interval I_k, initial values of $x(t_k)$ and $\lambda(t_k)$ have to be guessed at first. These values are used to integrate the system (5.1) and (3.2) under consideration of (3.7) in the specified time interval up to t_{k+1}^-. At an autonomous transition at time t_j, conditions (3.4), (3.63), and $m_{q_{j-1},q_j}(x(t_j)) = 0$ for the desired transition from discrete state q_{j-1} to q_j have to be fulfilled. At a controlled switching triggered by the necessity to change the discrete state according to (3.8), Eq. (3.5), (3.6), and $x(t_j^-) = x(t_j)$ have to hold. The errors in the conditions resulting from the initial guess of $x(t_k)$ and $\lambda(t_k)$ constitute the residuals for an optimization function, that varies the initial values until all residuals are zero. If this is achieved, then an optimal solution σ^* is found. The domain of attraction for initial values around the optimal values $x^*(t_k)$ and $\lambda^*(t_k)$ can be enlarged by applying a combination of direct optimization with indirect multiple shooting as introduced in [175, 210].

5.2.3 Numerical Example

The tree search Algorithm 5.1 is applied to the following hybrid system with 4 discrete states, continuous state dimension 2, a scalar control input $u_q(t)$ and the following linear dynamics:

$$\begin{aligned} \dot{x}(t) &= A_q x(t) + B u_q(t) \qquad (5.6)\\ \text{with: } A_1 &= \begin{pmatrix} -1 & 2 \\ -2 & -1 \end{pmatrix}, A_2 = \begin{pmatrix} -1 & -2 \\ 1 & -0.5 \end{pmatrix}, \\ A_3 &= \begin{pmatrix} -0.5 & -5 \\ 1 & -0.5 \end{pmatrix}, A_4 = \begin{pmatrix} -1 & 0 \\ 2 & -1 \end{pmatrix}, B = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \end{aligned}$$

The different dynamics result from the corresponding choices of the variables $\varsigma_q(t)$ with $q \in \mathcal{Q}$. For all discrete states q, the cost function is $\phi_q = \frac{1}{2}(x^T(t)\,x(t) + u_q^2(t))$. Initial and final time are $t_0 = 0$sec and $t_e = 2$sec. The system shall be driven from the initial state $x_0 = (-8, -6)^T$ to the terminal state $x_e = (0, 0)^T$. The control variable is constrained in all discrete states to $-10 \leq u_q(t) \leq 10$. This leads to the following optimal control:

$$u_q^*(t) = \begin{cases} -10 & \text{for} \qquad\quad \hat{u}(t) < -10 \\ \hat{u}(t) & \text{for} \quad -10 \leq \hat{u}(t) \leq 10 \\ 10 & \text{for} \quad\; 10 < \hat{u}(t) \end{cases}, \qquad (5.7)$$

where $\hat{u}(t) = -(\lambda_1(t) + \lambda_2(t))$ and $\lambda_1(t)$ and $\lambda_2(t)$ are the adjoint variables. The switching manifolds are defined as follows: $m_{12} = x_2(t) + 5 = 0$, $m_{13} = x_1(t) + 5 = 0$, $m_{21} = -x_2(t) - 8 = 0$, $m_{23} = -m_{32} = x_1(t) - x_2(t) = 0$, $m_{24} = -m_{42} = x_1(t) + 2 = 0$, $m_{31} = -x_1(t) - 8 = 0$, and $m_{34} = -m_{43} = x_2(t) + 2 = 0$. This implies that it cannot be switched directly from discrete state 1 to 4 with an autonomous switching. Allowing at most 3 autonomous switchings and a maximal number of 5 switchings for combining autonomous and controlled switching, the optimal state sequence found is $\varrho^* = (1, 2, 3, 4)$ with a minimum cost $J^* = 21.6980$. Fig. 5.1 shows the optimal discrete state sequence, the state trajectory in the x_1-x_2-plane, as well as the state and adjoint state over time. The state trajectories are continuous as required, whereas at least one of the adjoint state trajectories has a discontinuity at a switching instant according to (3.4).

The path through the search tree is illustrated in Fig. 5.2. The discrete state number is specified above the nodes and the edges are labeled with the minimal costs for the hybrid system, when it is relaxed from the following node on. A number below a node means that the node is a terminal node with the specified costs. The label *nf* denotes that the path to the next node is not feasible. *nt* marks that the next node cannot be terminal, since it does not correspond to the desired terminal discrete state but the last permitted autonomous switch leads to it. *ub* indicates that the minimal costs for the system relaxed from the previous node on is already higher than the lowest upper bound on the optimal costs. Fig. 5.2 shows that the complexity of computation is reduced since less discrete state sequences are analyzed than in a brute-force algorithm. Here, 5 relaxed and 2 original (only autonomous switching) HOCPs are solved, whereas the brute-force approach solved 23 original HOCPs. An implementation of the latter algorithm found the optimal schedule in 424s, which is 2.4 times longer than the 177s for the proposed tree search method. Computations were performed on a AMD Athlon 6000+ processor with 3GHz, 2GB RAM running Matlab 2008a.

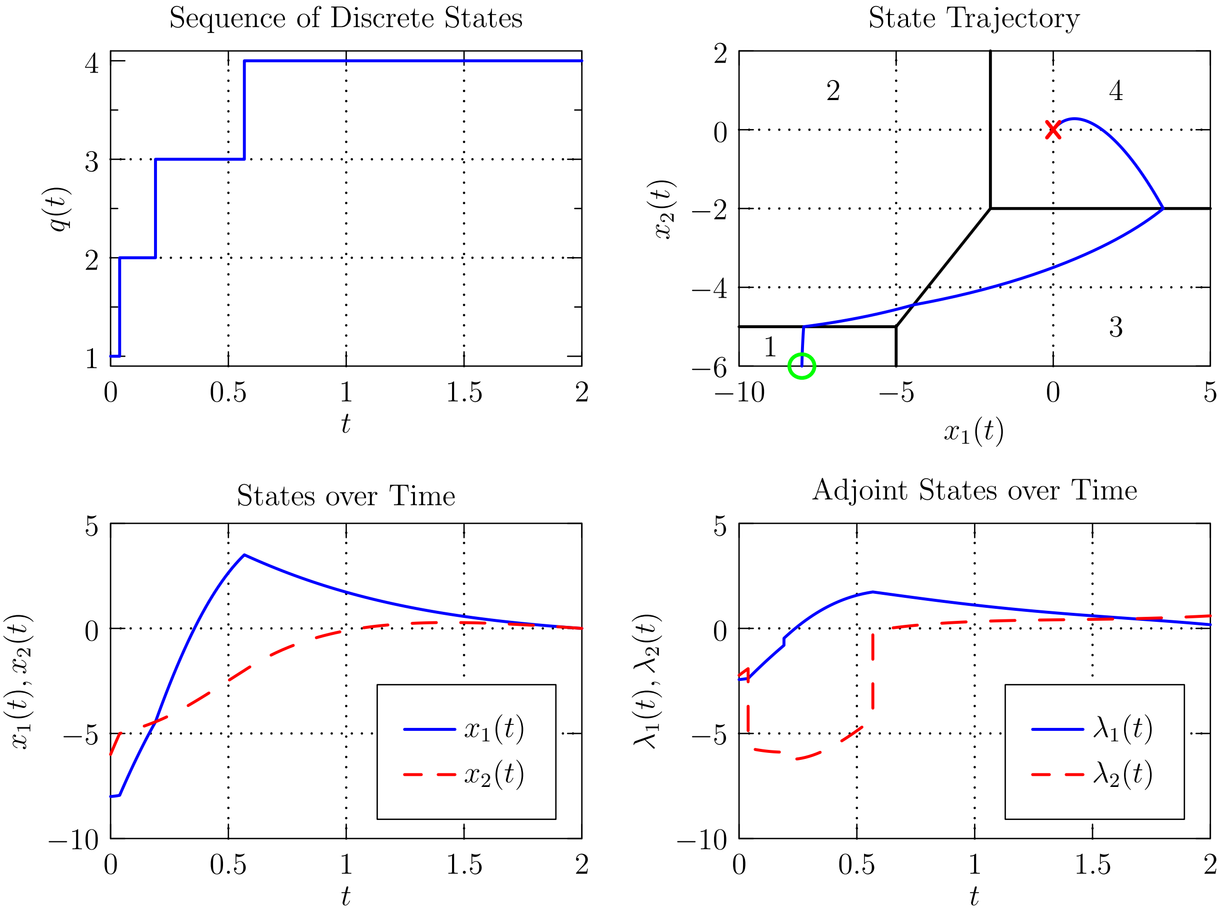


Fig. 5.1: Optimization result.

5.2.4 Discussion

The proposed tree search algorithm combines branch and bound principles with indirect multiple shooting for autonomously switched hybrid systems. The computational time is

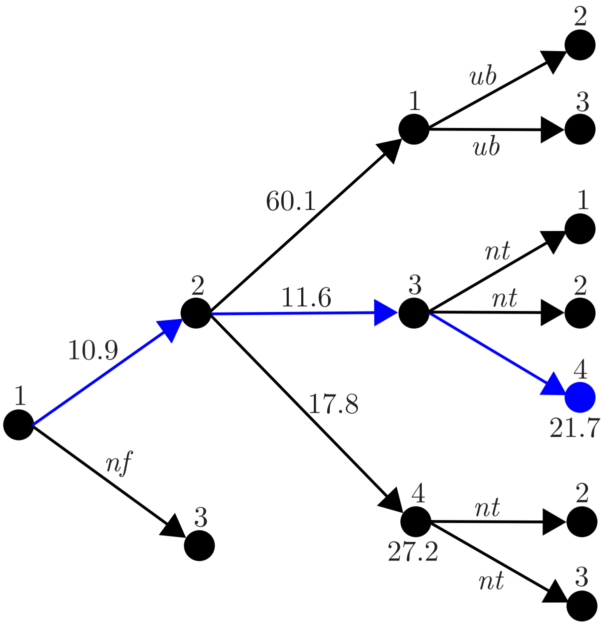


Fig. 5.2: Course of the tree search.

significantly reduced in the numerical example compared to algorithms that employ brute force search over sequences of discrete states based on indirect methods.

The considered HOCP also belongs to the class of mixed-integer programming problems. As discussed in Chap. 2, approaches to solve such problems efficiently include cutting-plane methods, decomposition methods, logic-based methods, and branch and bound methods. Several authors suppose that branch and bound forms the most efficient method. For example, in [53] the superior efficiency of a branch and bound method compared to some implementations of the other approaches has been shown for mixed-integer quadratic programming problems. In [99], the efficient solution capabilities of a branch and bound algorithm are extended to mixed-integer nonlinear programming problems. Given these experiences, it is expected that the proposed method provides a significant reduction in computation time compared to a standard brute-force approach for HOCPs. The efficiency of branch and bound concepts relies on pruning bundles of branches in an early stage of the search tree. Thus, it appears well-suited for hybrid systems which (a) contain switching manifolds that cannot be reached with the continuous control, so that following sub-branches are irrelevant, (b) have large differences in upper and lower cost bounds for different branches of the search tree, and (c) have tight relaxations, such that the solutions of the relaxed hybrid systems are close to the solutions of the corresponding original hybrid system. The more switches occur and discrete states exist, the larger the computational advantage of the method is in general compared to algorithms analyzing all possible discrete state sequences. Though the complexity is always reduced in our experience, it, however, cannot be guaranteed that the complexity is reduced as always for branch and bound methods. In comparison to approaches with discretized time, e.g. those in [162] and [16], the introduced algorithm has the advantage that the size of the search tree is only determined by the number of switches but does not grow exponentially with time.

If Assumption 5.5 holds, the algorithm finds the globally optimal solution. If there exist several local minima per discrete state sequence, the algorithm provides a locally optimal solution. By special approaches for global optimization with indirect methods, a global minimum can still be found in the case of several local minima, but this is beyond the scope here.

The optimization of the discrete state sequence for the parts with controlled switching uses the Hamiltonian minimization condition (3.8) with respect to the discrete control directly in the multiple shooting method. The results of the approach depend considerably on the initialization and the domain of convergence is relatively small. Nevertheless, the convergence speed is high and results are accurate. A difficulty in the implementation is to deal with the unknown number and times of controlled switchings, which also vary during the optimization. Instead of the chosen approach, any other indirect optimal control approach for controlled switching is appropriate as far as it can be combined with indirect multiple shooting for autonomous switching. For example, the mentioned approach in [136] seems promising regarding the ease of implementation and the size of the domain of convergence. There, a sequence of discrete states is selected initially and the lengths of the associated time intervals are varied until an optimal solution with optimal discrete state sequence is found.

5.3 An Algorithm for Gradient-Based Optimization of the Discrete State Sequence

In the following, an algorithm is introduced for HOCPs with partitioned continuous state space. The algorithm locally optimizes the discrete state sequence based on gradient information such that the computational complexity can be reduced by a factor of combinatorial complexity compared to a brute-force algorithm. This section is structured as follows: Sec. 5.3.1 briefly shows the considered HOCP with partitioned state space. Sec. 5.3.2 presents the algorithm, while its convergence is proved in Sec. 5.3.3. Sec. 5.3.4 illustrates the effectiveness of the algorithm with two numerical examples and in Sec. 5.3.5, a cooperative transportation scenario with multiple robots is solved with the novel algorithm. Sec. 5.3.6 provides a discussion of relevant aspects of the algorithm.

5.3.1 Hybrid Optimal Control Problem

The proposed algorithm is developed for hybrid systems with partitioned continuous state space. The definition of and the standing assumptions for this class of hybrid systems are presented in Sec. 2.2.3. The class of hybrid systems contains continuous and discrete states, continuous controls, continuous and discrete dynamics, and switching manifolds for autonomous switching, where switching manifolds are allowed to intersect. The corresponding HOCP is formulated in Definition 2.32. The necessary optimality conditions of the novel HMP derived in Sec. 3.3.2 are used in the algorithm. The optimality conditions provide gradients of the costs on switching manifolds and their intersections. The gradients enable the algorithm to optimize the discrete state sequence in parallel with the search for the continuous optimal control.

Definition 5.14: A switching point z_j is the vector consisting of the switching state-time pair $z_j = (x_j^T \;\; t_j)^T$, where the switching state is on the switching manifold $m_{q_{j-1},q_j}(x_j) = 0$. Here, j denotes the j-th switching. All switching points with switching times $t_0 < t_1 < ... < t_N < t_e$ are summarized in the vector

$$z = (z_0 \;\; z_1 \; ... \; z_N \; z_{N+1})^T , \tag{5.8}$$

where z_0 is the initial state-time point $(x_0^T \; t_0)^T$. Usually, the terminal point $z_{N+1} = (x_e^T \; t_e)^T$ is only partially specified, e.g. x_e or t_e might be free. □

5.3.2 Algorithm

This section introduces an algorithm that finds a locally optimal trajectory χ^* based on the optimality conditions of Theorem 3.4. The algorithm considers the optimal discrete state sequence ϱ^* as subject of optimal control that is varied and finally determined during a single optimization run. In contrast, earlier algorithms that use indirect shooting methods have combinatorial complexity as they can only solve the HOCP for a pre-specified sequence of discrete states ϱ. Thus, finding a locally optimal solution (χ^*, ϱ^*) requires to search over every possible discrete state sequence.

The HOCP $\min_{\upsilon \in \mathcal{U}} J(\upsilon)$ (2.5) is decomposed, such that the algorithm can be set up with two layers. From the decomposition, an HOCP $\min_z J(z)$ is obtained, which is solved for the optimal switching point vector z^* on the upper layer of the algorithm. The optimal switching points z^* satisfy the optimality conditions (3.62) and (3.63). Between two consecutive switching points z_j and z_{j+1}, only state and control trajectories, that satisfy the optimality conditions (2.2), (3.60), (3.61), and (3.64), are allowed and are determined in the lower layer.

In every iteration l, the algorithm produces a feasible sequence of discrete states $\varrho^l = (q_0^l, ..., q_{N(l)}^l)$ and switching points $z_j^l = (x_j^{lT} \; t_j^l)^T$ for $j = 1, ..., N(l)$. Here, feasibility means that an optimal control $\upsilon_{q_j^l}^* \in \mathcal{U}_{q_j^l}$ exists, which transfers system (2.2) from switching point z_j^l to z_{j+1}^l for $j \in \{0, 1, ..., N(l)\}$ without leaving the state space $\mathcal{X}_{q_j^l}$.

The algorithm is presented in the following:

Algorithm 5.2:

Step 0 **Initialization:** Choose a feasible discrete state sequence $\varrho^0 = (q_0^0, ..., q_{N(0)}^0)$ with switching points z^0. Set $l = 0$ and $j_z = 1$.

Step 1 **(Lower Layer) Optimal Control of Subproblems:** Solve

$$\min_{\upsilon_{q_j^l} \in \mathcal{U}_{q_j^l}} J_{q_j^l} = \min_{\upsilon_{q_j^l} \in \mathcal{U}_{q_j^l}} \int_{t_j^l}^{t_{j+1}^l} \phi_{q_j^l}(x_{q_j^l}(t), u_{q_j^l}(t))\, dt \big(+ g(x(t_e)) \big) \tag{5.9}$$

for every discrete state $q_j^l \in \varrho^l$ for $j \in \{0, 1, ..., N(l)\}$. For the last discrete state $q_{N(l)}$, the terminal cost $g(x(t_e))$ is added. The optimization uses z_j^l as initial value and z_{j+1}^l as bounding condition, and it takes the dynamics (2.2) and feasible control trajectories $\upsilon_{q_j^l} \in \mathcal{U}_{q_j^l}$ into account. Indirect shooting on first-order optimality conditions is used [25, 32, 136] as also discussed in Sec. 4.2.2 because it provides highly accurate solutions.

Step 2a (Upper Layer) Gradients of Costs on Switching Manifolds: Determine for all z_j^l with $j = \{1, ..., N\}$ the gradients of the costs $\nabla^{\mathrm{m}}_{z_j^l} J(z^l) = \left(\nabla^{\mathrm{m}}_{x_j^l} J(z^l) \ \nabla^{\mathrm{m}}_{t_j^l} J(z^l)\right)$ projected onto the switching manifolds $m_{q_{j-1}^l, q_j^l} = 0$ using (3.62) and (3.63):

$$\nabla^{\mathrm{m}}_{x_j^l} J^T(z^l) = \lambda(t_j^l) - \lambda(t_j^{l-}) + \sum_{M_{q_{j-1}^l, i} \in \mathcal{N}^1_{q_{j-1}^l}} \nabla_x m^T_{q_{j-1}^l, i}(x(t_j^l))\pi_{j,(q_{j-1}^l, i)}$$

$$+ \sum_{M_{i,k} \in \mathcal{N}^2_{q_{j-1}^l}} \nabla_x m^T_{i,k}(x(t_j^l))\pi_{j,(i,k)} \tag{5.10}$$

$$\nabla^{\mathrm{m}}_{t_j^l} J(z^l) = H_{q_{j-1}}(t_j^{l-}) - H_{q_j}(t_j^l) \ . \tag{5.11}$$

If $\nabla^{\mathrm{m}}_{z_j^l} J(z^l) = 0 \ \forall z_j^l$, then stop.

Step 2b (Upper Layer) Update of Switching Points: Set $j := j_z$, $d_{z_j^l} = -\nabla^{\mathrm{m}}_{z_j^l} J^T(z^l)$, $\alpha = c\beta^p$ with $c \in \mathbb{R}^+$, $\beta \in (0,1)$, $p = 0$, and $z_j^{l+1} = z_j^l + \alpha d_{z_j^l}$. If z_j^{l+1} is beyond the boundaries of $m_{q_{j-1}^l, q_j^l} = 0$, then z_j^{l+1} is projected to the corresponding neighboring switching manifolds, such that $\|z_j^{l+1} - z_j^l\| = \|\alpha d_{z_j^l}\|$ holds. For notational convenience, we restrict the consideration to the case, where the old sequence $\varrho^l = (*, q_{j-1}^l, q_j^l, *)$ and the new one $\varrho^{l+1} = (*, q_{j-1}^{l+1}, q_j^{l+1}, q_{j+1}^{l+1}, *)$ with $q_{j-1}^l = q_{j-1}^{l+1}$ and $q_j^l = q_{j+1}^{l+1}$ differ only in the discrete state q_j^{l+1} that is additionally visited in the sequence ϱ^{l+1}. Then, the updated switching point z_j^{l+1} is accepted if the Armijo-like criterion [4]

$$J(z^{l+1}) - J(z^l) \leq \begin{cases} -\tau_0 \alpha d^T_{z_j^l} d_{z_j^l} & \text{if } x_j^{l+1} \in M_{q_{j-1}^l, q_j^l} \\ -\tilde{\tau}_0 \alpha d^T_{\min} d_{\min} & \text{if } x_j^{l+1} \notin M_{q_{j-1}^l, q_j^l} \end{cases} \tag{5.12}$$

is satisfied with $\tau_0 \in (0,1)$, $\tilde{\tau}_0 = \beta\tau_0$, and

$$\|d_{\min}\| = \min\left\{\|d_{z_j^l}\|, \|d_{z_j^{l+1}}\|, \|d_{z_{j+1}^{l+1}}\|\right\} . \tag{5.13}$$

If z_j^{l+1} is rejected, then set $p = p + 1$ and repeat Step 1. If z_j^{l+1} is accepted, set $l := l + 1$, $p := 0$, and $j_z := j_z + 1$. If $j_z > N(l)$ (the algorithm has just updated the last switching point $z^l_{N(l)}$), then set $j_z := 1$. Goto Step 1. □

Projection of Switching Points

Here, the projection of switching points is described after some introductory comments. In general, the projection of switching points on nonlinear switching manifolds and for non-convex state partitions $\mathcal{X}_q$ can be handled numerically. To enable a simplified implementation of the algorithm, the switching manifolds are assumed to be affine and all partitions $\mathcal{X}_q$ to be convex. For affine switching manifolds, non-convex $\mathcal{X}_q$ can be convexified by introducing additional switching manifolds. For convex $\mathcal{X}_q$, also all switching manifolds are convex.

To update the switching point z_j^l, a candidate $\bar{z}_j^{l+1}$ is found by proceeding from z_j^l in the descent direction with step size α:

$$\bar{z}_j^{l+1} = z_j^l + \alpha d_{z_j^l} \ . \tag{5.14}$$

If the corresponding $\bar{x}_j^{l+1}$ fulfills $\bar{x}_j^{l+1} \in M_{q_{j-1}^l, q_j^l}$, the update of the switching point is valid: $z_j^{l+1} = \bar{z}_j^{l+1}$. If the switching state satisfies $\bar{x}_j^{l+1} \notin M_{q_{j-1}^l, q_j^l}$, it is on the unbounded extension of $M_{q_{j-1}^l, q_j^l}$. Then, $\bar{z}_j^{l+1}$ has crossed the boundary $\partial M_{q_{j-1}^l, q_j^l}$, which implies that the update step $\alpha d_{z_j^l}$ enforces a new sequence ϱ^{l+1} of discrete states. Note that for reasons of convergence only sequences ϱ^{l+1} are considered that are neighbors of the old sequence ϱ^l, i.e. there must exist trajectories χ^{l+1} with ϱ^{l+1} and χ^l with ϱ^l that are arbitrarily close to each other. The old sequence $\varrho^l = (*, \triangle, *)$ differs from the new one $\varrho^{l+1} = (*, \square, *)$ in the subsequences $\triangle = (q_j^l, ..., q_{j+\nu-1}^l)$ and $\square = (q_j^{l+1}, ..., q_{j+\varpi-1}^{l+1})$ with $\nu, \varpi \in \mathbb{N}_0$. Here, ν and ϖ denote how many discrete states disappear from the old sequence ϱ^l and how many are added to ϱ^{l+1}, respectively. For example, $\nu = 0$ and $\varpi = 1$ imply that $\triangle = \emptyset$ and $\square = \{q_j^{l+1}\}$, see also Fig. 5.3. For simplicity and without loss of generality, only $\nu = 0$ and $\varpi = 1$ is considered in the following.

Now, the goal is to find the switching point z_j^{l+1}, which belongs to the discrete state sequence $\varrho^{l+1} \neq \varrho^l$. For $m_{q_{j-1}^l, q_j^l}(x_j^l(t_j^l)) = 0$, assume that the trajectory of iteration l jumps from discrete state q_{j-1}^l to q_j^l. Then the sequence $(q_0^l, ..., q_{j-1}^l)$ remains the same for iteration $l+1$, since only the switching point z_j^l is shifted. From the switching point z_j^l and with the scaled descent direction $\alpha d_{z_j^l}$, the switching point z_j^{l+1} has to be found associated with the switching manifold $m_{q_{j-1}^{l+1}, q_j^{l+1}}(x_j^{l+1}(t_j^{l+1})) = 0$ with $q_{j-1}^l = q_{j-1}^{l+1}$ and $q_j^l \neq q_j^{l+1}$. To find z_j^{l+1}, a projection $P(z_j^l, \alpha d_{z_j^l})$ that maintains the length $\|\alpha d_{z_j^l}\|$ is applied. The projection works as follows: Basically a circle around z_j^l is considered with radius $\|\alpha d_{z_j^l}\|$ and tangent $\nabla_x m_{q_{j-1}^l, q_j^l}(x_j^l(t_j^l))$ at $\bar{z}_j^{l+1}$. Each intersection of the circle with switching manifolds delivers a candidate for a new switching point. Out of the set of candidate points, a subset is picked, which contains the switching points z_j^{l+1} and z_{j+1}^{l+1} leading from $q_{j-1}^l = q_{j-1}^{l+1}$ over q_j^{l+1} to $q_{j+1}^{l+1} = q_j^l$ (since $\nu = 0$ and $\varpi = 1$ in this example). From $q_{j+1}^{l+1} = q_j^l$ on, the rest of the old sequence $(q_{j+1}^l, ..., q_{N(l)+1}^l)$ with the old switching points completes the new sequence ϱ^{l+1}. If no intersection of the circle with a switching manifold restricting q_{j-1}^l exists, then the projection is declared to be infeasible and a new trial is started with a reduced step size $\|\alpha d_{z_j^l}\|$. The property of the projection $P(z_j^l, \alpha d_{z_j^l})$ to keep the length of the update step equal for all candidate points becomes important in the convergence analysis. A small example with 4 discrete states illustrates the update procedure in Fig. 5.3.

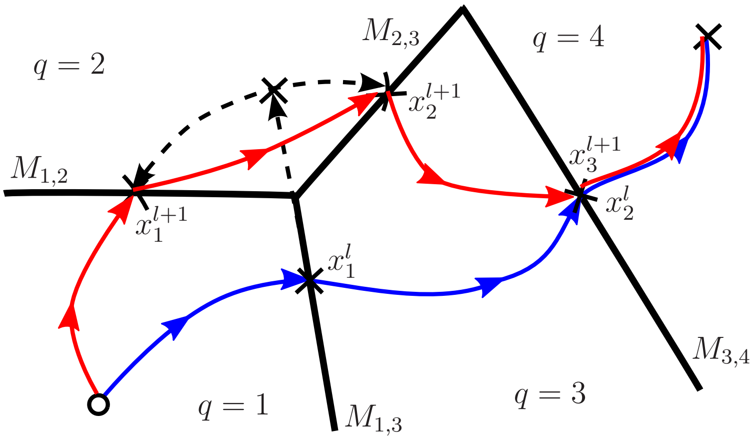


Fig. 5.3: Example for the update of a switching state that leads to a new discrete state sequence. Starting with the discrete state sequence $\varrho^l = (1, 3, 4)$ in iteration l, the gradient $\nabla^{\mathrm{m}}_{x_1^l} J(z^l)$ of the costs at the switching state x_1^l is determined. According to the gradient, the switching state x_1^l is shifted. Since the update exceeds the boundaries of the switching manifold $M_{1,3}$, it is projected to the switching manifolds $M_{1,2}$ and $M_{2,3}$ delivering the switching states x_1^{l+1} and x_2^{l+1} in iteration $l + 1$. The new discrete state sequence is $\varrho^{l+1} = (1, 2, 3, 4)$.

5.3.3 Convergence

In the following the convergence of the algorithm to a locally optimal solution is proven. Here, the standard concept of convergence of an algorithm [130] cannot be applied, since the dimension $N(l)$ of each switching point vector z^l (5.8) in the sequence of iterations $\{z^l\}_{l=1}^{\infty}$ may change at every iteration l. Furthermore, the gradient of the cost projected onto the switching manifolds $\nabla^{\mathrm{m}}_z J(z^l) = (\nabla^{\mathrm{m}}_{z_1^l} J(z^l), ..., \nabla^{\mathrm{m}}_{z_{N(l)}^l} J(z^l))$ is only piecewise continuous. However, the cost J is continuous on the whole state space according to Lemma 3.3. A suitable concept of a *convergent algorithm* similar to the one used in [7] is introduced.

Definition 5.15: The algorithm is convergent if every sequence $\{z^l\}_{l=1}^{\infty}$ fulfills the elementwise condition $\lim_{l \to \infty} \|z_j^l - z_i^*\| = 0$ for $j \in \{1, ..., N(l)\}$ and its corresponding $i \in \{1, ..., N^*\}$, where $\nabla^{\mathrm{m}}_z J(z^*) = 0$. □

Remark 5.16: For simplicity, it is assumed that an optimal switching point vector z^* satisfies $\nabla^{\mathrm{m}}_z J(z^*) = 0$. But note that there exist cases, where $\nabla^{\mathrm{m}}_{t_i^l} J(z^*) \neq 0$ for some $i \in \{1, ..., N^*\}$ according to (3.63). □

In the following, it is shown step-by-step that Algorithm 5.2 is convergent, compare also to the proof of the algorithm for systems with controlled switching in [7].

Lemma 5.1: Consider the Assumptions 2.31, 2.11, and 2.12 to hold. Then, the cost J is bounded for any feasible trajectory χ with initial condition x_0 at t_0, switching points z_j, and final condition x_e or t_e. □

Proof: From Assumption 2.31.5, $f_q(x(t), u_q(t))$ is Lipschitz continuous and bounded with respect to its arguments. For all $q \in \mathcal{Q}$, the trajectory χ is in a compact set. Here, it is assumed that, if x_e is specified, it is reached in finite time. As the cost J is continuous and $g(x(t_e))$ and $\phi_q(x(t), u_q(t))$ are bounded with respect to their arguments, $|J(z)|$ is bounded from above. ■

Definition 5.17: Algorithm 5.2 has *sufficient descent* if for every iteration l and every $\kappa > 0$, there exists $\eta > 0$ such that, if elementwise $\|z_j^l - z_i^*\| > \kappa$ for $j \in \{1, ..., N(l)\}$ and its corresponding $i \in \{1, .., N^*\}$, then

$$J(z^{l+1}) - J(z^l) \leq -\eta\,. \tag{5.15}$$

□

Proposition 5.2: Let the standing assumptions hold. If Algorithm 5.2 has sufficient descent and bounded cost, then it is convergent. □

Proof: If the algorithm has sufficient descent, then at every switching point z^l in the infinite sequence $\{z^l\}_{l=1}^{\infty}$ the cost J will decrease by a non-zero amount. As J is bounded, this must lead to convergence in the sense of Definition 5.15. ■

Now, it is shown that the applied Armijo-like criterion ensures that the algorithm has sufficient descent. At first, it is proven that an Armijo-like step is always possible if elementwise $\|z_j^l - z_i^*\| > \kappa$ for any $\kappa > 0$, for $j \in \{1, ..., N(l)\}$ and corresponding $i \in \{1, .., N^*\}$. Here, the piecewise differentiability of J projected onto switching manifolds is applied.

Proposition 5.3: Let the standing assumptions be fulfilled, let J be continuously differentiable in a neighborhood $B(z^l, \rho)$, $\rho > 0$, $\|z^{l+1} - z^l\| \leq \rho$, and let $\tau_0 \in (0, 1)$, $\beta \in (0, 1)$ be specified. Set $z^{l+1} = z^l + \beta^p d_{z^l}$ and $d_{z^l} = -\nabla_z^{\mathrm{m}} J^T(z^l)$, so that $\nabla_z^{\mathrm{m}} J(z^l)\, d_{z^l} < 0$, then there exists a finite $p \in \mathbb{N}_0$, such that

$$J(z^l + \beta^p d_{z^l}) - J(z^l) \leq -\tau_0 \beta^p d_{z^l}^T\, d_{z^l}\,, \tag{5.16}$$

i.e. the Armijo-like criterion is well-defined for updates $\beta^p d_{z^l}$ that leave the sequence $\varrho^l = \varrho^{l+1}$ unchanged. □

Proof: See [58]. ■

Condition (5.16) can easily be shown to hold also for variations at single switching points z_j^l with the gradient of the cost $\nabla_{z_j^l}^{\mathrm{m}} J(z^l)$.

Next, it is shown that the criterion also holds in a similar form for updates leading to a change in the discrete state sequence. Therefore, define the switching point $z_{s,j}^l = z_j^l + \beta^s d_{z_j^l}$ with $s \in \mathbb{R}$ on the intersection of two neighboring switching manifolds, that means the switching state $x_{s,j}^l \in \partial M_{q_{j-1}^l, q_j^l} \cap \partial M_{q_{j-1}^{l+1}, q_j^{l+1}}$. Furthermore, a switching point z^l, that remains unchanged except for a change of $\beta^p d_{z_j^l}$ in the j-th subvector, is denoted by $z^l(\beta^p d_{z_j^l})$.

Proposition 5.4: Consider the standing assumptions to be satisfied. For iteration l_k, let $x_i^* \notin M_{q_{j-1}^{l_k},q_j^{l_k}}$ $\forall i \in \{1, ..., N^*\}$, $\|z_i^* - z\| > \delta$ $\forall x \in M_{q_{j-1}^{l_k},q_j^{l_k}}$, $\delta > 0$, and $x_j^{l_k} \in M_{q_{j-1}^{l_k},q_j^{l_k}}$. Then there exists an iteration $l \in \mathbb{N}$, $l \geq l_k$, and a $p \in \mathbb{N}_0$ such that

$$J(z^{l+1}) - J(z^l) \leq -\tilde{\tau}_0 \beta^p d_{\min}^T d_{\min}\,, \tag{5.17}$$

where $\|d_{\min}\| = \min\left\{\|d_{z_j^l}\|, \|d_{z_j^{l+1}}\|, \|d_{z_{j+1}^{l+1}}\|\right\}$, and the updated sequence ϱ^{l+1} is different from the old one ϱ^l. □

The idea of the proof is the following: An update of a switching point leading to a new discrete state sequence is separated into two update steps: The first update step leads to a switching point on the old switching manifold and the second update step to a point on the new switching manifolds. Formulating the difference between the costs of the new and the old discrete state sequence in terms of Armijo steps in both parts of the update step and combining the results, it can be shown that (5.17) can always be fulfilled.

Proof: Assume first, that $x_j^{l_k} \in \text{Int}(M_{q_{j-1}^{l_k},q_j^{l_k}})$, where $\text{Int}(M_{q_{j-1}^{l_k},q_j^{l_k}})$ is the interior of $M_{q_{j-1}^{l_k},q_j^{l_k}}$. By Proposition 5.3 and $x_j^l \in \text{Int}(M_{q_{j-1}^{l_k},q_j^{l_k}})$ being the result of several Armijo steps (5.16) starting from $z_j^{l_k}$ with $l \geq l_k$, $\|z_j^l - z_{s,j}^l\|$ can be made arbitrarily small with increasing l. In particular for some l, there exists κ^l, s.t.

$$\|z_j^l - z_i^*\| > \kappa^l > \frac{1}{\beta^2}\|z_j^l - z_{s,j}^l\|\,. \tag{5.18}$$

Now, let $l, p \in \mathbb{N}$, such that

$$z_j^l + \beta^{p+2} d_{z_j^l} \in \{M_{q_{j-1}^l,q_j^l}, \mathbb{R}\}\,, \quad z_j^l + \beta^{p+1} d_{z_j^l} \notin \{M_{q_{j-1}^l,q_j^l}, \mathbb{R}\} \tag{5.19}$$

$$J(z^l(\beta^{p+y} d_{z_j^l})) - J(z^l) \leq -\tau_0 \beta^{p+y} d_{z_j^l}^T d_{z_j^l} \tag{5.20}$$

for every $y \in \{0, 1, 2\}$ and $d_{z_j^l} = -\nabla_{z_j^l}^{\text{m}} J^T(z^l)$, see Fig. 5.4. Eq. (5.19) implies

$$\beta^{p+2}\|d_{z_j^l}\| \leq \|z_j^l - z_{s,j}^l\| \leq \beta^{p+1}\|d_{z_j^l}\|\,, \tag{5.21}$$

which leads together with (5.18) to

$$\beta^p\|d_{z_j^l}\| \leq \frac{1}{\beta^2}\|z_j^l - z_{s,j}^l\| < \kappa^l\,. \tag{5.22}$$

The projection $P(z_j^l, \beta^p\|d_{z_j^l}\|)$ delivers the new switching points $z_v \in \{z_j^{l+1}, z_{j+1}^{l+1}\}$, which are used if the Armijo-like test (5.17) is successful. As the projection maintains the step length,

$$\|z_v - z_j^l\| = \beta^p\|d_{z_j^l}\| \overset{(5.22)}{<} \kappa^l \tag{5.23}$$

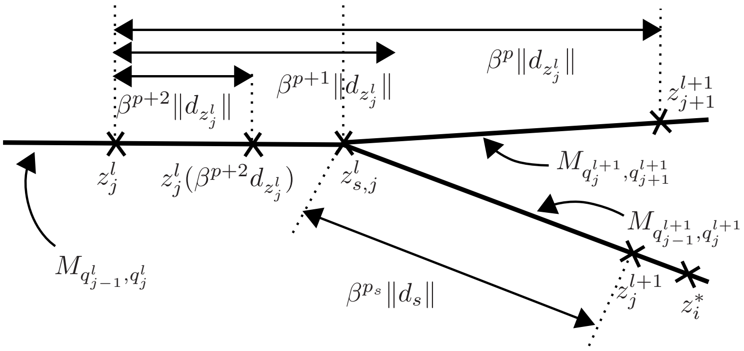

$\beta^p\|d_{z_j^l}\|$
$\beta^{p+1}\|d_{z_j^l}\|$
$\beta^{p+2}\|d_{z_j^l}\|$
z_{j+1}^{l+1}
z_j^l
$z_j^l(\beta^{p+2}d_{z_j^l})$
$z_{s,j}^l$
$M_{q_j^{l+1},q_{j+1}^{l+1}}$
$M_{q_{j-1}^{l+1},q_j^{l+1}}$
$M_{q_{j-1}^l,q_j^l}$
$\beta^{p_s}\|d_s\|$
z_j^{l+1}
z_i^*

Fig. 5.4: Switching points used in the proof.

is true $\forall z_v$. The following inequalities hold $\forall z_v$:

$$\|z_v - z_j^l\| = \|z_v - z_{s,j}^l + z_{s,j}^l - z_j^l\| \leq \|z_v - z_{s,j}^l\| + \|z_{s,j}^l - z_j^l\|$$

$$\Rightarrow \|z_v - z_{s,j}^l\| \geq \|z_v - z_j^l\| - \|z_{s,j}^l - z_j^l\| \overset{(5.23),(5.21)}{\geq} \beta^p(1-\beta)\|d_{z_j^l}\| \,. \tag{5.24}$$

Define z_s^l to be the vector of switching points that only differs from z^l in replacing z_j^l by $z_{s,j}^l$. By Proposition 5.3 it can be concluded that there exists a finite $l \in \mathbb{N}$ leading to a sufficiently small $\beta^p\|d_{z_j^l}\|$, such that $\beta^{p_s}\|d_s\|$ exists to satisfy:

$$J(z^{l+1}) - J(z_s^l) \leq -\tau_0\beta^{p_s}d_s^T d_s \,, \tag{5.25}$$

where d_s and β^{p_s} satisfy $\|d_s\| = \min(\|d_v\|)$, $v \in \{j, j+1\}$, $d_v = -\nabla_{z_v}^{\mathrm{m}} J^T(z_s^l)$, and the corresponding

$$\|z_v - z_{s,j}^l\| = \beta^{p_s}\|d_s\| \,. \tag{5.26}$$

By the continuity of J and $\|z_v - z_j^l\| < \|z_j^l - z_i^*\|$, it follows for sufficiently small $\beta^p\|d_{z_j^l}\|$ that $d_s^T d_v > 0$. Now, it remains to show that (5.17) holds:

$$\begin{aligned}
J(z^{l+1}) - J(z^l) &= J(z^{l+1}) - J(z_s^l) + J(z_s^l) - J(z^l) \\
&\leq J(z^{l+1}) - J(z_s^l) + J(z^l(\beta^{p+2}d_{z_j^l})) - J(z^l) \\
&\overset{(5.25),(5.20)}{\leq} -\tau_0\beta^{p_s}d_s^T d_s - \tau_0\beta^{p+2}d_{z_j^l}^T d_{z_j^l} \\
&\overset{(5.26),(5.24)}{\leq} -\tau_0\beta^p\|d_{z_j^l}\| \left[(1-\beta)\|d_s\| + \beta^2\|d_{z_j^l}\|\right] \\
&\leq -\tau_0\beta^p\|d_{z_j^l}\| \left[\beta(1-\beta)\|d_s\| + \beta^2\|d_{z_j^l}\|\right] \\
&\overset{(5.13)}{\leq} -\tau_0\beta\beta^p\|d_{z_j^l}\| \, \|d_{\min}\| \\
&\leq -\tilde{\tau}_0\beta^p d_{\min}^T d_{\min} \,.
\end{aligned} \tag{5.27}$$

Likewise, the same result can be obtained, if $x_j^l \in \partial M_{q_{j-1}^l,q_j^l} \cap \partial M_{q_{j-1}^{l+1},q_j^{l+1}}$ with $q_j^l \neq q_j^{l+1}$.■

Remark 5.18: For simplified notation, Proposition 5.4 is given for one additional discrete state q_j^{l+1} in the new sequence ϱ^{l+1} compared to the old one ϱ^l. The proof can be extended to handle all cases of possible combinations of additional and removed discrete states. □

Combining the results above, the sufficient descent property of the algorithm follows.

Proposition 5.5: Let the standing assumptions hold. For every $l \in \mathbb{N}$ and $\kappa^l > 0$ there exists $\eta > 0$, such that, if $\|z_j^l - z_i^*\| > \kappa^l$ for $j \in \{1, ..., N(l)\}$ and corresponding $i \in \{1, .., N^*\}$, then

$$J(z^{l+1}) - J(z^l) \leq -\eta\,. \tag{5.28}$$

□

Proof: If $\|z_j^l - z_i^*\| > \kappa^l$ for $j \in \{1, ..., N(l)\}$ and corresponding $i \in \{1, .., N^*\}$, then there exists $\epsilon > 0$ such that $\|\nabla^{\mathrm{m}}_{z_j^l} J(z^l)\| = \|d_{z_j^l}\| > \epsilon$. Propositions 5.3 and 5.4 imply that there always exists a finite $p \in \mathbb{N}_0$, such that a new switching point z^{l+1} can be determined with $\|z_v - z_j^l\| = \beta^p \|\nabla^{\mathrm{m}}_{z_j^l} J(z^l)\|\ \forall z_v \in \{z_j^{l+1}, ..., z_{j+\varpi}^{l+1}\}$. Here, z_v is any updated switching point and x_v may also be on $M_{q_{j-1}^l, q_j^l}$. If the update is across an intersection of switching manifolds and the algorithm does not reach the optimal switching point z_i^* exactly with this update, then there also exist $\epsilon_v > 0$, such that $\|\nabla^{\mathrm{m}}_{z_v} J(z^{l+1})\| = \|d_v\| > \epsilon_v$ holds for all z_v. Let $\epsilon_{\min}$ be $\min\{\epsilon, \epsilon_v\}\ \forall \epsilon_v$ and recall $\|d_{\min}\| = \min\left\{\|d_{z_j^l}\|, \|d_v\|\right\}$. In each iteration l, the sufficient descent condition

$$J(z^{l+1}) - J(z^l) \leq -\tilde{\tau}_0 \beta^p \epsilon_{\min}^2 = -\eta \tag{5.29}$$

holds, since either the descent property (5.16) or (5.17) is satisfied with $\tilde{\tau}_0 < \tau_0$ and $\|d_{\min}\| > \epsilon_{\min}$. ■

Theorem 5.6: Consider the Assumptions 2.31, 2.11, and 2.12 to hold. Then, a sequence $\{z^l\}_{l=1}^{\infty}$ of switching points, which is a solution of Algorithm 5.2, satisfies the elementwise equality:

$$\lim_{l \to \infty} \|z_j^l - z_i^*\| = 0 \tag{5.30}$$

for $j \in \{1, ..., N(l)\}$ and corresponding $i \in \{1, .., N^*\}$ with $\nabla^{\mathrm{m}}_z J(z^*) = 0$. □

Proof: The result is a direct consequence of Propositions 5.5 and 5.2. ■

Remark 5.19: In the current formulation of the algorithm and the convergence analysis, it is assumed that the optimal trajectory $\chi^*_{q_j^l}(\upsilon^*_{q_j^l}, x_0)$ which starts in z_j^l and reaches z_{j+1}^l stays in $\mathcal{X}_{q_j^l}$ for $t \in [t_j^l, t_{j+1}^l]$. If the assumption cannot be met for a specific problem, then it is necessary to add and remove switching points in-between two switching points. □

5.3.4 Numerical Examples

Linear System

In the first example, the partitioned state space contains 11 discrete states and each discrete state has a linear dynamics $\dot{x}(t) = A_q x(t) + B_q u(t)$ with $x(t) \in \mathbb{R}^2$, $u(t) \in U_q \subset \mathbb{R}^2$ and a running cost function $\phi_q = 0.5x^T(t)S_q x(t) + 0.5u^T(t)R_q u(t)$. The system is designed such that there exists a unique optimum. The algorithm changes the initial sequence $\varrho^0 = (1,3,4,6,7,10)$ to $\varrho = (1,3,4,7,10)$, $\varrho = (1,3,4,7,8,10)$, $\varrho = (1,3,4,5,7,8,10)$, and finally to the optimal sequence $\varrho^* = (1,3,4,5,8,10)$ with cost $J = 10.05$ in 0.19 h computational time.[4] The results are illustrated in Fig. 5.3.4, where it can be seen that the optimal state trajectory passes the intersection of the switching manifolds $m_{5,7} = 0$, $m_{5,8} = 0$, and $m_{7,8} = 0$. The parameters used in the different discrete states are provided in Sec. A.1.

The optimal solution and the computations are compared to results of a brute force implementation based on multiple shooting. For a global approach, the brute force algorithm investigates every of the 3976 possible discrete state sequences in the partitioned state space separately with at most 6 switchings beginning from q_0. The brute force solver finds the same optimum with $\varrho^* = (1,3,4,5,8,10)$ and $J = 10.05$ and it needs 275.3 h for the computation. This is about 1450 times longer than the proposed algorithm.

Nonlinear System

In the second example, the same partition of the state space as in example 1 is used. Now, the task is to control a unicycle [145]:

$$\dot{x}_1 = v_q \cos x_3 \tag{5.31}$$

$$\dot{x}_2 = v_q \sin x_3 \tag{5.32}$$

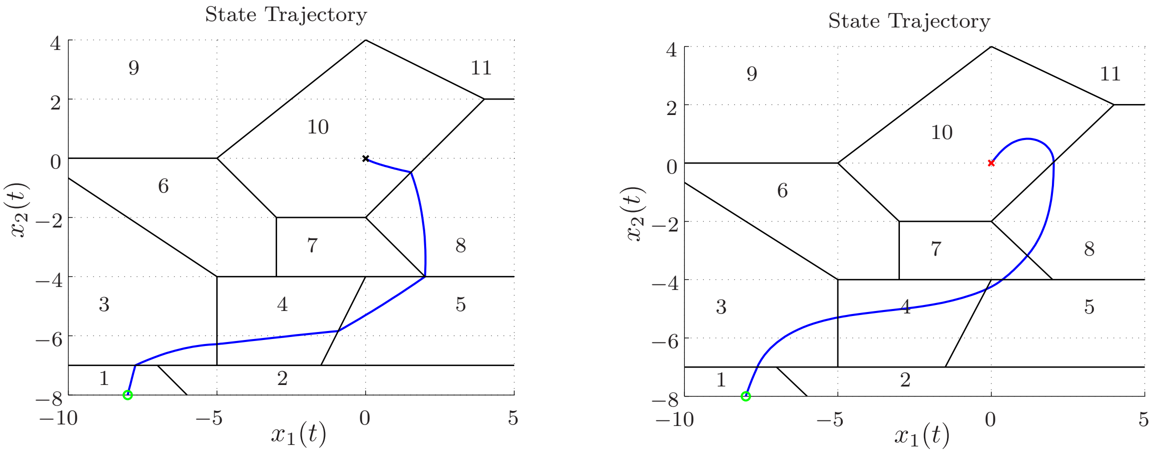
$$\dot{x}_3 = u_q \,. \tag{5.33}$$

Here, x_1 and x_2 are the coordinates of the unicycle, x_3 is its orientation, $u_q \in U_q$ is the steering input, and v_q is the magnitude of the velocity. Each partition $\mathcal{X}_q$ simulates a different surface, which causes the unicycle to drive with a fixed velocity v_q in that partition. The cost function is:

$$\phi_q(x(t), u(t)) = \frac{1}{2}x^T(t)S_q x(t) + T_q x(t) + \frac{1}{2}R_q u_q^2(t)\,. \tag{5.34}$$

Again, the parameters applied for the different dynamics, cost functions, and switching manifolds are presented in Sec. A.1. The algorithm is initialized with the discrete state sequence $\varrho^0 = (1,3,4,6,7,10)$, which is varied over $\varrho = (1,3,4,7,10)$ and $\varrho = (1,3,4,7,8,10)$ to the optimal sequence $\varrho^* = (1,3,4,5,7,8,10)$, see Fig. 5.3.4. The optimal costs are $J = 16.67$ and the computations took 2.7 h. The optimum was verified by a brute-force implementation, which took 348.7 h computational time.

[4] The computations were performed on a 1.6 GHz processor using Matlab 2009a.



(a) Example 1: There is no terminal cost function and no terminal state and discrete state specified, the initial states are $x_0 = (-8,\ -8)^T$ and $q_0 = 1$, and initial and final time are $t_0 = 0$ and $t_e = 2$.

(b) Example 2: The initial conditions are $x(t_0) = \left(-8,\ -8,\ \frac{\pi}{2}\right)^T$, $q_0 = 1$, and $t_0 = 0$. The terminal conditions are set to $x_1(t_e) = x_2(t_e) = 0$ with $x_3(t_e)$ unspecified and $t_e = 2$.

Fig. 5.5: The optimal discrete state sequence and the optimal state trajectory are shown for a linear and a nonlinear numerical example.

Autonomous Vehicle

An autonomous vehicle is modeled as a unicycle [145] with constant velocity and the velocity of a change in the orientation of the unicycle is the control input. The task consists in finding an optimal control that drives the autonomous vehicle around a parked car. The example is presented in detail in Sec. 4.2.

5.3.5 Cooperative Transportation Scenario

Hereafter, the algorithm is applied to a cooperative transportation scenario with multiple robots. The task in the transportation scenario demands several robotic vehicles to cooperatively move a round block from an initial position to a target position. The robots are constrained to dock to the block and once a robot has docked to the block, it is not allowed to separate from the block again. The continuous trajectories of the robots and the block, the docking times, and the sequence in which robots dock to the block are subject to optimal control.

Here, the optimization for the cooperative transportation task is performed centrally with global knowledge. This distinguishes our scenario with cooperative robotic vehicles from typical multi-agent scenarios [166], where computations, control, and collection of information are decentralized and one agent is usually not able to execute a task alone. Cooperative multi-vehicle systems are investigated with a global, optimal control approach in [135] and a good overview about different concepts based on local knowledge is provided in [113]. The special cooperative transportation scenario considered in this section is taken with some modifications from [104, 105]. There, the HOCP is solved by a direct optimal control method with three-layers. The computational complexity is exponential in the number of robots since each possible docking sequence is analyzed separately. The idea here is to show that the novel algorithm for a simultaneous variation of the discrete state sequence and the continuous control is able to solve the cooperative transportation task. Using the novel algorithm, the complexity to find a locally optimal solution is reduced by a factor of exponential complexity.

The combination of the transportation scenario and the proposed optimal control algorithm contains two major difficulties. The first difficulty is that the continuous state space in the transportation scenario is only partitioned in a subspace of the entire state space. This results from the constraint that a robot is not allowed to separate from the block. However, the algorithm needs a partitioned state space. The second difficulty consists in the conflict that the algorithm is implemented for hybrid systems with affine switching manifolds and the transportation scenario contains nonlinear switching manifolds. To handle both difficulties, the optimal control algorithm is modified. First, updates of the switching states are constrained to the partitioned subspace that is formed by the constraint that every robot has to dock to the block. Second, the projection of the gradients onto the switching manifolds is adapted for this special case of nonlinear switching manifolds.

The transportation problem is presented based on the work [205] as follows: At first, the transportation scenario is introduced. Then, necessary adaptations of optimal control algorithm are described. Finally, the simulation results are shown and discussed.

Formulation of the Transportation Scenario

The block and each robot are modeled as an omnidirectional vehicle with velocity-dependent friction:

$$\dot{x}^\nu(t) = A_q^\nu \, x^\nu(t) + B_q^\nu \, u^\nu(t) \,, \tag{5.35}$$

where

$$A_q^\nu = \begin{pmatrix} 0 & 0 & a_q^\nu & 0 \\ 0 & 0 & 0 & a_q^\nu \\ 0 & 0 & -a_q^\nu & 0 \\ 0 & 0 & 0 & -a_q^\nu \end{pmatrix} , \qquad B_q^\nu = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ b_q^\nu & 0 \\ 0 & b_q^\nu \end{pmatrix} . \tag{5.36}$$

Here, x_1^ν and x_2^ν are the position in an x_1-x_2-plane, x_3^ν and x_4^ν are the corresponding velocities, and the controls u_1^ν and u_2^ν are the accelerating and decelerating forces. The index ν is in $\{0, 1, ..., N_\nu\}$ with $N_\nu \in \mathbb{N}$, where $\nu = 0$ corresponds to the block and $\nu \in \{1, ..., N_\nu\}$ to one of the robots. The states and controls of the block and the robots are summarized in the state vectors $x = (x^{0,T}, x^{1,T}, ..., x^{N_\nu,T})^T$ and $u = (u^{0,T}, u^{1,T}, ..., u^{N_\nu,T})^T$. The resulting dynamics is:

$$\dot{x}(t) = A_q \, x(t) + B_q \, u(t) \tag{5.37}$$

with the block matrices

$$A_q = \begin{pmatrix} A_q^0 & 0 & \dots & 0 \\ 0 & A_q^1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A_q^{N_\nu} \end{pmatrix}, \qquad B_q = \begin{pmatrix} B_q^0 & 0 & \dots & 0 \\ 0 & B_q^1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & B_q^{N_\nu} \end{pmatrix}, \tag{5.38}$$

and 0 is of appropriate dimension. The discrete state $q \in \{1, ..., N_q\}$ with $N_q = \sum_{k=0}^{N_\nu} \begin{pmatrix} N_\nu \\ k \end{pmatrix}$ denotes which robots have docked to the block. As long as no robot has docked to the block, the block is uncontrollable, which is expressed by $b_q^0 = 0$. With an increase in the number of robots that have docked to the block, the value b_q^0 usually also increases. The term a_q^0 is always 1. If a robot $\nu \in \{1, ..., N_\nu\}$ has not docked to the block, then a_q^ν and b_q^ν are set to 1. If the robot has docked to the block, then a_q^ν and b_q^ν are 0. This means that when a robot has docked to the block, its dynamics is set to zero, and the robot and the block are considered in the following as one system with associated dynamics. This procedure simplifies the optimization as constraints for coupling the states of the block and the robot are avoided. The state space is not reduced after a docking since the algorithm and the applied HMP require a constant dimension of the state space.

The docking process of robot $\nu \in \{1, ..., N_\nu\}$ to the block is modeled as an autonomous switching. The nonlinear switching manifold is

$$m_{q_{j-1},q_j} = (x_1^\nu(t_j) - x_1^0(t_j))^2 + (x_2^\nu(t_j) - x_2^0(t_j))^2 - 1 = 0 \tag{5.39}$$

for $j \in \{1, ..., N_\nu\}$. The switching manifold has the shape of a circle, that represents the boundary of the block and half the size of a robot. Thus, a robot is modeled as a point with zero diameter.

The HOCP is to find a control trajectory υ and a sequence of discrete states ϱ such that the cost functional

$$J = \sum_{j=0}^{N_\nu} \int_{t_j}^{t_{j+1}} \frac{1}{2} x^T(t) S_{q_j} x(t) + \frac{1}{2} u^T(t) R_{q_j} u(t) \, dt \tag{5.40}$$

is minimized, where the weighting matrices S_{q_j} and R_{q_j} are again in block diagonal form:

$$S_{q_j} = \begin{pmatrix} S_{q_j}^0 & 0 & \dots & 0 \\ 0 & S_{q_j}^1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & S_{q_j}^{N_\nu} \end{pmatrix}, \qquad R_{q_j} = \begin{pmatrix} R_{q_j}^0 & 0 & \dots & 0 \\ 0 & R_{q_j}^1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & R_{q_j}^{N_\nu} \end{pmatrix}. \tag{5.41}$$

The individual weighting matrices

$$S_q^\nu = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & s_q^\nu & 0 \\ 0 & 0 & 0 & s_q^\nu \end{pmatrix}, \qquad R_q^\nu = \begin{pmatrix} r_q^\nu & 0 \\ 0 & r_q^\nu \end{pmatrix} \tag{5.42}$$

are set up analogously to the system matrices, where s_q^0 and r_q^0 are zero, when no robot has docked to the block yet. In the matrix S_q^ν, only the velocity of a robot or the block is penalized not the position.

For the minimization, the system dynamics (5.37), the described, autonomous switching structure, and the switching manifolds (5.39) have to be considered. For advanced simulations in a further evolution of the scenario, there are additional constraints specified at the docking of robot $\nu \in \{1, ..., N_\nu\}$ to the block:

$$m_j = \begin{pmatrix} x_3^\nu(t_j) - x_3^0(t_j) \\ x_4^\nu(t_j) - x_4^0(t_j) \end{pmatrix} = 0 \tag{5.43}$$

for $j \in \{1, ..., N_\nu\}$. For a more realistically physical behavior, the constraints (5.43) demand the velocity and direction of movement to be identical for the robot and the block.

Adaptations of the Optimal Control Algorithm

The optimal control Algorithm 5.2 for variable discrete state sequences cannot be applied directly to the transportation scenario. The first issue is that the state space is not partitioned since a robot is not allowed to separate from the block once it docked to it. Due to the non-partitioned state space, the usual updates of the switching states cannot be performed. Here, it may happen that an update of the switching state, where robot ν docks to the block, leads to a system trajectory, where robot ν and the block do not meet anymore, i.e. no switching occurs in that case. In this case, it may be impossible to obtain neighboring trajectories. However, the used version of the HMP and the algorithm require neighboring trajectories. To be able to still apply the algorithm to the transportation scenario, the updates of the switching states are constrained to a partitioned subspace, where every robot docks to the block. This subspace allows the robots to change the order of their dockings and enables the algorithm to optimize the discrete state sequence as in a partitioned state space. To remain in the partitioned subspace, some adaptations of the algorithm are required.

The second issue is that in the implementation of the algorithm only affine switching manifolds are considered. Here, the switching manifolds are nonlinear and form a circle. To find an updated switching state that is again on the switching manifold, a modified projection is applied.

The third issue is again related to the non-partitioned state space and the sparsely partitioned subspace. Here, it is difficult to detect when a change in the docking sequence of the robots is appropriate based on the update vectors. A specialized decision condition is described that is used in the transportation scenario to decide when the discrete state sequence is changed during the optimization.

Partitioned Subspace and Nonlinear Switching Manifold Here, modifications of the algorithm are introduced to ensure that the update of the switching state x_j^l in iteration l still leads to a docking of robot $\nu \in \{1, ..., N_\nu\}$ to the block. For a simplified notation, the iteration counter l is usually suppressed in the following. The gradient of the costs with respect to the switching state is:

$$\nabla^{\mathrm{m}}_{x_j} J(z) = \Big(\nabla^{\mathrm{m}}_{x_j^0} J(z), ..., \nabla^{\mathrm{m}}_{x_j^\nu} J(z), ..., \nabla^{\mathrm{m}}_{x_j^{N_\nu}} J(z)\Big)\,. \tag{5.44}$$

The update of the switching state for all robots, except for robot ν that docks to the block, is given by:

$$x_j^{\bar{\nu},l+1} = x_j^{\bar{\nu},l} - \alpha \nabla^{\mathrm{m}}_{x_j^{\bar{\nu},l}} J(z^l) \tag{5.45}$$

with $\bar{\nu} \in \{1, ..., N_\nu\}$ and $\bar{\nu} \neq \nu$. This update procedure cannot be applied for robot ν and the block. The reason is that the docking condition (5.39) would not be satisfied, since the gradients of robot ν and the block are in the tangent space of the nonlinear switching manifold. The goal is to find an update for the components of the switching state of the block x_j^0 and robot x_j^ν, such that the autonomous switching between the block and robot ν still occurs after the update and the costs decrease. In our experience, the update direction of the block has a dominating effect on reducing the overall costs. Therefore, it is decided to always calculate the update of the block by

$$x_j^{0,l+1} = x_j^{0,l} - \alpha \nabla^{\mathrm{m}}_{x_j^{0,l}} J(z^l)\,. \tag{5.46}$$

To guarantee the docking after the update, the update of the switching state for robot ν is separated into two parts. First, the switching state $x_j^{\nu,l}$ is updated with the update of the block $-\alpha \nabla^{\mathrm{m}}_{x_j^{0,l}} J(z^l)$, which leads to the intermediate update $\bar{x}_j^{\nu,l+1}$. Second, the intermediate switching state $\bar{x}_j^{\nu,l+1}$ is shifted along the circular boundary of the block in the relative update direction:

$$d_j^{\mathrm{rel}} := -\big(\nabla^{\mathrm{m}}_{x_j^\nu} J(z) - \nabla^{\mathrm{m}}_{x_j^0} J(z)\big)^T\,. \tag{5.47}$$

The projection on the nonlinear boundary of the block is implemented as follows, see also Fig. 5.6:

Algorithm 5.3: Nonlinear Projection

Step 0 **Initialization:** Let the factor $0 < \alpha \leq 1$ be given and determine $\bar{x}_j^{\nu,l+1}$ and d_j^{rel}.

Step 1 **Intersection:** Set up the circle

$$\big(x_{j,1}^{\mathrm{int}} - \bar{x}_{j,1}^{\nu,l+1}\big)^2 + \big(x_{j,2}^{\mathrm{int}} - \bar{x}_{j,2}^{\nu,l+1}\big)^2 - \big(\alpha d_j^{\mathrm{rel}}\big)^2 = 0 \tag{5.48}$$

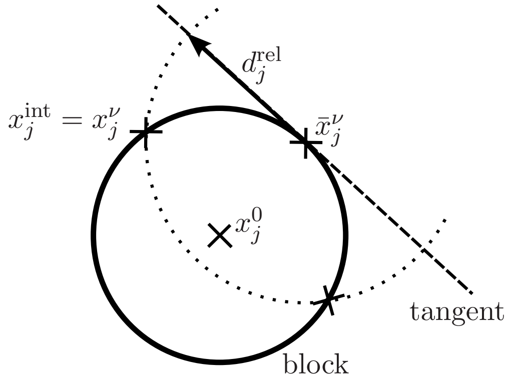


Fig. 5.6: (Compare to [205]) The updated switching state x_j^ν for robot ν is found by a nonlinear projection onto the switching manifold.

around the intermediate switching state $\bar{x}_j^{\nu,l+1}$. Calculate two intersection points x_j^{int} of the circle with the nonlinear switching manifold (5.39).[5]

Step 2 Select new Switching State: The new switching state $x_j^{\nu,l+1}$ is the intersection point x_j^{int} that satisfies:

$$d_j^{\text{rel},T}\left(x_j^{\text{int}} - \bar{x}_j^{\nu,l+1}\right) > 0\,. \tag{5.49}$$

□

To ensure that the updates remain in the partitioned subspace, i.e. an intersection of the circle (5.48) and the switching manifold exists, the factor α is reduced until the intersection occurs. Further, the update step length αd_j^{rel} is always chosen to be less or equal to $\sqrt{2}$, such that the new switching state lies on the same half of the circular boundary of the block, which has radius 1. This simplifies the numerical convergence, since the physical direction of the docking process ("robot ν docks to the block" vs. "the block docks to robot ν") does not change.

If the relative update direction d_j^{rel} would be applied without the nonlinear projection, the update of the switching state x_j^ν would be the same as for all other robots. To cope with the nonlinear switching manifold, the relative update direction d_j^{rel} is changed. Thus it cannot be ensured anymore that the costs decrease with the applied update. However, the change in the update direction is usually very small and the update direction of the block is usually the dominating one for lowering the costs. In our experience, feasible updates always existed that reduced the overall costs with the chosen update direction, when the step size was sufficiently small.

The optimal control subproblem from one switching point to the next is solved by indirect multiple shooting as described in Sec. 5.3. The underlying integration of the system dynamics is stopped, when the continuous state trajectory meets the nonlinear switching manifold. Here, it can happen during the optimization that the switching manifold is not met. In this case, the integration is continued until the final time t_e. As this causes

[5] The first intersection point is found numerically, e.g. using the Matlab function *fsolve*, and the second one is derived analytically afterwards.

problems for the solver used in the indirect multiple shooting, an additional stopping condition is introduced. The condition consists in an additional, affine switching manifold

$$m_j^{\mathrm{aff}}(x^\nu(t)) = \frac{1}{\|(x_{j,3}^\nu \; x_{j,4}^\nu)^T\|} \Big(x_{j,3}^\nu \left(x_1^\nu(t) - x_{j,1}^0 \right) + x_{j,4}^\nu \left(x_2^\nu(t) - x_{j,2}^0 \right) \Big) = 0 \,, \tag{5.50}$$

which is perpendicular to the velocity $(x_{j,3}^\nu \; x_{j,4}^\nu)^T$ of robot ν and passes through the center $(x_{j,1}^0 \; x_{j,2}^0)^T$ of the block. The integration stops anywhere on the union of the affine and the nonlinear switching manifold (5.39).

Changes in the Docking Sequence In a partitioned state space, a change in the sequence of discrete states occurs whenever the update of a switching state lies on another switching manifold compared to the previous iteration. In the cooperative transportation scenario, the state space is not partitioned and only sparsely partitioned on the partitioned subspace. Hence, there exist cases, where the update of a switching state does not lie on another switching manifold, but a change in the docking sequence of the robots would lead to lower costs based on the updated switching state. To accelerate and stabilize the numerical convergence of the algorithm, a procedure is developed that detects when changes in the discrete state sequence possibly lead to lower costs and that specifies how updated switching states are determined.

In the following, the procedure is explained based on the exemplary situation illustrated in Fig. 5.7. There, the docking sequence of the robots is $(1, 2, 3)$ in iteration l. In the figure, it is shown that robot 2 docks to the block second and the switching states are $x_2^{0,l}$ for the block, $x_2^{2,l}$ for robot 2, and $x_2^{3,l}$ for robot 3. Robot 3 docks third with the switching states $x_3^{0,l}$, $x_3^{2,l}$, and $x_3^{3,l}$. Now, switching state x_3 is to be updated. With the usual update of the block, the candidate switching state

$$\bar{x}_3^{0,l+1} = x_3^{0,l} - \alpha \nabla^{\mathrm{m}}_{x_3^{0,l}} J(z^l) \tag{5.51}$$

is obtained and similarly the candidate $\bar{x}_3^{3,l+1}$ for robot 3. The candidate $\bar{x}_3^{0,l+1}$ of the block lies in the opposite half-space defined by the affine manifold M_{sw} than the previous switching state $x_3^{0,l}$. One point on the manifold M_{sw} is $x_2^{0,l}$ and the manifold is perpendicular to line a, which is the straight connection of the switching states $x_2^{0,l}$ and $x_3^{0,l}$. Since manifold M_{sw} is crossed, it is decided to change the discrete state sequence, such that robot 3 docks before robot 2. Instead of $\bar{x}_3^{0,l+1}$, the intersection point $x_2^{0,l+1}$ of the block's trajectory and a circle with radius $\alpha \nabla^{\mathrm{m}}_{x_3^{0,l}} J(z^l)$ around the previous switching state $x_3^{0,l}$ is used as new switching state. This increases the chances that the new switching states form feasible optimal control subproblems. The switching state $x_2^{3,l+1}$ of robot 3 is found analogously. Since robot 2 docks after robot 3 now, an additional switching state $x_2^{2,l+1}$ is inserted in between the switching states $x_1^{2,l}$ and $x_2^{2,l}$ on the trajectory of robot 2. The switching state $x_2^{2,l+1}$ is inserted at the time, when the block's trajectory meets the new switching state $x_2^{0,l+1}$. The previous switching state $x_3^{3,l}$ is not considered further. If the update does not lead to a decrease in the costs, the update is repeated with reduced α.

Note that it cannot be guaranteed with this update procedure for changes in the docking sequence that the costs decrease with an update. The reason is that the update direction

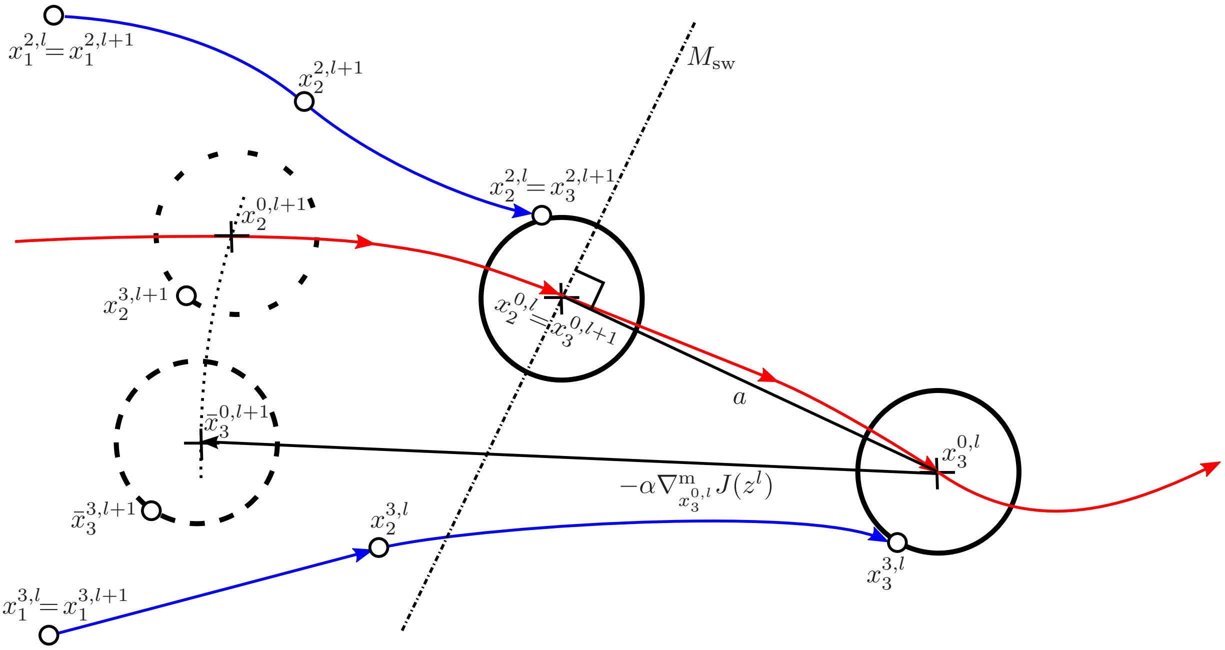

$x_1^{2,l}\!=\!x_1^{2,l+1}$
$x_2^{2,l+1}$
M_{sw}
$x_2^{2,l}\!=\!x_3^{2,l+1}$
$x_2^{0,l+1}$
$x_2^{3,l+1}$
$x_2^{0,l}\!=\!x_3^{0,l+1}$
a
$\bar{x}_3^{0,l+1}$
$x_3^{0,l}$
$-\alpha\nabla^{\mathrm{m}}_{x_3^{0,l}}J(z^l)$
$\bar{x}_3^{3,l+1}$
$x_2^{3,l}$
$x_3^{3,l}$
$x_1^{3,l}\!=\!x_1^{3,l+1}$

Fig. 5.7: (Compare to [205]) If during the update of the switching state $x_3^{0,l}$ the affine manifold b is crossed by the update vector, the docking sequence is changed, such that robot 2 docks before robot 3 in iteration l and robot 3 before robot 2 in iteration $l+1$.

$-\nabla^{\mathrm{m}}_{x_3^{0,l}}J(z^l)$ is modified by the the circular projection onto the block's trajectory. However, in our experience the modifications of the update direction are small, such that a descent direction for the costs is usually still achieved.

Simulation Results

In the first simulation example, 3 robots have to dock to the block and transport it cooperatively to the origin with $t_0 = 0$ and $t_e = 15$sec. During the docking process, the additional conditions (5.43) do not have to be considered. After the initialization, the costs are $J = 81.5$ and the docking sequence of the robots is $(1, 2, 3)$, see Fig. 5.8. Here, the trajectories from one switching point to the next are optimal, but the connection of the optimal trajectory parts is still suboptimal. The modified algorithm finds the optimal costs $J = 41.3$ and the optimal docking sequence to be $(1, 3, 2)$, which means that the discrete state sequence changed during the optimization, see Fig. 5.9. Further, it can be observed that the trajectories of the robots and the block are smoother than after the initialization, especially the trajectory of robot 3.

In a second example, 4 robots have to steer the block to the origin from $t_0 = 0$ to $t_e = 18$sec. This time, the robots have to satisfy the additional constraints (5.43), when they dock to the block. The additional constraints demand that the velocity and the direction of movement of the docking robot equals the ones of the block at the switching time. To find optimal solutions with the additional conditions, it is required to modify the adjoint transversality condition (3.62) by adding the term $\nabla_{x_j^l} m_j^T(x_j^l(t_j))\,\pi_j$ with $\pi_j \in \mathbb{R}^2$. This can be derived e.g. with classical variational techniques [32]. The trajectories after

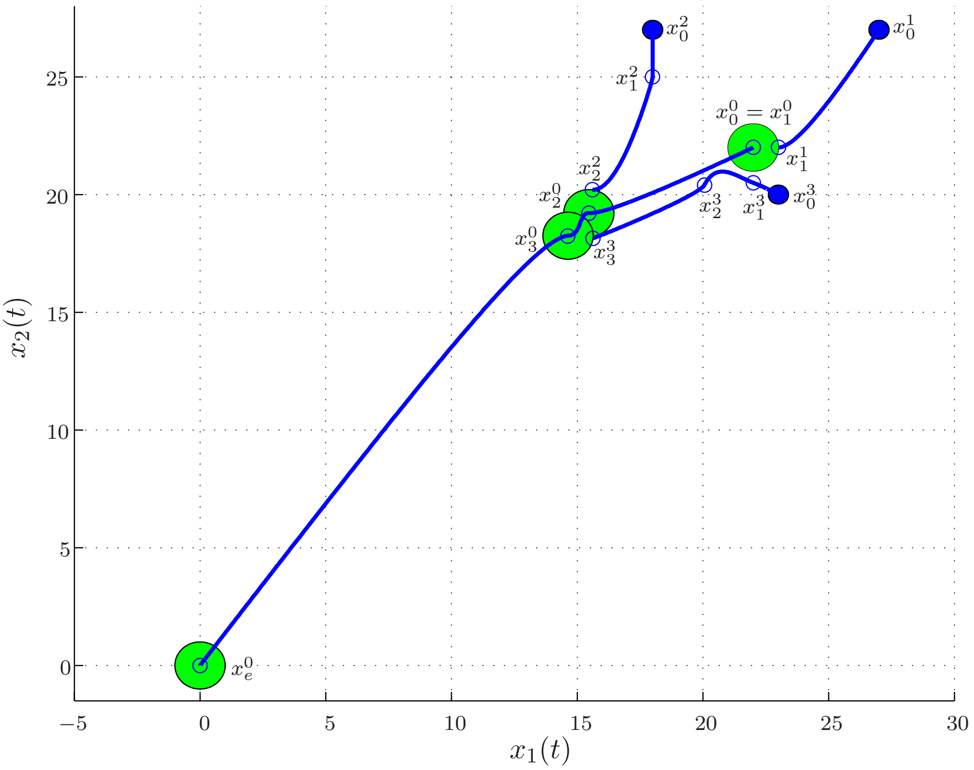


Fig. 5.8: ([205]) Trajectories and switching states of the block and 3 robots after the initialization. The robots dock to the block in the sequence $(1, 2, 3)$.

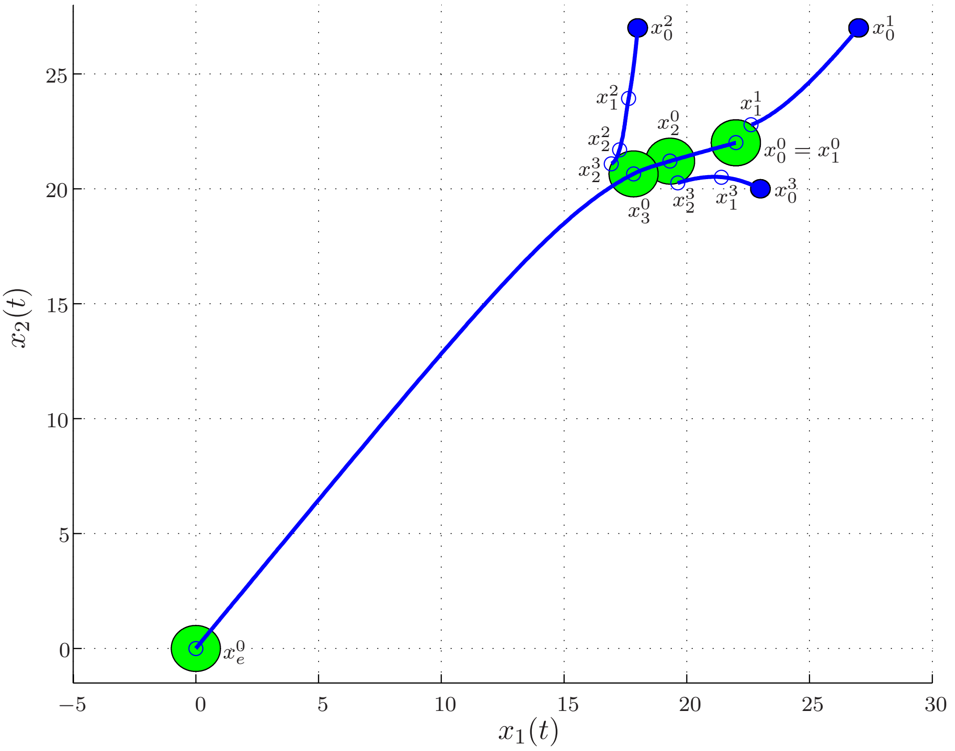


Fig. 5.9: ([205]) Trajectories and switching states of the block and 3 robots after the optimization. The docking sequence changed to $(1, 3, 2)$.

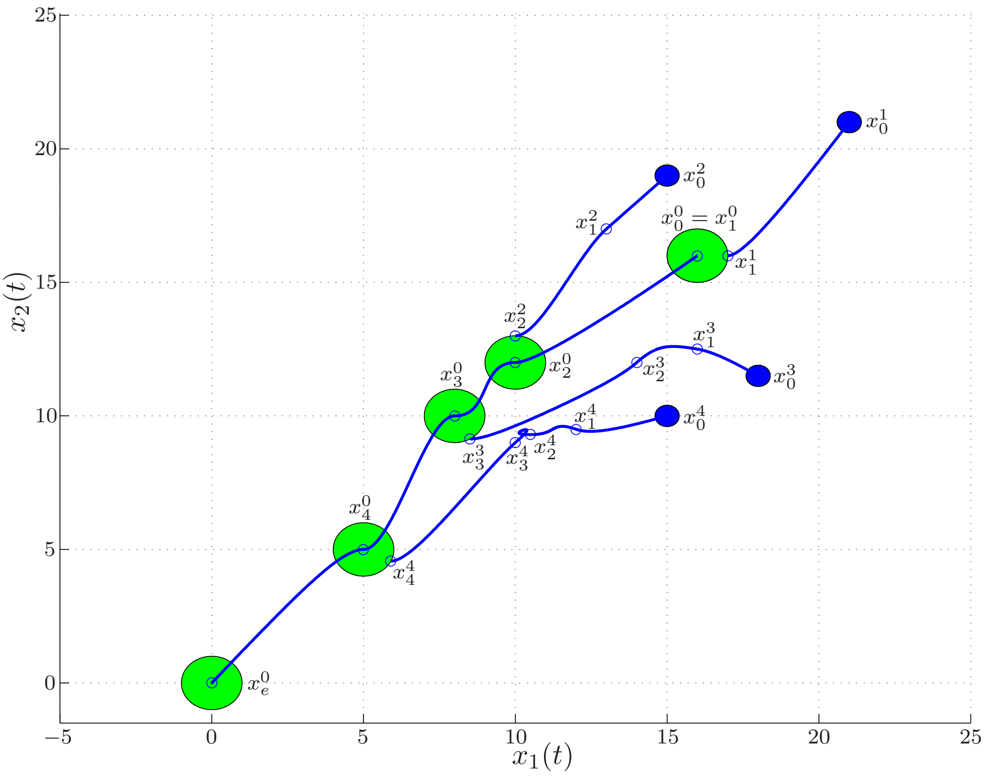


Fig. 5.10: ([205]) Trajectories and switching states of the block and 4 robots after the initialization. The robots dock to the block in the sequence $(1, 2, 3, 4)$.

the initialization are illustrated in Fig. 5.10. The initial costs are $J = 172.9$ and the docking sequence is $(1, 2, 3, 4)$. In between the second switching state x_2^4 and the third switching state x_3^4 of robot 4, the optimal sub-trajectory forms a loop, which clearly shows that the initialization is far from the overall optimal solution. After the optimization, the costs are $J = 131.8$ and the docking sequence remained $(1, 2, 3, 4)$, see Fig. 5.11. The overall trajectories are smoothed compared to the initialization and no loops are present anymore. However, the overall optimal solution has not yet been obtained entirely, which is verified by some nonzero gradients of the costs at the switching states. The parameters and the initialization of the two simulation examples are provided in Sec. A.2.

Discussion of Results

In our experience, the updates of the switching points with the gradient descent approach are reliable. This statement is based on the observation that the updates usually lead to lower costs whenever a solution is found in the underlying optimization. The limiting component in the algorithm is the indirect multiple shooting method used in the underlying optimization for the solution of the optimal control subproblems. For the cooperative transportation scenario, its domain of convergence showed to be small. That means that only small step sizes are possible for the updates. Sometimes the shooting method does not converge to the new switching point, though the updated switching point is reachable from the previous docking point.

In the example with 3 robots, the convergence to the optimal solution is still achieved,

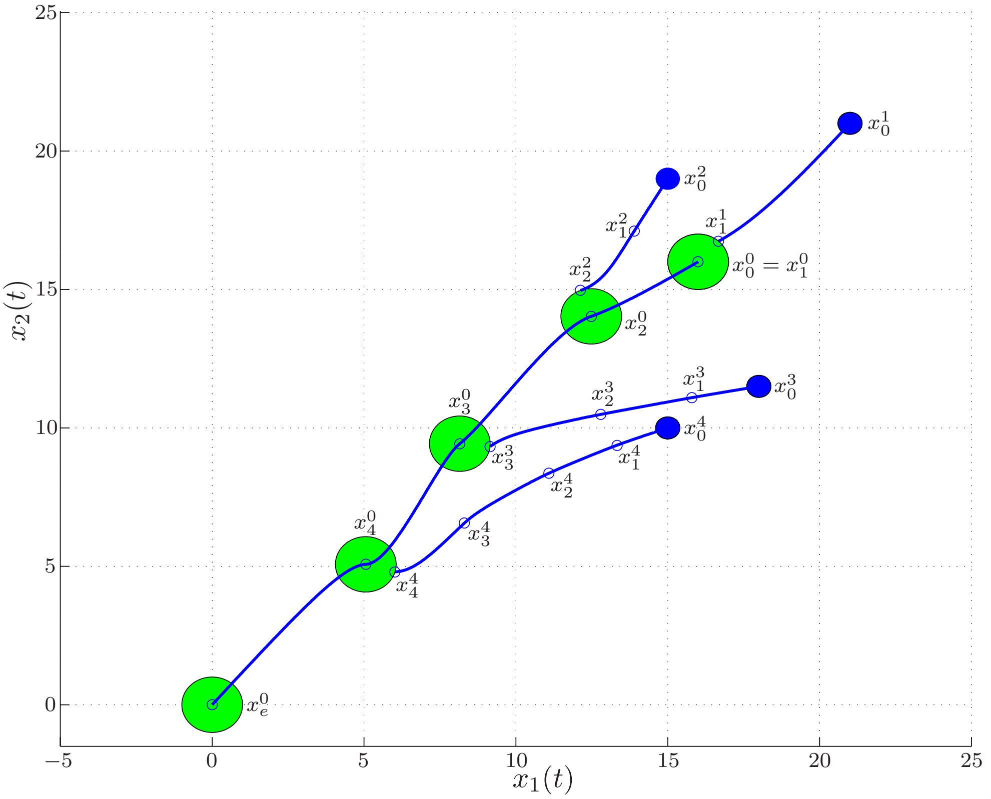


Fig. 5.11: ([205]) Trajectories and switching states of the block and 4 robots after the optimization. The robots still dock to the block in the sequence $(1, 2, 3, 4)$.

though the initialization is difficult and only small step sizes are successfully executed. With 4 robots and the additional constraints for the direction of movement and the velocity of the robots at the docking time, the algorithm approaches the optimal solution but does not converge entirely to the optimal solution. Here, the feasible region of the updates is smaller compared to the example with 3 robots and the indirect multiple shooting often does not find a solution.

In the implementation of the algorithm, it is considered that the underlying multiple shooting method does not converge to the desired switching state entirely. Nevertheless, the solution is accepted if it leads to lower costs and if the found switching state lies on the specified switching manifold. Otherwise, the step size is reduced and the optimization is repeated. With the additional constraints in the example with 4 robots, the error tolerance of the algorithm decreases, since the constraints must be met precisely (tolerance: 10^{-4}). The reduced error tolerance is one of the main reasons for the worse convergence compared to the example with 3 robots. Further, the convergence of the shooting method is supported by dividing the optimal control subproblems into subsubproblems by optimizing the trajectory of each robot to the updated switching point separately.

In future, the underlying shooting method shall be improved by initializing it for every updated switching point with the second initialization concept presented in Sec. 4.2. Even more promising, it is to replace the indirect multiple shooting method by the min-H algorithm introduced in Sec. 4.3 for the solution of the optimal control subproblems.

5.3.6 Discussion

For hybrid systems with partitioned state space, an algorithm is introduced that varies the discrete state sequence while searching for optimal hybrid trajectories. The algorithm uses an extended version of the HMP presented in Sec. 3.3 that accounts for corners and intersections of switching manifolds. Consequently, gradients of the cost can be calculated at those corners or intersections and the gradient information is exploited in the algorithm. Therefore, the algorithm avoids the problem of former algorithms based on the HMP that the computational complexity increases combinatorially with the number of switchings. In the current formulation, the algorithm finds locally optimal controls and discrete state sequences. In convex HOCPs, the global optimal solution can be found in a single run of the algorithm. In general for a global optimization, the complexity advantage of the algorithm based on the extended HMP compared to former algorithms depends on the number and distribution of locally optimal solutions in the given HOCP. To find a globally optimal solution, it is necessary to use methods from global optimization e.g. a smart distribution of different initializations such that the entire state space is analyzed for optimal solutions.

The gradient of the costs, that is applied for shifting a switching state, is provided by the adjoint transversality condition (3.62) for autonomous switching. It contains several normals of switching manifolds if the continuous state trajectory meets an intersection of switching manifolds. The associated Lagrange multipliers π are calculated numerically by a linear regression in this case. To obtain unique results, only a subset of linearly independent normals is used in the algorithm from the entire set of possibly linearly dependent normals that appear in the adjoint transversality condition.

If the continuous state trajectory passes one switching manifold at an autonomous switching (no intersection), two options are available in the implementation to determine the Lagrange multiplier π. Again a linear regression can be used, which implies that the gradients of the costs with respect to the switching state are tangential to the switching manifold. This implementation provides robust convergence properties. In the second option, the multiplier π is calculated from a rearrangement of terms in the Hamiltonian value condition resulting in (3.110). The advantage is that the Hamiltonian value condition is already satisfied here. A disadvantage is that an additional projection of the resulting gradients to the tangent space of the switching manifold is required. Further, to achieve convergence is slightly more difficult than in the first option since the Hamiltonians are very sensitive to small changes in the adjoint variables during the optimization process.

On the lower layer of Algorithm 5.2, two-point BVPs are solved with indirect multiple shooting. Some aspects of the applied indirect multiple shooting method are discussed in the following: The state and adjoint dynamics (2.2) and (3.60) are integrated until the state trajectory hits a switching manifold instead of stopping the integration at the switching time t_{j+1}^l. This implies that all switching times t_j^l for $j \in \{1, ..., N(l)\}$ are determined indirectly by the choice of x_j^l and not anymore directly by Armijo update steps of t_j^l. It is crucial for the success of the algorithm that the target switching state x_{j+1}^l is reachable from the initial switching state x_j^l. Leaving the switching time t_{j+1}^l unspecified during the forward integration stabilizes the numerical optimization as the chances to meet x_{j+1}^l are increased. Further, the Armijo update steps for the switching times t_j^l were very sensitive in our experience in contrast to the updates of the switching states x_j^l. This numerically destabilized the optimization and lead to sincere reachability problems, such that the updates of the switching times t_j^l were finally taken out. If the switching state x_{j+1}^l is not reachable from x_j^l, then the Armijo step is repeated with a reduced step size until a feasible switching state combination is found.

The stopping condition for the integration implies that each optimal control subproblem is one with free terminal time t_{j+1}^l for $j \in \{1, ..., N(l)\}$. Further, an optimal solution of the entire HOCP requires that the Hamiltonians just before and just after the autonomous switching are equal, that is (3.63)

$$H_{q_j^l}\big(x(t_{j+1}^{l-}), \lambda(t_{j+1}^{l-}), u_{q_j^l}(t_{j+1}^{l-})\big) = H_{q_{j+1}^l}\big(x(t_{j+1}^{l}), \lambda(t_{j+1}^{l}), u_{q_{j+1}^l}(t_{j+1}^{l})\big)\,,$$

if the continuous state trajectory avoids intersections of switching manifolds. The optimal solutions of the subproblems have to consider the Hamiltonian continuity condition in this case. To be able to solve the optimal control subproblems separately, the following observation is important: The optimal Hamiltonian is constant for almost every time for the considered class of hybrid systems, if the optimal trajectory does not pass through intersections of switching manifolds. This can be verified by analyzing its non-differentiable points and its derivative:

$$\frac{d}{dt}H_{q_j^*} = \nabla_t H_{q_j^*} + \nabla_x H_{q_j^*}\dot{x}^*(t) + \nabla_\lambda H_{q_j^*}\dot{\lambda}^*(t) + \nabla_u H_{q_j^*}\dot{u}_{q_j^*}^*(t) = 0\,.$$

The first term $\nabla_t H_{q_j^*}$ is zero since the Hamiltonian does not explicitly depend on time. The two terms $\nabla_x H_{q_j^*}\dot{x}^*(t)$ and $\nabla_\lambda H_{q_j^*}\dot{\lambda}^*(t)$ cancel each other, compare (3.60) and (2.9). In the last term, either the components of $\nabla_u H_{q_j^*}$ are zero when the corresponding components of the optimal control $u_{q_j^*}^*(t)$ are within their bounds in $U_{q_j^*}$ or the components of $\dot{u}_{q_j^*}^*(t)$ are zero when the corresponding components of $u_{q_j^*}^*(t)$ are on the boundary $\partial U_{q_j^*}$. At jumps of the optimal control and at autonomous switchings, the Hamiltonian is continuous, compare (3.63), (3.64), (2.9), and Assumption 2.31.5.

If the final time t_e of the entire HOCP is unspecified, there are no constraints on the terminal times t_{j+1}^l of the different subproblems. To find optimal terminal times t_{j+1}^l, the constant Hamiltonian values have to be zero at the terminal times: $H_{q_j^l}(x(t_{j+1}^{l-}), \lambda(t_{j+1}^{l-}), u_{q_j^l}(t_{j+1}^{l-})) = 0$ for $j \in \{1, ..., N(l)+1\}$. This condition is added to the indirect multiple shooting algorithm.

If the final time t_e is specified, then the following problem arises: On the one hand, the optimal control subproblems have to be solved one after the other starting with the first subproblem from t_0. Otherwise, it remains unknown how much time can be spent in the last subproblem. On the other hand, the optimal control subproblems need to be solved in backward order to determine the unknown Hamiltonians $H_{q_{j+1}^l}(x(t_{j+1}^l), \lambda(t_{j+1}^l), u_{q_{j+1}^l}(t_{j+1}^l))$ just after an autonomous switching. Currently, two alternatives to solve the conflict are implemented. The first alternative is suitable for HOCPs, where the optimal, terminal Hamiltonian $H_{q_e^*}(x^*(t_e), \lambda^*(t_e), u_{q_e^*}^*(t_e))$ is close to zero. In this case, the implementation demands the Hamiltonians $H_{q_j^l}(x(t_{j+1}^{l-}), \lambda(t_{j+1}^{l-}), u_{q_j^l}(t_{j+1}^{l-}))$ at the switching times t_{j+1}^l to be zero, since the Hamiltonian function is constant in $[t_0, t_e]$. In the second alternative, the Hamiltonians $H_{q_{j+1}^{l-1}}(x(t_{j+1}^{l-1}), \lambda(t_{j+1}^{l-1}), u_{q_{j+1}^{l-1}}(t_{j+1}^{l-1}))$ just after an autonomous switching from the previous iteration $l-1$ are stored and used as target values for the Hamiltonians $H_{q_j^l}(x(t_{j+1}^{l-}), \lambda(t_{j+1}^{l-}), u_{q_j^l}(t_{j+1}^{l-}))$ in the current iteration l. This alternative can provide more accurate results than the first alternative for HOCPs with specified terminal time t_e. However, the convergence is more delicate. A suggestion for future work is to extend the first alternative such that the Hamiltonians do not converge to zero but to an additional optimization variable. The optimization variable itself slowly approaches the optimal, constant Hamiltonian value in the optimization with indirect multiple shooting.

Overall, the gradient descent based optimization of the switching points and the discrete state sequence works reliable. If the current solution is not in the neighborhood of the optimal solution, the upper layer of the algorithm converges fast towards the solution. Here, the step size is usually constrained by the convergence properties of the lower layer. In the neighborhood of an optimal solution, the convergence rate of the gradient based optimization is low. In this phase of the optimization, it seems to be advantageous to switch to a Newton based update scheme for the switching points. The integration of a Newton method, which converges at least quadratically near the optimum, is a task for future work.

For low dimensional, slightly nonlinear optimal control subproblems, the indirect multiple shooting method is reliable. For more complex subproblems like the cooperative transportation scenario, the convergence of the indirect multiple shooting is difficult to achieve and sometimes only possible for very small step sizes or not at all. This restrains the convergence of the entire algorithm to a locally optimal solution. Future work consists in embedding the numerically more stable min-H method from Sec. 4.3 on the lower layer of the algorithm. The possibilities of the algorithm to change the switching points and the discrete state sequence largely depend on the tightness of constraints on the control. Tight constraints cause a numerically stable behavior of the multiple shooting method, but slow down the convergence of switching points to an optimal solution since updates are only feasible for small step sizes. In contrast, loose constraints allow for large step sizes, but the indirect multiple shooting method is less reliable.

The algorithm is developed for indirect methods. However, it is also possible to embed direct solution methods on the lower layer instead of the indirect multiple shooting. Using a direct method, it is required to extract the adjoints from the direct solution with high accuracy e.g. with approaches like in [66, 154, 175]. This is important to obtain the gradient information for correct update directions of the switching points.

5.4 Summary

This chapter presents two approaches that find an optimal control and an optimal discrete state sequence for HOCPs with autonomous switching. The complexity of the two novel indirect methods is lower than the combinatorial complexity of previous hybrid optimal control algorithms, that belong to indirect and direct methods. The complexity of DP is linear in the number of discrete states and thus in general still lower than the complexity of the proposed algorithms. Nevertheless, DP is only applicable to HOCPs with a low state dimension.

The first approach encodes the autonomous switching structure in a tree, relaxes autonomous switching to controlled switching, and uses branch and bound principles for an efficient optimization with indirect multiple shooting. The second approach considers HOCPs with partitioned continuous state space and the corresponding optimality conditions derived in Sec. 3.3. The optimality conditions provide among others gradients of the costs with respect to the switching points. The gradients are used on the upper layer of the optimal control algorithm to optimize the switching points and to find the optimal discrete state sequence. On the lower layer, purely continuous optimal control subproblems are solved by indirect multiple shooting. The second approach reduces the computational complexity to find a locally optimal solution by a factor of combinatorial complexity compared to previous approaches. The efficiency of the two proposed approaches is demonstrated in several numerical examples. Additionally, it is also shown that the second approach can even be applied with some modifications to a cooperative transportation task, which is only partitioned on a subspace of the complete state space.

In the following chapter, indirect optimal control algorithms are proposed that have a high potential for an online application because of efficient computations and embedded feedback control laws.

6 Optimal Feedback Control for Hybrid Systems

This chapter introduces two methods that have a high potential for online optimization by exploiting the structure of hybrid optimal control problems (HOCPs). Additionally, the methods provide feedback control laws for an increased robustness against model uncertainties and disturbances. The first method is a model predictive control (MPC) algorithm for nonlinear and hybrid systems with control and state constraints. It provides fast updates based on neighboring extremals. Neighboring extremals approximate the nonlinear HOCP with a linear-quadratic one in the neighborhood of a given, not necessarily optimal trajectory. The linear-quadratic HOCP is solved by a second-order gradient method, such that a locally optimal solution is found. The solution is used for an optimal, local, feedback-feedforward controller, which is applied while the next linear-quadratic HOCP for the receding horizon is solved. The local controller supports the gradient method by also approaching unfulfilled optimality conditions. The convergence to a locally optimal solution is proven under local convexity assumptions. The algorithm can also handle changes in the hybrid switching and state constraint structure under certain assumptions, when the changes occur at the beginning or end of the moving horizon. The effectiveness of the algorithm is demonstrated for an overtaking scenario of an autonomous car.

The second method uses optimality conditions and existing solution methods of linear-quadratic HOCPs to find solutions by parameterized, optimal control primitives and an optimal concatenation of primitives. Major parts of the computation are performed offline, such that online computations are limited to the solution of a system of algebraic equations to find the optimal parameters. (Sub)optimal feedback control is applied either by a recomputation of the optimal parameters within an MPC framework or by computing locally valid, (sub)optimal feedback control laws. The considered HOCPs include autonomous switching and interior point constraints. Nonlinear HOCPs are also considered if they can be reformulated as linear-quadratic HOCPs by input-to-state or input-to-output feedback linearization. The primitive-based optimization is applied to robotic manipulators using feedback linearization. The approach is combined with a heuristic, kinematic motion planner based on rapidly-exploring random trees (RRTs), such that collision-free, suboptimal trajectories are determined.

6.1 Introduction and State of the Art

Indirect methods for the online solution of HOCPs are either designed for specialized cases or do not exist at all. However, indirect methods are promising for online application due to the structural information about an HOCP that is provided by optimality conditions. Here, two approaches are introduced, which show a high potential for online optimization and which are derived from optimality conditions in order to achieve fast computations.

When controlling a technical system in an environment, where models and parameters are uncertain and events are difficult to predict, it is desirable to adapt the control strategy during runtime whenever new information is available. A classical form of closing the control loop is feedback control based on new measurements of the state. When predicted events change or unforeseen events occur, the control strategy usually needs to be replanned. A major difficulty for adapting the control strategy during runtime is to finish computations in real-time, especially if optimal feedback control and optimal replanning is considered. Here, two optimal control approaches are proposed that provide (sub)optimal feedback control and optimal replanning. The two approaches exploit structural information of a given HOCP for efficient computations, such that depending on the computing power and the complexity of the HOCP, an online solution may be achieved. The first approach is an MPC algorithm based on neighboring extremals. This approach includes replanning at every sampling time and two forms of continuous state feedback. The second approach optimally concatenates optimal control primitives, which allows for fast computations and an application in an MPC framework. The primitive-based approach is motivated by recent research in robotics, where primitives are used to achieve online (re)planning capabilities.

In this chapter, the contributions are the following:

(i) The proposed MPC approach based on neighboring extremals and for the solution of HOCPs is novel. Unlike existing approaches for subproblems of the HOCPs considered here, the presented MPC approach can handle changes in the sequence of hybrid switchings and state constraints under certain assumptions.

(ii) A former gradient method of second order is extended here to hybrid systems with autonomous switching, state constraints, and parameter updates and the convergence is proven.

(iii) Optimal feedback control laws are updated at each sampling point and they further improve the control in case of suboptimal trajectories besides optimally correcting deviations from the current (sub)optimal solution in contrast to former algorithms.

(iv) In the second approach, existing optimal solutions based on primitives are extended to HOCPs by optimally concatenating several optimal control primitives. The concatenation is such that only algebraic computations have to be executed online.

(v) The primitive-based optimal control approach is combined with a sampling-based motion planner to find collision-free trajectories for a robotic manipulator.

Hereafter, the state of the art in MPC and primitive-based control is reviewed and the two novel optimal control approaches are described shortly.

Model Predictive Control Approach MPC is attractive as it can be used intuitively and it can be applied to nonlinear, hybrid, and state and control constrained systems [41]. In MPC an optimal control problem is solved over a prediction horizon and the optimization is repeated for a moved horizon after applying the first part of the optimal control. The shorter the time between two optimization runs is, the faster is the response to disturbances and thus the higher is the robustness. A major limitation of the MPC algorithms for nonlinear and hybrid systems is that the online optimization often takes long, so that the desired robustness is not achieved or even no online solution is possible. Mainly three types of approaches have been pursued to overcome this limitation.

The first type, called *explicit MPC*, offline precomputes optimal feedback control laws, which are piecewise affine (PWA) for linear, constrained systems [17], such that only the appropriate feedback control laws need to be selected and evaluated online. The approach has been extended to hybrid systems consisting of PWA systems, see [2, 43] for overviews, and nonlinear systems [77]. A major drawback of explicit MPC is that the complexity increases exponentially with the state and control dimension and the prediction horizon. The complexity is partly reduced by using value functions for a simplified selection process of the feedback control laws [11] and by approximating the explicit MPC and combining it with some online optimization steps [191].

In contrast to explicit MPC, the two other types of approaches are also suitable for large state dimensions, strongly nonlinear systems, long prediction horizons, and changing parameter estimates during the execution of the optimal control. However, these approaches only achieve real-time capabilities for optimal control problems with moderate complexity. The second type exploits structural properties of MPC like the sparsity of Jacobian matrices to accelerate the online optimization with e.g. interior-point methods. Additionally, warm start techniques are used and the optimization is terminated early. Examples for linear systems are given in [177], for nonlinear systems in [49], and for hybrid systems in [88], where a mixed-integer optimal control problem is solved.

The third type is based on sensitivity updates or neighboring extremals. They utilize a linearization of the optimal control problem around the current solution to quickly approximate the solution for the next sample period. An optimal feedback control law is approximated in each time step of an MPC algorithm with only one optimization step based on a linearization around an approximately optimal solution [48]. This real-time method is proven to be convergent to the optimal solution and is set up for problems with fixed time horizon. A linear feedback based on sensitivities is added to the control for receding horizon problems [189]. The method relies on the assumption that an optimal nominal solution is calculated beforehand in each time step. Starting with sensitivity updates in [79], the method is extended in [184] to neighboring extremals for MPC with shrinking horizon. The neighboring extremals are iteratively used to find an optimal solution for varied initial states in the neighborhood of an optimal solution. All methods mentioned up to this point rely on direct optimization methods.

Based on indirect approaches, optimal feedback control laws are derived with neighboring extremals based on optimality conditions of second order around an optimal solution [60, 183]. Additionally, an optimal solution around a nominal optimal solution for a variation in the initial states is approximated with one iteration of a gradient method for purely continuous optimal control problems with state and control constraints [60]. An optimal solution is calculated offline with a fixed time horizon in [126, 127]. Online, an optimal feedback control law based on neighboring extremals is applied around the calculated optimal trajectory. The offline optimization is performed by solving the multi-point boundary-value problem (MPBVP) resulting from a linearization of the optimality conditions with a multiple shooting method. Only in [126, 183], the considered system class includes hybrid systems with autonomous switching. All approaches that rely on the second variation assume that during runtime no change occurs in the sequence of active state constraints and, if considered, discrete states.

Following the third type of approaches, a two-layered MPC algorithm is introduced based on neighboring extremals derived from a linearization of optimality conditions. The algorithm is a step towards the online solution of complex, nonlinear HOCPs with autonomous switching, a fixed sequence of discrete states, estimates of model parameters that are updated during runtime, and state and control constraints. The upper layer of the MPC algorithm contains an iterative gradient method of second order to determine a locally optimal solution. The gradient method in [32] is extended here to cope with parameter updates, hybrid systems, and state and control constraints. The novel gradient method uses the optimality conditions and the system of Riccati equations derived in [183] with small modifications like including updates of the model parameters from an estimator, e.g. a Kalman filter, at each sample time. The convergence of the gradient method is proven. On the lower layer, an optimal feedback-feedforward controller is applied. It optimally corrects deviations from the current (sub-)optimal trajectory and additionally reduces the deviation of the current, possibly suboptimal solution from an optimal solution. Under certain conditions, the MPC algorithm can also handle changes in the sequence of discrete states, state-constrained and unconstrained phases, when the changes occur at the beginning or the end of the receding horizon. The contributions are the discussed extensions of the gradient method, the proof of convergence, the updated, locally valid, optimal feedback control laws, and the MPC framework with changes in the sequence of discrete states and constraints. We introduced the MPC algorithm in [206, 213].

Approach based on Optimal Control Primitives Trajectory planning is one of the fundamental issues in robotics. The planning process is often carried out in high dimensional configuration spaces and a resulting trajectory has to minimize a cost criterion and avoid collisions with obstacles. Especially due to unforeseen events in the workspace of the robot like not predicted movements of obstacles, a replanning of the state trajectory during runtime has to be possible in real-time.

Optimal control algorithms [176] find optimal and collision-free trajectories, but are in general not able to finish their computations in real-time, which turns them unsuitable for online replanning. Similarly, randomized kinodynamic motion planners [95], which sample the control space based on RRTs and integrate the system dynamics, find (sub)optimal and collision-free solutions. However, they are often too slow and are usually not able to cope with a control space dimension greater than 4 due to an exponential increase in the number of samples with the dimension of the control space. Other sampling-based motion planning approaches, e.g. kinematic planners based on RRTs [91] or probabilistic road maps (PRMs) [81], are able to achieve real-time computations for example systems. In [80], sampling-based planners are improved such that the convergence to an optimal solution is guaranteed. A disadvantage of these sampling-based kinematic planners is that the number of samples still grows exponentially with the state space dimension. Further, the system dynamics are not considered, such that it cannot be ensured that the found trajectories are feasible. Approaches like potential field methods [87] are able to plan in real-time, but in general optimal trajectories are not obtained. A good overview on different motion planning concepts is given in [94].

To avoid the mentioned disadvantages, trajectory planning based on control primitives has been investigated recently. The idea of primitive-based control is to generate trajectories on the basis of several primitives, which encode simple and stereotypical motions. To achieve desired, complex trajectories, primitives are sometimes adapted by parameters and sequenced or superimposed.

Approaches that use primitive-based control for robotic manipulators or mobile robots are presented in the following: Primitives based on dynamical systems encoding the desired trajectory in a landscape of attractors are introduced in [75] and specified in [147] as *Dynamic Movement Primitives* (DMPs). The trajectory profiles can be adapted by parameters, which makes DMPs attractive for imitation learning approaches as in [90, 114, 148]. In [110], movement-patterns are used for motion capturing and learning, where interior points are extracted from a human movement trajectory. Spline optimization is applied in order to compute the trajectory, passing through the extracted and fixed interior points. Nonlinear contraction theory is applied in [125] for a smooth concatenation of DMPs for trajectories of an unmanned aerial vehicle (UAV). A similar approach is presented in [47], where drawing tasks are reproduced by switching between linear, dynamical systems delivering different parts of the trajectory.

Motion primitives inspired by experimental observations on the motion generation of vertebrates are introduced in [118]. There, primitives are defined as linear systems with different equilibria. By a suitable linear combination of the system outputs, desired motions of a manipulator with two degrees of freedom (DOF) are generated. In [54], a primitive-based hybrid control approach is presented, in which the dynamics of nonlinear systems are quantized in terms of time-parameterized trajectory primitives. The primitives are concatenated by a *maneuver automaton* to generate feasible trajectories for a UAV. A method based on the non-optimal concatenation of linear-quadratic regulators around set-points of the linearized dynamics is presented in [171] for planning stabilizing trajectories.

Here, trajectory planning based on optimal control primitives is proposed. At first, the solution method is formulated for HOCPs independently of robotic applications. The solvable class of HOCPs has linear dynamics and quadratic costs or is transformable to such a problem by input-to-state and input-to-output feedback linearization. Further, autonomous switching and interior point constraints are considered. In [52], the optimal solution of a finite-horizon, linear-quadratic optimal control problem is represented by a parametrized optimal control primitive. Here, these primitives are used and optimally concatenated, such that the HOCP is solved. The optimal parameters required for the solution are computed algebraically in several milliseconds. In a next step, the method is applied to trajectory planning for robotic manipulators. Collision-free trajectories are obtained if the primitive-based optimization is combined with a kinematic motion planner. To avoid collisions, samples from the solution of the motion planner are included iteratively as interior point constraints in the optimization to guide the trajectory around obstacles if necessary. Since the interior points obtained from the motion planner are not optimal, the overall trajectory is suboptimal. However, the trajectory is optimal with respect to the specific choice of interior points. The primitive-based optimal control approach is published in [218, 219].

6.2 Model Predictive Control based on Neighboring Extremals

In this section, an MPC algorithm with potential for efficient computations is proposed. It solves HOCPs with autonomous switching, interior point constraints, and state and control constraints based on neighboring extremals. In Sec. 6.2.1, the HOCP is formulated and Sec. 6.2.2 presents the gradient method. In Sec. 6.2.3, the MPC algorithm is introduced and in Sec. 6.2.4, it is applied to a numerical example. Sec. 6.2.5 discusses some observations of the method.

6.2.1 Hybrid Optimal Control Problem

Hybrid systems with continuous and discrete states, continuous controls, continuous and discrete dynamics, switching manifolds for autonomous switching, and state and control constraints are considered in the following. The hybrid system is defined in Definition 2.42 and satisfies the Assumption 2.43. The HOCP is formulated in Definition 2.44 with terminal and running costs and interior and terminal constraints. Additionally, all functions of the hybrid system and the HOCP depend explicitly on time and a set of parameters. Parameters $\bar{p} \in \mathcal{P}$ are assumed to be constant during an execution $\sigma = (\tau, \varrho, \chi, \upsilon)$ of the hybrid system. Parameters are considered for the proposed MPC algorithm since in practical applications of MPC, model parameters are often unknown and estimated in parallel to a model predictive controller. When updates of model parameters are provided by an estimator like a Kalman filter, the model predictive controller has to include the updated parameters in its computations.

For a simplified notation throughout this section, events are introduced. Events include interior point constraints, beginnings of a state constraint, and switches of the discrete state when the continuous state trajectory hits a switching manifold.

Definition 6.1: Assuming that only one event can occur at the event time t_j, the corresponding interior condition is

$$0 = m_j(x(t_j), \bar{p}, t_j) := \begin{cases} m_{q_{j-1},q_j}(x(t_j), \bar{p}, t_j) & \text{if aut. switching} \\ \big(h^{\mathrm{s,a}}_{q_{j-1}}(x(t_j), \bar{p}, t_j)^T, ..., \\ \qquad h^{(r-1),\mathrm{s,a}}_{q_{j-1}}(x(t_j), \bar{p}, t_j)^T\big)^T & \text{if state constraint} \\ \psi_{j-1}(x(t_j), \bar{p}, t_j) & \text{if interior constraint} \end{cases} \tag{6.1}$$

for $t_j \in [t_0, t_e]$, $j \in \{1, ..., N\}$, and $q_j \in \mathcal{Q}$, and the terminal condition is

$$0 = m_{N+1}(x(t_e), \bar{p}, t_e) := \psi_N(x(t_e), \bar{p}, t_e) \tag{6.2}$$

at the final time t_e. □

6.2.2 Optimal Control Algorithm

Before introducing the MPC algorithm, the applied basic optimal control algorithm is presented in this section. It solves a HOCP over a receding horizon at each sample time of the MPC algorithm with new estimates or measurements of the state and constant model parameters. The basic algorithm successively linearizes a set of first-order optimality conditions around a given trajectory to iteratively approach an optimal trajectory. The linearized optimality conditions lead to a linear MPBVP, which is solved by a gradient method of second order, compare [32]. We assume throughout this section that the optimal sequence of discrete states and events is initially given and does not change during the optimization.

Optimality Conditions Necessary optimality conditions are derived here with variational methods. Below, the dependence of variables on time is often omitted and partial derivatives $\nabla_x f_{q_j}$ are denoted by $f_{q_j,x}$ etc. The Hamiltonian is as defined in Definition 2.24:

$$H_{q_j}(x,\lambda,u,\mu,\bar{p},t) = \phi_{q_j}(x,u,\bar{p},t) + \lambda^T f_{q_j}(x,u,\bar{p},t) + \mu^T h_{q_j}(x,u,\bar{p},t)$$

for $t \in [t_j, t_{j+1})$, $j \in \{0,...,N\}$, $q \in \mathcal{Q}$, and $\mu \geq 0 \in \mathbb{R}^{n_{h_{q_j}}}$.

Theorem 6.1: Given a hybrid system $\mathbb{H}$ according to Definition 2.42 and a hybrid system execution σ fulfilling the Assumptions 2.43, 2.11 and 2.12, then all controls υ^* (locally) minimizing the cost functional:

$$\inf_{\upsilon \in \mathcal{U}} J(\upsilon) \tag{6.3}$$

lead to a locally optimal execution $\sigma^* = (\tau^*, \varrho^*, \chi^*, \upsilon^*)$, such that the following conditions are satisfied for a fixed $\bar{p} \in \mathcal{P}$:

$$\dot{x}^*(t) = f_{q_j}(x^*(t), u^*_{q_j^*}(t), \bar{p}, t) \qquad \text{for } t \in [t_j, t_{j+1}),\ j \in \{0,...,N\} \tag{6.4}$$

$$\dot{\lambda}^*(t) = -H^T_{q_j^*,x}(x^*(t), \lambda^*(t), u^*_{q_j^*}(t), \mu^*(t), \bar{p}, t) \qquad \text{for } t \in [t_j, t_{j+1}),\ j \in \{0,...,N\} \tag{6.5}$$

$$0 = H_{q_j^*,u}(x^*(t), \lambda^*(t), u^*_{q_j^*}(t), \mu^*(t), \bar{p}, t) \qquad \text{for } t \in [t_j, t_{j+1}),\ j \in \{0,...,N\} \tag{6.6}$$

$$0 = y_{N+1} := m^T_{N+1,x}(x^*(t_e), \bar{p}, t_e)\, \pi^*_{N+1} + g^T_x(x^*(t_e), \bar{p}, t_e) - \lambda^*(t_e) \tag{6.7}$$

$$0 = y_j := \lambda^*(t_j) - \lambda^*(t_j^-) + m^T_{j,x}(x^*(t_j), \bar{p}, t_j)\, \pi^*_j \qquad \text{for } j \in \{1,...,N\} \tag{6.8}$$

$$0 = z_{N+1} := H_{q_N^*}(x^*(t_e), \bar{p}, t_e) + g_t(x^*(t_e), \bar{p}, t_e) + m_{N+1,t}(x^*(t_e), \bar{p}, t_e)\, \pi^*_{N+1} \tag{6.9}$$

$$0 = z_j := H_{q_{j-1}^*}(x^*(t_j), \bar{p}, t_j^-) - H_{q_j^*}(x^*(t_j), \bar{p}, t_j) + m^T_{j,t}(x^*(t_j), \bar{p}, t_j)\, \pi^*_j \qquad \text{for } j \in \{1,...,N\} \tag{6.10}$$

$$0 = h^{\mathrm{a}}_{q_j^*}(x^*(t), u^*_{q_j^*}(t), \bar{p}, t) \qquad \text{for } t \in [t_j, t_{j+1}),\ j \in \{0,...,N\} \tag{6.11}$$

$$0 = \mu^{\mathrm{ina},*}(t) \qquad \text{for } t \in [t_0, t_e] \tag{6.12}$$

$$0 = m_j(x^*(t_j), \bar{p}, t_j) \qquad \text{for } j \in \{1,...,N+1\} \tag{6.13}$$

with constant and optimal Lagrange multipliers $\pi^*_j \in \mathbb{R}^{n_{m_j}}$ and the abbreviations y_{N+1}, y_j, z_{N+1}, and z_j. □

Note that condition (6.9) vanishes if the terminal time t_e is fixed.

Proof: The optimality conditions are found by standard calculus of variations, compare [32]. ■

Neighboring Extremals Assume that an optimal trajectory χ_1^* fulfilling the optimality conditions above is given. Under the standing assumptions and Assumption 6.3, there exists an optimal trajectory χ_2^* in the neighborhood of χ_1^* with $x_2(t_0) = x_1(t_0) + \delta x(t_0)$ and $\bar{p}_2 = \bar{p}_1 + d\bar{p}$. An estimator provides model parameters $\bar{p}$ and the difference in the values of the current and the previous iteration of the MPC algorithm is denoted by $d\bar{p}$. The optimal control algorithm to be presented successively approaches the trajectory χ_2^* by iteratively solving linear MPBVPs, which result from a linearization around a given trajectory starting with a linearization around χ_1^*. The approach also works for a non-optimal initial trajectory χ_1.

Proposition 6.2: Let a trajectory $\chi : [t_0, t_e] \to \mathcal{X}$ with discrete state sequence $\varrho = (0, 1, ..., N)$ be given[1], let the standing assumptions hold, and define

$$A_j := f_{j,x} - f_{j,u}\Big(E_{j,1}H_{j,ux} + E_{j,2}^T h_{j,x}^{\mathrm{a}}\Big) \tag{6.14}$$

$$B_j := f_{j,u}E_{j,1}f_{j,u}^T \tag{6.15}$$

$$C_j := H_{j,xx} - H_{j,xu}\Big(E_{j,1}H_{j,ux} + E_{j,2}^T h_{j,x}^{\mathrm{a}}\Big) - h_{j,x}^{\mathrm{a},T}\Big(E_{j,2}H_{j,ux} + E_{j,3}h_{j,x}^{\mathrm{a}}\Big) \tag{6.16}$$

$$D_j := f_{j,\bar{p}} - f_{j,u}\Big(E_{j,1}H_{j,u\bar{p}} + E_{j,2}^T h_{j,\bar{p}}^{\mathrm{a}}\Big) \tag{6.17}$$

$$G_j := H_{j,x\bar{p}} - H_{j,xu}\Big(E_{j,1}H_{j,u\bar{p}} + E_{j,2}^T h_{j,\bar{p}}^{\mathrm{a}}\Big) - h_{j,x}^{\mathrm{a},T}\Big(E_{j,2}H_{j,u\bar{p}} + E_{j,3}h_{j,\bar{p}}^{\mathrm{a}}\Big) \tag{6.18}$$

$$d_j := f_{j,u}\Big(E_{j,1}\theta_j - E_{j,2}^T\delta h_j^{\mathrm{a}}\Big) \tag{6.19}$$

$$\gamma_j := -h_{j,x}^{\mathrm{ina},T}\delta\mu^{\mathrm{ina}} + H_{j,xu}\Big(E_{j,1}\theta_j - E_{j,2}^T\delta h_j^{\mathrm{a}}\Big) + h_{j,x}^{\mathrm{a},T}\Big(E_{j,2}\theta_j - E_{j,3}\delta h_j^{\mathrm{a}}\Big) \tag{6.20}$$

with $\theta_j := h_{j,u}^{\mathrm{ina},T}\delta\mu^{\mathrm{ina}} - \delta H_{j,u}$, $f_j := f_j(x(t), u(t), \bar{p}, t)$, $H_j := H_j(x(t), \lambda(t), u(t), \mu(t), \bar{p}, t)$, $H_N(t_e) := H_N(x(t_e), \lambda(t_e), u_N(t_e), \mu(t_e), \bar{p}, t_e)$, $\Delta H_j := H_{j-1}(t_j^-) - H_j(t_j)$, and

$$E_j := \begin{pmatrix} E_{j,1} & E_{j,2}^T \\ E_{j,2} & E_{j,3} \end{pmatrix} := \begin{pmatrix} H_{j,uu} & h_{j,u}^{\mathrm{a},T} \\ h_{j,u}^{\mathrm{a}} & 0 \end{pmatrix}^{-1}. \tag{6.21}$$

Here and hereafter, it is assumed that $E_j^{-1} > 0$.[2] Then, a first-order Taylor series expansion

[1] The notation is chosen without loss of generality, since f_j and f_{j+1} with $j \in \{0, ..., N\}$ may also be identical.

[2] The assumption that E_j^{-1} fulfills $E_j^{-1} > 0$ and thus is invertible is reasonable, when singular cases are not considered. Singular cases are not considered here but can be solved by applying methods from the literature, e.g. [18, 139]. In general, singular cases do not appear when the elements of the control appear at least quadratically in the running cost functions and/or system differential equations.

leads to the hybrid, linear MPBVP with the dynamics

$$\begin{aligned} \delta\dot{x} &= A_j\delta x - B_j\delta\lambda + D_j d\bar{p} - d_j && (6.22)\\ \delta\dot{\lambda} &= \, - C_j\delta x - A_j^T\delta\lambda - G_j d\bar{p} + \gamma_j && (6.23)\end{aligned}$$

and the boundary conditions

$$\begin{aligned} dy_{N+1} &= g_{xt}(x(t_e),\bar{p},t_e)dt_e + g_{xx}(x(t_e),\bar{p},t_e)dx(t_e) + g_{x\bar{p}}(x(t_e),\bar{p},t_e)d\bar{p}\\ &\quad + m_{N+1,x}^T(x(t_e),\bar{p},t_e)\, d\pi_{N+1} - d\lambda(t_e) + \Big(\pi_{N+1}^T m_{N+1,xt}(x(t_e),\bar{p},t_e)dt_e\\ &\quad + \big(\pi_{N+1}^T m_{N+1,x}(x(t_e),\bar{p},t_e)\big)_x dx(t_e) + \big(\pi_{N+1}^T m_{N+1,x}(x(t_e),\bar{p},t_e)\big)_{\bar{p}} d\bar{p}\Big)^T && (6.24)\\ dy_j &= d\lambda(t_j) - d\lambda(t_j^-) + m_{j,x}^T(x(t_j),\bar{p},t_j)d\pi_j + \Big(\pi_j^T m_{j,xt}(x(t_j),\bar{p},t_j)dt_j\\ &\quad + \big(\pi_j^T m_{j,x}(x(t_j),\bar{p},t_j)\big)_x dx(t_j) + \big(\pi_j^T m_{j,x}(x(t_j),\bar{p},t_j)\big)_{\bar{p}} d\bar{p}\Big)^T && (6.25)\\ dz_{N+1} &= g_{tt}(x(t_e),\bar{p},t_e)dt_e + g_{tx}(x(t_e),\bar{p},t_e)dx(t_e) + g_{t\bar{p}}(x(t_e),\bar{p},t_e)d\bar{p}\\ &\quad + H_{N,x}(t_e)dx(t_e) + H_{N,t}(t_e)dt_e + H_{N,\bar{p}}(t_e)d\bar{p} + H_{N,u}(t_e)du(t_e)\\ &\quad + f_N^T(t_e)d\lambda(t_e) + h_N^T(t_e)d\mu(t_e) + m_{N+1,t}^T d\pi_{N+1}\\ &\quad + \pi_{N+1}^T\Big(m_{N+1,tt}dt_e + m_{N+1,tx}dx(t_e) + m_{N+1,t\bar{p}}d\bar{p}\Big) && (6.26)\\ dz_j &= \Delta H_{j,t}dt_j + \Delta H_{j,x}dx(t_j) + \Delta H_{j,\bar{p}}d\bar{p} + f_{j-1}^T(t_j^-)d\lambda(t_j^-) - f_j^T(t_j)d\lambda(t_j)\\ &\quad + h_{j-1}^T(t_j^-)d\mu(t_j^-) - h_j^T(t_j)d\mu(t_j) + m_{j,t}^T d\pi_j\\ &\quad + H_{j-1,u}(t_j^-)du(t_j^-) - H_{j,u}(t_j)du(t_j)\\ &\quad + \pi_j^T\Big(m_{j,tt}dt_j + m_{j,tx}dx(t_j) + m_{j,t\bar{p}}d\bar{p}\Big) && (6.27)\\ dm_j &= m_{j,t}dt_j + m_{j,x}dx(t_j) + m_{j,\bar{p}}d\bar{p} && (6.28)\end{aligned}$$

with $h_j(t) := h_j(x(t),u_j(t),\bar{p},t)$ and $m_j := m_j(x(t_j),\bar{p},t_j)$. □

Proof: The proof of conditions (6.24) - (6.28) follows directly from a first-order Taylor series expansion of (6.7) - (6.10), and (6.13). To show (6.22) and (6.23), the optimality conditions (6.4), (6.5), and (6.6) - (6.12) are linearized:

$$\begin{aligned} \delta\dot{x} &= f_{j,x}\delta x + f_{j,u}\delta u + f_{j,\bar{p}}d\bar{p} && (6.29)\\ \delta\dot{\lambda} &= \, - H_{j,xx}\delta x - f_{j,x}^T\delta\lambda - H_{j,xu}\delta u - h_{j,x}^T\delta\mu - H_{j,x\bar{p}}d\bar{p} && (6.30)\\ \delta h_j^{\mathrm{a}} &= h_{j,x}^{\mathrm{a}}\delta x + h_{j,u}^{\mathrm{a}}\delta u + h_{j,\bar{p}}^{\mathrm{a}}d\bar{p} && (6.31)\\ \delta\mu^{\mathrm{ina}} &= 0 && (6.32)\\ \delta H_{j,u}^T &= H_{j,ux}\delta x + f_{j,u}^T\delta\lambda + H_{j,uu}\delta u + h_{j,u}^T\delta\mu + H_{j,u\bar{p}}d\bar{p}\,. && (6.33)\end{aligned}$$

Solving (6.33) and (6.31) for δu and $\delta\mu^{\mathrm{a}}$, the following is obtained:

$$\begin{pmatrix}\delta u\\ \delta\mu^{\mathrm{a}}\end{pmatrix} = \, - E_j\begin{pmatrix}H_{j,ux}\delta x + f_{j,u}^T\delta\lambda + H_{j,u\bar{p}}d\bar{p} + \theta_j\\ h_{j,x}^{\mathrm{a}}\delta x + h_{j,\bar{p}}^{\mathrm{a}}d\bar{p} - \delta h_j^{\mathrm{a}}\end{pmatrix}. \qquad (6.34)$$

Inserting (6.34) into (6.29) and (6.30), Equations (6.22) and (6.23) result. ■

Note that the term $h_{j,u}^{\text{ina},T}\delta\mu^{\text{ina}}$ is kept though $\delta\mu^{\text{ina}}$ is zero in this case here, since it is needed in the MPC algorithm later.

Solution of the MPBVP Multiple shooting as in [127] can be used to solve the MPBVP. We used instead a gradient method based on a transition matrix method and a backward sweep method with Riccati-matrices, compare [32]. Since the Riccati approach converged more reliably for the numerical example in Sec. 6.2.4, we only show the backward sweep method in the following. The backward sweep of all terminal and interior conditions to the initial time t_0 is set up such that the size of some Riccati-matrices, see (6.59) - (6.70) grows at each interior condition in direction of the initial time, compare [183]. The system of Riccati equations is based on the approach

$$\begin{aligned}
\delta\lambda(t) &= S_j(t)\delta x(t) + R_{j,1}(t)d\boldsymbol{\pi}_{j+1} + R_{j,2}(t)d\boldsymbol{t}_{j+1} + W_{j,1}(t)d\bar{p} + e_j(t) && (6.35)\\
d\boldsymbol{m}_j &= R_{j,1}^T(t)\delta x(t) + Q_{j,1}(t)d\boldsymbol{\pi}_{j+1} + Q_{j,2}(t)d\boldsymbol{t}_{j+1} + W_{j,2}(t)d\bar{p} + \eta_{j,1}(t) && (6.36)\\
d\boldsymbol{z}_j &= R_{j,2}^T(t)\delta x(t) + Q_{j,2}^T(t)d\boldsymbol{\pi}_{j+1} + Q_{j,3}(t)d\boldsymbol{t}_{j+1} + W_{j,3}(t)d\bar{p} + \eta_{j,2}(t) && (6.37)
\end{aligned}$$

with $\boldsymbol{m}_j := (m_{N+1}^T \ \dots \ m_j^T)^T$, $\boldsymbol{z}_j := (z_{N+1}^T \ \dots \ z_j^T)^T$, $\boldsymbol{\pi}_j := (\pi_{N+1}^T \ \dots \ \pi_j^T)^T$, and $\boldsymbol{t}_j := (t_e \ \dots \ t_j)^T$.

Proposition 6.3: Let the standing assumptions hold and define $\|\Delta f_j\|^2_{R_{j-1,2}(t_j)} := \Delta f_j^T R_{j-1,2}(t_j)\Delta f_j$ and:

$$\begin{aligned}
\hat{S}_j &:= \left(\pi_j^T m_{j,x}\right)_x && (6.38)\\
\hat{R}_{j,1} &:= m_{j,x}^T && (6.39)\\
\hat{R}_{j,2} &:= \left(\pi_j^T m_{j,xt}\right)^T + \Delta H_{j,x}^T + \left(\pi_j^T m_{j,x}\right)_x f_{j-1}(t_j^-) && (6.40)\\
\hat{Q}_{j,2} &:= m_{j,t} + m_{j,x} f_{j-1}(t_j^-) && (6.41)\\
\hat{Q}_{j,3} &:= \Delta H_{j,t} + \Delta H_{j,x} f_{j-1}(t_j^-) + H_{j,x} f_j(t_j) + f_{j-1}^T(t_j^-)\hat{R}_{j,2}\\
&\quad + \pi_j^T\left(m_{j,tt} + m_{j,tx} f_{j-1}(t_j^-)\right)\\
&\quad + H_{j-1,u}(t_j^-)\dot{u}_{j-1}(t_j^-) - H_{j,u}(t_j)\dot{u}_j(t_j)\\
&\quad + h_{j-1}^T(t_j^-)\dot{\mu}_{j-1}(t_j^-) - h_j^T(t_j)\dot{\mu}_j(t_j) && (6.42)\\
\hat{W}_{j,1} &:= \left(\pi_j^T m_{j,x}\right)_{\bar{p}}^T && (6.43)\\
\hat{W}_{j,2} &:= m_{j,\bar{p}} && (6.44)\\
\hat{W}_{j,3} &:= \Delta H_{j,\bar{p}}(t_j) + f_{j-1}^T(t_j^-)\left(\pi_j^T m_{j,x}\right)_{\bar{p}} + \pi_j^T m_{j,t\bar{p}} && (6.45)\\
\hat{\eta}_{j,2} &:= H_{j-1,u}(t_j^-)\delta u(t_j^-) - H_{j,u}(t_j)\delta u(t_j)\\
&\quad + h_{j-1}^T(t_j^-)\delta\mu(t_j^-) - h_j^T(t_j)\delta\mu(t_j) && (6.46)
\end{aligned}$$

for $j \in \{1, ..., N\}$, where the expressions $f_{N+1}(t_e)$, $H_{N+1}(t_e)$, $\delta u_{N+1}(t_e)$, $\delta\mu_{N+1}(t_e)$, $\dot{u}_{N+1}(t_e)$, and $\dot{\mu}_{N+1}(t_e)$ are zero. Then the differential equations (6.22) and (6.23) of the

MPBVP can be expressed by the following system of Riccati differential equations:

$$\begin{aligned}
\dot{S}_j =& -C_j - A_j^T S_j - S_j A_j + S_j B_j S_j && (6.47)\\
\dot{R}_{j,1} =& -(A_j^T - S_j B_j) R_{j,1} && (6.48)\\
\dot{R}_{j,2} =& -(A_j^T - S_j B_j) R_{j,2} && (6.49)\\
\dot{Q}_{j,1} =& -R_{j,1}^T B_j R_{j,1} && (6.50)\\
\dot{Q}_{j,2} =& -R_{j,1}^T B_j R_{j,2} && (6.51)\\
\dot{Q}_{j,3} =& -R_{j,2}^T B_j R_{j,2} && (6.52)\\
\dot{W}_{j,1} =& -(A_j^T - S_j B_j) W_{j,1} - G_j - S_j D_j && (6.53)\\
\dot{W}_{j,2} =& -R_{j,1}^T (B_j W_{j,1} - D_j) && (6.54)\\
\dot{W}_{j,3} =& -R_{j,2}^T (B_j W_{j,1} - D_j) && (6.55)\\
\dot{e}_j =& -(A_j^T - S_j B_j) e_j + S_j d_j + \gamma_j && (6.56)\\
\dot{\eta}_{j,1} =& R_{j,1}^T (B_j e_j + d_j) && (6.57)\\
\dot{\eta}_{j,2} =& R_{j,2}^T (B_j e_j + d_j)\,. && (6.58)
\end{aligned}$$

Further, the corresponding interior conditions (6.25), (6.27), and (6.28) for $j \in \{1, ..., N\}$ are satisfied by:

$$\begin{aligned}
S_{j-1}(t_j^-) &= S_j(t_j) + \hat{S}_j && (6.59)\\
R_{j-1,1}(t_j^-) &= \begin{pmatrix} R_{j,1}(t_j) & \hat{R}_{j,1} \end{pmatrix} && (6.60)\\
R_{j-1,2}(t_j^-) &= \begin{pmatrix} R_{j,2}(t_j) & \hat{R}_{j,2} \end{pmatrix} && (6.61)\\
Q_{j-1,1}(t_j^-) &= \begin{pmatrix} Q_{j,1}(t_j) & 0 \\ 0 & 0 \end{pmatrix} && (6.62)\\
Q_{j-1,2}(t_j^-) &= \begin{pmatrix} Q_{j,2}(t_j) & R_{j-1,1}^T(t_j)\Delta f_j \\ 0 & \hat{Q}_{j,2} \end{pmatrix} && (6.63)\\
Q_{j-1,3}(t_j^-) &= \begin{pmatrix} Q_{j,3}(t_j) & R_{j-1,2}^T(t_j)\Delta f_j \\ \Delta f_j^T R_{j-1,2}(t_j) & \hat{Q}_{j,3} + \|\Delta f_j\|^2_{R_{j-1,2}(t_j)} \end{pmatrix} && (6.64)\\
W_{j-1,1}(t_j^-) &= W_{j,1}(t_j) + \hat{W}_{j,1} && (6.65)\\
W_{j-1,2}(t_j^-) &= \begin{pmatrix} W_{j,2}^T(t_j) & \hat{W}_{j,2}^T \end{pmatrix}^T && (6.66)\\
W_{j-1,3}(t_j^-) &= \begin{pmatrix} W_{j,3}^T(t_j) & \hat{W}_{j,3}^T + W_{j,1}^T(t_j)\Delta f_j \end{pmatrix}^T && (6.67)\\
e_{j-1}(t_j^-) &= e_j(t_j) && (6.68)\\
\eta_{j-1,1}(t_j^-) &= \begin{pmatrix} \eta_{j,1}^T(t_j) & 0 \end{pmatrix}^T && (6.69)\\
\eta_{j-1,2}(t_j^-) &= \begin{pmatrix} \eta_{j,2}^T(t_j) & \hat{\eta}_{j,2}^T + e_j^T(t_j)\Delta f_j \end{pmatrix}^T\,. && (6.70)
\end{aligned}$$

The terminal conditions (6.24) and (6.26) are:

$$S_{N+1}(t_e) = \hat{S}_{N+1} + g_{xx}(x(t_e), \bar{p}, t_e) \qquad (6.71)$$

$$R_{N+1,1}(t_e) = \hat{R}_{N+1,1} \tag{6.72}$$

$$R_{N+1,2}(t_e) = \hat{R}_{N+1,2} + g_{xt}(t_e) + g_{xx}(t_e)f_N(t_e) \tag{6.73}$$

$$Q_{N+1,1}(t_e) = 0 \tag{6.74}$$

$$Q_{N+1,2}(t_e) = \hat{Q}_{N+1,2} \tag{6.75}$$

$$Q_{N+1,3}(t_e) = \hat{Q}_{N+1,3} + g_{tt}(t_e) + g_{tx}(t_e)f_N(t_e) \tag{6.76}$$

$$W_{N+1,1}(t_e) = \hat{W}_{N+1,1} + g_{x\bar{p}}(t_e) \tag{6.77}$$

$$W_{N+1,2}(t_e) = \hat{W}_{N+1,2} \tag{6.78}$$

$$W_{N+1,3}(t_e) = \hat{W}_{N+1,3} + g_{t\bar{p}}(t_e) \tag{6.79}$$

$$e_{N+1}(t_e) = 0 \tag{6.80}$$

$$\eta_{N+1,1}(t_e) = 0 \tag{6.81}$$

$$\eta_{N+1,2}(t_e) = \hat{\eta}_{N+1,2}\,. \tag{6.82}$$

□

Proof: Differentiating (6.35) - (6.37) with respect to time and regrouping terms with respect to their dependence on δx, $d\boldsymbol{\pi}_{j+1}$, $d\boldsymbol{t}_{j+1}$, $d\bar{p}$ (and on none of these terms), the Riccati differential equations (6.47) - (6.58) are obtained. The remaining conditions are derived by inserting the relations (6.35) - (6.37), and

$$d\lambda(t_j) = \delta\lambda(t_j) + \dot{\lambda}(t_j)dt_j \tag{6.83}$$

$$d\lambda(t_j^-) = \delta\lambda(t_j^-) + \dot{\lambda}(t_j^-)dt_j \tag{6.84}$$

$$dx(t_j) = \delta x(t_j) + \dot{x}(t_j)dt_j = \delta x(t_j^-) + \dot{x}(t_j^-)dt_j \tag{6.85}$$

into (6.24) - (6.28) and regrouping terms. ■

Remark 6.2: At the initial time t_0, after integrating the Riccati-matrices and vectors backwards in time, the unknowns $d\boldsymbol{\pi}_1$ and $d\boldsymbol{t}_1$ are calculated with the given values $\delta x(t_0)$ and $d\bar{p}$ and the desired values $d\boldsymbol{m}_1$ and $d\boldsymbol{z}_1$:

$$\begin{pmatrix} d\boldsymbol{\pi}_1 \\ d\boldsymbol{t}_1 \end{pmatrix} = \begin{pmatrix} Q_{0,1}(t_0) & Q_{0,2}(t_0) \\ Q_{0,2}^T(t_0) & Q_{0,3}(t_0) \end{pmatrix}^{-1} \left[\begin{pmatrix} d\boldsymbol{m}_1 - \eta_{0,1}(t_0) \\ d\boldsymbol{z}_1 - \eta_{0,2}(t_0) \end{pmatrix} - \begin{pmatrix} R_{0,1}^T(t_0)\delta x(t_0) + W_{0,2}(t_0)d\bar{p} \\ R_{0,2}^T(t_0)\delta x(t_0) + W_{0,3}(t_0)d\bar{p} \end{pmatrix} \right]. \tag{6.86}$$

Afterwards, the MPBVP is solved by integrating (6.22) forward in time with (6.35) and (6.86). □

Iterative Optimization In the following, a second-order gradient algorithm similar to the one in [32] is shortly presented. It finds an optimal control trajectory satisfying (6.4) - (6.13) by iteratively improving an initially guessed control trajectory based on a linearization of the optimality conditions (6.22) - (6.28) around the current trajectory.

Algorithm 6.1:

Step 0 Find feasible initial trajectories υ^1 and μ^1 and values t_j^1 and π_j^1 for $j \in \{1, ..., N+1\}$ in the fixed discrete state and event sequence $\varrho = (0, ..., N)$. Set iteration counter $l = 0$ and choose $\tau_0 \in (0, 1)$ and $\beta \in (0, 1)$.

Step 1 Integrate (6.4) forward in time to obtain χ^{l+1}.

Step 2 Integrate (6.5) backwards in time to obtain λ^{l+1}.

Step 3 Evaluate the error

$$\Omega^{l+1} = \frac{1}{2} \sum_{j=0}^{N} \int_{t_j^{l+1}}^{t_{j+1}^{l+1}} \|H_{j,u}^{l+1}\|^2 + \|h_j^{\mathrm{a},l+1}\|^2 \, dt + \frac{1}{2} \sum_{j=1}^{N+1} \left(\|m_j^{l+1}\|^2 + \|z_j^{l+1}\|^2 \right), \quad (6.87)$$

where $\|a\|^2 = a^T a$ and a^{l+1} means that all arguments of a are evaluated for iteration $l + 1$.

Step 4 If $\Omega^{l+1} < \epsilon_\Omega$, then stop. Else if $l > 0$ and if the Armijo criterion [4]

$$\Omega^{l+1} - \Omega^l \leq -2\tau_0 \beta^p \Omega^l \quad (6.88)$$

is fulfilled, then set $l := l + 1$ and go to Step 5, else keep l and go back to Step 5 with $p := p + 1$, $p \in \mathbb{N}_0$, for a reduced step size.

Step 5 Find $\delta\upsilon$, $\delta\mu^{\mathrm{a}}$, dt_j, and $d\pi_j$ from the solution of the MPBVP around the trajectories χ^l and λ^l with modified Newton steps $\delta\mu^{\mathrm{ina}} = 0$, $\delta H_{j,u} = -\beta^p H_{j,u}^l$, $\delta h_j^{\mathrm{a}} = -\beta^p h_j^{\mathrm{a},l}$, $dm_j = -\beta^p m_j^l$, and $dz_j = -\beta^p z_j^l$.

Step 6 Determine υ^{l+1}, μ^{l+1}, t_j^{l+1}, and π_j^{l+1} with $\delta\upsilon$, $\delta\mu$, dt_j, and $d\pi_j$ and go to Step 1.□

Hereafter, the convergence of the algorithm is shown using the following assumption.

Assumption 6.3: For all iterations l, $t \in [t_j^l, t_{j+1}^l)$, and $j \in \{0, ..., N\}$, it is assumed that

$$Q_0(t_0) := \begin{pmatrix} Q_{0,1}(t_0) & Q_{0,2}(t_0) \\ Q_{0,2}^T(t_0) & Q_{0,3}(t_0) \end{pmatrix} \text{ is invertible,}$$

that no change in the discrete state and event sequence ϱ occurs, and that an optimal control υ^* exists in a convex region around the initial control υ^0, such that the optimality conditions (6.4) - (6.13) are satisfied. □

Remark 6.4: In general, the matrix $Q_0(t_0)$ composed of the terms $Q_{0,1}(t_0)$, $Q_{0,2}(t_0)$, and $Q_{0,3}(t_0)$ is invertible, which is supported by an analysis of the boundary conditions and the quadratic differential equations for the matrix $Q_0(t_0)$ and our experience. For less complex HOCPs than the one considered here, the existence of the inverse of $Q_0(t_0)$ can be proven. During the optimization, the discrete state and event sequence does not change, if the initialization is close enough to the optimal solution and shares the same discrete state and event sequence. In this case and if no singular controls exist, an optimal control exists in a convex region around the initialization. If the latter two assumptions are not fulfilled, either the sampling time needs to be reduced or even a new initialization has to be found.□

Proposition 6.4: Suppose the Assumptions 2.43, 2.11, 2.12, and 6.3 to hold. Then, there exists a $p \in \mathbb{N}_0$, such that

$$\Omega^{l+1} - \Omega^l \leq -2\tau_0 \beta^p \Omega^l \tag{6.89}$$

is true for all $l \geq 0$. □

Proof: The Armijo condition (6.89) is proven separately for the integral and non-integral part in (6.87). The separation is possible since improvements in one part do not influence the fulfillment in the other part up to first order. The proof is shown by contradiction. That means, at first, it is assumed that for any $p \in \mathbb{N}_0$

$$\begin{aligned}\frac{1}{2}\sum_{j=0}^{N}\Big(\int_{t_j^{l+1}}^{t_{j+1}^{l+1}} \|H_{j,u}^{l+1}\|^2 + \|h_j^{\mathrm{a},l+1}\|^2\,dt - \int_{t_j^l}^{t_{j+1}^l} \|H_{j,u}^{l}\|^2 + \|h_j^{\mathrm{a},l}\|^2\,dt\Big)&\\ > -\tau_0\beta^p \sum_{j=0}^{N}\int_{t_j^l}^{t_{j+1}^l} \|H_{j,u}^{l}\|^2 + \|h_j^{\mathrm{a},l}\|^2\,dt&\end{aligned} \tag{6.90}$$

holds. In the following of the proof, the differences $\delta x(t_0)$ and $d\bar{p}$ are set to zero, which is possible since after one iteration of the algorithm (under Assumption 6.3 and possibly while ignoring the Armijo condition (6.89)) $\delta x(t_0)$ and $d\bar{p}$ are zero for all further iterations l. This implies that $\lim_{p\to\infty}\delta\chi = \lim_{p\to\infty}\delta\lambda = 0$, $\lim_{p\to\infty} dt_j = 0$, and $\lim_{p\to\infty} d\pi_j = 0$. Values in iteration $l+1$ are approximated by a Taylor series around values in iteration l:

$$H_{j,u}^{l+1} = H_{j,u}^{l} + H_{j,ux}^{l}\delta x + f_{j,u}^{l,T}\delta\lambda + H_{j,uu}^{l}\delta u + h_{j,u}^{l,T}\delta\mu + H_{j,u\bar{p}}^{l}d\bar{p} + o(\beta^p) \tag{6.91}$$

$$h_j^{\mathrm{a},l+1} = h_j^{\mathrm{a},l} + h_{j,x}^{\mathrm{a},l}\delta x + h_{j,u}^{\mathrm{a},l}\delta u + h_{j,\bar{p}}^{\mathrm{a},l}d\bar{p} + o(\beta^p). \tag{6.92}$$

Eq. (6.90) is divided by β^p and passed to the limit $p \to \infty$, while inserting (6.91) and (6.92) and rearranging integral boundaries. The left hand side is then with $d\bar{p} = 0$ and $\delta\mu^{\mathrm{ina}} = 0$:

$$\begin{aligned}\lim_{p\to\infty}\frac{1}{2\beta^p}\Bigg[&\sum_{j=0}^{N} dt_{j+1}\Big(\|H_{j,u}^{l}(t_{j+1}^{l-})\|^2 + \|h_j^{\mathrm{a},l}(t_{j+1}^{l-})\|^2 - \|H_{j+1,u}^{l}(t_{j+1}^{l})\|^2 - \|h_{j+1}^{\mathrm{a},l}(t_{j+1}^{l})\|^2\Big)\\ &+2\sum_{j=0}^{N}\int_{t_j^l}^{t_{j+1}^l}\begin{pmatrix}H_{j,u}^{l,T}\\ h_j^{\mathrm{a},l}\end{pmatrix}^T\begin{pmatrix}H_{j,ux}^{l}\delta x + f_{j,u}^{l,T}\delta\lambda + H_{j,uu}^{l}\delta u + h_{j,u}^{\mathrm{a},l,T}\delta\mu^{\mathrm{a}}\\ h_{j,x}^{\mathrm{a},l}\delta x + h_{j,u}^{\mathrm{a},l}\delta u\end{pmatrix} dt + o(\beta^p)\Bigg].\end{aligned} \tag{6.93}$$

For simplicity, it is assumed here that the magnitude of the first term in (6.93) is smaller than the last term, which in our experience was always the case. However, it is also possible to add the term

$$\begin{aligned}\begin{pmatrix}\delta u_c\\ \delta\mu_c^{\mathrm{a}}\end{pmatrix} = &-\frac{dt_{j+1}}{2(t_{j+1}^l - t_j^l)}\,E_j\begin{pmatrix}H_{j,u}^{l,T}\\ h_j^{\mathrm{a},l}\end{pmatrix}\left\|\begin{pmatrix}H_{j,u}^{l,T}\\ h_j^{\mathrm{a},l}\end{pmatrix}\right\|^{-2}\\ &\left(\left\|\begin{pmatrix}H_{j,u}^{l,T}(t_{j+1}^{l-})\\ h_j^{\mathrm{a},l}(t_{j+1}^{l-})\end{pmatrix}\right\|^2 - \left\|\begin{pmatrix}H_{j+1,u}^{l,T}(t_{j+1}^{l})\\ h_{j+1}^{\mathrm{a},l}(t_{j+1}^{l})\end{pmatrix}\right\|^2\right)\end{aligned} \tag{6.94}$$

with dt_{j+1} taken from (6.27) to δu and $\delta\mu$ in (6.34), which compensates the first term in (6.93), as it is similarly introduced in Sec. 4.3.3. Inserting δu from (6.34) into (6.93), finally, the contradiction

$$-\sum_{j=0}^{N}\int_{t_j^l}^{t_{j+1}^l}\left\|\begin{pmatrix}H_{j,u}^{l,T}\\ h_j^{\mathrm{a},l}\end{pmatrix}\right\|^2 dt < -\tau_0\sum_{j=0}^{N}\int_{t_j^l}^{t_{j+1}^l}\left\|\begin{pmatrix}H_{j,u}^{l,T}\\ h_j^{\mathrm{a},l}\end{pmatrix}\right\|^2 dt \tag{6.95}$$

is obtained, which proves the first part of (6.89). Similarly for the second part, it is shown that the assumption, that for any $p \in \mathbb{N}_0$ the relation

$$\frac{1}{2}\left\|\begin{pmatrix}\boldsymbol{m}_0^{l+1}\\ \boldsymbol{z}_0^{l+1}\end{pmatrix}\right\|^2 - \left\|\begin{pmatrix}\boldsymbol{m}_0^{l}\\ \boldsymbol{z}_0^{l}\end{pmatrix}\right\|^2 > -\tau_0\beta^p\left\|\begin{pmatrix}\boldsymbol{m}_0^{l}\\ \boldsymbol{z}_0^{l}\end{pmatrix}\right\|^2 \tag{6.96}$$

holds, leads to a contradiction. A Taylor series expansion is formed with $\delta x(t_0) = 0$ and $d\bar{p} = 0$:

$$\boldsymbol{m}_0^{l+1} = \boldsymbol{m}_0^l + d\boldsymbol{m}_0^l + o(\beta^p) \tag{6.97}$$

$$\boldsymbol{z}_0^{l+1} = \boldsymbol{z}_0^l + d\boldsymbol{z}_0^l + o(\beta^p)\,. \tag{6.98}$$

Using (6.36), (6.37), and (6.86), dividing the left hand side of (6.96) by β^p and passing to the limit, the contradiction

$$-\left\|\begin{pmatrix}\boldsymbol{m}_0^{l}\\ \boldsymbol{z}_0^{l}\end{pmatrix}\right\|^2 < -\tau_0\left\|\begin{pmatrix}\boldsymbol{m}_0^{l}\\ \boldsymbol{z}_0^{l}\end{pmatrix}\right\|^2 \tag{6.99}$$

compared to (6.96) results, such that (6.89) is proven. ■

Theorem 6.5: Let the Assumptions 2.43, 2.11, 2.12, and 6.3 hold. Then Algorithm 6.1 converges to a solution trajectory υ^*, such that the optimality conditions (6.4) - (6.13) are satisfied. □

Proof: The error Ω is bounded below by zero. Since (6.89) always holds and $\Omega^l \geq 0$, the difference $\Omega^{l+1} - \Omega^l$ always decreases by a non-zero amount. Together with $\Omega \geq 0$ and the standing assumptions, this leads to the convergence of Ω to zero. Thus, the optimality conditions (6.7) - (6.11), and (6.13) are fulfilled. The remaining conditions are satisfied due to the construction of the algorithm. ■

6.2.3 Model Predictive Control Algorithm

In this section, it is described how the presented optimal control algorithm is applied in an MPC context.

Concept The index ρ is the current time step of the MPC algorithm and the sample time is T_s. The initial and final times of the current time step are t_0^ρ and t_e^ρ, which gives a prediction horizon of $T_p = t_e^\rho - t_0^\rho \gg T_s$. The MPC algorithm has 3 main steps:

Algorithm 6.2:

Step 0 **Initialization:** Initialize with trajectories $\upsilon^{0,\text{ini}}$ and $\mu^{0,\text{ini}}$ and values $t_j^{0,\text{ini}}$ and $\pi_j^{0,\text{ini}}$ for $j \in \{1, ..., N^{0,\text{ini}}+1\}$ in the optimal discrete state and event sequence $\varrho^{0,\text{ini}} = \{0, ..., N^{0,\text{ini}}\}$. Set iteration counter $\rho := 0$.

Step 1 **Optimization:** Run Algorithm 6.1 starting from current trajectories $\upsilon^{\rho,\text{ini}}$ and $\mu^{\rho,\text{ini}}$, and values $t_j^{\rho,\text{ini}}$ and $\pi_j^{\rho,\text{ini}}$ with $x(t_0^\rho)$, $\varrho^{\rho,\text{ini}}$, t_0^ρ, and t_e^ρ until the optimality conditions are satisfied with the desired accuracy ϵ_Ω.

Step 2 **Control:** Apply the control υ^ρ for the interval $[t_0^\rho, t_0^\rho + T_s)$ to the system. Additionally, a local, feedback-feedforward control $\delta u^\rho = -F_0^\rho - F_1^\rho \delta x - F_2^\rho d\bar{p} - F_3^\rho dt_e$ is applied to increase the robustness against disturbances and support the gradient method in approaching an optimal solution.

Step 3 **Shift Forward:** The time interval $[t_0^\rho, t_e^\rho]$ and initial position x_0^ρ are shifted to $[t_0^{\rho+1}, t_e^{\rho+1}]$ and $x_0^{\rho+1}$, such that initial values $\upsilon^{\rho+1,\text{ini}}$, $\mu^{\rho+1,\text{ini}}$, $t_j^{\rho+1,\text{ini}}$, $\pi_j^{\rho+1,\text{ini}}$, $\varrho^{\rho+1,\text{ini}}$ for iteration $\rho+1$ are provided. Set $\rho := \rho + 1$ and go to Step 1. □

Control Step The control applied in the time interval $[t_0^\rho, t_0^\rho + T_s)$ consists of the open-loop control υ^ρ and the closed-loop control $\delta\upsilon^\rho$. This structure is well-suited for disturbance rejection. The closed-loop part $\delta\upsilon^\rho$ allows us to increase the sample time T_s or to apply the MPC algorithm to systems with faster dynamics compared to MPC with purely open-loop control υ^ρ. The first 3 terms of the closed-loop control

$$\delta u^\rho(t) = -F_0^\rho(t) - F_1^\rho(t)\delta x(t) - F_2^\rho(t) d\bar{p} - F_3^\rho(t) dt_e \tag{6.100}$$

for $t \in [t_0, t_e]$ are derived from inserting (6.35) and (6.86) into (6.34). Here, it is important to use the full Newton steps, i.e. $\delta H_{j,u}^\rho = -H_{j,u}^\rho$ etc., such that the remaining error ϵ_Ω in the optimality conditions is decreased as much as possible. The required Riccati-matrices and vectors are taken from the optimization step, such that no additional integration is necessary and the closed-loop control is calculated in real-time. Without the term $F_3^\rho dt_e$ with $dt_e := t - t_0^\rho$, the prediction horizon $t_e^\rho - t$ would decrease, which would degrade the quality of the MPC and lead to jumps in the control, when the prediction horizon jumps back to T_p at time $t_0^{\rho+1}$. To largely reduce those jumps and to maintain the real-time capabilities of the closed-loop control, the changes caused by the moving final time are linearly approximated. The procedure is the following: The optimal control problem is solved in the optimization step as if t_e^ρ was a free final time. Then, the row in $R_{j,2}^{\rho,T}$, $Q_{j,2}^{\rho,T}$, $Q_{j,3}^\rho$, $W_{j,3}^\rho$, and $\eta_{j,2}^\rho$ corresponding to $dz_{N^\rho+1}$ and the column in $Q_{j,3}^\rho$ corresponding to dt_e are canceled. The canceled row in $R_{j,2}^{\rho,T}$ is stored in $b_1^{\rho,T}$ and the canceled columns in $Q_{j,2}^\rho$ and $Q_{j,3}^\rho$ in b_2^ρ. This leads to

$$F_3^\rho(t) = E_{j,1}^\rho(t) f_{j,u}^{\rho,T} \left(b_1^\rho(t) - \left(R_{j,1}^\rho(t)\; R_{j,2}^\rho(t)\right) \begin{pmatrix} Q_{0,1}^\rho(t_0) & Q_{0,2}^\rho(t_0) \\ Q_{0,2}^{\rho,T}(t_0) & Q_{0,3}^\rho(t_0) \end{pmatrix}^{-1} b_2^\rho(t_0) \right). \tag{6.101}$$

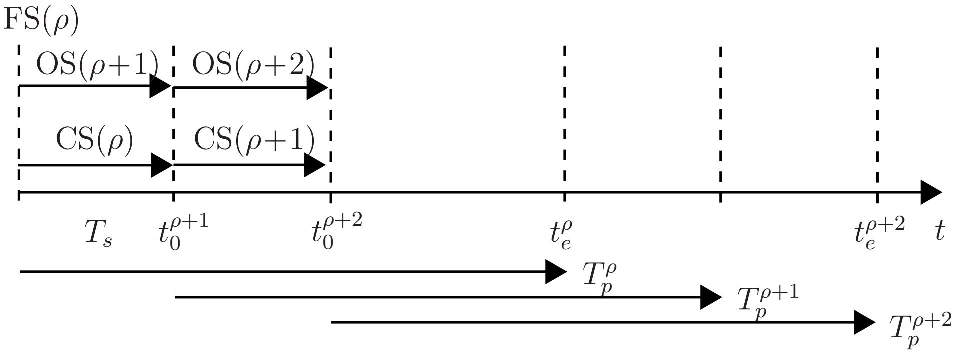


Fig. 6.1: Time flow of the numerical implementation of the MPC algorithm: OS(ρ): Optimization step for iteration ρ, CS(ρ): Control step in iteration ρ, FS(ρ): Shift forward step for iteration $\rho + 1$.

Shift Forward Step In this step, the initialization for the optimization step in iteration $\rho+1$ is performed. The initial and final times are set to $t_0^{\rho+1} := t_0^{\rho} + T_s$ and $t_e^{\rho+1} := t_e^{\rho} + T_s$. The part of υ^{ρ} and μ^{ρ} in interval $[t_0^{\rho}, t_0^{\rho+1})$ is discarded for determining $\upsilon^{\rho+1}$ and $\mu^{\rho+1}$ from υ^{ρ} and μ^{ρ}. In the additional interval $(t_e^{\rho}, t_e^{\rho+1}]$, $\upsilon^{\rho+1}$ and $\mu^{\rho+1}$ are set to $u^{\rho}(t_e^{\rho})$ and $\mu^{\rho}(t_e^{\rho})$, which delivered good results in simulations. The optimization step for iteration $\rho + 1$ is started at time t_0^{ρ}, see Fig. 6.1, such that $x_0^{\rho+1}$ cannot be measured and $\bar{p}^{\rho+1}$ cannot be estimated. Therefore, $x_0^{\rho+1}$ and $\bar{p}^{\rho+1}$ are predicted in the shift forward step.

Discrete Sequence Changes Until now, the discrete state and event sequence was assumed to be fixed to prove the convergence of the MPC algorithm. However, especially in MPC, changes in the discrete sequence are likely since considered events will drop out of and new events will enter the moving horizon T_p. Only discrete sequence changes at the beginning or at the end of the moving horizon are considered here. If Assumption 6.3 about the local convexity is still valid in the case of a new event entering the moving horizon and if the new event has just entered the moving horizon, then it is possible (without proof) by a set of rules to find a locally optimal control and discrete event sequence with Algorithm 6.2. In the simulations in Sec. 6.2.4, the rules worked fine. If the algorithm does not converge either the sample time T_s is too large or the optimal solution jumps such that the local convexity assumption is violated. In the latter case, a close to optimal solution as reinitialization for the MPC algorithm has to be recalculated with an optimal discrete state and event sequence, e.g. by the optimal control algorithm proposed in Sec. 5.2.

6.2.4 Overtaking Maneuver of an Autonomous Vehicle

The MPC algorithm is applied to a numerical example to show that the method is able to solve a complex nonlinear HOCP with several changes in the sequence of discrete states and constraints. The purpose is not to illustrate that it is capable to find optimal solutions online if the complexity of the HOCP is sufficiently low. Here, the optimal control problem is hybrid due to the cost functions and not due to the system dynamics. The task is to

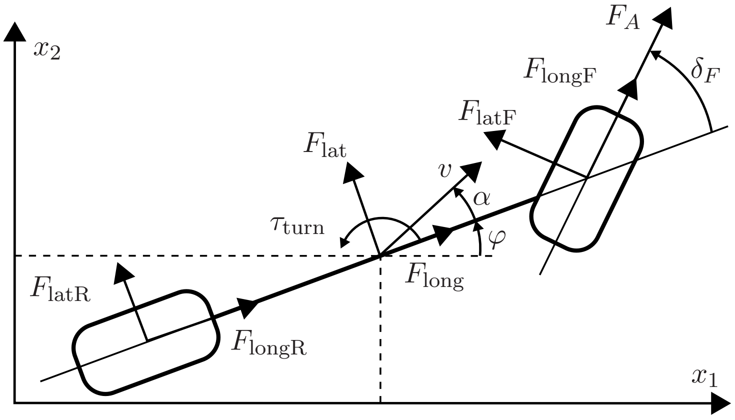


Fig. 6.2: Single track model of the autonomous car with states, controls, and forces.

find optimal trajectories for an autonomous car with single track dynamics [158]

$$\dot{x}_1 = v\cos(\alpha+\varphi) \tag{6.102}$$

$$\dot{x}_2 = v\sin(\alpha+\varphi) \tag{6.103}$$

$$\dot{v} = \frac{1}{m}(F_{\text{long}}\cos\alpha + F_{\text{lat}}\sin\alpha) \tag{6.104}$$

$$\dot{\alpha} = \frac{1}{mv}(-F_{\text{long}}\sin\alpha + F_{\text{lat}}\cos\alpha) - \omega_{\text{turn}} \tag{6.105}$$

$$\dot{\varphi} = \omega_{\text{turn}} \tag{6.106}$$

$$\dot{\omega}_{\text{turn}} = \frac{1}{J_z}\tau_{\text{turn}}\,, \tag{6.107}$$

where x_1 and x_2 are the position of the car, $v > 0$ its velocity, α the angle between the velocity and the longitudinal axis, φ the orientation, ω_{turn} the turn rate, F_a the acceleration force, δ_F the steering angle, m the mass, and J_z the rotational inertia, see Fig. 6.2. The forces and the torque

$$F_{\text{long}} = F_{\text{longR}} + F_{\text{longF}}\cos\delta_F - F_{\text{latF}}\sin\delta_F \tag{6.108}$$

$$F_{\text{lat}} = F_{\text{latR}} + F_{\text{longF}}\sin\delta_F + F_{\text{latF}}\cos\delta_F \tag{6.109}$$

$$\tau_{\text{turn}} = -F_{\text{latR}}l_R + (F_{\text{longF}}\sin\delta_F + F_{\text{latF}}\cos\delta_F)l_F \tag{6.110}$$

are acting on the center of gravity of the car with l_F and l_R denoting the distance between center and front and rear axle. The longitudinal forces

$$F_{\text{longF}} = F_A - \frac{k_R m\bar{g}}{2} \tag{6.111}$$

$$F_{\text{longR}} = -\frac{k_R m\bar{g}}{2} - \frac{c_w A_c \varsigma}{2}v^2 \tag{6.112}$$

contain the acceleration force F_A and the roll and air resistance. Here, k_R is the rolling resistance coefficient, $\bar{g}$ the gravitational acceleration, ς the air density, c_w the drag coefficient, and A_c the reference area of the car. The lateral forces F_{latF} and F_{latR} originate from a small angle approximation of the 'magic formula tire model' from [121]:

$$F_{\text{latF}} = C_F\big(-\alpha + \delta_F - \frac{l_F\omega_{\text{turn}}}{v}\big) \tag{6.113}$$

$$F_{\text{latR}} = C_R\big(-\alpha + \frac{l_R\omega_{\text{turn}}}{v}\big) \tag{6.114}$$

with front and rear cornering stiffnesses C_F and C_R. The sum of forces acting on a tire is constrained by 'Kamm's circle' [196], which is the maximal static tire to road friction $\mu_f\frac{m\bar{g}}{2}$ with friction coefficient μ_f:

$$F_{\text{longF}}^2 + F_{\text{latF}}^2 \leq \big(\mu_f\frac{m\bar{g}}{2}\big)^2 \tag{6.115}$$

$$F_{\text{longR}}^2 + F_{\text{latR}}^2 \leq \big(\mu_f\frac{m\bar{g}}{2}\big)^2\,. \tag{6.116}$$

However, these mixed state and control constraints are neglected in the following optimization. The reasons are twofold: First, the constraints increase the complexity for the MPC algorithm drastically. Second, no routine is implemented that steers the optimization back to feasible solutions if in some part of the optimization, no control exists that satisfies the control constraints introduced below and the mixed state and control constraints at the same time. The control constraints are:

$$F_{\text{minA}} \leq F_A \leq F_{\text{maxA}} \tag{6.117}$$

$$\delta_{\text{minF}} \leq \delta_F \leq \delta_{\text{maxF}}\,. \tag{6.118}$$

The autonomous car A has to drive on the right lane and overtake a car B driving ahead if this leads to lower costs. Car B drives with constant speed v_B. The discrete decision whether to overtake or not is modeled by dividing the road into two discrete states with different cost functions. Discrete state 1 is formed such that if the center point of the car is in the corresponding region, then the complete car is on the right lane. The switching manifold

$$m_{1,2}(x(t)) = x_2(t) - 0.5w_c \tag{6.119}$$

separates the two discrete states and it is illustrated in Fig. 6.3. The width of a car is denoted by w_c and the width of a lane by w_r. To avoid a collision, car A has to obey the length l_c and width w_c of car B and additionally a safety margin l_{safe} in longitudinal direction and w_{safe} in lateral direction. Since car A can only overtake on the left side of car B, the right road boundary and the safety region are combined to the right road constraint $h_R(x(t),t)$. For the optimization, the combination is created smoothly with a raised cosine function over the length l_{tr} of the transition region. For a concise notation, the dependence on time t is omitted in the following equation and the difference $x_1(t) - x_1^B(t)$ is written as Δx_1. The left and right boundaries of the road are

$$h_L(x(t),t) = x_2 - w_r + 0.5w_c \tag{6.120}$$

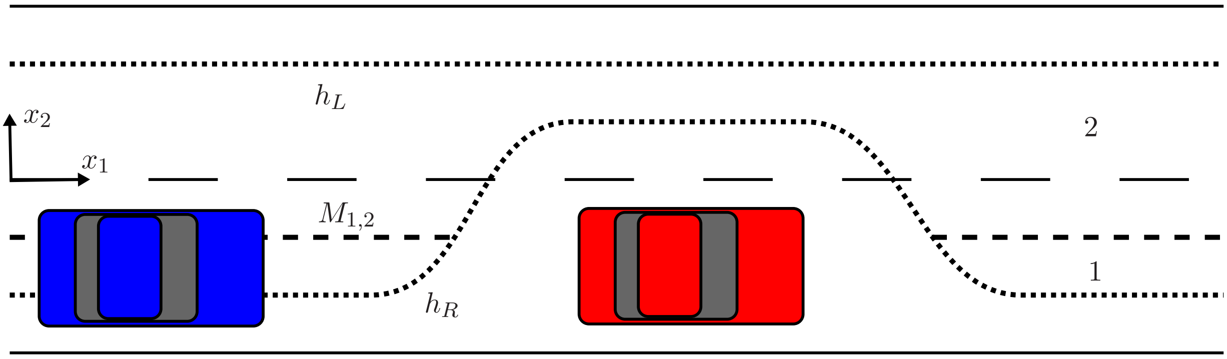


Fig. 6.3: Road with right and left boundaries and a separation in discrete states 1 and 2. If optimal, the following car overtakes the preceding car while considering the constraints.

$$h_R(x(t),t) = \begin{cases} k_1 - x_2 & \text{if} \quad \Delta x_1 \leq -l_l - l_{\text{tr}} \\ \frac{k_2+k_1}{2} + \frac{k_2-k_1}{2}\cos\left(\pi\frac{\Delta x_1 + l_l}{l_{\text{tr}}}\right) - x_2 & \text{if} \quad -l_l - l_{\text{tr}} < \Delta x_1 \leq -l_l \\ k_2 - x_2 & \text{if} \quad -l_l < \Delta x_1 \leq l_l \\ \frac{k_2+k_1}{2} + \frac{k_2-k_1}{2}\cos\left(\pi\frac{\Delta x_1 - l_l}{l_{\text{tr}}}\right) - x_2 & \text{if} \quad l_l < \Delta x_1 \leq l_l + l_{\text{tr}} \\ k_1 - x_2 & \text{if} \quad l_l + l_{\text{tr}} < \Delta x_1 \end{cases} \tag{6.121}$$

with $l_l = l_c + l_{\text{safe}}$, $k_1 = -w_r + 0.5w_c$, $k_2 = x_2^B + w_c + w_{\text{safe}}$, $x_1^B(t) = x_1^B(t_0) + v^B(t - t_0)$, and $x_2^B(t) = x_2^B(t_0)$. The state constraints $h_R(x(t),t) \leq 0$ and $h_L(x(t),t) \leq 0$ are shown with the lines $h_R(x(t),t) = 0$ and $h_L(x(t),t) = 0$ in Fig. 6.3. The running cost functions are chosen to

$$\phi_1(x(t),u(t)) = \frac{1}{2}\big(10\delta_F^2(t) + 10^{-6}F_A^2(t)\big) - v(t) \tag{6.122}$$

$$\phi_2(x(t),u(t)) = \frac{1}{2} + m_{1,2}(x(t)) + \phi_1(x(t),u(t)), \tag{6.123}$$

which is a compromise between driving far and applying low control magnitudes. The cost function in discrete state 2 is set up, such that the costs are higher than in discrete state 1 to ensure that the car is usually driving on the right lane. Furthermore, ϕ_2 is such that the costs increase with the distance to the switching manifold to push the car back to the right lane after it has passed the obstacle. Note that the switching manifold $m_{1,2}(x(t))$ is positive if $x(t)$ is in discrete state 2.

Choosing real-world values for the coefficients, see Sec. A.3, and a prediction horizon $T_p = 1.5$sec, the autonomous car A overtakes the preceding car B. In the first graphic in Fig. 6.4, a situations is shown, where an event happens. In this case, a state constraint is hit at the end of the prediction horizon, which was not reached in the preceding iteration. For a better overview, the coordinate system moves with car B, which means that the traveled distance $v_B t$ of car B with $t_0 = 0$ is subtracted from the position $x_1(t)$ of car A in the figures. The short, solid line corresponds to the predicted trajectory of car A. The following graphics, except the last one, present all situation that appear during simulations where an event happens. Illustrated events include a change in the sequence of discrete states at the end or the beginning of the prediction horizon, hitting a state constraint at the end of the prediction horizon, or moving along a state constraint for the first time. In the last graphic, the complete trajectory of car A in a time span of 6sec illustrates the successful application of Algorithm 6.1 to the overtaking scenario. The sampling time of the MPC algorithm is $T_s = 0.1$sec. The computations were executed on a 3 GHz processor using Matlab for simulation and optimization and C++ for automatic differentiation. The computations took 127sec and most of the computational time was spent in the optimization steps, when the changes in the discrete event sequence from one iteration to the next occurred.

In [88], a solution to a similar HOCP is computed in 4.3sec - 168sec depending on the accuracy. Considering our prediction horizon, the repetitive solution of the HOCP in the MPC algorithm, and that the entire problem is solved at once without applying MPC in [88], the computational times of the two optimal control algorithms are in a comparable range. The algorithm in [88] is successfully applied to the real-time control of several other example systems with less complex dynamics. Though a comparison is difficult because of the different HOCPs, it appears that the computational times in the real-time algorithm in [79] are also in a similar range compared to the computational times here and in [88].

6.2.5 Discussion

A model predictive control algorithm for hybrid systems with autonomous switching and state and control constraints is presented. It provides fast updates based on a gradient method of second order using neighboring extremals. The algorithm was successfully applied to a complex numerical example with changes in the sequence of discrete states and active state constraints.

The MPC algorithm is initialized by running the second-order gradient methods iteratively on increasing time intervals until the desired length of the prediction horizon is reached. This procedure is required for a successful initialization since the domain of convergence of the method is small. Alternatively, the initialization could be performed by the optimal solution of a different optimal control algorithm. The initialization from one iteration of the MPC algorithm to the next consists of the solution from the previous iteration and an extrapolation for adapting to the receding prediction horizon and the new switching times.

The proposed algorithm is especially suitable for the practical application to HOCPs with model uncertainties for three reasons: (i) By the model predictive control approach, the computations are repeated after the time step T_s on a moved horizon, which approximates a feedback control. (ii) At each sampling time T_s, the algorithm explicitly adapts the optimal solution to updates of the model parameters provided by a an estimator running in parallel. (iii) In between two sampling times, the MPC algorithm applies an optimal open-loop control and additionally an optimal feedback control. The optimal feedback control also improves the solution in the case that the optimization has been stopped before the optimal solution was reached with the desired accuracy.

The gradient method of second order has a quadratic convergence rate in a sufficiently small neighborhood around the optimal solution. There, the approximation of the HOCP with a linear-quadratic HOCP, that is used by the gradient method, is close to reality and therefore the convergence rate is high. In practice, the method also converges fast

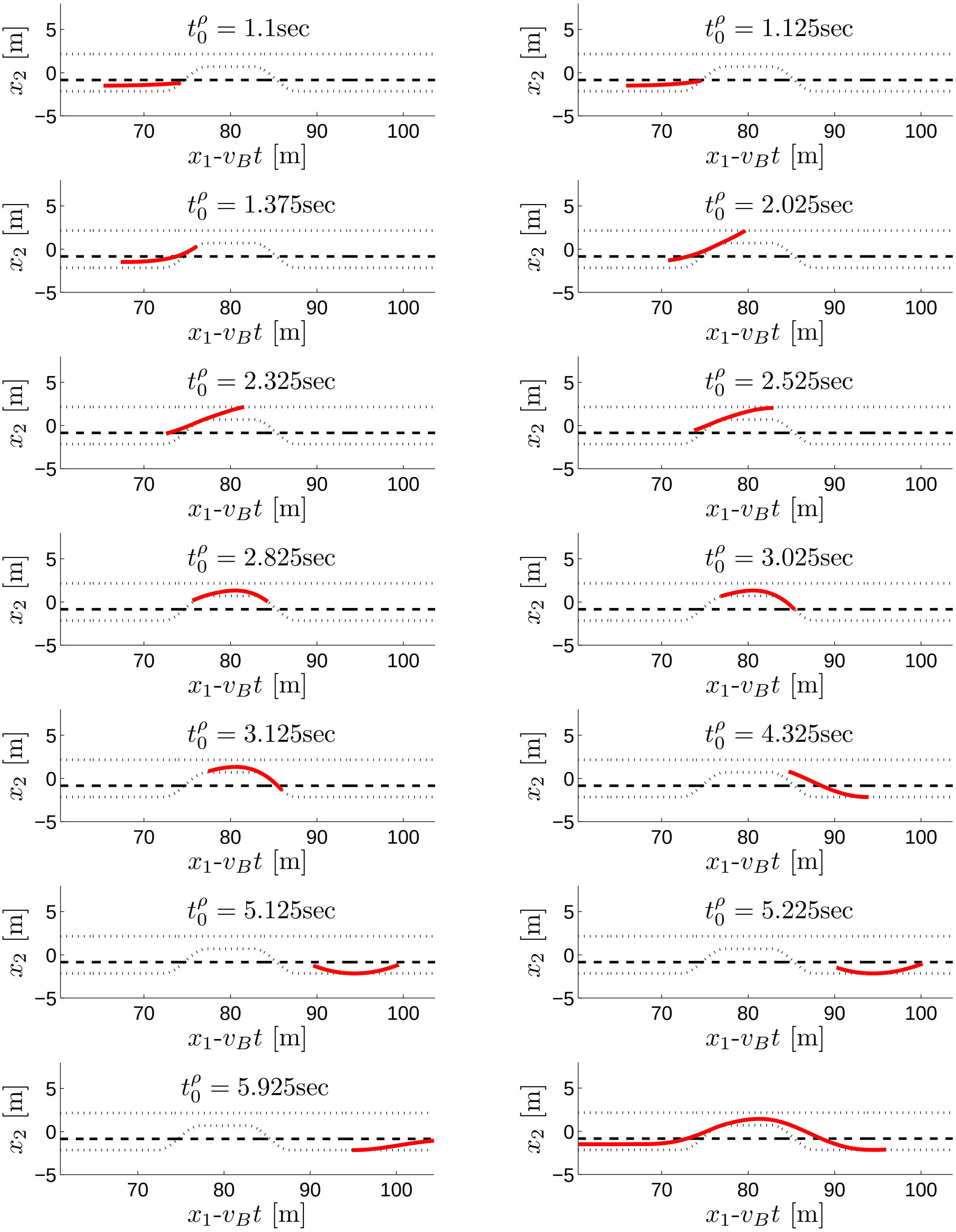


Fig. 6.4: In all graphics, except for the last one, the optimal trajectory for the prediction horizon is illustrated at an iteration of the MPC algorithm, where an event occurs that has not occurred in the iteration before. The types of events are hitting and moving along a state constraint, and a change in the sequence of discrete states. In the last graphic, the entire driven trajectory of the autonomous car is shown.

for small sampling times T_s and small deviations $\delta x(t_0)$ from the initial conditions of the HOCP in the previous iteration. For larger sampling times T_s and larger deviations $\delta x(t_0)$, however, the convergence rate of the gradient method degrades and it even may happen that convergence is not achieved. In this case, the initialization for this iteration is not in a valid domain of the linearization and the HOCP is not convex in this region. If such a case is observed in the computations, the algorithm steps back, applies a reduced sampling step size T_s, and repeats the optimization. This procedure is especially necessary when new events occur at the end of the prediction horizon in order to find an optimal solution. In real-time, such a procedure is not possible. The nonlinearity of the model equations of the autonomous car and the complexity of the HOCP in general require sampling times around $T_s = 0.1$sec for a successful application. At event times, the sampling time T_s has to be reduced to achieve convergence, sometimes by more than an order of magnitude.

In the numerical example here, real-time computations are not yet achieved and the majority of the computational time is spent in the cases, in which new events occur. It depends on the computing power and the complexity of the HOCP including the desired sampling time T_s and the nonlinearity of the system dynamics, if calculations can be finished in real-time.

In order to reduce the computational load and increase the domain of convergence in future, a gradient method of first order can be combined with the currently used gradient method of second order. Alternatively, elements from the min-H algorithm in Sec. 4.3 can be included. It is expected that this increases the stability and convergence rate for changes of the event sequence. Furthermore, the implementation should be streamlined for fast computations, e.g. keeping the Riccati-matrices on the same size during the complete optimization time interval $t_e - t_0$ as in Sec. 4.3, replacing the quasi-continuous integrations by discrete-time ones, considering symmetries for integrations, and adapt the implementation for parallel computations. Additionally, the complete program can be implemented in C++, which will reduce the computational time for major parts of the program that are currently written in Matlab.

6.3 Trajectory Planning based on the Optimal Concatenation of Optimal Control Primitives

In the following, an online capable algorithm for the solution of HOCPs based on optimal control primitives is presented. For a numerical example, the approach is combined with a sampling-based motion planner, which delivers interior point constraints for guiding a robotic manipulator around obstacles. The section is organized as follows: Sec. 6.3.1 specifies the HOCP under consideration. Sec. 6.3.2 introduces parametric control primitives and how they are used to solve HOCPs. In Sec. 6.3.3, the primitives are used to find optimal trajectories for robotic manipulators. The planning scheme for collision-free trajectories of a manipulator is described in Sec. 6.3.4 and evaluated in Sec. 6.3.5.

6.3.1 Hybrid Optimal Control Problem

The hybrid system considered in this section includes continuous and discrete states, continuous controls, continuous and discrete dynamics, and switching manifolds for autonomous switching. Definition 2.38 and Assumption 2.39 are applied in this case with the difference that the set of allowed control values U_q is unbounded and $\mathcal{U}$ is the set of piecewise continuous control functions $\upsilon : [t_0, t_e] \to U$. The sequence of discrete states is assumed to be fixed: $\varrho = (0, 1, ..., N)$. The HOCP is specified as in Definition 2.40 with terminal and running costs and interior and terminal constraints.

For the moment, it is assumed that the dynamics, switching manifolds, and constraints are linear. Additionally an output $y \in \mathbb{R}^{n_y}$ is defined:

$$\dot{x}(t) = A_q x(t) + B_q u_q(t) \tag{6.124}$$

$$y(t) = O_q x(t)\,. \tag{6.125}$$

Here, interior constraints are combined with the switching manifolds in one expression

$$\begin{pmatrix} m_{j-1,j}(x(t_j)) \\ \psi_{j-1}(x(t_j)) \end{pmatrix} := O_{j-1} x(t_j) - y_j = 0 \tag{6.126}$$

for $j \in \{1, ..., N\}$ and terminal constraints are defined as

$$\psi_e(x(t_e)) := O_e x(t_e) - y_e = 0\,. \tag{6.127}$$

The terminal and running costs are:

$$g(x(t_e)) = \frac{1}{2}\,\|x(t_e) - x_e\|^2_{W_e} \tag{6.128}$$

$$\phi_j(x(t), u(t)) = \frac{1}{2}\,\begin{pmatrix} x(t) & u(t) \end{pmatrix} \begin{pmatrix} C_j & V_j \\ V_j^T & Z_j \end{pmatrix} \begin{pmatrix} x(t) \\ u(t) \end{pmatrix}\,, \tag{6.129}$$

where $\|x(t_e) - x_e\|^2_{W_e} = (x(t_e) - x_e)^T W_e (x(t_e) - x_e)$, $V_j \in \mathbb{R}^{n_x \times n_u}$, the matrices $W_e, S_j \in \mathbb{R}^{n_x \times n_x}$ are symmetric and positive semi-definite, $Z_j \in \mathbb{R}^{n_u \times n_u}$ is symmetric and positive definite. Applying calculus of variations [32], the optimality conditions for the specified HOCP are derived:

$$\dot{x}(t) = A_j x(t) + B_j u(t) \tag{6.130}$$

$$\dot{\lambda}(t) = -C_j x(t) - V_j u(t) - A_j^T \lambda(t) \tag{6.131}$$

$$0 = Z_j u(t) + V_j^T x(t) + B_j^T \lambda(t) \tag{6.132}$$

$$\lambda(t_e) = W_e x(t_e) - W_e x_e + O_e^T \nu_e \tag{6.133}$$

$$\lambda(t_j^-) = \lambda(t_j) + \nabla_x \begin{pmatrix} m_{j-1,j}(x(t_j)) \\ \psi_{j-1}(x(t_j)) \end{pmatrix}^T \pi_j \tag{6.134}$$

$$H_{j-1}(x(t_j^-), \lambda(t_j^-), u_{j-1}(t_j^-)) = H_j(x(t_j), \lambda(t_j), u_j(t_j)) \tag{6.135}$$

$$x(t_0) = x_0 \tag{6.136}$$

with the constant multiplier $\nu_e \in \mathbb{R}^{n_y}$. The control in the state and adjoint differential equations is replaced by the expression

$$u(t) = -Z_j^{-1}\big(V_j^T x(t) + B_j^T \lambda(t)\big) . \tag{6.137}$$

This leads to the following reformulation of the state and adjoint differential equations:

$$\begin{pmatrix} \dot{x}(t) \\ \dot{\lambda}(t) \end{pmatrix} = \Xi_j \begin{pmatrix} x(t) \\ \lambda(t) \end{pmatrix} , \tag{6.138}$$

where Ξ denotes the Hamiltonian matrix

$$\Xi_j = \begin{pmatrix} A_j - B_j Z_j^{-1} V_j^T & -B_j Z_j^{-1} B_j^T \\ -C_j + V_j Z_j^{-1} V_j^T & V_j Z_j^{-1} B_j^T - A_j^T \end{pmatrix} .$$

6.3.2 Solution based on Optimal Control Primitives

The HOCP posed above is solved in the following with a parametric solution. The parametric solution is globally optimal and large parts can be computed offline, such that only few algebraic computations need to be carried out online. The method is divided into two parts. The purely continuous optimal control subproblems in between switching points or interior point constraints are solved by optimal control primitives offline. Online the primitives are evaluated with the relevant parameters and the primitives are optimally concatenated.

Primitives for Purely Continuous Optimal Control Problems

In [52], no autonomous switching and no interior point constraints are considered. There, it is shown that the solution of purely continuous optimal control problems with equations as in (6.138) can be expressed by a stabilizing and anti-stabilizing solution with the two parameter vectors $\tilde{\eta}_j \in \mathbb{R}^{n_x}$ and $\tilde{\rho}_j \in \mathbb{R}^{n_x}$. In the following, the solution derived in [52] is shortly presented. The assumptions are that (A_j, B_j) is controllable for $j \in \{0, ..., N\}$ and Ξ_j has no eigenvalues on the imaginary axis. The parametric state and adjoint trajectories are then defined as

$$\begin{pmatrix} x(t) \\ \lambda(t) \end{pmatrix} = \begin{pmatrix} I \\ S_{\oplus,j} \end{pmatrix} e^{A_{\oplus,j} t} \tilde{\eta}_j + \begin{pmatrix} I \\ S_{\ominus,j} \end{pmatrix} e^{A_{\ominus,j}(t-t_e)} \tilde{\rho}_j , \tag{6.140}$$

where I is the identity matrix of size $n_x \times n_x$. The matrix $S_{\oplus,j} \in \mathbb{R}^{n_x \times n_x}$ denotes the positive semi-definite and symmetric maximal solution and $S_{\ominus,j} \in \mathbb{R}^{n_x \times n_x}$ denotes the negative semi-definite and symmetric minimal solution of the continuous-time algebraic Riccati equation (CARE)

$$S_j A_j + A_j^T S_j - \big(V_j + S_j B_j\big) Z_j^{-1} \big(V_j^T + B_j^T S_j\big) + C_j = 0 . \tag{6.141}$$

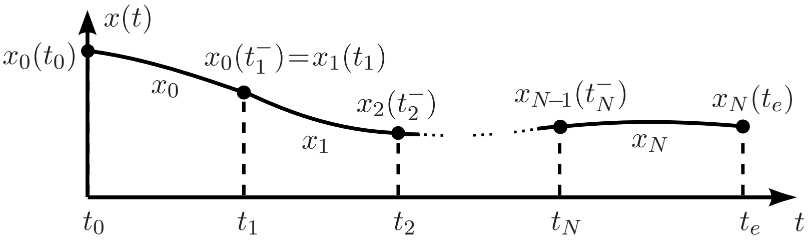


Fig. 6.5: Concatenation of $N+1$ trajectories.

With the two extremal solutions, the closed-loop system matrices $A_{k,j} \in \mathbb{R}^{n_x \times n_x}$, $k \in \{\oplus, \ominus\}$ are

$$A_{k,j} = A_j - B_j K_{k,j} \ . \tag{6.142}$$

The gain matrices $K_{k,j} \in \mathbb{R}^{n_u \times n_x}$ are defined as

$$K_{k,j} = Z_j^{-1}\big(V_j^T + B_j^T S_{k,j}\big) \ , \tag{6.143}$$

and the eigenvalues of $A_{\oplus,j}$ and $A_{\ominus,j}$ are stable and anti-stable, respectively. The main idea of the parametric solution is to combine the stabilizing solution $S_{\oplus,j}$ and anti-stabilizing solution $S_{\ominus,j}$ of the CARE to achieve an optimal point-to-point movement from an initial to a desired final point. The resulting parameterized optimal control law is

$$u_j^*(t) = -K_{\oplus,j} e^{A_{\oplus,j} t} \tilde{\eta}_j - K_{\ominus,j} e^{A_{\ominus,j}(t-t_e)} \tilde{\rho}_j \ , \tag{6.144}$$

which we define as a control primitive for linear-quadratic optimal control problems.

Optimal Concatenation of Primitives

The optimal control primitives (6.144) solve purely continuous optimal control problems. Hereafter, an approach is introduced that combines several optimal control primitives optimally, such that the considered HOCP with autonomous switching and interior point constraints is solved.

From one autonomous switching to the next, the optimal solution is given in a parametrized form by an optimal control primitive

$$u_j^* = -K_{\oplus,j} e^{A_{\oplus,j}(t-t_j)} \tilde{\eta}_j - K_{\ominus,j} e^{A_{\ominus,j}(t-t_{j+1})} \tilde{\rho}_j \ . \tag{6.145}$$

For a clear notation, the parts of the continuous state trajectory with constant discrete state $j \in \{0, ..., N\}$ are denoted x_j, compare Fig. 6.5. To determine the optimal parameters of each primitive, it is required to consider the entire HOCP. The optimal parameters are found by evaluating the optimality conditions (6.133) - (6.136). In the following, these optimality conditions are analyzed with respect to optimal control primitives.

To solve the HOCP optimally with the optimal control primitives, there are $2n_x(N+1)$ unknown parameters $\tilde{\eta}_j$ and $\tilde{\rho}_j$ and N unknown switching times t_j to be determined. The $2n_x(N+1)+N$ conditions that specify the unknown parameters and switching times are described below. The optimal state trajectories have to satisfy the initial state constraints (6.136) and the terminal state and adjoint constraints (6.127) and (6.133), which are in total $2n_x$ conditions. Further, it is assumed that $n_{\psi_{j-1}}$ interior point constraints ψ_{j-1} with $0 \leq n_{\psi_{j-1}} < n_x$ have to be fulfilled at the j-th autonomous switching. This means that the continuous state $x_{j-1}(t_j^-)$ has to satisfy the $n_{\psi_{j-1}}+1$ conditions (6.126) at the switching times t_j. The continuous state must be continuous at autonomous switching times:

$$x_{j-1}(t_j^-) = x_j(t_j)\,. \tag{6.146}$$

The adjoint transversality conditions (6.134) have to be considered, which deliver $n_x - n_{\psi_{j-1}} - 1$ additional conditions. The N conditions for the optimal switching times t_j are given by the Hamiltonian continuity conditions (6.135), which are quadratic in the parameters $\tilde{\eta}_j$ and $\tilde{\rho}_j$ and exponential in the switching times t_j. This set of equations can be solved efficiently by standard optimization solvers, such that an online solution is usually possible.

In the case, that the switching times t_j are fixed beforehand, the Hamiltonian continuity conditions (6.135) do not have to be considered. This implies that only a set of linear equations has to be solved to find the optimal parameters $\tilde{\eta}_j$ and $\tilde{\rho}_j$. In general, these computations can be solved online. In the following, the set of linear equations is derived for one switching time t_1, n_{ψ_0} interior point constraints ψ_0, and n_x terminal constraints ψ_e. For a concise notation, the abbreviations $E_{\oplus,j} = e^{A_{\oplus,j}(t_{j+1}-t_j)}$ and $E_{\ominus,j} = e^{-A_{\ominus,j}(t_{j+1}-t_j)}$ are introduced. Inserting (6.140) in the boundary conditions (6.136) and (6.127), the initial and terminal conditions are:

$$x_0(t_0) = x_0 = \tilde{\eta}_0 + E_{\ominus,0}\tilde{\rho}_0 \tag{6.147}$$

$$x_e(t_e) = y_e = O_e E_{\oplus,1}\tilde{\eta}_1 + O_e\tilde{\rho}_1\,. \tag{6.148}$$

At the switching time t_j, the combined autonomous switching condition and interior point condition (6.126) and the continuity condition (6.146) of the continuous state lead to the equations:

$$y_1 = O_0 E_{\oplus,0}\tilde{\eta}_0 + O_0\tilde{\rho}_0 \tag{6.149}$$

$$0 = E_{\oplus,0}\tilde{\eta}_0 + \tilde{\rho}_0 - \tilde{\eta}_1 - E_{\ominus,1}\tilde{\rho}_1\,. \tag{6.150}$$

Consider the matrix $\bar{O}_0$ with dimension $(n_x - n_{\psi_0} - 1) \times n_x$ to be of rank $n_x - n_{\psi_0} - 1$ and to satisfy rank $\begin{pmatrix} O_0 \\ \bar{O}_0 \end{pmatrix} = n_x$. The adjoint transversality conditions yield the last $n_x - n_{\psi_0} - 1$ conditions

$$0 = \bar{O}_0\big((S_{\oplus,0}E_{\oplus,0}\tilde{\eta}_0 + S_{\ominus,0}\tilde{\rho}_0) - (S_{\oplus,1}\tilde{\eta}_1 + S_{\ominus,1}E_{\ominus,1}\tilde{\rho}_1)\big)\ , \tag{6.151}$$

The resulting system of linear equations is:

$$\begin{pmatrix} x_0 \\ y_e \\ y_1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} I & E_{\ominus,0} & 0 & 0 \\ 0 & 0 & O_e E_{\oplus,1} & O_e \\ O_0 E_{\oplus,0} & O_0 & 0 & 0 \\ E_{\oplus,0} & I & -I & -E_{\ominus,1} \\ \bar{O}_0 S_{\oplus,0} E_{\oplus,0} & \bar{O}_0 S_{\ominus,0} & -\bar{O}_0 S_{\oplus,1} & -\bar{O}_0 S_{\ominus,1} E_{\ominus,1} \end{pmatrix} \begin{pmatrix} \tilde{\eta}_0 \\ \tilde{\rho}_0 \\ \tilde{\eta}_1 \\ \tilde{\rho}_1 \end{pmatrix}\,. \tag{6.152}$$

When the choice of initial, interior and terminal constraints is bounded and reachable for the system (6.130), the system of linear equations (6.152) can be solved uniquely and the optimal parameters $\tilde{\eta}_0$, $\tilde{\rho}_0$, $\tilde{\eta}_1$, and $\tilde{\rho}_1$ are obtained.

In the presented example, the final states were fully assigned and thus the weighting function for the terminal state $g(x(t_e))$ is zero. Otherwise, the weighting function for the terminal state would have been included in the system of linear equations (6.152). In this case, the additional condition $\lambda_e(t_e) = W_e x_e(t_e) - W_e x_e$ from (6.133) has to be considered. With (6.140), the condition is

$$\big(S_{\oplus,1} - W_e\big) E_{\oplus,1} \tilde{\eta}_1 + \big(W_e - S_{\ominus,1}\big) \tilde{\rho}_1 + W_e x_e = 0 \ . \tag{6.153}$$

The condition replaces the second line in the system of equations (6.152).

Optimal Costs

In the following, it is shown how the minimal costs of the optimal solution is computed. The integral over the running costs (6.129) can be analytically solved for a fixed discrete state $j \in \{0, ..., N\}$, see [52]. With the parametric equations (6.140), the integral over running costs (6.129) for the control primitive υ_j^* with $j \in \{0, ..., N-1\}$ is

$$\int_{t_j}^{t_{j+1}} \phi_j \, dt = \begin{pmatrix} \tilde{\eta}_j \\ \tilde{\rho}_j \end{pmatrix}^T \begin{pmatrix} S_{\oplus,j} - E_{\oplus,j}^T S_{\oplus,j} E_{\oplus,j} & S_{\ominus,j} E_{\ominus,j} - E_{\oplus,j}^T S_{\ominus,j} \\ E_{\ominus,j}^T S_{\oplus,j} - S_{\oplus,j} E_{\oplus,j} & E_{\ominus,j}^T S_{\ominus,j} E_{\ominus,j} - S_{\ominus,j} \end{pmatrix} \begin{pmatrix} \tilde{\eta}_j \\ \tilde{\rho}_j \end{pmatrix} . \tag{6.154}$$

The cost of the final, optimal primitive υ_N^* is:

$$g(x(t_e)) + \int_{t_N}^{t_e} \phi_N \, dt = \begin{pmatrix} \tilde{\eta}_N \\ \tilde{\rho}_N \\ x_e(t_e) \end{pmatrix}^T$$
$$\begin{pmatrix} S_{\oplus,N} + E_{\oplus,N}^T (W_e - S_{\oplus,N}) E_{\oplus,N} & S_{\ominus,N} E_{\ominus,N} + E_{\oplus,N}^T (W_e - S_{\ominus,N}) & -E_{\oplus,N}^T W_e \\ E_{\ominus,N}^T S_{\oplus,N} + (W_e - S_{\oplus,N}) E_{\oplus,N} & E_{\ominus,N}^T S_{\ominus,N} E_{\ominus,N} - S_{\ominus,N} + W_e & -W_e \\ -W_e E_{\oplus,N} & -W_e & W_e \end{pmatrix} \begin{pmatrix} \tilde{\eta}_N \\ \tilde{\rho}_N \\ x_e(t_e) \end{pmatrix} . \tag{6.155}$$

If the final state is completely specified, the weighting matrix W_e is zero. The entire costs are the sum over the costs for the individual optimal control primitives.

Optimal Feedback for Optimal Control Primitives

The parametric representation of optimal control primitives υ_j^* is derived from feedback and feedforward matrices. However, when the optimal control primitives are applied to the system, the control is up to now open-loop. Deviations $\delta\chi$ from the optimal state trajectory χ^* due to disturbances or model and parameter uncertainties are not corrected by an optimal, feedback control law. In the following, three possibilities for including (sub)optimal feedback control are presented.

Recalculation of Optimal Solution with MPC The calculation of the optimal parameters is usually fast. Especially in the case of fixed switching times t_j, the computations are usually finished in millisecond, compare Tab. 6.3.5, since only a system of linear equations as in (6.152) has to be solved. Due to the fast computations, it is possible for systems with appropriate dynamics to recalculate the optimal parameters online based on the current state $x(t)$ at the current time t. The recalculation provides an updated optimal solution including optimal controls υ_j^* and optimal switching times t_j and states $x(t_j)$ from the current state $x(t)$ to the final time t_e and possibly state x_e. Executing the recalculations in an MPC framework, an optimal feedback control is approximated. The shorter the sampling time of the MPC algorithm, the better the approximation is as long as the computations can be finished sufficiently fast.

Optimal, Local Feedback Control The second possibility is to create an optimal feedback control that is added to the optimal control primitive $u_j^*(t)$ at time t. The additional control $\delta u_j(t)$ is valid in a neighborhood around the initially calculated, optimal state and adjoint trajectories (6.140). The feedback control relies on a linearization around the state and adjoint trajectories. The linearization leads to a set of Riccati matrices as used in Sec. 6.2. The Riccati-matrices have to be integrated from the final time t_e backwards to the initial time t_0. The integration can be performed offline once the optimal parameters are determined. Online the optimal feedback control is given by

$$\delta u_j^*(t) = -Z_j^{-1}\Big(\big(V_j^T + B_j^T S_j(t)\big)\delta x(t) + B_j^T\big(R_j(t)d\pi_{j+1} + e_j(t)\big)\Big) \tag{6.156}$$

with the corresponding Riccati-matrices $S_j(t)$ and $R_j(t)$ and Riccati-vector $e_j(t)$. If the deviations $\delta x(t)$ are such that the valid domain of the linearization is left, the nominal state and adjoint trajectories and the Riccati-matrices need to be adapted as in Sec. 6.2.

Suboptimal Feedback Control In the case that a recalculation of optimal parameters takes too long and a precalculation or online calculation of optimal feedback control laws is not possible, a suboptimal feedback control law can still be derived. The idea is that the stabilizing solution $S_{\oplus,j}$ of the CARE (6.141) is used for a feedback control

$$\delta u_j(t) = -Z_j^{-1}\Big(V_j^T + B_j^T S_{\oplus,j}\Big)\delta x(t)\,. \tag{6.157}$$

The control law (6.157) can be evaluated online with the already precomputed solution $S_{\oplus,j}$. The feedback control $\delta u_j(t)$ is optimal for purely continuous, linear-quadratic optimal control problems with infinite horizon and without terminal constraints. For the given linear-quadratic HOCP, the control $\delta u_j(t)$ is therefore only suboptimal.

Primitives for Nonlinear Hybrid Optimal Control

Up to now, only HOCPs with linear dynamics are considered. The proposed solution method based on optimal control primitives and their optimal concatenation is extended in the following to nonlinear dynamics of the form:

$$\dot{x}(t) = f_j(x(t)) + \sum_{i=1}^{n_u} p_{j,i}(x(t))\, u_{j,i}(t) \tag{6.158}$$

$$y(t) = o_j(x(t))\,, \tag{6.159}$$

where $p_{j,i}(x(t)) \in \mathbb{R}^{n_x}$ and $u_{j,i}(t)$ is the i-th component of the control vector $u_j(t)$. It is assumed that the dynamics (6.158) is input-to-state feedback linearizable or that the output system (6.158) and (6.159) is input-to-output feedback linearizable depending on which form of feedback linearization is required in the HOCP.

Feedback linearization is shortly presented below with a scalar input $u_{j,i}$ based on [86]. Feedback linearization for multi-input systems is discussed in [159, 167]. If an input-to-state feedback linearization is required, a transformation

$$\bar{x} = \gamma_j(x(t)) \tag{6.160}$$

is needed that transforms the system (6.158) into a controllable canonical form. Here, it is assumed that the transformation exists for a domain of the continuous state space, which contains the optimal state trajectory completely. Additionally, the transformation is assumed to be sufficiently often differentiable for the analysis here with respect to the continuous state $x(t)$. The transformation has to satisfy

$$\gamma(x(t)) = \Big(\gamma_{j,1}(x(t)),\ L_{f_j}\gamma_{j,1}(x(t)),\ L_{f_j}^2\gamma_{j,1}(x(t)),\ ...,\ L_{f_j}^r\gamma_{j,1}(x(t))\Big)^T, \tag{6.161}$$

where $\gamma_{j,1}(x(t))$ is the first component of the vector function $\gamma_j(x(t))$. The Lie derivative is defined as $L_{f_j}\gamma_{j,1}(x(t)) := \nabla_x\gamma_{j,1}(x(t))\, f_j(x(t))$ and $L_{f_j}^k\gamma_{j,1}(x(t)) := L_{f_j}L_{f_j}^{k-1}\gamma_{j,1}(x(t))$ is the recursive application of the Lie derivative for k times. The relative degree $r \in \mathbb{N}_0$ denotes the Lie derivative, where $L_{p_{j,i}}L_{f_j}^k\gamma_{j,1}(x(t))$ with $k \in \{0, ..., r-2\}$ is zero and $L_{p_{j,i}}L_{f_j}^{r-1}\gamma_{j,1}(x(t))$ is not zero for the first time. After the transformation of (6.158) into a controllable canonical form

$$\dot{\bar{x}}(t) = \begin{pmatrix} 0 & 1 & 0 & ... & 0 \\ 0 & 0 & 1 & ... & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & ... & 0 \end{pmatrix} \bar{x}(t) + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \bar{u}_{j,i}(t)\,, \tag{6.162}$$

the linearizing control is

$$u_{j,i}(t) = \frac{\bar{u}_{j,i}(t) - L_{f_j}^r\gamma_{j,1}(x(t))}{L_{p_{j,i}}L_{f_j}^{r-1}\gamma_{j,1}(x(t))} \tag{6.163}$$

and $\bar{u}_{j,i}(t)$ is the control of the transformed system. Next, the HOCP is formulated for the transformed system (6.162), which implies that the cost function is set up for the variables

$\bar{x}(t)$ and $\bar{u}_j(t)$.

$$J = \frac{1}{2}\left\|\bar{x}(t_e) - \bar{x}_e\right\|_{W_e}^2 + \sum_{j=0}^{N} \int_{t_j}^{t_{j+1}} \frac{1}{2} \begin{pmatrix} \bar{x}(t) \\ \bar{u}_j(t) \end{pmatrix}^T \begin{pmatrix} C_j & V_j \\ V_j^T & Z_j \end{pmatrix} \begin{pmatrix} \bar{x}(t) \\ \bar{u}_j(t) \end{pmatrix} dt\,. \tag{6.164}$$

Now, the solution proposed in Sec. 6.3.2 for linear-quadratic HOCPs can be applied to find the optimal control $\bar{v}_j$ of the transformed system.

In the case of an input-to-output linearization, the procedure is similar to obtain a controllable canonical form. This time the output $y(t)$ (6.159) is differentiated until no component of $L_{p_j} L_{f_j}^{r-1} o_j(x(t))$ is zero for the first time with $r \in \mathbb{N}_0$. Here, $p_j(x(t))$ is the matrix created by the column vectors $p_{j,i}(x(t))$ for $i \in \{1, ..., n_u\}$. This leads to the linearizing control:

$$u_j(t) = \left(L_{p_j} L_{f_j}^{r-1} o_j(x(t))\right)^{-1} \left(\bar{u}_j(t) - L_{f_j}^{r} o_j(x(t))\right). \tag{6.165}$$

Remark 6.5: Please note that the inversion might not be well defined in some cases, e.g. in the case that the dimension of the output n_y is not equal to the dimension of the original control n_u. In this case, it seems to be most straightforward to introduce additional outputs, which are added to $y(t)$, such that $n_y = n_u$. In case that the state $x(t)$ is such that the matrix $L_{p_j} L_{f_j}^{r-1} o_j(x(t))$ is singular, some components of the control $u_j(t)$ may be chosen arbitrarily and the remaining components are determined uniquely again. □

The corresponding HOCP depends on the output $y(t)$ and the transformed control $u_j(t)$:

$$J = \frac{1}{2}\left\|y(t_e) - y_e\right\|_{W_e}^2 + \sum_{j=0}^{N} \int_{t_j}^{t_{j+1}} \frac{1}{2} \begin{pmatrix} y(t) \\ \bar{u}_j(t) \end{pmatrix}^T \begin{pmatrix} C_j & V_j \\ V_j^T & Z_j \end{pmatrix} \begin{pmatrix} y(t) \\ \bar{u}_j(t) \end{pmatrix} dt\,. \tag{6.166}$$

Again, the primitive-based method from Sec. 6.3.2 for linear-quadratic HOCPs is used to find an optimal solution. In the next section, the optimal control method based on optimal control primitives is applied to find optimal trajectories for a robotic manipulator.

6.3.3 Primitive-Based Trajectory Planning for Robotic Manipulators

In this section, optimal trajectories are planned with optimal control primitives and their optimal concatenation for robotic manipulators, which consist of kinematic chains. The primitive-based optimal control method is suitable for planning trajectories with specified interior-point constraints. At the interior constraints, the system dynamics possibly changes due to different contact situations and due to loading or releasing some objects to be transported or tools for manipulation, such that the overall dynamics is hybrid.

The dynamics of a manipulator is given by [112]:

$$M_j\big(\tilde{q}(t)\big)\,\ddot{\tilde{q}}(t) + \tilde{c}_j\big(\tilde{q}(t), \dot{\tilde{q}}(t)\big) + \tilde{g}_j\big(\tilde{q}(t)\big) = \tilde{\tau}_j(t)\;, \tag{6.167}$$

where $\tilde{q}(t) \in \mathbb{R}^{n_{\tilde{q}}}$ are the joint angles, $\tilde{\tau}(t) \in \mathbb{R}^{n_u}$ is the control torque with $n_u = n_{\tilde{q}}$, $M_j(\tilde{q}(t)) \in \mathbb{R}^{n_{\tilde{q}} \times n_{\tilde{q}}}$ is the positive-definite and symmetric inertia matrix, $\tilde{c}_j(q(t), \dot{q}(t)) \in \mathbb{R}^{n_{\tilde{q}}}$ are the centrifugal and coriolis torques, $\tilde{g}_j(q(t)) \in \mathbb{R}^{n_{\tilde{q}}}$ is the gravity torque. By introducing an additional state variable $\tilde{p}(t) := \dot{\tilde{q}}(t)$, the system of second-order differential equations (6.167) is converted to a system of first-order differential equations:

$$\begin{pmatrix}\dot{\tilde{q}}(t)\\ \dot{\tilde{p}}(t)\end{pmatrix} = \begin{pmatrix}\tilde{p}(t)\\ -M_j(\tilde{q}(t))^{-1}\Big(\tilde{c}_j(\tilde{q}(t),\tilde{p}(t)) + \tilde{g}_j(\tilde{q}(t)) - \tilde{\tau}_j(t)\Big)\end{pmatrix} \tag{6.168}$$

$$y(t) = o_j(\tilde{q}(t)) \tag{6.169}$$

where $y(t) \in \mathbb{R}^{n_y}$ with $1 \leq n_y \leq n_{\tilde{q}}$ is the output of the system. It is assumed here, that the dimension n_y of the output equals the dimension $n_{\tilde{q}}$ of the joint vector. If this is not the case, please see Remark 6.5. The function $o_j(\tilde{q}(t))$ is known as forward kinematics, specifying the transformation from joint space to task space coordinates. To apply the primitive-based optimization for trajectory planning, the nonlinear dynamics has to be linearized first.

Joint Space Primitives

To optimize a robot's movements in joint space, an input-to-state feedback linearization is required. In (6.168), the system dynamics are already given in controllable canonical form, such that no coordinate transformation is required. It is shown that the robot dynamics (6.168) are input-to-state feedback linearizable [159] and the feedback control is

$$\tilde{\tau}_j(t) = M_j(\tilde{q}(t))u_j(t) + \tilde{c}_j(\tilde{q}(t),\tilde{p}(t)) + \tilde{g}_j(\tilde{q}(t))\ . \tag{6.170}$$

The virtual control of the linearized system is $\bar{u}_j(t)$ and the relation $\dot{\tilde{p}}(t) = \bar{u}_j(t)$ holds. The dynamics of the state $x(t) = \begin{pmatrix}\tilde{q}^T(t) & \tilde{p}^T(t)\end{pmatrix}^T$ with respect to the virtual control input $\bar{u}_j(t)$ can be written in the form (6.124) with

$$A_j = \begin{pmatrix}0 & I\\ 0 & 0\end{pmatrix},\ B_j = \begin{pmatrix}0\\ I\end{pmatrix}\ ,\ \text{and } I \in \mathbb{R}^{n_{\bar{u}} \times n_{\bar{u}}}\ . \tag{6.171}$$

Based on the linearized dynamics (6.124) and the running cost function (6.129), the stabilizing and anti-stabilizing solutions of (6.141) and further matrices according to (6.142) and (6.143) can be computed. This leads to the optimal, virtual control primitives $\bar{u}_j^*(t)$ for $j \in \{0, ..., N\}$ as specified in (6.144). The parameters $\tilde{\eta}_j$ and $\tilde{\rho}_j$ are chosen optimally as described in Sec. 6.3.2 depending on the autonomous switchings and the interior point constraints. From the optimal, virtual control primitives $\bar{u}_j^*(t)$, the optimal control primitives

$$\tilde{\tau}_j^*(t) = M_j(\tilde{q}(t))\bar{u}_j^*(t) + \tilde{c}_j(\tilde{q}(t),\tilde{p}(t)) + \tilde{g}_j(\tilde{q}(t)) \tag{6.172}$$

are derived, which are applied to the physical manipulator.

In the HOCP, the state trajectory χ and the virtual control trajectories $\bar{\upsilon}_j$ are optimized. With the input-to-state feedback linearization of the manipulator dynamics, the continuous states $\tilde{q}(t)$ and $\tilde{p}(t)$ of the original system are equal to the state $x(t)$ of the linearized system. Further, the virtual control $\bar{u}_j(t)$ equals the joint acceleration $\ddot{\tilde{q}}(t)$. This implies that the trajectories $\tilde{q}$, $\dot{\tilde{q}}$, and $\ddot{\tilde{q}}$ of the joint angle, the joint velocity, and the joint acceleration are chosen optimally. However, it is not possible to optimize the original, physical control trajectories $\tilde{\tau}_j$, e.g. minimize the integral over the square of the control values $\tilde{\tau}_j(t)$ for $j \in \{0, ..., N\}$ and $t \in [t_j, t_{j+1}]$. Optimizing in joint space, it is also possible to consider interior point constraints, which are given in Cartesian coordinates, also called task space coordinates. This is achieved by transferring the constraints in task space with the forward kinematics (6.159) to constraints in joint space, e.g. a constraint is $y_j - o_{j-1}(\tilde{q}(t_j)) = 0$ with the task space coordinates y_j that have to be fulfilled at time t_j.

Task Space Primitives

In some cases, it may be desired to find optimal trajectories in task space, e.g. when the movement of an end-effector attached to a manipulator shall be optimized. In this case, an input-to-output feedback linearization is used. It is shown that the output $y(t)$ defined in (6.159) with the state dynamics (6.158) is input-to-output linearizable in each component with relative degree $r = 2$. The result of the feedback linearization for the multi-input multi-output system (6.159) and (6.158) is given by

$$\begin{aligned}\ddot{y}(t) = {} & \dot{\mathcal{J}}_j(\tilde{q}(t))\,\tilde{p}(t) - \mathcal{J}_j(\tilde{q}(t))M_j(\tilde{q}(t))^{-1}\big(\tilde{c}_j(\tilde{q}(t), \tilde{p}(t)) + \tilde{g}_j(\tilde{q}(t))\big) \\ & + \mathcal{J}_j(\tilde{q}(t))M_j(\tilde{q}(t))^{-1}\tilde{\tau}_j(t)\,. \end{aligned} \tag{6.173}$$

Here, $\mathcal{J}_j(\tilde{q}(t)) = \frac{\partial o_j(\tilde{q}(t))}{\partial \tilde{q}}$ is the Jacobian matrix, which is the derivative of the forward kinematics $o_j(\tilde{q}(t))$ with respect to the joint angle $\tilde{q}$. The input-to-output linearization provides the linear relation $\ddot{y}(t) = \bar{u}_j(t)$ between the Cartesian acceleration of the end-effector and the virtual control $\bar{u}_j(t)$ for the linearized system. The linearization is formulated as an HOCP with the linear dynamics (6.124) and the continuous state $x(t) = \begin{pmatrix} y^T(t) & \dot{y}^T(t) \end{pmatrix}^T$. The matrices A_j and B_j are as defined in (6.171) and the output matrix is $O_j = \begin{pmatrix} I & 0 \end{pmatrix}$ with the identity matrix $I \in \mathbb{R}^{n_y \times n_y}$. The procedure presented in Sec. 6.3.2 is applied to compute the parameters for the optimal task space primitives. The main difference compared to optimal control primitives in joint space is that the state $x(t)$ contains the Cartesian position $y(t)$ and velocity $\dot{y}(t)$ and the interior point constraints specify the position of the end-effector instead of the joint angles. Solving (6.173) for $\tilde{\tau}_j(t)$, the control primitive, that is applied to the original manipulator, is

$$\begin{aligned}\tilde{\tau}_j(t) = {} & M_j(\tilde{q}(t))\mathcal{J}_j(\tilde{q}(t))^{-1}\Big(\bar{u}_j(t) - \dot{\mathcal{J}}_j(\tilde{q}(t))\,\tilde{p}(t) \\ & + \mathcal{J}_j(\tilde{q}(t))M_j(\tilde{q}(t))^{-1}\big(\tilde{c}_j(\tilde{q}(t), \tilde{p}(t)) + \tilde{g}_j(\tilde{q}(t))\big)\Big) \end{aligned} \tag{6.174}$$

for $j \in \{0, ..., N\}$. In the case of task space primitives, optimality refers to the trajectories of the position $y(t)$, velocity $\dot{y}(t)$, and acceleration $\ddot{y}(t)$ of the end-effector in Cartesian coordinates. Here, it is assumed that $n_{\bar{u}} = n_y$ and that the inversion of the Jacobian $\mathcal{J}$ is always possible. If this is not the case, please consider Remark 6.5.

6.3.4 Obstacle Avoidance

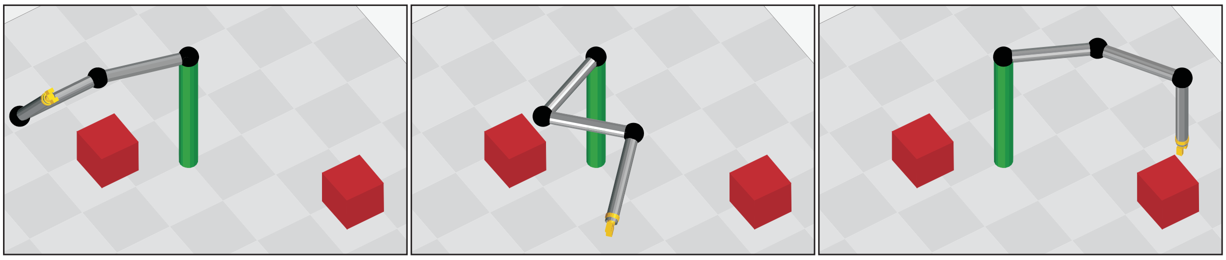
In the previous sections, it has been shown how optimal control primitives are concatenated in order to obtain an optimal trajectory with respect to autonomous switchings and a number of specified interior point constraints. Here, interior point constraints with free or fixed time are added to the HOCP to ensure that collision-free trajectories are found in the presence of static or moving obstacles in the manipulator workspace.

It is in general unknown where to place the interior points to obtain a collision-free trajectory. Therefore, the primitive-based control method is combined with a heuristic motion planning algorithm as we presented in [198]. The main idea is to depict samples of kinematic paths derived from a *Rapidly-Exploring Random Tree* (RRT) algorithm if collisions were detected. The complete algorithm proceeds as follows: In a first step, the RRT algorithm finds a collision-free kinematic path from the initial to the desired final state. Afterwards, one single primitive is considered generating the trajectory between the initial state and final state and the resulting trajectory is checked for collisions. If a collision was detected at time t_j, the configuration-time sample from the kinematic path whose time stamp is closest to t_j is added as an interior point. Afterwards, the trajectory generated by the concatenation of two control primitives at the interior point is computed. If a collision occurs again, another interior point is added to the planning process. This procedure is repeated until a collision-free overall trajectory is found. If no collision-free trajectory is found, then the optimization is repeated starting with another kinematic path, that is found by the RRT algorithm. The RRT algorithm will in general find different solutions since samples are generated stochastically. When a collision-free solution is found by the combined RRT and primitive-based approach, the resulting trajectory is optimal for the specific choice of interior points. However, the interior points are not chosen optimally since they originate from the RRT solution. Consequently, the overall trajectory is suboptimal for the specified HOCP.

6.3.5 Robotic Manipulation Example

The proposed method is applied to a simple manipulation task with the manipulator 'ViSHaRD3', which has 3 degrees of freedom (DOF). The dynamics and parameterization of the robot are specified in [173]. The planning process is carried out in joint space under consideration of two static obstacles in the workspace. Referring to Fig. 6.6, the task objective is to control the manipulator from its initial state $x_0 = \begin{pmatrix} \tilde{q}_0^T & \tilde{p}_0^T \end{pmatrix}^T$ to the final state $x_e = \begin{pmatrix} \tilde{q}_e^T & \tilde{p}_e^T \end{pmatrix}^T$. The initial and final joint velocities $\tilde{p}(t_0)$ and $\tilde{p}(t_e)$ are equal to zero. Further, the trajectory must pass through the interior configuration $\tilde{q}_c$ with an unspecified velocity $\tilde{p}(t_c)$ at the specified time t_c. The joint velocities $\tilde{p}(t_c)$ at the interior constraint are optimized during the planning process. Here, autonomous switching is not considered in this example.

In Fig. 6.7, the planning process is illustrated. At first, the optimization algorithm searches for an optimal trajectory with two primitives that meet at the desired interior configuration $\tilde{q}_c$. The resulting trajectory in joint space is checked with the forward kinematics for collisions with the two obstacles in the workspace. Here, a collision is found with the first obstacle, see Fig. 6.7(a). For the next optimization run, the configuration-time sample, which is closest to the collision time, is added from the collision-free RRT solution, see Fig. 6.7(b). To be able to extend the simulation example straightforwardly to moving obstacles, the additional interior point is considered with a fixed time t_j. The



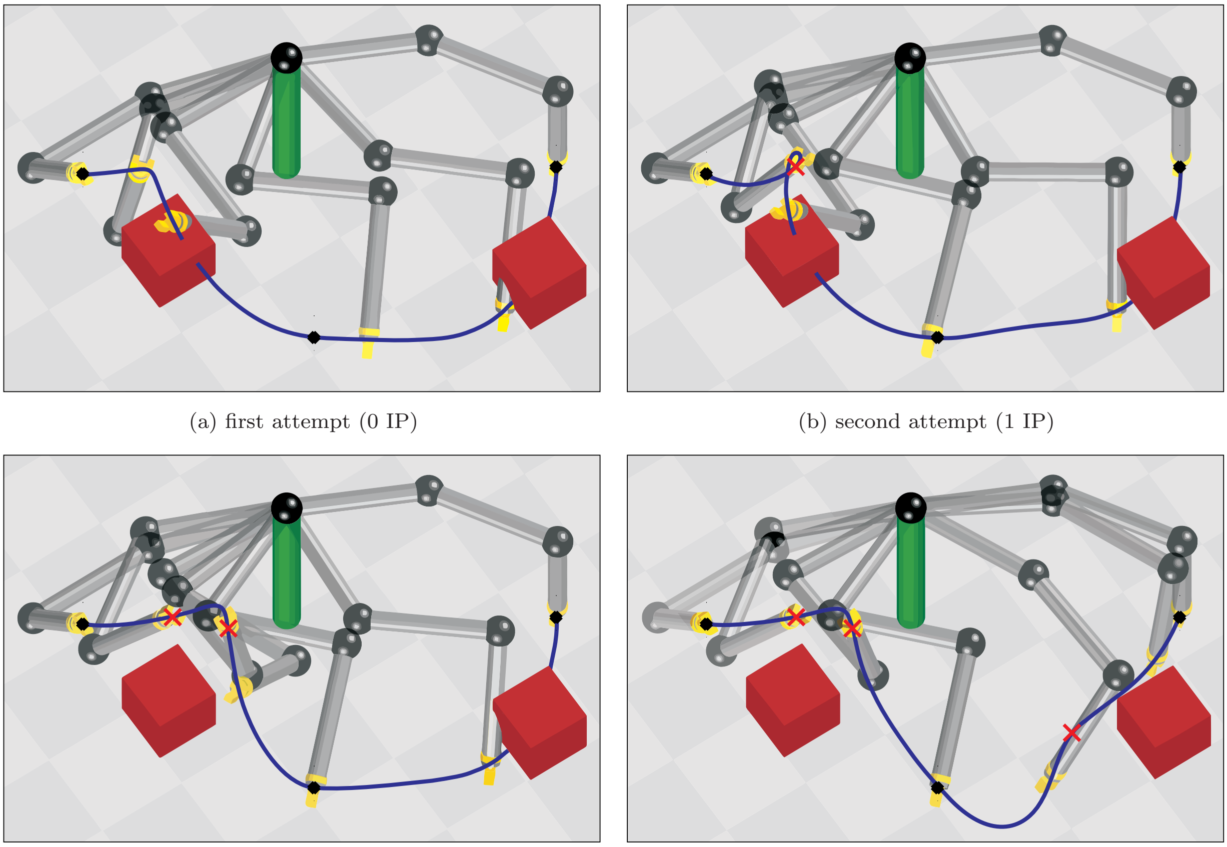
(a) $t = 0s$ (b) $t = t_c = 3$sec (c) $t = t_e = 6$sec

Fig. 6.6: Initial, interior, and final configuration of the manipulator.

additional interior point is marked with a red cross. Since the state trajectory still collides with the first obstacle, a second interior point is included, such that the robot does not collide with the first obstacle anymore, compare Fig. 6.7(c). Now, the first observed collision is detected with the second obstacle. By adding a third interior point, a completely collision-free trajectory is computed as illustrated in Fig. 6.7(d).

The resulting state trajectories, joint accelerations, as well as the interior configurations added by the heuristic planning algorithm (marked as red crosses) are shown in Fig. 6.8. There, the joint accelerations show corners at all interior states but are continuous. This is related to the fact that the interior configurations are assigned and the angular velocities are required to be continuous. Eq. (6.132) is solved for the control $\bar{u}_j(t)$ just before and just after the interior constraint t_j while considering (6.146), (6.134), and (6.171). The result shows that the virtual control $\bar{u}_{j-1}(t_j^-)$ just before the interior constraint equals $\bar{u}_j(t_j)$ just after the constraint. With the linear relation $\bar{u}_j(t) = \dot{\bar{p}}(t)$ obtained from the input-to-state linearization, it is concluded that the angular accelerations are always continuous.

Table 6.3.5 lists the computational effort for the parameter computation of all primitives, which generate the overall trajectory. There, it is shown how much time is spent for the computation of the optimal concatenation of primitivess, collision checking and integration of the differential equation (6.168). The small computational effort results from an offline precomputation of the closed-loop matrices $A_\ominus$ and $A_\oplus$ as well as the extremal solutions $P_\oplus$ and $P_\ominus$ of the CARE (6.141). The constraint equations (6.147) - (6.151) for the primitives are automatically derived and arranged in the form of (6.152), if interior points must be met. Thus, the computations that must still be carried out online are the evaluation of the matrix exponentials in (6.152), the solution of the system of linear equations for the unknown parameters, and checking the resulting trajectory for collisions. In Table 6.3.5, the expression 'kinematic path' refers to the time that is required to find a solution with the RRT algorithm. A set of parameters are computed in $16.5 \cdot 10^{-3}$sec and the complete trajectory including the insertion of additional interior point constraints is found in 0.81sec excluding the time for collision checking. The major part of the computations is spent for collision checking and the related integration of the differential equation (6.168). For computations, MATLAB2008a was used on a 2GHz Intel Core 2 Duo T5750 processor with 2GB RAM.



(a) first attempt (0 IP)

(b) second attempt (1 IP)

(c) third attempt (2 IPs)

(d) final attempt (3 IPs)

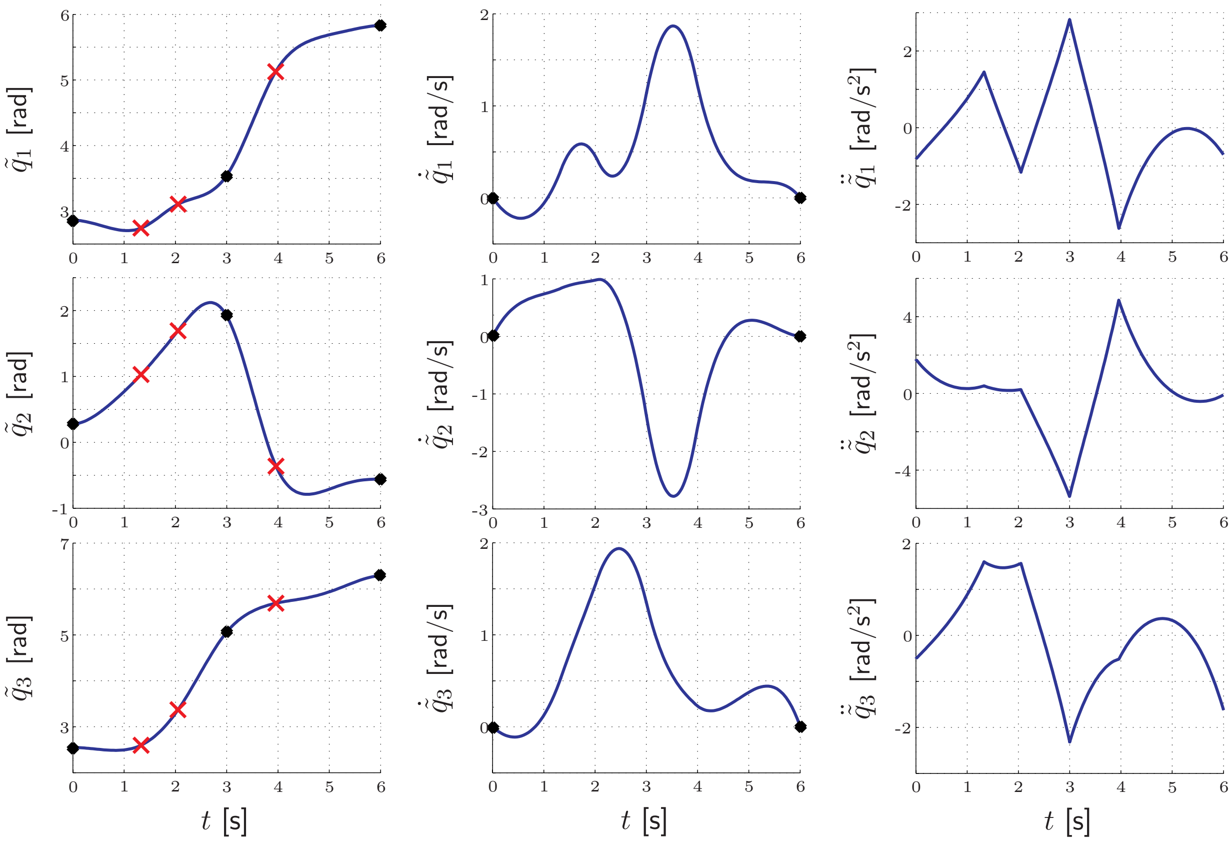
Animation of the trajectory

Fig. 6.7: An illustration of the robot and the task space trajectories of the endeffector is shown for the different steps of the planning process. In the planning process, interior point (IP) constraints from an RRT solution are iteratively added until a collision-free trajectory is found. The resulting solution can be seen in graph (d), which corresponds to the joint angles in Fig. 6.8(a).

6.3.6 Discussion

An optimal control method for HOCPs with a fixed discrete state sequence and autonomous switching is introduced that finds optimal trajectories by the concatenation of optimal control primitives. Optimal control primitives are derived in [52] for purely continuous, linear-quadratic optimal control problems with terminal constraints based on a Riccati-solution with feedback and feedforward terms. Here, the approach is extended to HOCPs and optimal control problems with interior point constraints. Thereby, linear-quadratic HOCPs as well as input-to-state and input-to-output feedback linearizable HOCPs are considered. For the extension, several optimal control primitives from [52] are concatenated such that all optimality conditions of an HOCP are satisfied at the switching times and times of interior point constraints.

The optimal control primitives are given in a parameterized form, such that the feedback and feedforward Riccati-matrices can be computed beforehand. For the solution of a



(a) joint angles (b) angular velocities (c) angular accelerations

Fig. 6.8: Joint angles (a), angular velocities (b), and angular accelerations (c) obtained from concatenated primitives with boundary values and intermediate configuration (black dots). The interior point constraints added by the planning algorithm to guarantee a collision-free trajectory are marked with red crosses.

specific HOCP, the parameters are determined based on the corresponding optimality conditions and imposed initial, interior, and terminal constraints. It depends on the system dynamics, the sampling rate of the MPC algorithm, and the HOCP, if the optimal parameters can be calculated in real-time as for example required in an MPC framework. In the case of affine constraints and fixed switching times, the optimal parameters are found by solving a system of linear equations. This can be computed very fast and the computation time is reliably predictable, such that an online solution is probable. In the case of free switching times or nonlinear constraints, a system of nonlinear equations has to be solved for the unknown parameters. For the solution, it is in general required to use iterative, static optimization algorithms, e.g. based on a Newton method. Now, the computational time is likely to be an order or orders of magnitude higher and is less reliably predictable.

In the case of a linear-quadratic HOCP, the variables that can be optimized are the continuous state and the continuous control. In the case of an HOCP, that requires an input-to-state or input-to-output feedback linearization, the continuous state and continuous control of the transformed system are the variables to be optimized. In some of these cases as in the case of manipulators, the continuous state of the original system equals

Tab. 6.1: Average costs and computational times.

Description	Quantity
Kinematic path	9.4 sec
Parameter computation	16.5 msec
Trajectory computation	0.81 sec
Collision Checking	$17,7$ sec
Total computational time	28 sec
Optimal costs J^*	62.5

the continuous state of the feedback linearization, which implies that the original state is also optimized. Further, some parts of the state's derivative, e.g. the acceleration of a manipulator, are considered in the optimization since the virtual control corresponds to these parts of the derivative. In no case, constraints on the control can be considered.

A drawback of using feedback linearization is the high sensitivity to inaccuracies in the model parameters, which occur in practice. Especially friction terms in manipulators are difficult to determine. Instead of applying the optimal control directly to the manipulator as open-loop control, the optimal state trajectory is given as reference to a closed-loop controller to ensure that the manipulator shows the desired behavior. In the case of small modeling inaccuracies, i.e. the optimal solution for the correct model is in a convex neighborhood around the calculated solution, the closed-loop control approach still leads to suboptimal state trajectories.

The trajectory planning scheme is applied to a robotic manipulator in simulations. The planned trajectories are collision-free, include interior point constraints, and may contain autonomous switching, though autonomous switching is not considered in the example. The RRT algorithm used for obstacle avoidance finds collision-free, kinematic paths in the joint space, where the samples of the path are connected by straight lines. This means that the RRT algorithm does not consider the robot dynamics, such that it is not guaranteed that the found path is feasible for the manipulator. Inserting some samples from the RRT solution as interior point constraints into the primitive-based optimization, it is not ensured that a feasible solution exists. The principle idea of the obstacle avoidance has been introduced in our publication [198]. There, it is observed that the RRT solution is usually feasible when the set of possible paths through obstacles is large relative to the system dynamics. If the space between obstacles is narrow, then often problems with the feasibility occur. In the simulations here, the obstacles were wide enough apart, such that all RRT solutions were feasible. Further, the approach can be extended to calculate collision-free trajectories in real-time, e.g. only one primitive in the extreme case. A replanning of the complete trajectory during runtime is likely to be possible if interior points have not changed much, such that a further application of the kinematic motion planner can be avoided.

Instead of the used RRT method, a randomized kinodynamic planner could be applied as proposed in [198]. The kinodynamic planner considers the system dynamics and kinematics and finds feasible paths in joint space by sampling the control input. A major drawback of the kinodynamic planner is that the computational time increases significantly compared to the used RRT algorithm due to the additional integrations at each sample. A further limitation of the randomized kinodynamic planner is the exponential increase in complexity with the continuous control dimension, such that an application to more than 4 continuous controls is usually not possible.

The trajectories found for manipulators with obstacle avoidance are suboptimal. The reason is that the samples from the RRT solution, which are included into the optimal control problem as interior point constraints for collision avoidance, are not chosen optimally. However, the trajectories are optimal for the specific choice of interior point constraints. This is ensured by the optimality conditions that hold at the concatenation points of two primitives.

6.4 Summary

The structure of HOCPs and the associated optimality conditions is exploited in two optimal control approaches. The approaches calculate solutions fast, such that depending on the computational power and the problem complexity an online solution and online replanning may be possible. Further, the approaches include state feedback based on a linearization of the optimality conditions around the currently optimal trajectory and based on a repetitive solution with updated measurements in an MPC framework. The two indirect methods clearly outperform direct methods and dynamic programming if parameters change during the optimization, if replanning is required due to new or shifted events, and if the state dimension is medium-to-high.

The first approach consists of an MPC algorithm that uses a gradient method of second order for the optimization. The gradient method is extended to HOCPs here and the convergence is proven. If the computation is stopped before the optimal solution is reached completely, the optimal feedback control law further approaches the optimal control in the periods between two sampling times of the MPC algorithm. In the case that a new event happens at the end of the prediction horizon or an event vanishes at the beginning of the prediction horizon, the algorithm can still find the optimal control for the changed discrete state sequence if the change remains in a convex region of the HOCP.

The second approach relies on parametric optimal control primitives as solution from the literature for linear-quadratic optimal control problems. Here, HOCPs are solved by concatenating optimal control primitives and ensuring that optimality conditions are satisfied at the concatenation points of two primitives. Large parts of the computations of the optimal control primitives can be precomputed and the optimal parameters for the specific HOCP are found online by the solution of a set of algebraic equations. The approach is extended to nonlinear HOCPs that are input-to-state or input-to-output feedback linearizable. The extension is applied to the optimal control of a robotic manipulator, while avoiding obstacles. For finding collision-free trajectories, sampling points of the solution from a kinematic motion planner are inserted iteratively into the optimization as interior point constraints if necessary. Feedback is included by recalculating optimal primitives for each iteration of an MPC approach or by including (sub)optimal feedback control laws.

7 Conclusion and Future Directions

In the following, a summary and conclusions on the contribution of this thesis and an outlook on future work are provided.

7.1 Conclusion

Hybrid optimal control problems can basically be solved by three types of approaches: dynamic programming, direct methods, and indirect methods. All three methods have their strengths and weaknesses. That means for a user that the specific hybrid optimal control problem (HOCP) needs to be analyzed to decide which method is favorable for the solution. The thesis focuses on indirect methods and the goal was to increase the applicability and competitiveness of indirect methods in comparison to direct methods and dynamic programming. The reasons for choosing indirect methods were several advantageous properties that direct methods and dynamic programming do not have and the potential for further improvements with impact.

The contribution of the thesis is to reduce dominant limitations of indirect methods that were presented in the introduction, such that the applicability of indirect methods is increased for HOCPs. In the following, the contributions are discussed and a new evaluation of strengths and weaknesses of indirect methods compared to direct methods and dynamic programming is provided:

(i) **Optimality Conditions:** The set of HOCPs for which optimality conditions exists is expanded in three cases, which are included in two novel versions of the hybrid minimum principle (HMP).

First, optimality conditions are provided for controlled switching with simultaneous resets of the continuous state. Second, optimality conditions are derived for the case that the continuous dynamics, the running cost function, the switching manifolds for autonomous switching, and the reset maps for the continuous state at switchings explicitly depend on time. The first two extensions result in a version of the HMP that considers a generalized class of HOCPs.

Third, a version of the HMP is formulated for hybrid systems with partitioned continuous state space, where switching manifolds intersect. There, optimality conditions are provided for the case that an optimal state trajectory passes an intersection of switching manifolds. This novel version of the HMP is further applied in an algorithm to reduce the computational complexity for determining an optimal sequence of discrete states. The proposed version implicitly includes optimality conditions for switching manifolds with corners as corners are a degenerated case of intersecting switching manifolds.

With the novel versions of the HMP, important cases of HOCPs can now be analyzed with indirect methods. Direct methods can still consider a larger class of HOCPs, however, found optima can only be verified for the cases, for which optimality conditions exist. The class of HOCPs that can be investigated with dynamic programming is smaller than the corresponding class for indirect methods.

(ii) **Initialization:** Two concepts for the initialization of existing indirect multiple shooting algorithms with direct methods are proposed as well as a novel min-H algorithm that is initialized straightforwardly.

The first concept based on direct methods is an extension of existing approaches for purely continuous optimal control problems to HOCPs. In this concept, a direct multiple shooting method solves the entire specified HOCP at once. The solution is used to initialize an indirect multiple shooting algorithm, which again solves the complete HOCP at once. The results are highly accurate, however the robustness is relatively low. In the second concept, the HOCP is decomposed into purely continuous optimal control subproblems. Each subproblem is solved separately from the other subproblems by direct multiple shooting first followed by indirect multiple shooting, which is initialized by the solution of the direct method. A gradient descent method ensures that the subproblems are connected optimally. The second initialization concept is robust and less sensitive to initialization errors than the first concept. However, convergence is not as fast as in the first concept due to the gradient method for optimizing the connection of subproblems, see also (iv). To combine both advantages, it is suggested to apply the second concept first and use the solution to initialize an indirect multiple shooting algorithm for the solution of the entire HOCP. The direct methods are initialized straightforwardly by providing initial values for the control and possibly state trajectory and by the sequence of discrete states. From the solution of the direct method, values of the adjoints and the sequence of constrained and unconstrained arcs is extracted for the initialization of the indirect method.

In a second approach, a min-H algorithm as a special class of indirect gradient methods is introduced for HOCPs with autonomous switching and interior point constraints. The algorithm is initialized by a control history, that leads to a desired sequence of discrete states, and by some multipliers. In general, the multipliers are successfully initialized with zeros. The algorithm can be extended to HOCPs with state constraints. The sequence of constrained and unconstrained arcs is determined and varied during the optimization, such that the sequence is not required for the initialization. More details of the algorithm are discussed in (iii).

Overall, the two initialization approaches, simplify the initialization of indirect methods significantly, such that the information that has to be provided initially is comparable to the information required in direct methods. Only dynamic programming is still easier to initialize.

(iii) **Domain of Convergence:** The previously small domain of convergence of indirect methods due to local convergence properties and numerical instabilities is considerably enlarged by a novel min-H algorithm in two ways: The method is proved to be globally convergent to a local optimum and the numerical stability is increased.

The min-H algorithm uses a direct minimization of the Hamiltonian function at every point of time for updates of the control. The minimization yields an optimal control and the current control iteratively approaches the optimal control. This allows us to prove that the method is globally convergent under mild conditions for HOCPs with autonomous switching.

The min-H algorithm integrates the state and adjoint differential equations each in their stable direction, which reduces numerical instabilities significantly compared to existing indirect multiple shooting algorithms as verified in simulations. However, for sensitive HOCPs with nonlinear dynamics, numerical instabilities may still arise in the backward integration of the Riccati-differential equations. The differential equations are partly quadratic and result from a linearization of the HOCP around the currently considered solution trajectory.

In the following, the convergence properties and the numerical stability of the novel min-H algorithm is compared to other indirect methods, direct methods, and dynamic programming. Though the numerical stability of indirect methods has been increased with the min-H algorithm, the numerical stability of direct methods and dynamic programming is still not reached. Besides the min-H algorithm, only dynamic programming shows global convergence. In contrast, global convergence is not achieved in other existing gradient and min-H methods that are developed for purely continuous optimal control problems. Similarly, indirect multiple shooting is only locally convergent since all updates are based on gradient information. Thus, it cannot be ensured for all pairs of state and adjoint values that the algorithm converges to a locally optimal solution. Direct methods, in general, also rely on gradient information for the optimization. That implies that the optimization may stop when no improvement of the costs can be found but optimality conditions, e.g. Karush-Kuhn-Tucker conditions, are not satisfied. Therefore, direct methods are only locally convergent.

(iv) **Complexity of Autonomous Switching:** Two algorithms are proposed to reduce the combinatorial complexity of previous indirect methods for the solution of HOCPs with autonomous switching.

Instead of analyzing every possible discrete state sequence separately, a branch and bound method is proposed in the first algorithm to reduce the search space for an optimal discrete state sequence. For the branch and bound approach, the possible discrete state sequences are organized in a tree and a relaxation from autonomous switching to controlled switching is introduced. A solution method for controlled switching based on indirect multiple shooting is applied combining results from literature. This method shows a linear increase in the computational complexity with the number of discrete states, such that a solution can be obtained efficiently. In the course of the optimization with the branch and bound method, the relaxations of the HOCP are tightened step-by-step until an optimal solution for the original HOCP is found. The algorithm is especially efficient if the relaxations are tight and pruning of branches can start early in the tree.

In the case of HOCPs with partitioned continuous state space, the complexity can be further reduced. Based on the associated HMP for a partitioned state space, a novel algorithm is presented that varies and optimizes the discrete state sequence based on gradient information simultaneously with the continuous optimization. The algorithm decomposes the HOCP into purely continuous optimal control subproblems, which are connected by switching points. On the lower layer of the algorithm, the subproblems are solved optimally be indirect multiple shooting. On the upper layer, the switching points are optimized by a gradient method. If switching points are shifted beyond the boundaries of a switching manifold, the switching points are projected to neighboring switching manifolds and the discrete state sequence is adapted accordingly. By this procedure, the discrete state sequence is locally optimized in parallel with the continuous optimization. Consequently, the computational complexity is reduced by a factor of combinatorial complexity compared to previous indirect methods.

Below, the computational complexity of the novel indirect methods for autonomous switching is compared to direct methods and dynamic programming. Using the proposed branch and bound algorithm, the computational complexity of indirect methods is similar to the complexity of direct methods that frequently use branch and bound approaches. The computational complexity of direct methods for the solution of HOCPs with partitioned state space is in general combinatorial even though branch and bound methods are applied. In this case, the introduced algorithm for intersecting switching manifolds provides the lower complexity. Dynamic programming only shows a linear increase in complexity with the number of discrete states. However, dynamic programming is not applicable to optimal control problems with intersecting switching manifolds and with more than 4 continuous states due to the curse of dimensionality.

(v) **Online Methods:** Based on optimality conditions, two approaches with a high potential for the online solution of HOCPs are introduced. Here, HOCPs include autonomous switching and interior point constraints and in the first approach, state constraints and parameter updates are additionally considered.

First, a model predictive control (MPC) algorithm is presented that calculates updates with a gradient method of second order. The gradient method is based on neighboring extremals that are found by a linearization of the optimality conditions around the previous, optimal solution. Under local convexity assumptions, the MPC algorithm can also handle changes in the sequence of discrete states and state-constrained and unconstrained phases that occur at the beginning or end of the moving horizon. In between sampling times, a local, optimal feedback control law is applied. It optimally corrects perturbations leading away from the optimal solution and even improves the solution in case that the previous solution is only suboptimal. The feedback control law is updated at every sampling time of the MPC algorithm. Depending on the computational power and the complexity of the specific HOCP at hand, the MPC algorithm may be online capable.

For linear-quadratic HOCPs and feedback linearizable HOCPs, a second online approach is proposed based on optimal control primitives. Optimal control primitives are derived in literature in parametrized form for linear-quadratic, purely continuous optimal control problems. Here, these primitives are concatenated by considering the associated optimality conditions, such that an HOCP is solved optimally. Large parts of the required computations are calculated offline. Due to the parametrized form, only algebraic computations have to be executed online to determine the optimal parameters. The computations are finished in milliseconds, such that the method has a high potential for online application. The primitive-based approach is especially intended to be used in trajectory planning for manipulators. The method is combined with a sampling-based motion planner to find collision-free trajectories. For the application to manipulators, a feedback linearization is required. This implies that the original control for the nonlinear hybrid system is not considered in the optimization, such that energy minimal solutions cannot be found. Instead, the joint angles, velocities, and accelerations are optimized.

Comparing direct methods and dynamic programming with the proposed online approaches, the following can be concluded: The MPC approach reduces the number of computations by partly using analytic results in contrast to direct methods. For HOCPs, no direct methods are known that are online capable and are based on sensitivities. Dynamic programming and direct approaches based on explicit MPC precompute the solution, such that online only minor computations are executed, which outperforms the MPC algorithm with respect to the computational time. However, the situation changes when parameters vary since dynamic programming and explicit MPC have to recalculate the solution in this case. Further, dynamic programming, and similarly also explicit MPC, is not applicable for a continuous state dimension of 4 and higher in contrast to the MPC algorithm and the primitive-based approach. The online computational times of the primitive-based approach are almost comparable with the online calculation times of dynamic programming. Dynamic programming is still faster for finding collision-free trajectories, however, if replanning is necessary due to unforeseen events, the primitive-based method is faster. Direct methods are not competitive with the primitive-based approach, when the approaches are compared for the solution of the same HOCPs.

The following limitations from the ones mentioned in the introduction of the thesis are not improved: It is still not possible to provide globally optimal solutions and globally valid, optimal feedback control laws with indirect methods. Indirect methods are based on necessary optimality conditions that only contain local information by design except for special cases like linear-quadratic optimal control problems. Therefore, no global statements can be given for general HOCPs.

Considering the discussed contributions of the thesis, it is concluded that indirect methods are in general competitive to direct methods and dynamic programming. In some categories like numerical stability, indirect methods are still not as good as direct methods and dynamic programming though the distance could be reduced. In other categories, like overall computational complexity, online solution, and accuracy of solutions, indirect methods outperform direct methods and dynamic programming.

7.2 Future Directions

There is still a large potential for further improvements of indirect methods. Future challenges and possible extensions to the novel results and insights gained in this thesis are discussed in the following. After some general remarks, the discussion is divided into the five categories that are used in the conclusion.

Especially for direct methods, there exist several software packages that find optimal controls for HOCPs. In contrast, available software packages are missing for the solution of HOCPs with indirect methods. To increase the popularity and to fully benefit from the features of indirect methods, it is important for the future to implement and provide software packages that include the state of the art in indirect methods. Further, the packages have to be implemented, such that a user can run the included optimal control algorithms intuitively and does not need any specialist knowledge. Such an implementation is possible by using common integration routines, automatic differentiation for calculating derivatives etc.

(i) **Optimality Conditions:** Two cases for future research are described: First, consider HOCPs with partitioned continuous state space, where switching manifolds intersect. Assume that there exists a discrete control to choose the subsequent discrete state, when a state trajectory meets an intersection of switching manifolds. Then, the question arises what the optimal discrete control is. Preliminary results indicate that the associated optimality conditions contain a minimization condition of the integrals over the Hamiltonians for the remaining time horizon in the different discrete states.

Next consider HOCPs with autonomous switching and intersecting switching manifolds but without partitioned state space. Here, a future challenge is to derive optimality conditions for the case that a state trajectory passes an intersection of switching manifolds and possibly a discrete control exists there to select the subsequent discrete state. It is especially interesting to explore how far needle variational techniques are valid here and how optimality conditions can be used in indirect methods to determine an optimal discrete state sequence.

(ii) **Initialization:** A possible extension of the novel min-H algorithm for an increased numerical stability is to initialize it by a direct method, e.g. direct multiple shooting. A solution of low accuracy from the direct method seems to be sufficient for a successful initialization as the numerical stability of the min-H algorithm is already relatively high. The initialization is straightforward since only the optimal control trajectory found by the direct method has to be transferred to the min-H algorithm.

(iii) **Domain of Convergence:** Three ideas for an increased domain of convergence are presented. First of all, it is desirable to further extend the domain of convergence of the min-H algorithm by increasing the numerical stability of the backward integration of Riccati-matrices. Therefore, it is essential to understand what causes the Riccati-matrices to tend to infinity in complex HOCPs. It seems especially promising to scale down the cost functions and constraints and to start the optimization on a reduced horizon, which is extended step-by-step.

Additionally, a future task is to develop and implement an optimal control algorithm based on collocation and quasilinearization for HOCPs. In such an algorithm, the state and adjoint differential equations and possibly the constraints are not fulfilled at the beginning, while the optimality condition for the control is always satisfied. The interesting question is how the numerical stability of a collocation-based quasilinearization algorithm compares to the proposed min-H algorithm and indirect multiple shooting. Solutions from collocation methods have a lower accuracy than the min-H algorithm and especially than indirect multiple shooting. However, it seems worth to analyze if such a collocation approach provides solutions with a higher numerical stability, especially as an initialization of an indirect multiple shooting algorithm for further refinement is straightforward.

The most common solution methods for multi-point boundary value problems originating from optimality conditions are multiple shooting, collocation, and gradient methods. The methods differ in the optimality conditions that are always satisfied during the optimization and the optimality conditions that are approached by the algorithm. This implies, as pointed out in [32], that there exist more possibilities to set up solution methods. It might be worth to analyze the numerical stability and applicability of the additional solution methods in future.

(iv) **Complexity of Switching:** Three possible improvements of existing algorithms for a low computational complexity are suggested. The introduced algorithm that varies the discrete state sequence during the continuous optimization uses a gradient descent method on the upper layer to optimize the switching points. This leads to a stable but slow convergence of the approach near the optimal solution. A future task is to modify the algorithm, such that Newton update steps are applied near the optimal solution, and to prove convergence of the Newton steps.

Next, there exist several algorithms for HOCPs with controlled switching that are linear in their complexity with respect to the number of discrete states if the sequence of discrete states is optimized. A further task is to analyze how these algorithms are transferable to HOCPs with controlled switching and simultaneous resets of the continuous state and how the complexity changes.

Another task contains the extension of the tree search algorithm based on branch and bound to HOCPs with autonomous switching and resets of the continuous state.

(v) **Online Methods:** A promising idea is to replace the second-order gradient method used in the MPC algorithm by the novel min-H algorithm. This has several advantages and nevertheless keeps the existing strengths of the approach. The convergence of the min-H algorithm is at least quadratic in a sufficiently small neighborhood around the optimal solution. But in contrast to the second-order gradient method, the min-H approach is globally convergent and does not require convex regions of the HOCP. The domain of convergence will be increased by the min-H algorithm, especially when changes in the sequence of discrete states and state constraints occur at the beginning or end of the moving horizon. It seems probable that the sampling time of the MPC algorithm can be increased with the min-H algorithm due to the large domain of convergence. The min-H approach also provides feedback control laws that can be applied in between two sampling instants of the MPC algorithm. When streamlining the implementation for faster computations, it seems likely that real-time computations can be achieved.

A Parameters for Simulations

In the following, parameters of the models, constraints, switching manifolds, and for the optimization are presented as these parameters were applied in the simulations shown in Sec. 5.3.4, 5.3.5, and 6.2.4. The parameters for the numerical examples in Sec. 3.2.2, 4.2.3, 4.3.4, and 5.2.3 are directly provided in the corresponding sections and the parameters for the 3-DOF manipulator can be found in [173, 216, 218].

A.1 Parameters for Variable Sequences

Below, parameters for numerical examples are given, that are solved by the optimal control Algorithm 5.2, which varies the discrete state sequence based on gradients of the optimal costs.

The switching manifolds for the linear and the nonlinear example are set up as

$$m_{i,k} = m_1 x_1 + m_2 x_2 + m_0 \tag{A.1}$$

for all $i, k \in \mathcal{Q}$, where discrete states i and k are direct neighbors. The parameters are shown in Tab. A.1 - A.4.

Linear Example

The system matrices for the linear dynamics and the weighting matrices for the cost functions have the form:

$$A_q = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}, \qquad B_q = \begin{pmatrix} b & 0 \\ 0 & b \end{pmatrix} \tag{A.2}$$

$$S_q = \begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix}, \qquad R_q = \begin{pmatrix} r & 0 \\ 0 & r \end{pmatrix}, \tag{A.3}$$

Tab. A.1: Switching manifolds $m_{i,k}$ for autonomous switching from discrete state i to k in the numerical examples with linear and nonlinear dynamics.

	$m_{1,2}$	$m_{1,3}$	$m_{2,1}$	$m_{2,3}$	$m_{2,4}$	$m_{2,5}$	$m_{3,1}$	$m_{3,2}$	$m_{3,4}$	$m_{3,6}$	$m_{3,9}$	$m_{4,2}$
m_0	14	7	-14	7	7	7	-7	-7	5	11	0	-7
m_1	1	0	-1	0	0	0	0	0	1	1	0	0
m_2	1	1	-1	1	1	1	-1	-1	0	1.5	1	-1

Tab. A.2: Switching manifolds $m_{i,k}$ for autonomous switching from discrete state i to k in the numerical examples with linear and nonlinear dynamics.

	$m_{4,3}$	$m_{4,5}$	$m_{4,6}$	$m_{4,7}$	$m_{5,2}$	$m_{5,4}$	$m_{5,7}$	$m_{5,8}$	$m_{6,3}$	$m_{6,4}$	$m_{6,7}$	$m_{6,9}$
m_0	-5	-4	4	4	-7	4	4	4	-11	-4	3	0
m_1	-1	2	0	0	0	-2	0	0	-1	0	1	0
m_2	0	-1	1	1	-1	1	1	1	-1.5	-1	0	1

Tab. A.3: Switching manifolds $m_{i,k}$ for autonomous switching from discrete state i to k in the numerical examples with linear and nonlinear dynamics.

	$m_{6,10}$	$m_{7,4}$	$m_{7,5}$	$m_{7,6}$	$m_{7,8}$	$m_{7,10}$	$m_{8,5}$	$m_{8,7}$	$m_{8,10}$	$m_{8,11}$	$m_{9,3}$	$m_{9,6}$
m_0	5	-4	-4	-3	2	2	-4	-2	2	-2	0	0
m_1	1	0	0	-1	1	0	0	-1	-1	0	0	0
m_2	1	-1	-1	0	1	1	-1	-1	1	1	-1	-1

Tab. A.4: Switching manifolds $m_{i,k}$ for autonomous switching from discrete state i to k in the numerical examples with linear and nonlinear dynamics.

	$m_{9,10}$	$m_{9,11}$	$m_{10,6}$	$m_{10,7}$	$m_{10,8}$	$m_{10,9}$	$m_{10,11}$	$m_{11,8}$	$m_{11,9}$	$m_{11,10}$
m_0	4	0	-5	-2	-2	-4	-4	2	0	4
m_1	0.8	1	-1	0	1	-0.8	0.5	0	-1	-0.5
m_2	-1	0	-1	-1	-1	1	1	-1	0	-1

Tab. A.5: Values used in the system matrices A_q and B_q, the cost matrices S_q and R_q, and the control constraints for the different discrete states q in the numerical example with linear dynamics.

Discrete State q	a_{11}	a_{12}	a_{21}	a_{22}	b	s	r	$u_{q,\max}$
1	-1	0.5	-2	-1	1	0.6	1	10
2	-1	-2	0.5	-0.1	1	1	1	2
3	-0.5	-0.1	-0.1	-0.1	1	0.5	0.1	10
4	-0.5	3	0.5	-1	1	0.5	0.1	10
5	-2	-2	0.5	-2	1	0.1	0.1	10
6	-0.1	0.2	-2	-2	1	8	1	2
7	-0.5	-2	0.5	-1	1	1	0.5	10
8	-0.5	-1	5	-0.5	1	0.1	0.1	10
9	-0.1	0.5	-0.5	-0.2	1	10	1	2
10	-4	2	0	-4	1	2	1	10
11	-1	1	1	-2	1	2	1	2

where the discrete state q ranges from 1 to $N_q = 11$. The only exception is discrete state $q = 10$ as a scalar control is applied here and thus $B_{10} = \begin{pmatrix} b \; b \end{pmatrix}^T$ is a vector and $R_{10} = r$ is scalar. The control constraints are

$$|u_{q,1}| \leq u_{q,\max} \quad \text{and} \quad |u_{q,2}| \leq u_{q,\max} \tag{A.4}$$

The parameters for each discrete state q are shown in Tab. A.5.

Nonlinear Example

The nonlinear differential equations (5.31) - (5.33) are parametrized by the velocity v_q in discrete state $q \in \{1, ..., 11\}$. The weighting matrices for the cost functions (5.34) have the form:

$$S_q = \begin{pmatrix} s_1 & 0 & 0 \\ 0 & s_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} , \qquad T_q = \begin{pmatrix} t_1 & 0 & 0 \end{pmatrix} , \qquad R_q = \begin{pmatrix} r_1 \end{pmatrix} . \tag{A.5}$$

The control constraints are

$$|u_{q,1}| \leq u_{q,\max} \tag{A.6}$$

Tab. A.6: Values used in the system and cost equations for the different discrete states q in the numerical example with nonlinear dynamics.

Discrete State q	v_q	s_1	s_2	t_1	r_1	$u_{q,\max}$
1	10	1	1	0	10	5
2	2	1	1	0	10	5
3	15	1	0.1	0	1	10
4	20	0.01	0.01	0	1	10
5	10	0.1	0.1	0	1	10
6	1	20	10	−20	10	5
7	2	20	1	−20	10	10
8	15	1	1	0	0.5	10
9	0.1	20	20	0	10	5
10	5	1	1	0	0.1	10
11	1	1	1	0	10	5

The parameters for each discrete state q are shown in Tab. A.6.

A.2 Parameters for Cooperative Transportation Scenario

Example with 3 Robots

The system matrices and the weighting matrices are defined as follows:

$$A_q^\nu = \begin{pmatrix} 0 & 0 & a_q^\nu & 0 \\ 0 & 0 & 0 & a_q^\nu \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \qquad B_q^\nu = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ b_q^\nu & 0 \\ 0 & b_q^\nu \end{pmatrix} \tag{A.7}$$

$$S_q^\nu = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & s_q^\nu & 0 \\ 0 & 0 & 0 & s_q^\nu \end{pmatrix}, \qquad R_q^\nu = \begin{pmatrix} r_q^\nu & 0 \\ 0 & r_q^\nu \end{pmatrix}. \tag{A.8}$$

The parameters a_q^ν, b_q^ν, s_q^ν, and r_q^ν for $\nu \in \{0, ..., 3\}$ are shown in Tab. A.7 and A.8. The controls satisfy the constraints $|u_1^\nu| \leq 10$ and $|u_2^\nu| \leq 10$. The initial states of the block and

Tab. A.7: Values used in the system matrices A_q^ν and B_q^ν for the different discrete states q, when the docking sequence of the robots $\nu \in \{1, ..., 3\}$ is in ascending order, i.e. $(1, 2, 3)$.

Discrete State q	a_q^0	a_q^1	a_q^2	a_q^3	b_q^0	b_q^1	b_q^2	b_q^3
1	0	1	1	1	0	1	1	1
2	1	0	1	1	0.8	0	1	1
3	1	0	0	1	1	0	0	1
4	1	0	0	0	1	0	0	0

Tab. A.8: Values used in the weighting matrices S_q^ν and R_q^ν for the different discrete states q, when the docking sequence of the robots $\nu \in \{1, ..., 3\}$ is in ascending order, i.e. $(1, 2, 3)$.

Discrete State q	s_q^0	s_q^1	s_q^2	s_q^3	r_q^0	r_q^1	r_q^2	r_q^3
1	0	0.5	0.5	0.5	0.8	0.8	0.8	0.8
2	0.5	0	0.5	0.5	0.8	0.8	0.8	0.8
3	0.5	0	0	0.5	0.8	0.8	0.8	0.8
4	0.5	0	0	0	0.8	0.8	0.8	0.8

the robots are:

$$x_0^0 = \begin{pmatrix} 22 \\ 22 \\ 0 \\ 0 \end{pmatrix}, \quad x_0^1 = \begin{pmatrix} 27 \\ 27 \\ 0 \\ 0 \end{pmatrix}, \quad x_0^2 = \begin{pmatrix} 18 \\ 27 \\ 0 \\ 0 \end{pmatrix}, \quad x_0^3 = \begin{pmatrix} 23 \\ 20 \\ 0 \\ 0 \end{pmatrix}. \tag{A.9}$$

The final state of the block and all robots is $x_e^\nu = (0,\ 0,\ 0,\ 0)^T$ for $\nu \in \{0, ..., 3\}$.

Tab. A.9: Values used in the system matrices A_q^ν and B_q^ν for the different discrete states q, when the docking sequence of the robots $\nu \in \{1, ..., 4\}$ is in ascending order, i.e. $(1, 2, 3, 4)$.

Discrete State q	a_q^0	a_q^1	a_q^2	a_q^3	a_q^4	b_q^0	b_q^1	b_q^2	b_q^3	b_q^4
1	0	1	1	1	1	0	1	1	1	1
2	1	0	1	1	1	0.4	0	1	1	1
3	1	0	0	1	1	0.9	0	0	1	1
4	1	0	0	0	1	1	0	0	0	1
5	1	0	0	0	0	1.2	0	0	0	0

Example with 4 Robots

For the example with 4 robots, the system matrices contain additional terms for damping except for the block. The matrices are defined as follows:

$$A_q^\nu = \begin{pmatrix} 0 & 0 & a_q^\nu & 0 \\ 0 & 0 & 0 & a_q^\nu \\ 0 & 0 & -a_q^\nu & 0 \\ 0 & 0 & 0 & -a_q^\nu \end{pmatrix}, \qquad B_q^\nu = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ b_q^\nu & 0 \\ 0 & b_q^\nu \end{pmatrix} \tag{A.10}$$

$$S_q^\nu = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & s_q^\nu & 0 \\ 0 & 0 & 0 & s_q^\nu \end{pmatrix}, \qquad R_q^\nu = \begin{pmatrix} r_q^\nu & 0 \\ 0 & r_q^\nu \end{pmatrix}. \tag{A.11}$$

The values of the parameters a_q^ν, b_q^ν, s_q^ν, and r_q^ν for $\nu \in \{0, ..., 4\}$ are given in Tab. A.9 and A.10. The controls satisfy the constraints $|u_1^\nu| \leq 30$ and $|u_2^\nu| \leq 30$. The block and the robots are initialized by:

$$x_0^0 = \begin{pmatrix} 16 \\ 16 \\ 0 \\ 0 \end{pmatrix}, \quad x_0^1 = \begin{pmatrix} 21 \\ 21 \\ 0 \\ 0 \end{pmatrix}, \quad x_0^2 = \begin{pmatrix} 15 \\ 19 \\ 0 \\ 0 \end{pmatrix}, \quad x_0^3 = \begin{pmatrix} 18 \\ 11.5 \\ 0 \\ 0 \end{pmatrix}, \quad x_0^4 = \begin{pmatrix} 15 \\ 10 \\ 0 \\ 0 \end{pmatrix}. \tag{A.12}$$

The final state of the block and all robots is $x_e^\nu = (0,\ 0,\ 0,\ 0)^T$ for $\nu \in \{0, ..., 4\}$.

A.3 Parameters for Autonomous Driving

The parameters used in the overtaking scenario of the autonomous car are given in Tab. A.11, A.12, and A.13 and the initialization is shown in Tab. A.14.

Tab. A.10: Values used in the weighting matrices S_q^ν and R_q^ν for the different discrete states q, when the docking sequence of the robots $\nu \in \{1, ..., 4\}$ is in ascending order, i.e. $(1, 2, 3, 4)$.

Discrete State q	s_q^0	s_q^1	s_q^2	s_q^3	s_q^4	r_q^0	r_q^1	r_q^2	r_q^3	r_q^4
1	0	1	1	1	1	0.2	1	1	3	5
2	0.2	0	1	1	1	0.2	1	1	1	1
3	0.2	0	0	1	1	0.2	1	1	1	1
4	0.2	0	0	0	1	0.2	1	1	1	1
5	0.2	0	0	0	0	0.2	1	1	3	5

Tab. A.11: Parameter values for the autonomous car.

Variable	m	J_z	l_F	l_R	$\bar{g}$	k_R	c_w	A_c	ς
Value	1000	13225	1.25	1.25	9.81	0.015	0.3	1.91	1.22
Unit	kg	$\mathrm{kg\,m^2}$	m	m	$\mathrm{m/sec^2}$	-	-	$\mathrm{m^2}$	$\mathrm{kg/m^3}$

Tab. A.12: Parameter values for the autonomous car.

Variable	C_F	C_R	μ_f	F_{minA}	F_{maxA}	δ_{minF}	δ_{maxF}
Value	50	50	0.9	-8000	1000	-35	35
Unit	kN/rad	kN/rad	-	N	N	°	°

Tab. A.13: Parameter values for the environment of the autonomous car.

Variable	w_r	w_c	w_{safe}	l_c	l_{safe}	l_{tr}
Value	3	1.7	0.5	4	1	5
Unit	m	m	m	m	m	m

Tab. A.14: Initialization of the overtaking scenario.

Variable	$x_{1,0}$	$x_{2,0}$	v_0	α_0	φ_0	ω_{turn}	$x_{1,0}^B$	$x_{2,0}^B$	v_0^B
Value	60	-1.5	15	0	0	0	80	-1.5	10
Unit	m	m	m/sec	rad	rad	rad/sec	m	m	m/sec

Notation

At first, some general notation used in this thesis is presented. Then, the notation is shown that is used for describing the theory and algorithms. At last, the notation applied in numerical examples is introduced.

Abbreviations

CARE	continuous-time algebraic Riccati equation
DMP	dynamic movement primitive
DOF	degree of freedom
DP	dynamic programming
HJB	Hamilton-Jacobi-Bellman
HMP	hybrid minimum principle
HOCP	hybrid optimal control problem
(LB-)MIOP	(logic-based) mixed-integer optimization problem
(MP)BVP	(multi-point) boundary value problem
MPC	model predictive control
PRM	probabilistic road map
PWA	piecewise affine
RRT	rapidly-exploring random tree
SQP	sequential quadratic programming
UAV	unmanned aerial vehicle

General

∂A	boundary of a set A
$\mathrm{Int}(A)$	interior of a set A
$\nabla_b\, a$	derivative of a with respect to the elements of b; result is a row vector if a is scalar and a Jacobian if a is a vector
$\nabla_{bb}\, a$	second derivative (Hessian) of a with respect to the elements of b
$\nabla_b \partial A(c)$	normal of the boundary of the set A with respect to b at the position c
$\Vert\cdot\Vert\,,\ \Vert\cdot\Vert_2$	Euclidean norm of a vector
$\vert\cdot\vert$	absolute value of a scalar
$o(a)$	error of order a^2 with the property $\lim_{a\to 0}\frac{o(a)}{a} = 0$
$a(t)$	value of a at the continuous time instant t
$a[s]$	value of a at the sample s of the discretized time instant t_s

Subscripts and Superscripts

a^0	initial value of variable a at iteration 0 of an algorithm
a_0, a^{ini}	initial value of variable a for dynamic systems
a_e	terminal value of variable a
a^*	optimal value of variable or function a
a^T	transpose of vector or matrix a
a^{-1}	inverse of variable or function a
$\dot{a}$	time derivative of a
$a^{(r)}$	r-th time derivative of variable or function a
a^-	limit of a from the left
a^{a}	function a is active
a^{ina}	function a is inactive
a^{c}	function a is associated with controls
a^{s}	function a is associated with states
a^{sc}	function a is associated with states and controls

Sets

Sets of Numbers

$\emptyset$	empty set
$\mathbb{N}$	natural numbers
$\mathbb{N}_0$	natural numbers combined with 0: $\mathbb{N}_0 := \mathbb{N} \cup \{0\}$
$\mathbb{Z}$	integer numbers
$\mathbb{R}$	real numbers

Sets of Variables and Parameters

I_j	$\subset \mathbb{R}$	interval of possible switching times for the j-th switch
$\mathcal{N}$	$\subset \mathbb{N}$	set of nodes in a tree
$\mathcal{N}_q^1$	$\subset \mathbb{N}$	direct neighbors of first order of discrete state q
$\mathcal{N}_q^2$	$\subset \mathbb{N}$	direct neighbors of second order of discrete state q
Ω_q	$\subset \mathbb{N}$	set of admissible discrete controls
$\mathcal{P}$	$\subseteq \mathbb{R}^{n_p}$	parameter space
$\mathcal{Q}$	$\subset \mathbb{N}$	discrete state space
Θ_j	$\subseteq \mathbb{R}^r$	subspace of $\mathcal{X}_j$ spanned by the row vectors of $\nabla_x \vartheta_j$
$\bar{\Theta}_j$	$\subseteq \mathbb{R}^{n_x - r}$	complimentary space to subspace Θ_j
$\mathcal{R}$	$\subset \mathbb{R}^{n_x}$	region of reachability
U_q	$\subset \mathbb{R}^{n_u}$	continuous control space associated with discrete state q
$\mathcal{X}_q$	$\subseteq \mathbb{R}^{n_x}$	continuous state space associated with discrete state q

Sets of Functions and Collections of Sets

$C^1, C^2, ...$	set of at least once, twice, ... continuously differentiable functions
E	subset of M containing extensions from a node in a tree
$\mathcal{F}$	collection of continuous vector fields f_q
G	set of nodes generated in the last iteration in a tree
$\mathbb{H}$	hybrid system
$\mathcal{H}$	collection of constraint functions h_q
$\mathcal{I}$	collection of reachable time intervals on switching manifolds
L	set of live nodes in a tree
Λ	collection of reset functions $\varphi_{i,k}$
$\hat{M}$	set of possible discrete state sequences with at most $\bar{N}$ switchings
$\mathcal{M}$	collection of switching manifolds $m_{i,k}$
Ω	collection of admissible discrete control sets Ω_q
σ	hybrid system execution
Σ	set of measurable, binary control functions ς
R	relaxed hybrid system (from autonomous to controlled switching)
S	set of nodes to be currently analyzed in a tree
U	collection of continuous control spaces U_q
$\mathcal{U}$	set of measurable, continuous control functions u
$\mathcal{U}_q$	set of measurable, continuous control functions u_q for discrete state q
$\mathcal{X}$	collection of continuous state spaces $\mathcal{X}_q$
Ψ	collection of interior point constraints ψ_j
$\hat{\Psi}$	information about feasible solutions used in the tree search algorithm

Variables

Scalars and Vectors

$d_1, ..., d_{n_d}$	discrete states
d_j	inhomogeneous term of the linearized state differential equation
$d_{\min}$	vector with absolute value of minimal derivative d_z
d_z	derivative of costs with respect to a switching point
$d\nu$	change in the Lagrange multiplier ν
$d\bar{p}$	change in the parameter vector from one iteration to the next
$d\hat{\pi}_j$	change in the Lagrange multiplier $\hat{\pi}_j$ for interior constraints ψ_j
dt_j	change in the switching time t_j
δ_j^i	difference in the switching time due to a needle variation
$\delta\lambda$	change in the adjoint λ due to a change in the control
$\delta\mu$	change in the Lagrange multiplier μ
δu	update of the control
δx	change in the state x due to a change in the control
δu^i	difference between perturbed and optimal control
δx^i	difference between perturbed and optimal state

Δu	difference between current and optimal control
$\Delta_{j-1,p}$	difference between $x^*(t_j)$ and $x^*(t_j - \delta_j^i)$
$\Delta_{p,j}$	difference between $x^*(t_j)$ and $x^*(t_j - \kappa_j^i)$
e_j	Riccati vector
ϵ^i	sequence of small, positive values determining the length of a needle variation in the optimal control trajectory υ^*
φ	angle between $\xi^\kappa(t_j^-)$ and $\nabla_x m_{p,j}$
η_{con}	auxiliary variable at the switching time t_j
η_j	auxiliary variable at the switching time t_j (Chap. 3)
η_j	Riccati vector
$\eta_{j,\cdot}$	Riccati vector with growing dimension
$\hat{\eta}_{j,2}$	auxiliary Riccati vector
$\eta_{j-1,j}$	auxiliary variable at the switching time t_j
$\eta_{j-1,p}$	auxiliary variable at the switching time t_j
$\eta_{p,j}$	auxiliary variable at the switching time t_j
γ_j	inhomogeneous term of the linearized adjoint differential equation
$\gamma_{j-1,p}$	difference between perturbed and optimal state trajectory from $t_j - \delta_j^i$ to $t_j - \kappa_j^i$
$\gamma_{p,j}$	difference between perturbed and optimal state trajectory from $t_j - \kappa_j^i$ to t_j
κ_c	constant Lagrange multiplier at the entry of a state constraint
κ_j^i	difference in the switching time due to a needle variation
λ	adjoint state
$\bar{\lambda}$	adjoint state
$\tilde{\lambda}$	adjoint obtained from a backward integration
$\hat{\lambda}$	adjoint used for an update of the backward integration
λ_k	adjoint state as optimization variable at time t_k in indirect multiple shooting
μ	Lagrange multiplier for state constraints
ν	constant Lagrange multiplier for the terminal constraint
p	discrete state passed additionally due to a needle variation
$\bar{p}$	parameter vector
π_j	constant Lagrange multiplier at an autonomous switching or interior condition (Chap. 6)
$\boldsymbol{\pi}_j$	vector of constant Lagrange multipliers at interior conditions
$\hat{\pi}_j$	constant Lagrange multiplier at an interior constraint (Chap. 4)
$\tilde{\pi}_j$	constant Lagrange multiplier at interior condition $\tilde{m}_j$ (Chap. 4)
q (also: i, j, k, l)	discrete state
θ	neighboring angle of the angle between f_{j-1} and $\nabla_x m_{j-1,p}$
ϑ	angle between $\xi^\delta(t_j^-)$ and $\nabla_x m_{j-1,p}$
θ_j	auxiliary variable
r_j	derivative $\nabla_x m_{j,j+1}^T$ of switching manifold
ϱ	sequence of discrete states
ς_i	vector of binary variable for selecting the dynamics (Chap. 5)
ς_k	error equations in indirect multiple shooting (Chap. 4)

t	time
t_c	entry time of a state constraint
t_c^{out}	exit time of a state constraint
t_j	j-th switching time
$\boldsymbol{t}_j$	vector of switching times from the j-th switch to the final time
t_k	switching time for a needle variation in the discrete control (Chap. 3)
t_k	time of a shooting node in indirect multiple shooting
t_s	time of a sample s in direct multiple shooting
$\bar{t}_s$	average time of two time samples t_{s+1} and t_s
t_υ	time of a needle variation in the optimal control υ^*
Δt_j	time between the switching times t_{j+1} and t_j
$\Delta t[s]$	time between the key samples s_{j+1} and s_j
τ	sequence of initial, switching, and final times
$u_{j,i}$	i-th component of the control u_j
u_p	continuous control in discrete state p
u_q	continuous control in discrete state q
u_υ	constant control value of a needle variation in the optimal control υ^*
$\bar{u}$	continuous control
$\upsilon_1, ..., \upsilon_5$	constant control values for needle variations
$\upsilon_{\cdot,j}$	auxiliary vectors and scalars
ω	sequence of discrete controls
ω_q	discrete control
x, $\bar{x}$, $\hat{x}$	continuous state
$\tilde{x}$	continuous state extended by accumulated running costs
x^0	accumulated running costs
x_1	arbitrary value or first component of the state variable x
x_2	arbitrary value or second component of the state variable x
x_j	continuous state in discrete state j (Chap. 6)
$x_{j-1,p}^i$	continuous state on the switching manifold $m_{j-1,p}$
$x_{p,j}^i$	continuous state on the switching manifold $m_{p,j}$
x_k	continuous state as optimization variable at time t_k in indirect multiple shooting
x_s	continuous state on the intersection of several switching manifolds
ξ	derivative of the difference between perturbed and optimal state
ξ^i	quotient of the difference between perturbed and optimal state and the length of the needle variation
ξ_{con}	ξ for variations along an intersection of switching manifolds
ξ_n	ξ for variations maintaining the nominal discrete state sequence
ξ_p	ξ for variations with a modified discrete state sequence
y	continuous output
ψ	neighboring angle of the angle between f_{j-1} and $\nabla_x m_{p,j}$
z	switching point vector
z_j	switching point for the j-th switching
$\bar{z}_j$	candidate switching point for the j-th switching
ζ_j, $\tilde{\zeta}_j$	auxiliary value at switching times

Matrices

A_j	$\mathbb{R}^{n_x \times n_x}$	system matrix of the linearized state differential equation
B_j	$\mathbb{R}^{n_x \times n_x}$	coupling matrix of the linearized state differential equation
B^+, B^-		matrix for boundary conditions
B		normalized and orthogonalized matrix with full rank in a space
C_j	$\mathbb{R}^{n_x \times n_x}$	coupling matrix of the linearized adjoint differential equation
D_j	$\mathbb{R}^{n_x \times n_{\bar{p}}}$	parameter influence matrix of the linearized state dynamics
E_j	$\mathbb{R}^{\bar{n} \times \bar{n}}$	inverse of extended Hessian of the Hamiltonian ($\bar{n} = n_u + n_{h_j^{\mathrm{a}}}$)
$E_{\oplus,j}/E_{\ominus,j}$	$\mathbb{R}^{n_x \times n_x}$	positive/negative auxiliary matrix
F_j	$\mathbb{R}^{n_x \times n_x}$	system matrix of the linearized adjoint differential equation
$\Phi_j(\tau, \tau_0)$	$\mathbb{R}^{n_x \times n_x}$	state transition matrix from time τ_0 to τ in discrete state j
G_j	$\mathbb{R}^{n_x \times n_{\bar{p}}}$	parameter influence matrix of the linearized adjoint dynamics
K_j, $\tilde{K}_j$	$\mathbb{R}^{n_u \times n_x}$	auxiliary matrix
L_j, $\tilde{L}_j$	$\mathbb{R}^{n_u \times n_x}$	auxiliary matrix
P		projection matrix or nonlinear projection
Q_j	$\mathbb{R}^{n_{\psi_j} \times n_{\psi_j}}$	Riccati matrix
$Q_{j,\cdot}$		Riccati matrix with growing dimension
$\hat{Q}_{j,\cdot}$		auxiliary Riccati matrix
R_j	$\mathbb{R}^{n_x \times n_{\psi_j}}$	Riccati matrix
$R_{j,\cdot}$		Riccati matrix with growing dimension
$\hat{R}_{j,\cdot}$		auxiliary Riccati matrix
S_j	$\mathbb{R}^{n_x \times n_x}$	Riccati matrix
$\hat{S}_j$	$\mathbb{R}^{n_x \times n_x}$	auxiliary Riccati matrix
T_j	$\mathbb{R}^{n_{\psi_j} \times n_x}$	Riccati matrix
W_j	$\mathbb{R}^{n_u \times n_u}$	auxiliary matrix
$W_{j,\cdot}$		Riccati matrix with growing dimension
$\hat{W}_{j,\cdot}$		auxiliary Riccati matrix
Ξ_j	$\mathbb{R}^{n_u \times n_x}$	auxiliary matrix

Trajectories

$\chi = \chi(\upsilon, t_0)$	$[t_0, t_e) \to \mathcal{X}$	trajectory of the continuous state
$\chi_{q_j} = \chi_{q_j}(\upsilon_{q_j}, t_j)$	$[t_j, t_{j+1}) \to \mathcal{X}_{q_j}$	trajectory of the continuous state in discrete state q_j
χ^i	$[t_0, t_e) \to \mathcal{X}$	perturbed trajectory of the continuous state
υ	$[t_0, t_e) \to U$	trajectory of the continuous control
υ_{q_j}	$[t_j, t_{j+1}) \to U_{q_j}$	trajectory of the continuous control in discrete state q_j

Functions and Functionals

c_j	$\mathbb{R}^{n_u} \to \mathbb{R}^{n_{c_u}}$	control constraints in discrete state j
$c_{q,i}$	$\mathcal{X}_q \times \mathbb{R} \to \mathbb{R}$	switching costs for a switch from discrete state q to i
dm_j		change in interior condition m_j
dy_j		change in adjoint transversality condition (Chap. 6)
$d\psi_j$		change in interior and terminal constraints
dz_j		change in Hamiltonian continuity condition (Chap. 6)
δH_j		change in Hamiltonian
ΔH_j		difference of Hamiltonians before and after a switching
f_q	$\mathcal{X}_q \times U_q \times \mathbb{R} \to \mathbb{R}^{n_x}$	continuous vector field specifying the continuous dynamics in discrete state q
$\tilde{f}_q$	$\mathcal{X}_q \times U_q \times \mathbb{R} \to \mathbb{R}^{n_{\tilde{x}}}$	continuous dynamics extended by running cost function
ϕ_q	$\mathcal{X}_q \times U_q \times \mathbb{R} \to \mathbb{R}$	running cost function in discrete state q
$\varphi_{i,k}$	$\mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}^{n_x}$	reset function for a switch from discrete state i to k
$\tilde{\varphi}_{i,k}$	$\mathbb{R}^{n_{\tilde{x}}} \times \mathbb{R} \to \mathbb{R}^{n_{\tilde{x}}}$	reset function extended by switching costs
g	$\mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}$	terminal cost function
$\tilde{g}$	$\mathbb{R}^{n_{\tilde{x}}} \times \mathbb{R} \to \mathbb{R}$	terminal cost function extended by accumulated running costs
γ_j	$\mathcal{X}_j \to \mathbb{R}^{n_x}$	transformation for a feedback linearization
$\gamma[s]$	$\mathbb{R}^{n_x} \times U \times \mathbb{R} \to \mathbb{R}$	equality constraints in direct multiple shooting
Γ	$\mathcal{Q} \times \mathcal{X} \times \Omega \to \mathcal{Q}$	discrete transition map
h_q	$\mathcal{X}_q \times U_q \times \mathbb{R} \to \mathbb{R}^{n_{h_q}}$	function for state and control constraints
H_q	$\mathcal{X}_q \times \mathbb{R}^{n_x} \times U_q \times \mathbb{R}$ $\to \mathbb{R}$	Hamiltonian for discrete state q (also in extended version in case of state constraints)
J		cost functional to be minimized
J_j		cost functional to be minimized in discrete state j
$L_f\gamma$		Lie derivative of γ in direction of f
$m_{i,k}$	$\mathbb{R}^{n_x} \times \mathbb{R} \to \mathbb{R}$	switching manifold for a switch from discrete state i to k
m_j	$\mathcal{X}_j \times \mathcal{P} \times \mathbb{R} \to \mathbb{R}^{n_j}$	interior condition (Chap. 6)
$\boldsymbol{m}_j$		interior condition with growing dimension (Chap. 6)
$\bar{m}$	$\mathbb{R}^{n_x-1} \to \mathbb{R}$	alternative, local description together with a variable $y \in \mathbb{R}^{n_x-1}$ of the zero level set of a switching manifold
$\tilde{m}_j$	$\mathcal{X}_j \to \mathbb{R}^{n_j}$	interior condition combining a switching manifold and interior constraints
o_j	$\mathcal{X}_j \to \mathbb{R}^{n_y}$	nonlinear output function
$\Omega(\cdot)$		cost functional used for controlling the convergence of an algorithm
$p_{j,i}$	$\mathcal{X}_j \to \mathbb{R}^{n_x}$	coefficient of the i-th component of the control
$\varpi(\cdot)$		error in fulfilling constraints
ϑ_j	$\mathcal{X}_j \to \mathbb{R}^{r \cdot n_{h_j^{\mathrm{a}}}}$	vector of active constraints and their time derivatives up to $r-1$
$T(t_e)$		target condition for the final time

y_j		adjoint transversality condition (Chap. 6)
ψ_j	$\mathcal{X}_q \times \mathbb{R} \to \mathbb{R}^{n_{\psi_j}}$	interior and terminal constraints
z_j		Hamiltonian continuity condition (Chap. 6)
$\boldsymbol{z}_j$		vector of Hamiltonian continuity conditions (Chap. 6)
ζ_j	$\mathcal{X}_q \times \mathbb{R} \to \mathbb{R}^{n_x}$	artificial function for proof of convergence

Parameters

Counters

α	number of the submanifold $m_{i,k}^{\alpha}$ (Chap. 3)
α	step size $c\beta^p$ (Chap. 5)
c	current number in the sequence of state constraints
c	constant (Chap. 5)
i	current number in the sequence of needle variations in the optimal control υ^*
j	current number in the sequence of discrete states; often used as replacement for the discrete state q_j
j_z	current switching point to be updated
ι	number of continuous differentiations that are at least possible for a function
k	current sampling number in indirect multiple shooting
l	iteration counter in an algorithm
n_d	number of direct neighbors of discrete state q
n_{h_q}	dimension of the constraint function h_q
n_j	dimension of the interior condition $\tilde{m}_j$
$n_{\bar{p}}$	dimension of the parameter space
n_u	dimension of the continuous control space
n_x	dimension of the continuous state space
n_y	dimension of the continuous output
n_ψ	dimension of an interior or terminal constraint
ν	number of the submanifold $m_{i,l}^{\nu}$ or counter
N	number of autonomous and controlled switchings in an execution of a hybrid system
$\bar{N}$	number of maximally allowed switchings
N_c	number of state constraints met in an execution σ
N_{cc}	basis of the combinatorial complexity $(N_{cc}-1)^N$ (nearly the average number of direct neighbors of a discrete state)
N_k	number of samples in indirect multiple shooting
$N_{\max}$	number of nodes in a tree denoting possible discrete state sequences
N_q	number of discrete states
N_s	number of samples in direct multiple shooting
p	variable number to specify the step size β^p
ϖ	counter

r	relative degree
ρ	iteration counter for an MPC algorithm
s	current sampling number in direct multiple shooting
s_c	key sample for the entry onto a state constraint
s_j	key sample for an autonomous switching
ς	counter

Constants

A_j	system matrix (Chap. 6)
$A_{\oplus,j}/A_{\ominus,j}$	positive/negative closed loop system matrix
B_j	input matrix (Chap. 6)
$B(x,\rho)$	ball around x with radius ρ
β	constant basis of the step size β^p
C_j	weighting matrix for running costs (Chap. 6)
δ	small constant
ϵ	step size
ϵ_Ω	stopping criterion specifying the accuracy of a solution
I	identity matrix of appropriate dimension
η	small constant for the sufficient descent criterion
$\tilde{\eta}_j$	parameter vector for optimal solution (Chap. 6)
κ	small constant
$K_{\oplus,j}/K_{\ominus,j}$	positive/negative gain matrix
L	Lipschitz constant
ν_l, ν_1, ν_2	coefficients for a convex combination
ν_1', ν_2', ν_3'	coefficients for a convex combination
O_j	output matrix (Chap. 6)
$\bar{O}_j$	complementary output matrix (Chap. 6)
π	constant Pi
ρ	small constant for the sufficient descent criterion
$\tilde{\rho}_j$	parameter vector for optimal solution (Chap. 6)
s	scalar factor (Chap. 5)
$S_{\oplus,j}/S_{\ominus,j}$	positive/negative solution of algebraic Riccati equation
T_p	prediction horizon
T_s	sampling time
τ_0	constant used in the proofs of convergence
$\tilde{\tau}_0$	constant used in the proofs of convergence
V_j	weighting matrix for running costs (Chap. 6)
W_e	weighting matrix for terminal costs (Chap. 6)
Ξ_j	Hamiltonian matrix (Chap. 6)
Z_j	weighting matrix for running costs (Chap. 6)
ζ	constant specifying the relation $\lim_{i\to\infty} \frac{\delta_j^i}{\kappa_j^i}$

Notation in Exemplified Systems

Functions

$\tilde{c}_j$	centrifugal and Coriolis torque
$\tilde{g}_j$	gravitational torque
$\mathcal{J}_j$	Jacobian of a manipulator
M_j	inertia matrix
o_j	forward kinematics

Variables

α	angle between the velocity and the longitudinal axis of a car
δ_F	steering angle
$\Delta\vartheta$	costs of a gear shift
$\Delta\tau$	time that a gear shift takes
ϵ	relative value after a gear shift
F_a	acceleration of a car
F_{lat}	lateral force
F_{latF}	lateral force at front wheel
F_{latR}	lateral force at rear wheel
F_{long}	longitudinal force
F_{longF}	longitudinal force at front wheel
F_{longR}	longitudinal force at rear wheel
φ	orientation of a car
λ_ϑ	adjoint associated to state ϑ
λ_τ	adjoint associated to state τ
λ_v	adjoint associated to state v
λ_y	adjoint associated to state y
ω_{turn}	turn rate of a car
$\tilde{p}$	joint velocity
q	discrete state (e.g. gear or lane)
$\tilde{q}$	joint angle
θ	orientation of a car
ϑ	accumulated running costs
t	time
τ	time as a state variable
τ_{turn}	turning torque
$\tilde{\tau}$	control torque for a manipulator
u	input force for acceleration (control)
v	velocity

x	position
x_1	position
x_2	position
y	position
z	switching point

Parameters

a	constant
A	system matrix
A_c	reference area of a car
b	constant
B	input matrix or vector
$c_1, ..., c_4$	constants
c_w	drag coefficient
C_F/C_R	front/rear cornering stiffness
d	friction coefficient (damping)
$\delta_{\text{minA}}/\delta_{\text{maxA}}$	minimal/maximal steering angle
$F_{\text{minA}}/F_{\text{maxA}}$	minimal/maximal acceleration force
$\bar{g}$	gravitational constant
γ	constant
J_z	rotational inertia of a car
k_1, k_2	coefficient
k_R	coefficient of rolling resistance
l_1	threshold for distance
l_c	length of car
l_F/l_R	distance from center of mass to front/rear axle
l_{safe}	longitudinal safety distance
l_{tr}	length of transition region
m	mass
N_ν	number of robots
R_q	cost matrix for the control
S_q	cost matrix for the state (quadratic)
ς	air density
T_q	cost matrix for the state (linear)
v_B	velocity of car B
w_c	width of car
w_r	width of road
w_{safe}	lateral safety distance

Bibliography

[1] Z. Aktas and H.J. Stetter. A classification and survey of numerical methods for boundary value problems in ordinary differential equations. *Int. J. for Numerical Methods in Engineering*, 11:771–796, 1977.

[2] A. Alessio and A. Bemporad. *Nonlinear Model Predictive Control: Towards New Challenging Applications*, volume LNCIS 384, chapter A Survey on Explicit Model Predictive Control, pages 345–369. Springer, 2009.

[3] A.D. Ames, H. Zheng, R.D. Gregg, and S. Sastry. Is there life after zeno? taking executions past the breaking (zeno) point. In *American Control Conf.*, pages 2652–2657, 2006.

[4] L. Armijo. Minimum of functions having lipschitz-continuous first partial derivatives. *Pacific J. of Mathematics*, 16:1–3, 1966.

[5] U.M. Ascher, R.M.M. Mattheij, and R.D. Russell. *Numerical Solution of Boundary Value Problems in Ordinary Differential Equations.* Prentice–Hall, 1988.

[6] U.M. Ascher and L.R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations.* SIAM, 1998.

[7] H. Axelsson, Y. Wardi, M. Egerstedt, and E. Verriest. A provably convergent algorithm for transition-time optimization in switched systems. In *44th IEEE Conf. on Decision and Control and European Control Conf.*, pages 1397–1402, 2005.

[8] H. Axelsson, Y. Wardi, M. Egerstedt, and E. Verriest. A gradient descent approach to optimal mode scheduling in hybrid dynamical systems. *J. of Opt. Theory and Applications*, 136(2):167–186, 2008.

[9] A. Back, J. Guckenheimer, and M. Myers. A dynamical simulation facility for hybrid systems. In *Hybrid Systems*, pages 255–267. Springer, 1993.

[10] E. Balas. Disjunctive programming and a hierarchy of relaxations for discrete optimization problems. *SIAM J. on Algebraic and Discrete Methods*, 6(3):466–486, 1985.

[11] M. Baotic, F. Borelli, A. Bemporad, and M. Morari. Efficient online computation of constrained optimal control. *SIAM J. on Control and Optimization*, 47(5):2470–2489, 2008.

[12] R.E. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

[13] A. Bemporad, P. Borodani, and M. Mannelli. Hybrid control of an automotive robotized gearbox for reduction of consumptions and emissions. In O. Maler and A. Pnueli, editors, *Hybrid Systems: Computation and Control*, LNCS 2623, pages 81–96. Springer, 2003.

[14] A. Bemporad, F. Borrelli, and M. Morari. On the optimal control law for linear discrete time hybrid systems. In C.J. Tomlin and M.R. Greenstreet, editors, *Hybrid Systems: Computation and Control, LNCS 2289*, pages 106–119. Springer, 2002.

[15] A. Bemporad and N. Giorgetti. Logic-based solution methods for optimal control of hybrid systems. *IEEE Trans. on Automatic Control*, 51(6):963–976, 2006.

[16] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35:407–427, 1999.

[17] A. Bemporad, M. Morari, V. Dua, and E.N. Pistikopoulos. The explicit linear quadratic regulator for constrained systems. *Automatica*, 38:3–20, 2002.

[18] S.C. Bengea and R.A. DeCarlo. Optimal control of switching systems. *Automatica*, 41(1):11–27, 2005.

[19] L.D. Berkovitz. *Optimal Control Theory*. Springer, 1974.

[20] Joh. Bernoulli, Jac. Bernoulli, and L. Euler. *Abhandlungen über Variationsrechnung*. Verlag von W. Engelmann, 1894.

[21] D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

[22] J.T. Betts. Survey of numerical methods for trajectory optimization. *J. of Guidance, Control, and Dynamics*, 21(2):193–207, 1998.

[23] J.T. Betts. *Practical Methods for Optimal Control using Nonlinear Programming*. SIAM, 2001.

[24] L.T. Biegler. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers & Chemical Engineering*, 8(3–4):243–247, 1984.

[25] H.G. Bock. Numerical solution of nonlinear multipoint boundary value problems with applications to optimal control. *ZAMM*, 58:407–409, 1978.

[26] H.G. Bock and K. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *9th IFAC World Congress*, pages 242–247, 1984.

[27] V. Boltyanski. Sufficient conditions for lagrange, mayer, and bolza optimization problems. *Mathematical Problems in Engineering*, 7:177–203, 2001.

[28] M. S. Branicky. Multiple lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. on Automatic Control*, 43(4):475–482, 1998.

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

[30] M.S. Branicky and S.K.Mitter. Algorithms for optimal hybrid control. In *34th IEEE Conf. on Decision and Control*, pages 2661–2666, 1995.

[31] M. Broucke, M.D. Di Benedetto, S. Di Gennaro, and A. Sangiovanni-Vincentelli. Efficient solution of optimal control problems using hybrid systems. *SIAM J. on Control and Optimization*, 43(6):1923–1952, 2005.

[32] A.E. Bryson and Y.C. Ho. *Applied Optimal Control: Optimization, Estimation and Control*. Hemisphere Publishing Corp., 1975.

[33] G. Bätz, M. Sobotka, D. Wollherr, and M. Buss. Robot basketball: Ball dribbling - a modified juggling task. In T. Kröger and F.M. Wahl, editors, *Advances in Robotics Research*, pages 323–334. Springer, 2009.

[34] J. Buck. Rhythmic flashing of fireflies ii. *The Quarterly Review of Biology*, 63(3):265–289, 1988.

[35] R. Bulirsch. Die Mehrzielmethode zur numerischen Lösung von nichtlinearen Randwertproblemen und Aufgaben der optimalen Steuerung. Technical report, Carl-Cranz Gesellschaft, DLR, Germany, 1971.

[36] R. Bulirsch, F. Montrone, and H.J. Pesch. Abort landing in the presence of windshear as a minimax optimal control problem, part 2: Multiple shooting and homotopy. *J. of Opt. Theory and Applications*, 70(2):223–254, 1991.

[37] R. Bulirsch, E. Nerz, H.J. Pesch, and O. von Stryk. Combining direct and indirect methods in optimal control: Range maximization of a hang glider. In R. Bulirsch, A. Miele, J. Stoer, and K.H. Well, editors, *Optimal Control*, pages 273–288. Birkhäuser Verlag, 1991.

[38] M. Buss, O. von Stryk, R. Bulirsch, and G. Schmidt. Towards hybrid optimal control. *at-Automatisierungstechnik*, 48(9):448–459, 2000.

[39] P.E. Caines, F.H. Clarke, X. Liu, and R.B. Vinter. A maximum principle for hybrid optimal control problems with pathwise state constraints. In *45th IEEE Conf. on Decision & Control*, pages 4821–4825, 2006.

[40] P.E. Caines and M.S. Shaikh. Optimality zone algorithms for hybrid systems: Efficient algorithms for optimal location and control computation. In *Hybrid Systems: Computation and Control*, LNCS 3927, pages 123–137, 2006.

[41] E.F. Camacho and C. Bordons. *Model Predictive Control*. Springer, 2004.

[42] E.F. Camacho, D.R. Ramírez, D. Limón, D.M. Peña, and T. Álamo. Model predictive control techniques for hybrid systems. In *3rd IFAC Conf. on Analysis and Design of Hybrid Systems*, pages 1–13, 2009.

[43] E.F. Camacho, D.R. Ramírez, D. Limón, D.M. Peña, and T. Álamo. Model predictive control techniques for hybrid systems. *Annual Reviews in Control*, 34(1):21–31, 2010.

[44] C. Carathéodory. *Variationsrechnung und partielle Differentialgleichungen erster Ordnung*. Teubner, 1935.

[45] F.H. Clarke and R.B. Vinter. Optimal multiprocesses. *SIAM J. on Control and Optimization*, 27(5):1072–1091, 1989.

[46] R. Pallu de la Barrière. *Optimal Control Theory*. W.B. Saunders Company, 1967.

[47] D. Del Vecchio, R. M. Murray, and P. Perona. Decomposition of human motion into dynamics based primitives with application to drawing tasks. *Automatica*, 39:2085–2098, 2003.

[48] M. Diehl, H.G. Bock, and J.P. Schlöder. A real-time iteration scheme for nonlinear optmization in optimal feedback control. *SIAM J. on Control and Optimization*, 43(5):1714–1736, 2005.

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

[50] M. Egerstedt, Y. Wardi, and H. Axelsson. Transition-time optimization for switched-mode dynamical systems. *IEEE Trans. on Automatic Control*, 51(1):110–115, 2006.

[51] H.O. Fattorini. *Infinite Dimensional Optimization and Control Theory*. Cambridge University Press, 1999.

[52] A. Ferrante, G. Marro, and L. Ntogramatzidis. A parametrization of the solutions of the finite-horizon LQ problem with general cost and boundary conditions. *Automatica*, 41:1359–1366, 2005.

[53] R. Fletcher and S. Leyffer. Numerical experience with lower bounds for miqp branch-and-bound. *SIAM J. on Optimization*, 8(2):604–616, 1995.

[54] E. Frazzoli. *Robust Hybrid Control for Autonomous Vehicle Motion Planning*. PhD thesis, Massachusetts Inst. of Technology, 2001.

[55] G.N. Galbraith and R.B. Vinter. Optimal control of hybrid systems with an infinite set of discrete states. *J. of Dynamical and Control Systems*, 9(4):563–584, 2003.

[56] M. Garavello and B. Piccoli. Hybrid necessary principle. In *44th IEEE Conf. on Decision and Control and European Control Conf.*, pages 723–728, 2005.

[57] M.R. Garey and D.S. Johnson. *Computers and Intractability: A guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[58] C. Geiger and C. Kanzow. *Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben*. Springer, 1999.

[59] M. Gerdts. *Optimal Control of Ordinary Differential Equations and Differential Algebraic Equations*. PhD thesis, Department of Mathematics, University of Bayreuth, 2007.

[60] R. Ghaemi, J. Sun, and I.V. Kolmanovsky. Neighboring extremal solution for nonlinear discrete-time optimal control problems with state inequality constraints. *IEEE Trans. on Automatic Control*, 54(11):2674–2679, 2009.

[61] F. Glover. Improved linear integer programming formulations of nonlinear integer problems. *Management Science*, 22:455–560, 1975.

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

[63] C.J. Goh and K.L. Teo. Control parametrization: A unified approach to optimal control problems with general constraints. *Automatica*, 24(1):3–18, 1988.

[64] H. Gonzalez, R. Vasudevan, M. Kamgarpour, S.S. Sastry, R. Bajcsy, and C.J. Tomlin. A descent algorithm for the optimal control of constrained nonlinear switched dynamical systems. In *Hybrid Systems: Computation and Control*, pages 51–60, 2010.

[65] R.G. Gottlieb. Rapid convergence to optimum solutions using a min-h strategy. *J. of Guidance, Control, and Dynamics*, 5(2):322–329, 1967.

[66] W. Grimm and A. Markl. Adjoint estimation from a direct multiple shooting method. *J. of Optimization Theory and Applications*, 92(2):263–283, 1997.

[67] L. Grüne. An adaptive grid scheme for the discrete hamilton-jacobi-bellman equation. *Numerische Mathematik*, 75:319–337, 1997.

[68] L. Grüne and O. Junge. Optimal stabilization of hybrid systems using a set oriented approach. In *17th Int. Symp. on Mathematical Theory of Networks and Systems*, pages 2089–2093, 2006.

[69] I.E. Grossmann. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering*, 3:227–252, 2002.

[70] R.F. Hartl, S.P. Sethi, and R.G. Vickson. A survey of the maximum principle for optimal control problems with state constraints. *SIAM Review*, 37(2):181–218, 1995.

[71] G.C. Haynes, A. Khripin, G. Lynch, J. Amory, A. Saunders, A.A. Rizzi, and D.E. Koditschek. Rapid pole climbing with a quadrupedal robot. In *IEEE Int. Conf. on Robotics and Automation*, pages 2767–2772, 2009.

[72] S. Hedlund and A. Rantzer. Optimal control of hybrid systems. In *38th Conf. on Decision and Control*, pages 3972–3977, 1999.

[73] E. Hellström, M. Ivarsson, J. Aslund, and L. Nielsen. Look-ahead control for heavy trucks to minimize trip time and fuel consumption. *Control Engineering Practice*, 17(2):245–254, 2009.

[74] M.R. Hestens. *Calculus of Variations and Optimal Control Theory.* J. Wiley and Sons, 1966.

[75] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Learning attractor landscapes for learning motor primitives. In *Advances in Neural Information Processing Systems*, pages 1523–1530. MIT Press, 2003.

[76] A.D. Ioffe and V.M. Tihomirov. *Theory of Extremal Problems.* North-Holland Publishing, 1979.

[77] T.A. Johansen. Approximate explicit receding horizon control of constrained nonlinear systems. *Automatica*, 40:293–300, 2004.

[78] K.H. Johansson, M. Egerstedt, J. Lygeros, and S. Sastry. On the regularization of zeno hybrid automata. *System & Control Letters*, 38:141–150, 1999.

[79] J.V. Kadam and W. Marquardt. Sensitivity-based solution updates in closed-loop dynamic optimization. In *7th Int. Symp. on Dynamics and Control of Process Systems*, 2004.

[80] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *Int. J. on Robotics Research*, 30(7):846–894, 2011.

[81] L. Kavraki, P. Svestka, J. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration space. *IEEE Trans. on Robotics and Automation*, 12(4):566–580, 1996.

[82] H.J. Kelley. Gradient theory of optimal flight paths. *ARS Journal*, 30:947–954, 1960.

[83] H.J. Kelley, R.E. Kopp, and H.G. Moyer. A trajectory optimization technique based upon the theory of the second variation. *Progress in Astronautics and Aeronautics*, 14:559–582, 1964.

[84] P. Kenneth and R. McGill. Two-point boundary value problem techniques. *Advances in Control Systems*, 3:69–109, 1966.

[85] C.S. Kenny and R.B. Lepnik. Numerical integration of the differential matrix riccati equation. *IEEE Trans. on Automatic Control*, 30(10):962–970, 1985.

[86] H.K. Khalil. *Nonlinear Systems.* Prentice Hall, 2002.

[87] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. of Robotics Research*, 5(1):90–98, 1986.

[88] C. Kirches. *Fast Numerical Methods for Mixed–Integer Nonlinear Model–Predictive Control.* PhD thesis, Ruprecht–Karls–Universität Heidelberg, 2010.

[89] C. Kirches, H.G. Bock, J.P. Schlöder, and S. Sager. Block-structured quadratic programming for the direct multiple shooting method for optimal control. *Optimization Methods and Software*, 26(2):239–257, 2011.

[90] J. Kober, B. J. Mohler, and J. Peters. Learning perceptual coupling for motor primitives. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 834–839, 2008.

[91] J.J. Kuffner and S.M. LaValle. Rrt-connect: an efficient approach to single-query path planning. In *IEEE Int. Conf. on Robotics and Automation*, pages 995–1001, 2000.

[92] G.A. Kurina and R. März. Feedback solutions of optimal control problems with dae constraints. *SIAM J. on Control and Optimization*, 46(4):1277–1298, 2007.

[93] L.S. Lasdon, S.K. Mitter, and A.D. Waren. The conjugate gradient method for optimal control problems. *IEEE Trans. on Automatic Control*, 12(2):132–138, 1967.

[94] S.M. LaValle. *Planning Algorithms*. Cambridge University Press, 2006.

[95] S.M. LaValle and J.J. Kuffner. Randomized kinodynamic planning. *Int. J. of Robotics Research*, 20(5):378–400, 2001.

[96] J.M. Lee. *Introduction to Smooth Manifolds*. Springer, 2002.

[97] S. Lee and I.E. Grossmann. New algorithms for nonlinear generalized disjunctive programming. *Computers & Chemical Engineering*, 24(9–10):2125–2141, 2000.

[98] D.B. Leineweber, I. Bauer, H.G. Bock, and J.P. Schlöder. An efficient multiple shooting based reduced sqp strategy for large-scale dynamic process optimization - part i: Theoretical aspects, part ii: Software aspects and applications. *Computers & Chemical Engineering*, 27:157–174, 2003.

[99] S. Leyffer. Integrating sqp and branch-and-bound for mixed integer nonlinear programming. *Computational Optimization and Applications*, 18:295–309, 2001.

[100] D. Liberzon and A.S. Morse. Basic problems in stability and design of switched systems. *IEEE Control Systems*, 19(5):59–70, 1999.

[101] A.G. Longmuir and E.V. Bohn. Second-variation methods in dynamic optimization. *J.l of Optimization Theory and Applications*, 3(3):164–173, 1969.

[102] J. Lu, L.Z. Liao, A. Nerode, and J.H. Taylor. Optimal control of systems with continuous and discrete states. In *32nd IEEE Conf. on Decision and Control*, pages 2292 – 2297, 1993.

[103] R. Luus. *Iterative Dynamic Programming*. Chapman & Hall, 2000.

[104] H. Mangesius. Hybrid optimal control for transportation planning with multiple vehicles. Master's thesis, Institute of Automatic Control Engineering, Technische Universität München, 2009.

[105] H. Mangesius, M. Sobotka, and O. Stursberg. Solution of a multi-agent transport problem by hybrid optimization. In *10th Int. Workshop on Discrete Event Systems*, 2010.

[106] H. Maurer. On optimal control problems with boundary state variables and control appearing linearly. *SIAM J. on Control and Optimization*, 15(3):345–362, 1977.

[107] R.K. Mehra and A.E. Bryson. Conjugate gradient methods with an application to v/stol flight-path optimization. Technical report, NASA, 1967.

[108] A. Miele. Recent advances in gradient algorithms for optimal control problems. *J. of Optimization Theory and Applications*, 17(5/6):361–430, 1975.

[109] S. Mitra, D. Liberzon, and N. Lynch. Verifying average dwell time of hybrid systems. *ACM Trans. on Embedded Computing Systems*, 8(1):3.1–3.37, 2008.

[110] H. Miyamoto, S. Schaal, F. Gandolfo, H. Gomi, Y. Koike, R. Osu, E. Nakano, Y. Wada, and M. Kawato. A kendama learning robot based on bi-directional theory. *Neural Networks*, 9(8):1281–1302, 1996.

[111] A.S. Morse. Supervisory control of families of linear set-point controllers–part 1 : Exact matching. *IEEE Trans. on Automatic Control*, 41(10):1413–1431, 1996.

[112] R. M. Murray, Z. Li, and S. S. Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1st. edition, March 1994.

[113] R.M. Murray. Recent research in cooperative control of multivehicle systems. *J. of Dynamics, Systems, Measurment, and Control*, 129:571–583, 2007.

[114] J. Nakanishi, J. Morimoto, G. Endo, S. Schaal, and M. Kawato. Learning from demonstration and adaptation of biped locomotion with dynamical movement primitives. In *Workshop on Robot Learning by Demonstration, IEEE Int. Conf. on Intelligent Robots and Systems*, 2003.

[115] A. Nerode and W. Kohn. Models for hybrid systems: Automata, topologies, controllability, observability. In R. Grossman, A. Nerode, A. Ravn, and H. Rischel, editors, *Hybrid Systems*, volume 736 of *Lecture Notes in Computer Science*, pages 317–356. Springer, 1993.

[116] A. Neumaier. *Acta Numerica 2004*, chapter Complete Search in Continuous Global Optimization and Constraint Satisfaction, pages 271–369. Cambridge University Press, 2004.

[117] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 1999.

[118] F. Nori and R. Frezza. Nonlinear control by a finite set of motion primitives. *6th IFAC Symp. on Nonlinear Control Systems*, 2004.

[119] I. Nowak. *Relaxation and Decomposition Methods for Mixed Integer Nonlinear Programming*. Springer, 2005.

[120] H.J. Oberle. Numerical computation of singular control problems with application to optimal heating and cooling by solar energy. *Applied Mathematics and Optimization*, 5:297–314, 1979.

[121] H.B. Pacejka. *Tyre Models for Vehicle Dynamics.* Swets & Zeitlinger, 1993.

[122] D. Pekarek, A.D. Ames, and J.E. Marsden. Discrete mechanics and optimal control appplied to the compass gait biped. In *46th IEEE Conf. on Decision and Control*, pages 5376–5382, 2007.

[123] D.L. Pepyne and C.G. Cassandras. Optimal control of hybrid systems in manufacturing. *Proc. of the IEEE*, 88(7):1108–1123, 2000.

[124] E. Perea-López, B.E. Ydstie, and I.E. Grossmann. A model predictive control strategy for supply chain optimization. *Computers and Chemical Engineering*, 27(8–9):1201–1218, 2003.

[125] B. E. Perk and J.-J. E. Slotine. Motion primitives for robotic flight control. *Computing Research Repository*, 2006.

[126] H.J. Pesch. Real-time computation of feedback controls for constrained optimal control problems, part 1: Neighboring extremals. *Optimal Control Applications and Methods*, 10:129–145, 1989.

[127] H.J. Pesch. Real-time computation of feedback controls for constrained optimal control problems, part 2: A correction method based on multiple shooting. *Optimal Control Applications and Methods*, 10:147–171, 1989.

[128] H.J. Pesch. *Computational Optimal Control*, chapter Solving Optimal Control and Pursuit-Evasion Game Problems of High Complexity, pages 43–61. Birkhäuser Verlag, 1994.

[129] H.J. Pesch and R. Bulirsch. The maximum principle, bellman's equation, and caratheodory's work. *J. of Optimization Theory and Applications*, 80(2):199–225, 1994.

[130] E. Polak. *Computational Methods in Optimization*, volume 77 of *Math. in Science and Eng.* Academic Press, 1971.

[131] E. Polak. An historical survey of computational methods in optimal control. *SIAM Review*, 15(2):553–584, 1973.

[132] L.S. Pontryagin, V.G. Boltyanskii, R.V. Gamkrelidze, and E.F. Mishchenko. *The Mathematical Theory of Optimal Processes.* New York: Wiley, 1963.

[133] W.B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality.* John Wiley & Sons, 2007.

[134] A. Rantzer and M. Johansson. Piecewise linear quadratic optimal control. *IEEE Trans. on Automatic Control*, 45(4):629–637, 2000.

[135] C. Reinl, M. Glocker, and O. von Stryk. Optimalsteuerung kooperierender Mehrfahrzeugsysteme. *at-Automatisierungstechnik*, 68, 2008.

[136] P. Riedinger, J. Daafouz, and C. Iung. About solving hybrid optimal control problems. In *17th IMACS World Congress*, 2005.

[137] P. Riedinger, C. Iung, and F. Kratz. An optimal control approach for hybrid systems. *European J. of Control*, 9(5):449–458, 2003.

[138] P. Riedinger, F. Kratz, C. Iung, and C. Zanne. Linear quadratic optimization for hybrid systems. In *38th Conf. on Decision and Control*, pages 3059–3064, 1999.

[139] P. Riedinger, C. Zanne, and F. Kratz. Time optimal control of hybrid systems. In *American Control Conf.*, pages 2466–2470, 1999.

[140] M. Rungger. *On the Numerical Solution of Nonlinear and Hybrid Optimal Control Problems*. PhD thesis, Universität Kassel, 2011.

[141] M. Rungger and O. Stursberg. Optimal control for deterministic hybrid systems using dynamic programming. In *3rd IFAC Conf. on Analysis and Design of Hybrid Systems*, pages 316–321, 2009.

[142] M. Rungger and O. Stursberg. Continuity of the value function for exit time optimal control problems of hybrid systems. In *49th IEEE Conf. on Decision and Control*, pages 4210–4215, 2010.

[143] M. Rungger and O. Stursberg. A numerical method for hybrid optimal control based on dynamic programming. *Nonlinear Analysis: Hybrid Systems*, 5(2):254–274, 2011.

[144] S. Sager. *Numerical Methods for Mixed–Integer Optimal Control Problems*. PhD thesis, Universität Heidelberg, 2006.

[145] R.G. Sanfelice and E. Frazzoli. On the optimality of dubins paths across heterogeneous terrain. In *Hybrid Systems: Computation and Control*, LNCS 4981, pages 457–470. Springer, 2008.

[146] R.W.H. Sargent. Optimal control. *J. of Computational and Applied Mathematics*, 124:361–371, 2000.

[147] S. Schaal. Dynamic movement primitives - a framework for motor control in humans and humanoid robots. In *The Int. Symp. on Adaptive Motion of Animals and Machines*, 2003.

[148] S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert. Control, planning, learning, and imitation with dynamic movement primitives. In *Workshop on bilateral Paradigms on Humans and Humanoids, IEEE Int. Conf. on Intelligent Robots and Systems*, 2003.

[149] T. Schauß, M. Scheint, M. Sobotka, W. Seiberl, and M. Buss. Effects of compliant ankles on bipedal locomotion. In *IEEE Int. Conf. on Robotics and Automation*, pages 2761–2766, 2009.

[150] M. Scheint, M. Sobotka, and M. Buss. Compliance in gait synthesis: Effects on energy and gait. In *IEEE Int. Conf. on Humanoid Robots*, pages 259–264, 2008.

[151] A. Schild, X.C. Ding, M. Egerstedt, and J. Lunze. Design of optimal switching surfaces for switched autonomous systems. In *48th IEEE Conf. on Decision and Control*, pages 5293–5298, 2009.

[152] A. Schoellig, P.E. Caines, M. Egerstedt, and R. Malhamé. A hybrid bellman equation for systems with regional dynamics. In *46th IEEE Conf. on Decision and Control*, pages 3393–3398, 2007.

[153] C.E. Seah and I. Hwang. Stochastic linear hybrid systems: Modeling, estimation, and application in air traffic control. *IEEE Trans. on Control Systems Technology*, 17(3):563–575, 2009.

[154] H. Seywald and R.R. Kumar. Finite difference scheme for automatic costate calculation. *J. of Guidance, Control, and Dynamics*, 19(1):231–239, 1996.

[155] M.S. Shaikh and P.E. Caines. On the optimal control of hybrid systems: Analysis and zonal algorithms for trajectory and schedule optimization. In *42nd IEEE Conf. on Decision and Control*, pages 2133–2149, 2003.

[156] M.S. Shaikh and P.E. Caines. On the optimal control of hybrid systems: Optimization of trajectories, switching times, and location schedules. In *Hybrid Systems: Computation and Control*, LNCS 2623, pages 466–481. Springer, 2003.

[157] M.S. Shaikh and P.E. Caines. On the hybrid optimal control problem: Theory and algorithms. *IEEE Trans. on Automatic Control*, 52(9):1587–1603, 2007. Corrigendum: 54(6):1440, 2009.

[158] D. Sieber. *Optimal Constrained Trajectory Planning for an Autonomous Vehicle.* Bachelor Thesis, Technische Universität München, 2009.

[159] J.-J. E. Slotine and W. Li. *Applied Nonlinear Control.* Prentice-Hall, 1991.

[160] J. Stoer and R. Bulirsch. *Numerische Mathematik 2.* Springer, 2005.

[161] O. Stursberg. Dynamic optimization of processing systems with mixed degrees of freedom. In *7th Int. IFAC Symp. Dynamics and Control of Process Systems*, 2004.

[162] O. Stursberg. A graph search algorithm for optimal control of hybrid systems. In *43rd IEEE Conf. on Decision and Control*, pages 1412–1417, 2004.

[163] O. Stursberg and S. Engell. Optimal control of switched continuous systems using mixed-integer programming. In *15th IFAC World Congress*, pages 1489–1494, 2002.

[164] O. Stursberg and S. Panek. Control of switched hybrid systems based on disjunctive formulations. In *Hybrid Systems: Computation and Control*, LNCS 2289, pages 421–435. Springer, 2002.

[165] H.J. Sussmann. A maximum principle for hybrid optimal control problems. In *38th IEEE Conf. on Decision and Control*, pages 425–430, 1999.

[166] K.P. Sycara. Multiagent systems. *AI Magazine*, 19(2):79–92, 1998.

[167] I.A. Tall. Multi-input control systems: Explicit feedback linearization. In *49th IEEE Conf. on Decision and Control*, pages 5378–5383, 2010.

[168] B.D. Tapley and J.M. Lewalle. Comparison of several numerical optimization methods. *J. of Optimization Theory and Applications*, 1(1):1–32, 1967.

[169] F. Taringoo and P.E. Caines. The sensitivity of hybrid systems optimal cost functions with respect to switching manifold parameters. In *Hybrid Systems: Computation and Control*, pages 475–479. Springer, 2009.

[170] F. Taringoo and P.E. Caines. Gradient-geodesic hmp algorithms for the optimization of hybrid systems based on the geometry of switching manifolds. In *49th IEEE Conf. on Decision and Control*, pages 1534–1539, 2010.

[171] R Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts. Lqr-trees: Feedback motion planning via sum-of-squares verification. *Int. Journal of Robotics Research*, 29:1038–1052, 2010.

[172] T.H. Tsang, D.M. Himmelblau, and T.F. Edgar. Optimal control via collocation and non-linear programming. *Int. J. on Control*, 21(5):763–768, 1975.

[173] M. Ueberle. *Design, Control, and Evaluation of a Family of Kinesthetic Haptic Interfaces*. PhD thesis, Technische Universität München, 2006.

[174] O. von Stryk. Numerical solution of optimal control problems by direct collocation. In R. Bulirsch, A. Miele, J. Stoer, and K.H. Well, editors, *Optimal Control*, volume 111 of Int. Ser. Numerical Mathematics, pages 129–143. Birkhäuser Verlag, 1993.

[175] O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37:357–373, 1992.

[176] O. von Stryk and M. Schlemmer. Optimal control of the industrial robot manutec r3. In R. Bulirsch and D. Kraft, editors, *Computational Optimal Control, Int. Series of Numerical Mathematics*, volume 115, pages 367–382. Birkhäuser, 1994.

[177] Y. Wang and S. Boyd. Fast model predictive control using online optimization. In *17th IFAC World Congress*, pages 6974–6979, 2008.

[178] Y. Wardi, M. Egerstedt, M. Boccadoro, and E. Verriest. Optimal control of switching surfaces. In *43rd IEEE Conf. on Decision and Control*, pages 1854–1859, 2004.

[179] D.S. Watkins. *Fundamentals of Matrix Computations.* John Wiley and Sons, 2002.

[180] S. Wei, K. Uthaichana, M. Zefran, R.A. DeCarlo, and S. Bengea. Applications of numerical optimal control to nonlinear hybrid systems. *Nonlinear Analysis: Hybrid Systems*, 1(2):264–279, 2007.

[181] H.P. Williams. *Model Building in Mathematical Programming.* John Wiley, 1978.

[182] H.S. Witsenhausen. A class of hybrid-state continuous-time dynamic systems. *IEEE Trans. on Automatic Control*, 11(2):161–167, 1966.

[183] L.J. Wood. Second-order optimality conditions for the bolza problem with path constraints. In *12th IEEE Conf. on Decision and Control*, pages 606–613, 1973.

[184] L. Würth, R. Hannemann, and W. Marquardt. Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization. *J. of Process Control*, 19(8):1277–1288, 2009.

[185] X. Xu and P.J. Antsaklis. A dynamic programming approach for optimal control of switched systems. In *39th IEEE Conf. on Decision and Control*, pages 1822–1827, 2000.

[186] X. Xu and P.J. Antsaklis. Optimal control of switched systems via non-linear optimization based on direct differentiations of value functions. *Int. J. of Control*, 75(16–17):1406–1426, 2002.

[187] X. Xu and P.J. Antsaklis. Optimal control of switched systems based on parameterization of the switching instants. *IEEE Trans. on Automatic Control*, 49(1):2–16, 2004.

[188] J. Zabczyk. *Mathematical Control Theory: An Introduction.* Birkhäuser, 1992.

[189] V.M. Zavala and L.T. Biegler. The advanced-step nmpc controller: Optimality, stability and robustness. *Automatica*, 45(1):86–93, 2009.

[190] V. Zeidan. The riccati equation for optimal control problems with mixed state-control constraints: Necessity and sufficiency. *SIAM J. on Control and Optimization*, 32(5):1297–1321, 1994.

[191] M.N. Zeilinger, C.N. Jones, and M. Morari. Real-time suboptimal model predictive control using a combination of explicit mpc and online optimization. In *47th IEEE Conf. on Decision and Control*, pages 4718–4723, 2008.

[192] F. Zhang and Z. Shi. Optimal and adaptive battery discharge strategies for cyber-physical systems. In *48th IEEE Conf. on Decision and Control and 28th Chinese Control Conf.*, pages 6232–6237, 2009.

[193] H. Zhang, W.S. Gray, and O.R. González. Performance analysis of digital flight control systems with rollback error recovery subject to simulated neutron-induced upsets. *IEEE Trans. on Control Systems Technology*, 16(1):46–59, 2008.

[194] J. Zhang, K. H. Johansson, J. Lygeros, and S. Sastry. Zeno hybrid systems. *Int. J. of Robust and Nonlinear Control*, 11(2):435–451, 2001.

[195] Q. Zhang and G. Yin. Nearly-optimal asset allocation in hybrid stock investment models. *J. of Optimization Theory and Applications*, 121(2):419–445, 2004.

[196] A. Zomotor. *Fahrwerktechnik: Fahrverhalten*. Vogel Buchverlag, 1991.

Own Publications and Supervised Student Projects

[197] B. Buchholz. Offline Routen Optimierung für LKW. Master's thesis, Technische Universität München, 2010.

[198] H. Ding, G. Schnattinger, B. Passenberg, and O. Stursberg. Improving motion of robotic manipulators by an embedded optimizer. In *IEEE Conf. on Automation Science and Engineering*, pages 204–209, 2010.

[199] M. Eibl. Erstellung einer Simulationsumgebung für prädiktiv gesteuerte Verkehrsteilnehmer. Master's thesis, Technische Universität München, 2009.

[200] D. Engelhardt. Prädiktive Optimierungsstrategien mittels Mixed-Integer Optimization für die ökonomische Optimierung des Betriebs schwerer LKW. Master's thesis, Technische Universität München, 2010.

[201] S. Gläser. *Optimale Steuerung von stückweise affine Systemen durch hybride Optimierung*. Bachelor Thesis, Technische Universität München, 2011.

[202] M. Herrmann. Optimierung der Gangwahl bei autonomen Fahrzeugen. Master's thesis, Technische Universität München, 2010.

[203] D. Krings. Online-Identifikation eines dynamischen Modells des Fahrverhaltens eines LKWs. Master's thesis, Technische Universität München, 2009.

[204] M. Kröninger. Initialisierungskonzepte für indirekte Optimierungsmethoden. Master's thesis, Technische Universität München, 2010.

[205] A. Kunz. *Hybrid Optimal Control of a Cooperative Multi Agent System*. Bachelor Thesis, Technische Universität München, 2011.

[206] A. Mergel. Fast updates for model predictive control of hybrid systems. Master's thesis, Technische Universität München, 2010.

[207] B. Passenberg, P.E. Caines, M. Leibold, O. Stursberg, and M. Buss. Optimal control for hybrid systems with partitioned state space. *IEEE Trans. on Automatic Control*, 2012, provisionally accepted.

[208] B. Passenberg, P.E. Caines, M. Sobotka, O. Stursberg, and M. Buss. The minimum principle for hybrid systems with partitioned state space and unspecified discrete state sequence. In *49th IEEE Conf. on Decision and Control*, pages 6666–6673, 2010.

[209] B. Passenberg, P. Kock, and O. Stursberg. Combined time and fuel optimal driving of trucks based on a hybrid model. In *European Control Conf.*, pages 4955–4960, 2009.

[210] B. Passenberg, M. Kroeninger, G. Schnattinger, M. Leibold, O. Stursberg, and M. Buss. Initialization concepts for optimal control of hybrid systems. In *18th IFAC World Congress*, pages 10274–10280, 2011.

[211] B. Passenberg, M. Leibold, O. Stursberg, and M. Buss. The minimum principle for time-varying hybrid systems with state switching and jumps. In *50th IEEE Conf. on Decision and Control and European Control Conf.*, pages 6723–6729, 2011.

[212] B. Passenberg, M. Leibold, O. Stursberg, and M. Buss. A globally convergent min-h algorithm for optimal control of hybrid systems. *SIAM J. on Control and Optimization*, 2011, submitted.

[213] B. Passenberg, A. Mergel, M. Leibold, O. Stursberg, and M. Buss. An efficient algorithm based on neighboring extremals for model predictive control of hybrid systems. *Automatica*, 2011, submitted.

[214] B. Passenberg, M. Sobotka, O. Stursberg, M. Buss, and P.E. Caines. An algorithm for discrete state sequence and trajectory optimization for hybrid systems with partitioned state space. In *49th IEEE Conf. on Decision and Control*, pages 4223–4229, 2010.

[215] B. Passenberg and O. Stursberg. Graph search for optimizing the discrete location sequence in hybrid optimal control. In *3rd IFAC Conf. on Analysis and Design of Hybrid Systems*, pages 304–309, 2009.

[216] G. Schnattinger. Randomized optimal motion planning for articulated robots with moving obstacles. Master's thesis, Technische Universität München, 2009.

[217] D. Sieber. *Optimal Constrained Trajectory Planning for an Autonomous Vehicle.* Bachelor Thesis, Technische Universität München, 2009.

[218] M. Steinegger. Trajectory planning for robots based on motion primitives. Master's thesis, Technische Universität München, 2010.

[219] M. Steinegger, B. Passenberg, M. Leibold, and M. Buss. Trajectory planning for manipulators based on the optimal concatenation of lq control primitives. In *50th IEEE Conf. on Decision and Control and European Control Conf.*, pages 2837–2842, 2011.

[220] R. Weller. Prädiktive Optimierungsstrategien mittels Dynamischer Programmierung für die ökonomische Optimierung des Betriebs schwerer LKW. Master's thesis, Technische Universität München, 2010.